



ORACLE

Каталог продуктов семейства Oracle Database

Самая быстрая, надежная, безопасная
и масштабируемая СУБД в мире

2020 год

Содержание

Введение	5
Технологии Oracle	6
Обзор технологий Oracle	7
Обзор продуктов семейства Oracle Database.....	8
СУБД Oracle Database.....	19
Oracle DB — конвергентная и мультимодельная СУБД.....	23
Построение информационных систем с повышенными требованиями по безопасности и защите информации	34
Архитектура обеспечения максимальной безопасности баз данных Oracle.....	46
Реализация систем высокой доступности средствами Oracle.....	59
Автономные базы данных.....	67
Oracle Sharding.....	71
Database Fleet Maintenance.....	75
Edition Based Redefinition (EBR).....	76
Глобальные распределенные сервисы (Global Data Services, GDS)	79
Облачные вычисления. Публичное облако Oracle (Public Cloud).....	81
Oracle Infrastructure as a Service (OCI), DBaaS.....	88
Как создать частное облако от Oracle	97
Инженерные системы.....	104
Продукты Oracle для управления качеством приложений.....	107
Хранилища данных и аналитические системы на основе СУБД Oracle.....	108
Консолидация баз данных	113
Обновление версии и миграция СУБД	117
Продукты Oracle	121
Опции Oracle DB Enterprise Edition	121
Active Data Guard	122
Advanced Analytics	125
Advanced Compression.....	126
Advanced Security.....	133



Опция Oracle Database In-Memory	137
Database Vault	140
In-Memory Database Cache	143
Label Security.....	147
Multitenant опция.....	150
OLAP	152
Partitioning	153
Real Application Clusters.....	154
Real Application Clusters One Node	157
Real Application Testing.....	160
Spatial and Graph	164

Средства управления инфраструктурой и СУБД (ОМС, OEM).. 167

Oracle Enterprise Manager.....	168
Data Masking and Subsetting Pack	171
Cloud Management Pack for Oracle Database	174
Diagnostics Pack	177
Tuning Pack.....	179
Database Lifecycle Management Pack.....	182
Oracle Management Cloud (OMC)	184

Прочие серверные продукты..... 191

Golden Gate.....	192
Oracle Big Data SQL	193
Oracle Secure Backup	194
Key Vault	196
Cloud File System.....	198
Oracle VM и виртуализация	199
Audit Vault and Database Firewall	200
Oracle Linux.....	202
Oracle Recovery Manager (RMAN)	205

Средства разработки 207

Oracle Application Express (APEX)	208
Oracle SQL Developer	212



Встроенные и мобильные СУБД.....	216
Oracle Mobile Server.....	217
Berkeley DB.....	217
NoSQL Database	221
Шлюзы.....	223
Шлюзы (Oracle Gateways)	224
Новые возможности релизов СУБД Oracle.....	225
Новые возможности Oracle Database 12.1.....	226
Новые возможности СУБД Oracle Database 12.1 — 20c.....	240
Программно-аппаратные комплексы (Инженерные системы)	256
Машина баз данных Oracle Exadata	257
Облачная машина — Exadata Cloud at Customer Gen 2.....	263
Oracle Zero Data Loss Recovery Appliance.....	265
Oracle Database Appliance (ODA)	267
Продукты для управления качеством приложений	269
Oracle Application Testing Suite	270
Real User Experience Insight	273
Application Replay	277





Введение

Информация является самым мощным ресурсом, находящимся в распоряжении современной организации. Именно у организаций, способных эффективно использовать деловую информацию, наиболее велики шансы получения прибыли и выживания в суровом экономическом климате нашего времени. Эффективность использования информации целиком определяется качеством построения информационной инфраструктуры организации. Данные — это всего лишь биты и байты, размещенные в файловой системе, и только корпоративная СУБД способна превратить эти биты и байты в информацию, необходимую для правильной организации бизнеса. Корпорация Oracle предлагает передовое программное обеспечение для организации эффективного хранения, управления, интеграции и использования бизнес-информации.

Компания Oracle производит большой спектр программных и аппаратных продуктов. Это и готовые приложения (Oracle E-Business Suite, CRM и т. д.), и сервера приложений, и средства для коллективной работы, и различные СУБД, и облачные сервисы. Однако этот каталог посвящен только продуктам, входящим в семейство Oracle Database (оно описано во второй части каталога). Также здесь описаны аппаратно-программные комплексы (инженерные системы) — Exadata, ZDLRA, Oracle Cloud&Customer, которые ускоряют работу с СУБД Oracle.

Данный каталог состоит из двух частей. В первой части описаны технологии Oracle Database, которые позволяют создавать высококачественные приложения и решать наиболее типичные задачи предприятия. Т. е. в этом разделе мы идем от задач предприятий. Во второй части описаны продукты семейства Oracle Database, их функции и назначение. Если Вас интересует конкретный продукт этого семейства, вы можете найти его описание в разделе 2 каталога.



Технологии Oracle



Обзор технологий Oracle

Oracle Database: СУБД для построения корпоративных систем управления базами данных

Oracle Database — СУБД, ориентированная на применение в корпоративных сетях распределенной обработки данных (Enterprise Grid), в облачных системах (Cloud computing), а также для построения корпоративных информационных систем. Она позволяет сократить расходы на информационные технологии благодаря автоматизации управления, использованию недорогих модульных компонентов и кластеризации серверов в целях эффективного использования ресурсов.

Архитектура СУБД Oracle рассчитана на работу с огромными объемами данных и большим (десятки и сотни тысяч) числом пользователей; она демонстрирует широкие возможности обеспечения высокой готовности, производительности, масштабируемости, информационной безопасности и самоуправляемости. СУБД Oracle может быть развернута на любой платформе, начиная от небольших серверов-лезвий и заканчивая симметричными многопроцессорными компьютерами и мэйнфреймами. Уникальная способность СУБД Oracle работать со всеми типами данных, от традиционных таблиц до XML-документов и картографических данных, позволяет рассматривать ее в качестве оптимального выбора для работы с приложениями оперативной обработки транзакций, поддержки принятия решений и управления коллективной работой с информацией.

Oracle Enterprise Manager: ПО для управления базами данных, серверами приложений, облаками, приложениями и всей ИТ-инфраструктурой предприятия

Oracle Enterprise Manager (OEM) позволяет администраторам управлять работой сложных информационных систем, построенных преимущественно на основе технологий Oracle, а также включающих программные продукты других компаний. OEM оснащен широким набором функциональных возможностей и позволяет снизить стоимость и сложность управления бизнес-приложениями в распределенных вычислительных средах. Он предоставляет разнообразные функции диагностики многоуровневых интернет/интранет-приложений, инструменты тонкой настройки производительности баз данных, серверов приложений и приложений, а также полнофункциональную среду, обеспечивающую управление прикладными программными системами и облачными инфраструктурами. Возможна интеграция OEM с другими инструментами управления.

Функциональность модуля OEM Cloud Control позволяет системным администраторам управлять корпоративными информационно-вычислительными ресурсами в сетях распределенной обработки данных (Enterprise Grid) и в облаках (Cloud computing), составленных из различных типов аппаратных средств, ПО и устройств хранения данных. Oracle Cloud Control обеспечивает расширенные возможности экономического управления большим числом недорогих серверов на основе автоматизации сложных процессов управления и устранении ручных операций, потенциально приводящих к ошибкам.

Oracle Web Logic: интегрированная платформа ПО промежуточного слоя

Сегодня множество организаций во всем мире используют сервер приложений Oracle — Web Logic для реализации Java приложений с трехуровневой архитектурой (СУБД — сервер приложений — тонкий клиент), поддержания работы веб-сайтов, корпоративных порталов и транзакционных приложений. Сервер приложений Oracle Web Logic представляет собой интегрированную платформу ПО промежуточного слоя (middleware), созданную на основе сервис-ориентированной архитектуры (Service-Oriented Architecture, SOA) и технологии сетей распределенной обработки данных (Grid Computing).

После покупки компании BEA Oracle продолжает доработку и развитие Web Logic, и сегодня он по праву считается лучшим сервером приложений на рынке. Он полностью поддерживает стандарты J2EE



и технологию сетевых вычислений, оснащен встроенным ПО для создания порталов, возможностями веб-кэширования, средствами бизнес-анализа, инструментарием быстрой разработки приложений, средствами интеграции корпоративных приложений и бизнес-процессов, организации беспроводного доступа, поддержкой веб-сервисов. Организации, осуществляющие перевод своих информационных систем на модель Grid Computing на основе Oracle Web Logic, добиваются сокращения затрат на комплексы серверов приложений и административных расходов, а также отмечают улучшение управляемости своих систем.

Oracle Identity Management — средства для централизованного управления пользователями информационных систем предприятия

Oracle Identity Management (OIM) — это набор средств для централизованного управления пользователями множества СУБД, прикладных систем, серверов приложений и других программных систем предприятия. Это интегрированная инфраструктура обеспечения безопасности в системах, построенных на технологиях и программных продуктах Oracle и других поставщиков ПО.

Oracle E-Business Suite, Oracle CRM, а также другие приложения, разработанные на платформе Oracle, используют OIM в качестве комплексной инфраструктуры обеспечения информационной безопасности. Так, многие приложения использует Oracle Internet Directory (OID) для управления пользовательскими предпочтениями, персональными контактами и адресной книгой.

В целом OIM рассматривается как комплексная инфраструктура безопасности для всего стека продуктов Oracle, включая Oracle Database и Oracle Web Logic. Так, Oracle WL поддерживает сервис безопасности JAAS (Java Authentication and Authorization Service), который может быть сконфигурирован таким образом, чтобы использовать регистрационные записи (роли и пользователи), определенные в OID.

Обзор продуктов семейства Oracle Database

В семействе Oracle Database можно выделить четыре группы программных продуктов.

- Системы управления базами данных (СУБД).
- Oracle Enterprise Manager (OEM) + его компоненты и Oracle Management Cloud (OMC) как средства управления и контроля ИТ-инфраструктуры предприятия.
- Шлюзы к базам данных (Database Gateways).
- Иные продукты (например, Oracle Secure Backup, Audit Vault and Database Firewall, GoldenGate).

Серверы баз данных Oracle в свою очередь можно условно разделить на три большие группы.

- Различные редакции СУБД Oracle Database и опциональные программные компоненты (опции) для редакции Oracle Database Enterprise Edition.
- Встраиваемые СУБД (TimesTen In-Memory Database, NoSQL, Mobile Server).
- Унаследованные СУБД (RDB Enterprise Edition, CODASIL DBMS).

Каждая СУБД имеет свою технологическую нишу и предназначена для решения определенного класса задач. В настоящее время семейство продуктов Oracle Database перекрывает все потребности рынка СУБД: от систем реального времени (TimesTen In-Memory Database) до мощных коммерческих систем обработки данных (СУБД Oracle Database), от СУБД для мобильных устройств до СУБД для мейнфреймов (рис. 1).



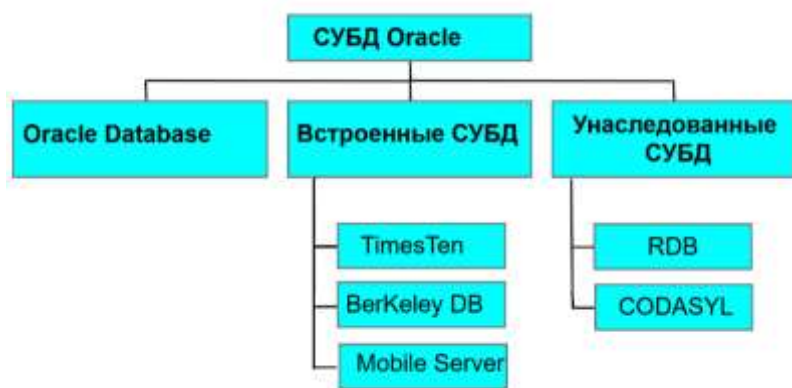


Рис. 1. СУБД семейства Oracle Database

СУБД Oracle Database

СУБД Oracle Database поставляется в нескольких редакциях, удовлетворяющих бизнес-требованиям к информационным потребностям организаций различного уровня. Oracle также предлагает несколько дополнительных продуктов, которые расширяют возможности Oracle Database Enterprise Edition для решения специальных задач. Ниже рассматриваются характеристики и функции, доступные в каждой редакции СУБД Oracle Database.

Oracle Database Standard Edition 2 (SE2)

Редакция Oracle Database Standard Edition 2 (SE2) ориентирована на организации небольшого и среднего масштаба, либо подразделения в составе крупной организации. Она может быть установлена на компьютерах малой вычислительной мощности, оснащенных не более чем двумя процессорными разъемами (сокетами). Каждая база данных Standard Edition 2 (SE2) может использовать до 16 потоков. И, поскольку большинство заказчиков располагают несколькими базами данных на одном 2-сокетном сервере для полноценного использования вычислительного потенциала CPU, лицензионные обязательства заказчика не зависят от того, сколько баз данных работает на одном сервере – необходима лицензия только на 2 сокета.

Несмотря на то, что Oracle Database SE2 позиционируется как сервер уровня рабочей группы, по своим возможностям и своему составу эта редакция СУБД Oracle может выступать и как центральный сервер баз данных в небольшой организации.

В то же время, ограниченные возможности параллельной обработки, недоступность опций БД, трудности в создании систем с высокой отказоустойчивостью могут потребовать в будущем перехода к Oracle Database Enterprise Edition, при этом достаточно только обновить управляющее программное обеспечение и выполнить скрипт обновления метаданных СУБД.

Oracle Database Enterprise Edition (EE)

Редакция Oracle Database Enterprise Edition (EE) обеспечивает эффективное, надежное, безопасное управление данными для решения критически важных задач в режиме оперативной обработки транзакций, для хранилищ данных и аналитических систем, для управления контентом и веб-приложениями.

Эта редакция реализует всю вычислительную и функциональную мощь Oracle Database. Она может работать на большинстве вычислительных платформ, поддерживать десятки тысяч пользователей и обрабатывать многотерабайтные хранилища данных, используя все процессорные ресурсы сервера или кластера серверов в режиме параллельной обработки. Oracle Database EE предназначена для создания информационных систем



среднего и крупного масштаба с высокими требованиями к надежности, доступности, быстродействию, безопасности, управляемости и масштабируемости.

Помимо функциональности, включенной непосредственно в редакцию Oracle Database EE, возможно использование дополнительных программных компонентов-опций, расширяющих функциональность сервера баз данных (например, для повышения безопасности систем или для систем повышенной надежности). Oracle Database EE может использоваться на отдельных серверах или в кластерных конфигурациях без ограничения на число процессорных разъемов.

Oracle Database Personal Edition (PE)

Oracle Database Personal Edition (PE) предоставляет одному пользователю средства разработки и развертывания приложений, которые требуют полной совместимости с базами данных Oracle Database SE2 и EE. Для этого в Oracle Database PE реализована вся функциональность Oracle Database Enterprise Edition, включая опциональные компоненты (за исключением опции Real Applications Clusters). Oracle Database PE работает только в среде MS Windows.

Oracle Database Express Edition

Редакция Oracle Database Express Edition (XE) используется для баз данных начального уровня, занимающих небольшой объем. Она основана на коде старших редакций, бесплатна для разработки приложений, развертывания и распространения, является простой в установке и администрировании. Oracle Database XE работает только в среде MS Windows и Linux и используется в основном для ознакомительных целей или обучения.

Oracle Database XE может быть установлена на компьютер с любым количеством процессоров и любым объемом дискового пространства, но редакция XE позволяет хранить не более 12 Гб пользовательских данных, использовать до 2 Гб оперативной памяти и только два логических ядра процессора из установленных на машине. Кроме того, полноценная техническая поддержка для СУБД этой редакции недоступна. В то же время существует возможность простого обновления редакции XE до SE2 или EE без применения дорогостоящих и сложных процедур миграции.

Oracle Database Enterprise Edition на инженерных системах

Oracle Database Enterprise Edition on Engineered Systems (EE-ES) представляет собой установленную на инженерных системах (Oracle Exadata или Oracle Database Appliance) СУБД Oracle Database Enterprise Edition. Помимо стандартного встроенного функционала Oracle Database EE, на инженерных системах доступны дополнительные функциональные возможности.

Сервисы СУБД Oracle в облаке

Кроме использования СУБД Oracle на компьютерах в ЦОД заказчика (ОРЕХ), можно заказать СУБД Oracle в публичном облаке Oracle как сервис (CAPEX). В облаке также можно заказать автономные БД Oracle (АТР, ADW, Autonomous JSON DB) (см. ниже). Существуют следующие сервисы.

Oracle Database Standard Edition в облаке Oracle Cloud

Oracle Database Cloud Service Standard Edition (DBCS SE) представляет собой сервис базы данных Oracle Database Standard Edition, доступный в облачной инфраструктуре Oracle Cloud.

Oracle Database Enterprise Edition в облаке Oracle Cloud

Oracle Database Cloud Service Enterprise Edition (DBCS EE) представляет собой сервис базы данных Oracle Database Enterprise Edition, доступный в облачной инфраструктуре Oracle Cloud.



Oracle Database Cloud Service Enterprise Edition High Performance

Oracle Database Cloud Service Enterprise Edition High Performance (DBCS EE-HP) представляет собой сервис базы данных Oracle Database Enterprise Edition вместе с некоторыми опциями (Database Options) и пакета управления (Management Packs), доступный в облачной инфраструктуре Oracle Cloud.

Oracle Database Cloud Service Enterprise Edition Extreme Performance

Oracle Database Cloud Service Enterprise Edition Extreme Performance (DBCS EE-EP) представляет собой сервис базы данных Oracle Database Enterprise Edition вместе со всеми опциями (Database Options) и пакетами управления (Management Packs), доступный в облачной инфраструктуре Oracle Cloud.

Oracle Database Exadata Cloud Service

Oracle Database Exadata Cloud Service представляет собой сервис базы данных Oracle Database Enterprise Edition вместе со всеми опциями (Database Options) и пакетами управления (Management Packs), установленный на инженерной системе Exadata и доступный в облаке Oracle.

Также этот сервис может быть доступен на площадке заказчика, если он имеет подписку на инженерную систему Oracle Exadata Cloud at Customer.

Отличия редакций и сервисов СУБД Oracle

Анализ отличий основных редакций Oracle Database можно найти в таблице 1. Более подробно Oracle Database Enterprise Edition и ее опциональные компоненты описаны ниже.

Таблица 1. Отличия редакций Oracle Database

Функция или опция	SE2	EE	EE-ES	DBCS SE	DBCS EE	DBCS EE-HP	DBCS EE-EP	ExaCS	Замечания
Консолидация									
Oracle Multitenant — количество PDB (подключаемых БД)	3	252	4096	3	3	4096	4096	4096	Числовое значение в каждом столбце указывает максимальное количество подключаемых баз данных (PDB), которые можно создать в каждой редакции. Для всех редакций, если нет лицензии на Oracle Multitenant, может быть до 3 PDB, созданных пользователем, в данной контейнерной базе данных в любое время. EE: опция за дополнительную плату; если есть лицензия на Oracle Multitenant, то можно создать до 252 PDB. EE-ES: опция за дополнительную плату; если есть лицензия на Oracle Multitenant, то можно создать до 4096 PDB.



									DBCS EE-HP, DBCS EE-EP и ExaCS: опция в комплекте; можно создать до 4096 PDB.
CDB Fleet Management	N	N	Y	N	Y	Y	Y	Y	
PDB Snapshot Carousel	N	N	Y	N	Y	Y	Y	Y	
Refreshable PDB switchover	N	N	Y	Y	Y	Y	Y	Y	
Инструменты для разработки									
SQLJ	Y	Y	Y	Y	Y	Y	Y	Y	Требуется наличие ПО Oracle Programmer.
Oracle Developer Tools for Visual Studio .NET	Y	Y	N/A	N/A	N/A	N/A	N/A	N/A	Поддерживается только на ОС Windows.
Microsoft Distributed Transaction Coordinator support	Y	Y	N/A	N/A	N/A	N/A	N/A	N/A	Поддерживается только на ОС Windows.
Native .NET Data Provider — ODP.NET	Y	Y	N/A	N/A	N/A	N/A	N/A	N/A	Поддерживается только на ОС Windows.
.NET Stored Procedures	Y	Y	N/A	N/A	N/A	N/A	N/A	N/A	Поддерживается только на ОС Windows.
Отказоустойчивость									
Application Continuity	N	Y	Y	N	N	N	Y	Y	EE и EE-ES: требуется опция Active Data Guard или Real Application Cluster.
Oracle Sharding	N	Y	Y	N	Y	Y	Y	Y	EE и EE-ES: нет ограничений на количество шардов или резервных шардов, если каждый шард имеет лицензию на Active Data Guard, Real Application Cluster или GoldenGate. Без лицензии на одну из этих опций можно использовать три шарда с резервными шардами с обычным Data Guard.
Oracle Failsafe	Y	Y	N	N	N	N	N	N	Поддерживается только на ОС Windows.
Oracle RAC One Node	N	Y	Y	N	N	N	N	Y	EE и EE-ES: отдельная платная опция.
Standard Edition High Availability	Y	N	N	N	N	N	N	N	
Oracle Data Guard — Redo Apply	N	Y	Y	N	Y	Y	Y	Y	
Oracle Data Guard — Far Sync Standby	N	Y	Y	N	N	N	Y	Y	EE и EE-ES: требует наличия опции Oracle Active Data Guard.
Oracle Data Guard — SQL Apply	N	Y	Y	N	Y	Y	Y	Y	
Oracle Data Guard — Snapshot Standby	N	Y	Y	N	Y	Y	Y	Y	
Oracle Data Guard — Real-Time	N	Y	Y	N	N	N	Y	Y	EE и EE-ES: требует наличия



Cascading Standbys									опции Oracle Active Data Guard.
Oracle Data Guard — Automatic Correction of Non-logged Blocks at a Data Guard Standby Database	N	N	Y	N	N	N	Y	Y	EE-ES: требует наличия опции Oracle Active Data Guard.
Oracle Active Data Guard	N	Y	Y	N	N	N	Y	Y	EE и EE-ES: отдельная опция.
Rolling Upgrades — Patch Set, Database, and Operating System	N	Y	Y	N	Y	Y	Y	Y	
Rolling Upgrade Using Active Data Guard	N	Y	Y	N	N	N	Y	Y	EE-ES: требует наличия опции Oracle Active Data Guard.
Online index rebuild	N	Y	Y	N	Y	Y	Y	Y	
Online table reorganization	N	Y	Y	N	Y	Y	Y	Y	Операция <i>ALTER TABLE... MOVE ONLINE</i>
Online table redefinition	N	Y	Y	N	Y	Y	Y	Y	Использование пакета <i>DBMS_REDEFINITION</i>
Duplexed backup sets	N	Y	Y	N	Y	Y	Y	Y	
Block change tracking for fast incremental backup	N	Y	Y	N	N	N	Y	Y	
Unused block compression in backups	N	Y	Y	N	Y	Y	Y	Y	
Block-level media recovery	N	Y	Y	N	Y	Y	Y	Y	
Lost Write Protection	N	Y	Y	N	Y	Y	Y	Y	
Automatic Block Repair	N	Y	Y	N	N	N	Y	Y	EE-ES: требует наличия опции Oracle Active Data Guard.
Parallel backup and recovery	N	Y	Y	N	Y	Y	Y	Y	
Tablespace point-in-time recovery	N	Y	Y	N	Y	Y	Y	Y	
Trial recovery	N	Y	Y	N	Y	Y	Y	Y	
Fast-start fault recovery	N	Y	Y	N	Y	Y	Y	Y	
Flashback Table	N	Y	Y	N	Y	Y	Y	Y	
Flashback Database	N	Y	Y	N	Y	Y	Y	Y	
Flashback Transaction	N	Y	Y	N	Y	Y	Y	Y	
Flashback Transaction Query	N	Y	Y	N	Y	Y	Y	Y	
Optimization for Flashback Data Archive	N	Y	Y	N	N	Y	Y	Y	Базовая функциональность Flashback Data Archive доступна для всех редакций. EE и EE-ES: оптимизации для Flashback Data Archive требуют наличия опции Oracle Advanced Compression.
Online Datafile Move	N	Y	Y	N	Y	Y	Y	Y	
Transaction Guard	N	Y	Y	N	Y	Y	Y	Y	
Cross-platform Backup and Recovery	N	Y	Y	N	Y	Y	Y	Y	



Global Data Services	N	Y	Y	N	N	N	Y	Y	EE-ES: требует наличия опции Oracle Active Data Guard.
Recovering tables and table partitions from RMAN backups									
Интеграция									
Database Gateways	Y	Y	Y	Y	Y	Y	Y	Y	Database Gateway for ODBC, включая все редакции. Все шлюзы Database Gateway доступны для всех редакций, но требуют отдельных лицензий.
Messaging Gateway	N	Y	Y	N	Y	Y	Y	Y	
Sharded Queues	N	Y	Y	N	N	Y	Y	Y	EE-ES: требует наличия опции Oracle Partitioning.
Управляемость									
Oracle Cloud Management Pack for Oracle Database	N	Y	Y	N	N	Y	Y	Y	EE и EE-ES: отдельная опция.
Oracle Data Masking and Subsetting Pack	N	Y	Y	N	Y	Y	Y	Y	EE и EE-ES: отдельная опция.
Oracle Database Lifecycle Management Pack for Oracle Database	N	Y	Y	N	N	Y	Y	Y	EE и EE-ES: отдельная опция.
Oracle Diagnostics Pack	N	Y	Y	N	Y	Y	Y	Y	EE и EE-ES: отдельная опция.
Oracle Tuning Pack	N	Y	Y	N	Y	Y	Y	Y	EE и EE-ES: отдельная опция, также требуется Oracle Diagnostics Pack.
Oracle Real Application Testing	N	Y	Y	N	Y	Y	Y	Y	EE и EE-ES: отдельная опция.
Database Resource Manager	N	Y	Y	N	Y	Y	Y	Y	
I/O Resource Management	N	N	Y	N	N	N	N	Y	
Instance Caging	Y	Y	Y	Y	Y	Y	Y	Y	
SQL Plan Management	Y	Y	Y	Y	Y	Y	Y	Y	SE2 и DBCS SE: доступен только один SQL plan baseline для SQL-запроса, SQL plan evolution отключен.
Automatic SQL Plan Management	N	N	Y	N	N	N	N	Y	EE-ES: доступен на Exadata. Недоступен на Oracle Database Appliance.
SQL Tuning Sets	N	Y	Y	N	Y	Y	Y	Y	
Oracle Fleet Patching and Provisioning (formerly Rapid Home Provisioning)	N	Y	Y	N/A	N/A	N/A	N/A	N/A	EE и EE-ES: когда развертывание производится на несколько машин, каждый сервер должен иметь одну из лицензий: Oracle Database Lifecycle Management Pack, Oracle RAC или Oracle RAC One Node.



Работа в сети									
Infiniband Support	N	Y	Y	N	N	N	N	Y	
Oracle Connection Manager	N	Y	Y	N	Y	Y	Y	Y	
Network Compression	N	Y	Y	N	N	Y	Y	Y	EE и EE-ES: требует наличия опции Oracle Advanced Compression.
Производительность									
Automatic Indexing	N	N	Y	N	N	N	N	Y	EE-ES: доступен на Exadata. Недоступен на Oracle Database Appliance.
SQL Quarantine	N	N	Y	N	N	N	N	Y	EE-ES: доступен на Exadata. Недоступен на Oracle Database Appliance.
Real-Time Statistics	N	N	Y	N	N	N	N	Y	EE-ES: доступен на Exadata. Недоступен на Oracle Database Appliance.
High-Frequency Automatic Optimizer Statistics Collection	N	N	Y	N	N	N	N	Y	EE-ES: доступен на Exadata. Недоступен на Oracle Database Appliance.
High-Frequency Automatic SPM Evolve Advisor Task	N	N	Y	N	N	N	N	Y	EE-ES: доступен на Exadata. Недоступен на Oracle Database Appliance.
Client Side Query Cache	N	Y	Y	N	Y	Y	Y	Y	
Query Results Cache	N	Y	Y	N	Y	Y	Y	Y	
PL/SQL Function Result Cache	N	Y	Y	N	Y	Y	Y	Y	
Oracle TimesTen Application-Tier Database Cache	N	Y	Y	N	N	N	N	N	EE и EE-ES: отдельная платная опция.
Database Smart Flash Cache	N	Y	Y	N/A	N/A	N/A	N/A	N/A	EE: поддерживается только на ОС Solaris и Oracle Linux. EE-ES: доступен на Oracle Client-Side Database Appliance. Недоступен на Exadata
Support for Oracle Exadata Storage Server Software	N	N	Y	N	N	N	N	Y	EE-ES: доступен на Exadata. Недоступен на Oracle Database Appliance.
Adaptive Execution Plans	N	Y	Y	N	Y	Y	Y	Y	
Concurrent Execution of UNION and UNION ALL Branches	N	Y	Y	N	Y	Y	Y	Y	
Oracle Database In-Memory	N	Y	Y	N	N	N	Y	Y	EE и EE-ES: отдельная платная опция.
In-Memory Column Store	N	Y	Y	N	N	N	Y	Y	EE и EE-ES: требует наличия опции Oracle Database In-Memory.
In-Memory Column Cache on Storage Servers	N	N	Y	N	N	N	N	Y	EE-ES: доступна на Exadata и требует наличия опции



									Oracle Database In-Memory. Недоступна на Oracle Database Appliance. Требуется Exadata, Supercluster или Exadata Cloud Service.
In-Memory Column Store on Active Data Guard standby	N	N	Y	N	N	N	Y	Y	EE-ES: доступна на Exadata и требует наличия опций Oracle Database In-Memory и Oracle Active Data Guard. Недоступна на Oracle Database Appliance.
Fault Tolerant In-Memory Column Store	N	N	Y	N	N	N	N	Y	EE-ES: доступна на Exadata и Infiniband-based Oracle Database Appliance. Требуется наличия опций Oracle Database In-Memory и Oracle Real Application Cluster.
Automatic In-Memory Column Store	N	N	Y	N	N	N	Y	Y	
In-Memory Column Store support for External Tables	N	N	Y	N	N	N	Y	Y	EE-ES: доступен на Exadata. Недоступен на Oracle Database Appliance.
In-Memory Aggregation	N	Y	Y	N	N	N	Y	Y	EE и EE-ES: требует наличия опции Oracle Database In-Memory.
Database In-Memory Base Level	N	Y	Y	N	Y	Y	Y	Y	Дает возможность поэкспериментировать с технологией Oracle Database In-Memory без покупки опции Oracle Database In-Memory. Имеет ограничение: размер области колоночного хранения (параметр INMEMORY_SIZE) не превышает 16 Гб CDB; в кластерной конфигурации Oracle RAC ограничение 16 Гб действует для каждого узла.
Memoptimized Rowstore	N	N	Y	N	N	N	Y	Y	EE-ES: доступен на Exadata. Недоступен на Oracle Database Appliance.
Attribute Clustering	N	Y	Y	N	Y	Y	Y	Y	
Zone Maps	N	N	Y	N	N	N	N	Y	EE-ES: доступен на Exadata и требует наличия опции Oracle Partitioning. Недоступен на Oracle Database Appliance.
Масштабируемость									
Oracle Real Application Clusters (Oracle RAC)	N	Y	Y	N	N	N	Y	Y	EE и EE-ES: отдельная опция.



Automatic Workload Management	Y	Y	Y	N	N	N	Y	Y	EE и EE-ES: требует наличия опции Oracle Real Application Cluster.
Quality of Service Management	Y	Y	Y	N	N	N	Y	Y	EE и EE-ES: требует наличия опции Oracle Real Application Cluster.
Oracle NoSQL Database Enterprise Edition	N	Y	Y	N/A	N/A	N/A	N/A	N/A	
Защита данных									
Column-Level Encryption	N	Y	Y	N	N	Y	Y	Y	EE и EE-ES: требует наличия опции Oracle Advanced Security.
Tablespace Encryption	N	Y	Y	Y	Y	Y	Y	Y	EE и EE-ES: требует наличия опции Oracle Advanced Security.
Oracle Advanced Security	N	Y	Y	N	N	Y	Y	Y	EE и EE-ES: отдельная опция.
Oracle Database Vault	N	Y	Y	N	N	Y	Y	Y	EE и EE-ES: отдельная опция.
Oracle Label Security	N	Y	Y	N	N	Y	Y	Y	EE и EE-ES: отдельная опция.
Enterprise User Security	N	Y	Y	N	Y	Y	Y	Y	
Centrally Managed Users	N	Y	Y	N	Y	Y	Y	Y	EE, EE-ES, DBCS EE, DBCS EE-HP, DBCS EE-EP, ExaCS: аутентификация и авторизация пользователей БД с помощью Microsoft Active Directory.
Fine-grained Auditing	N	Y	Y	N	Y	Y	Y	Y	
Privilege Analysis	N	Y	Y	N	Y	Y	Y	Y	
Real Application Security	N	Y	Y	N	Y	Y	Y	Y	
Redaction	N	Y	Y	N	N	Y	Y	Y	EE и EE-ES: требует наличия опции Oracle Advanced Security.
Transparent Sensitive Data Protection	N	Y	Y	N	Y	Y	Y	Y	
Virtual Private Database	N	Y	Y	N	Y	Y	Y	Y	
Keystore for Each Pluggable Database	N	N	Y	N	Y	Y	Y	Y	
Работа с геоинформацией									
Oracle Spatial and Graph	Y	Y	Y	Y	Y	Y	Y	Y	Не требует отдельной лицензии.
Property Graph and RDF Graph Technologies (RDF/OWL)	Y	Y	Y	Y	Y	Y	Y	Y	Не требует отдельной лицензии.
Parallel spatial index builds	N	Y	Y	N	Y	Y	Y	Y	
Multimaster replication of SDO_GEOMETRY objects	N	Y	Y	N	Y	Y	Y	Y	



Partitioned spatial indexes	N	Y	Y	N	N	Y	Y	Y	EE и EE-ES: требуется наличие опции Oracle Partitioning.
Хранилища данных и BI									
Oracle Partitioning	N	Y	Y	N	N	Y	Y	Y	EE и EE-ES: отдельная опция.
Oracle On-Line Analytical Processing (OLAP)	N	Y	Y	N	N	Y	Y	Y	EE и EE-ES: отдельная опция.
Oracle Machine Learning (formerly Advanced Analytics)	Y	Y	Y	Y	Y	Y	Y	Y	Теперь не требует отдельного лицензирования.
Oracle Advanced Compression	N	Y	Y	N	N	Y	Y	Y	EE и EE-ES: отдельная опция.
Advanced Index Compression	N	Y	Y	N	N	Y	Y	Y	EE и EE-ES: требует наличия опции Oracle Advanced Compression.
Prefix Compression (also called Key Compression)	N	Y	Y	N	Y	Y	Y	Y	
Hybrid Columnar Compression	N	Y	Y	N	Y	Y	Y	Y	EE: требует систему хранения ZFS SA, Axiom или FS1.
Hybrid Columnar Compression Row-Level Locking	N	Y	Y	N	N	N	N	Y	EE: требует систему хранения ZFS SA или FS1 и опцию Oracle Advanced Compression. EE-ES: требует наличия опции Oracle Advanced Compression.
Exadata Flash Cache Compression	N	N	Y	N	N	N	N	Y	EE-ES: доступна на Exadata и требует опции Oracle Advanced Compression. Недоступна на Oracle Database Appliance.
Heat Map	N	Y	Y	N	N	Y	Y	Y	EE и EE-ES: требует наличия опции Oracle Advanced Compression или Oracle Database In-Memory.
Automatic Data Optimization	N	Y	Y	N	N	Y	Y	Y	EE и EE-ES: требует наличия опции Oracle Advanced Compression или Oracle Database In-Memory.
Basic Table Compression	N	Y	Y	N	Y	Y	Y	Y	
Deferred Segment Creation	N	Y	Y	N	Y	Y	Y	Y	
Bitmapped index, bitmapped join index, and bitmap plan conversions	N	Y	Y	N	Y	Y	Y	Y	
Parallel query/DML	N	Y	Y	N	Y	Y	Y	Y	
Parallel statistics gathering	N	Y	Y	N	Y	Y	Y	Y	
Parallel index build/scans	N	Y	Y	N	Y	Y	Y	Y	
Parallel Data Pump Export/Import	N	Y	Y	N	Y	Y	Y	Y	
In-Memory Parallel Execution	N	Y	Y	N	Y	Y	Y	Y	
Parallel Statement Queuing	N	Y	Y	N	Y	Y	Y	Y	
Parallel capture and apply via	N	Y	Y	N	Y	Y	Y	Y	



XStream									
Transportable tablespaces, including cross-platform and full transportable export and import	N	Y	Y	N	Y	Y	Y	Y	
Summary management — Materialized View Query Rewrite	N	Y	Y	N	Y	Y	Y	Y	

СУБД Oracle Database

Oracle Database — это объектно-реляционная система управления базами данных, являющаяся результатом более чем 40-летней деятельности корпорации Oracle в области технологий баз данных.

СУБД Oracle в одинаковой степени оптимизирована и для приложений оперативной обработки транзакций, и для аналитических приложений. На практике это означает, что один и тот же продукт (например, Oracle Database Enterprise Edition) можно с успехом использовать и как сервер оперативных баз данных, обрабатывающий интенсивный поток относительно простых и коротких транзакций, поступающих от множества пользователей, так и в качестве сервера хранилища данных, который позволяет концентрировать большие объемы данных и выполнять над ними сложные аналитические вычисления.

Работает на всех платформах

Одной из основных характеристик СУБД Oracle Database является функционирование системы на большинстве программно-аппаратных платформ, в том числе на больших ЭВМ, UNIX-серверах, персональных компьютерах и т. д. Другой важной характеристикой является поддержка Oracle всех возможных вариантов программно-аппаратных архитектур, в том числе симметричных многопроцессорных систем, кластеров, систем с массовым параллелизмом, архитектур мейнфреймов. Очевидна значимость этих характеристик для современных организаций, где эксплуатируется множество компьютеров различных моделей и производителей. В таких условиях фактором успеха является максимально возможная типизация предлагаемых решений, ставящая своей целью существенное снижение стоимости владения программным обеспечением. Унификация систем управления базами данных — один из наиболее значимых шагов на пути достижения этой цели.

Поддержка Oracle большинства популярных компьютерных платформ и архитектур достигается за счет жесткой технологической схемы разработки кода СУБД. Разработку серверных продуктов выполняет единое подразделение корпорации Oracle, изменения вносятся централизованно, после чего все версии подвергаются тщательному тестированию на базовой платформе, а затем переносятся на все остальные платформы, где также детально проверяются. Возможность переноса Oracle Database обеспечивается специфической структурой исходного программного кода сервера баз данных. Приблизительно 80 % программного кода Oracle — это код на языке программирования C, который (с известными ограничениями) является платформонезависимым. Примерно 20 % кода, представляющего собой ядро СУБД, реализовано на машинно зависимых языках, и эта часть кода перерабатывается для различных платформ.

СУБД Oracle Database скрывает от пользователя детали реализации механизмов управления данными на каждой из платформ, что дает основание говорить о практически полной унификации базового программного обеспечения. Дополнительно к этому архитектура Oracle позволяет переносить прикладные системы, реализованные на одной платформе, на другие платформы без изменений как в структурах баз данных, так и исходных кодов серверной части приложений.



Oracle Database обеспечивает лучшую производительность и масштабируемость, чем любая другая база данных в мире. Модель многоверсионного согласованного чтения, разработанная компанией Oracle, гарантирует самую высокую пропускную способность. В сервере баз данных Oracle Database операции записи не блокируют операции чтения и операции чтения не блокируют операции записи. Сервер Oracle Database всегда автоматически блокирует только те записи, которые изменяются. При его использовании, в отличие от других СУБД, невозможно такое неприятное явление, как эскалация блокировки с уровня строки до уровня таблицы, приводящее к падению производительности из-за того, что пользователи, желающие поменять другие строки таблицы, будут вынуждены ждать завершения транзакции, заблокировавшей таблицу.

Управляет всеми вашими данными

Максимальный размер одной базы данных Oracle — 8 экзбайт, или 8 миллионов терабайт, в подавляющем числе случаев этого вполне достаточно, чтобы хранить всю вашу информацию.

СУБД Oracle Database опирается на стандарт SQL-3, позволяющий описывать определения новых типов объектов, состоящих из атрибутов (скалярных, то есть других типов, множеств объектов, ссылок на объекты) и обладающих ассоциированными с ним методами. Любая колонка таблицы может быть любого типа, поддерживаются также вложенные таблицы и массивы объектов переменной длины.

СУБД Oracle Database не просто предоставляет расширенный набор встроенных типов данных, но и позволяет конструировать новые типы данных со спецификацией методов доступа к ним. Это означает фактически, что разработчики получают не просто систему для хранения и обработки атрибутивных данных в виде таблиц, а инструмент, позволяющий строить структурированные типы данных, непосредственно отображающие сущности предметной области.

Одна из отличительных особенностей Oracle — возможность хранения и обработки различных предопределенных типов данных. Данная функциональность интегрирована в ядро СУБД и поддерживается модулем Multimedia в составе Oracle Database. Он обеспечивает работу с текстовыми документами, включая различные виды поиска, в том числе контекстного, работу с графическими образами более 20 форматов, работу с аудио- и видеоинформацией. Модуль Spatial and Graph предназначен для работы с пространственными данными и служит надежной основой для создания геоинформационных систем.

Для XML-документов в Oracle Database существует специальный тип данных XMLTYPE, работа с которым осуществляется при помощи модуля XML DB. XML DB, в частности, позволяет создавать XML-документы из реляционных таблиц. Результат любого SQL-запроса может быть преобразован в XML. Oracle предоставляет пять инструментальных наборов для работы с XML (XDK), доступных для Java, C, C++, PL/SQL и Java Beans. Oracle Database — первая промышленная СУБД, в которой реализована поддержка языка XQuery.

СУБД Oracle Database также поддерживает хранение и обработку документов в формате JSON (JavaScript Object Notation).

Интегрирует всю вашу информацию

СУБД Oracle Database позволяет эффективно работать с распределенными базами данных Oracle и обеспечивает двухфазную фиксацию транзакций к нескольким базам данных. Работа с другими базами данных может быть реализована через ODBC с помощью механизма Heterogeneous Services или через специализированные для каждой базы данных шлюзы Database Gateways.

Модуль Transactional Event Queues (TEQ) — это встроенный в Oracle Database механизм хранения и обработки очередей сообщений. Компонент TEQ относится к классу MOM (Message Oriented Middleware). Наличие такого компонента позволяет построить на базе СУБД полнофункциональную инфраструктуру для обработки сообщений и исключает необходимость приобретения для этой цели дополнительных средств третьих фирм (таких как IBM MQ Series), обеспечивая в то же время связь с ними в неоднородных средах за счет продукта Oracle Messaging Gateway.



TEQ обеспечивает асинхронный режим обмена сообщениями между приложениями. TEQ предлагает два режима рассылки сообщений: одно ко многим (point-to-multipoint) и публикация-подписка (publish-subscribe). AQ позволяет устанавливать приоритеты сообщениям и задавать их порядок в очереди (FIFO или на основе приоритета), группировать сообщения, определять правила доставки и время жизни сообщений, автоматически преобразовывать формат сообщений, получать по e-mail асинхронные уведомления о прибытии интересующего сообщения, передавать сообщения по протоколу HTTP(S).

Поддерживает широкий спектр интерфейсов программирования

Разработчику приложений Oracle предлагает широкий выбор программных средств для создания приложений, работающих с базой данных Oracle. Вы можете реализовать логику вашего приложения целиком в базе данных, используя мощные возможности СУБД Oracle: языки программирования PL/SQL и Java, триггеры, ограничения целостности данных, хранимые процедуры: внутренние, написанные на языках PL/SQL, Java, и внешние, написанные на любом из языков, которые вы предпочитаете: C, C++, Pascal, Fortran и т. д.

PL/SQL — платформонезависимый процедурный язык для транзакционной обработки данных, тесно интегрированный с SQL. В нормальном режиме программные модули PL/SQL выполняются виртуальной машиной PL/SQL, то есть интерпретатором. PL/SQL-процедуры могут быть скомпилированы в двоичный код платформы, на которой работает СУБД Oracle Database. В этом случае они будут исполняться напрямую без интерпретатора, что позволяет увеличить скорость их выполнения в несколько раз.

Oracle Call Interface поддерживает разработку программ с применением вызовов низкоуровневых функций на языке C для доступа к базам данных. Это позволяет создавать эффективные программы, требующие минимальных ресурсов. Возможность разработки оптимизированных по скорости и используемой памяти приложений достигается за счет использования вызовов функций, которые предоставляют полный контроль за выполнением операторов SQL и PL/SQL. Компонент OLE DB предоставляет возможность доступа к базам данных Oracle приложений, разработанных на C++, Microsoft Visual Basic, OLE 2.0. Полная поддержка языка макроопределений в Visual Basic позволяет получать данные из баз данных Oracle непосредственно в электронных таблицах Microsoft Excel.

Oracle Data Provider для программной платформы .NET (ODP.NET) обеспечивает оптимизированный доступ к данным для библиотеки ADO.NET в базе данных Oracle. ODP.NET позволяет разработчикам использовать преимущества расширенной функциональности базы данных Oracle, включая Real Application Cluster, самонастраивающийся кэш операторов и механизма Fast Connection Failover. Существует три типа драйверов ODP.Net: Managed Driver, ODP.NET Unmanaged Driver и ODP.NET Core. ODP.NET Managed Driver полностью разработан на .NET. Разработчики развертывают одну сборку в пакете развертывания размером менее 10 МБ. ODP.NET Unmanaged Driver содержит больше функций, чем управляемый драйвер, поскольку он может получить доступ к функциям, встроенным в клиент базы данных Oracle Client. ODP.NET Core разработан для многоплатформенных приложений .NET Core.

ODP.NET поддерживает Entity Framework Core, .NET Framework 4.8 и .NET Core 3.x.

Oracle Database снабжен всеми необходимыми средствами для сетевого подключения клиентских рабочих мест, для обеспечения работы клиентов по технологии OLE (модуль OLE DB), набором ODBC-драйверов (ODBC Driver) и библиотеками для разработки программ на языках третьего уровня, использующих для доступа к базе данных Oracle Call Level Interface (OCI).

Instant Client — это простейший способ развернуть полное клиентское приложение Oracle. В отличие от стандартной установки Oracle Client, насчитывающей сотни файлов, Instant Client состоит всего из трех библиотек, обеспечивающих полноценную работу клиентского приложения с базой данных Oracle.

Oracle Application Express — средство быстрой разработки веб-приложений, работающих с базой данных. Для создания и эксплуатации приложений Application Express на клиентском месте ничего не требуется,



кроме веб-браузера. Сами приложения хранятся и исполняются внутри базы данных Oracle. Application Express предназначен для непрограммистов. Простой интерфейс и множество шаблонов позволяют создавать веб-приложения буквально на лету.

Надежно защищает данные

Oracle Database предоставляет уникальные возможности по обеспечению высокой отказоустойчивости ваших приложений. Технология **Real Application Clusters** обеспечивает одновременную работу базы данных на многих узлах кластера и служит основой для создания отказоустойчивых приложений, защищенных от аппаратных сбоев и плановых простоев.

Oracle Database имеет широкий набор возможностей для защиты ваших данных от потерь.

Технология ASM (**Automatic Storage Management**) поддерживает двойное и тройное зеркалирование данных, защищая их от сбоев жесткого диска.

Автоматическая процедура резервирования базы данных на жесткий диск гарантирует, что резервные копии, необходимые для восстановления базы данных, всегда будут под рукой, исключает вероятность ошибок администратора и улучшает время восстановления базы данных. Быстрое инкрементальное резервирование базы данных, не требующее ее полного сканирования, с последующим применением к полной резервной копии позволяет значительно сократить время и дисковое пространство, необходимые для ежедневных оперативных операций резервирования базы данных.

Уникальные технологии **Flashback** СУБД Oracle позволяют администраторам легко диагностировать и исправлять ошибки оператора, связанные с удалением таблиц, неправильными транзакциями и изменениями отдельных строк, нескольких таблиц или всей базы данных. В Oracle Database время, необходимое для исправления ошибки оператора, равно или даже меньше времени, за которое эта ошибка произошла.

Oracle Database включает в себя технологию **Data Guard** для защиты баз данных от последствий катастроф, таких как землетрясение или наводнение. Используя Data Guard, вы можете иметь несколько удаленных копий основной базы данных, синхронизирующихся с ней автоматически, и одним нажатием кнопки мыши переводить обработку данных с основной базы на резервную. Опция **Active Data Guard** позволяет одновременно с резервированием использовать синхронные копии базы данных для чтения данных. Это дает возможность разгрузить основную базу данных и более полно утилизировать имеющиеся вычислительные ресурсы. Технология **Fast-Start Failover** обеспечивает в случае сбоя основной базы данных автоматическое переключение на резервную базу данных без участия администратора.

Oracle Database спроектирована таким образом, чтобы работа критически важных приложений не зависела от изменений в конфигурации базы данных. Для того чтобы сервер Oracle Database начал использовать добавленную оперативную память или добавленный жесткий диск, вам не надо перезапускать ваши системы. Многие операции по реорганизации данных, такие как создание новых индексов, добавление новых столбцов в таблицу, удаление столбцов и др., можно осуществлять в базе данных без ограничения доступа пользователей к реорганизуемым данным. Более того, Oracle Database поддерживает пошаговое обновление базы данных и операционной системы (Rolling Upgrade) без остановки работы пользователей, обеспечивая реальную работу ваших приложений в режиме 24/7.

За прошедшее десятилетие Oracle Database прошла множество независимых сертификаций на соответствие требованиям безопасности и является самой защищенной промышленной базой данных в мире. Oracle Database обеспечивает ограничение доступа к данным не только на уровне таблиц и столбцов, но и на уровне строк. Программная опция **Advanced Security** позволяет шифровать данные, передаваемые по сети. Табличные данные или целые табличные пространства могут храниться на диске в зашифрованном виде. В случае кражи базы данных или ее резервной копии злоумышленникам не удастся расшифровать секретные



данные с помощью каких-либо средств взлома. Процесс шифрования и дешифрования данных полностью прозрачен для программных приложений и не требует их переделки.

СУБД Oracle развивается и совершенствуется уже около 40 лет и остается выбором номер один на рынке баз данных. Использование базы данных Oracle не только позволяет сразу получить экономический эффект, но и гарантирует эффективное использование ваших вложений в будущем.

Oracle DB — конвергентная и мультимодельная СУБД

Современные приложения работают с данными разных типов и форматов (структурированные, неструктурированные, полуструктурированные), используют различные алгоритмы и технологии обработки данных (OLTP, IoT, блокчейн, аналитика и т.д.). Если раньше для работы с разными типами данных и нагрузками использовались различные специализированные СУБД, то теперь единая конвергентная СУБД Oracle позволяет эффективно реализовать всю эту обработку.

Даже в традиционной реляционной модели существуют два типа физического представления данных: построчный и колоночный. Построчный оптимален для работы в транзакционных системах, а колоночный используется для аналитических систем.

Другим показательным примером может быть использование популярного формата JSON для работы с документами. Данный формат стал де-факто стандартом для обмена информацией между разными приложениями. СУБД Oracle обеспечивает обработку JSON-документов на уровне своего ядра.

Современные информационные системы состоят из множества независимых модулей, которые, **во-первых**, требуют централизованного управления, **во-вторых**, механизмов быстрого и надежного обмена информацией для обеспечения прикладных процессов. Платформа СУБД должна поддерживать такие современные **парадигмы разработки**, как микросервисы, события, REST и SaaS, и быть интегрированной с ними.

Использование множества специализированных технологий для каждого специфического модуля приводит к фрагментации данных, разработки, администрирования, соблюдению единых политик безопасности, а также усложняет процессы интеграции данных, а значит, производительности и оперативности. Все это приводит к снижению общей эффективности и повышению материальных затрат.

Стратегия Oracle при проектировании СУБД как единой конвергентной системы основывается на следующих принципах.

- Поддержка множества типов нагрузки (OLTP, OLAP, IoT, Blockchain, Machine Learning и т.д.).
- Встроенная поддержка востребованных типов, моделей данных и алгоритмов их обработки (реляционные, JSON, XML, Graph, Spatial и т.д.).
- Средства для обеспечения различных парадигм разработки, а также интеграция с внешними востребованными технологическими решениями (микросервисы, событийность, REST, SaaS, CI/CD и т.д.).

Помимо такой универсальности, есть множество других причин, почему наши заказчики предпочитают хранить все типы информации в своей базе данных Oracle.

- **Единое администрирование, настройка и управление.** Данные, хранящиеся в базе данных, напрямую связаны с метаданными. СУБД Oracle автоматически поддерживает внутри себя систему управления метаданными – словарь БД (Oracle Dictionary). Также база данных Oracle Database предлагает проверенные и надежные способы для резервного копирования, восстановления, настройки производительности.



- **Простота разработки приложений.** Oracle для специализированного типа контента включает в себя языковые расширения SQL, API для PL/SQL и Java, а также множество алгоритмов через встроенные операторы SQL. Для специальных форматов СУБД Oracle включает в себя определенные языки запросов, такие как SQL для реляционных данных, XQuery для XML, графы SPARQL для RDF, команды доступа DICOM для медицинских изображений и службы REST для доступа к таблицам базы данных и объектам JSON.
- **Высокая доступность.** Архитектура максимальной доступности Oracle позволяет создавать конфигурации с нулевой потерей для любых данных. В отличие от конфигураций типичных систем, где информация об атрибутах хранится в базе данных с указателями на неструктурированные данные в файловой системе, требуется только одна процедура восстановления в случае сбоя.
- **Масштабируемая архитектура.** Во многих случаях способность индексировать, разбивать данные на секции, выполнять операции с помощью триггеров, использовать представления, делать настройки на уровне таблиц и параметризовать базу данных в целом позволяет приложениям поддерживать значительно большие наборы данных, которые построены на основе СУБД, а не на файловой системе.
- **Безопасность.** Oracle Database обеспечивает детализированную защиту (на уровне строк и столбцов). Одинаковые механизмы безопасности используются для всех видов информации. При использовании многих файловых систем службы каталогов не позволяют детализировать уровни контроля доступа. Иногда невозможно ограничить доступ отдельным пользователям, во многих системах предоставление пользователю доступа к любому контенту в каталоге дает доступ ко всему содержимому в нем.

Все вместе позволяет быстро строить эффективные системы, соответствующие самым последним индустриальным тенденциям. Как правило, современное приложение работает со множеством типов данных: реляционными, текстами, изображениями, видео, геокоординатами и т. д. Поэтому СУБД должна эффективно поддерживать эти типы данных и многие другие, которые могут потребоваться приложению в будущем.

СУБД Oracle оптимизирована под все типы нагрузок

Важной особенностью СУБД Oracle является ее оптимизация для разнообразных прикладных нагрузок. В одной базе данных могут одновременно выполняться массированная высококонкурентная транзакционная обработка и аналитическая обработка как новых, так и архивных данных.

Транзакционные системы

Реляционная схема данных с построчным хранением традиционно используется в приложениях транзакционных систем. СУБД Oracle предоставляет возможности для самых требовательных из них.

Высокая скорость параллельной обработки архитектурно гарантируется отсутствием блокировок между сессиями, производящими изменение и чтение одних и тех же данных. Согласованность чтения данных в одном SQL или транзакции исключает «грязные» данные.

Широкие возможности индексирования, секционирования, компрессии обеспечивают высочайшую производительность при массированном произвольном доступе, характерном для транзакционных (OLTP) систем.

Уникальные алгоритмы, реализованные в СУБД Oracle на программно-аппаратном комплексе Exadata, обеспечивают беспрецедентную скорость операций ввода-вывода к данным, расположенным на общей внешней системе хранения. С помощью технологии доступа через протокол RDMA время чтения данных с носителя для OLTP-систем сокращено **до 19 микросекунд**. Тот же RDMA позволяет сократить время записи в журнал транзакций СУБД в 8 раз по сравнению с альтернативными системами.



Хранилища данных и аналитические системы

Десятки лет СУБД Oracle предлагает развитые и производительные средства для обеспечения работы хранилищ данных и аналитических систем. К этому относятся массивованная параллельная загрузка и обработка данных, разнообразные средства секционирования, автоматические механизмы индексации данных и группирования, специальные алгоритмы обработки, bitmap-индексы, различные алгоритмы сжатия и дедупликации и т. д.

Уже несколько лет Oracle добавляет в СУБД расширенные возможности по использованию колоночного формата хранения и обработки данных, традиционного для хранилищ и аналитических систем.

На сервере СУБД с помощью опции In-Memory данные прозрачно для приложения преобразуются в колоночный формат, компрессируются и сохраняются в специальном кэше, расположенном в оперативной памяти. Сканирование данных производится с помощью векторных SIMD-инструкций процессора, множество значений в одном столбце обрабатывается за одну команду.

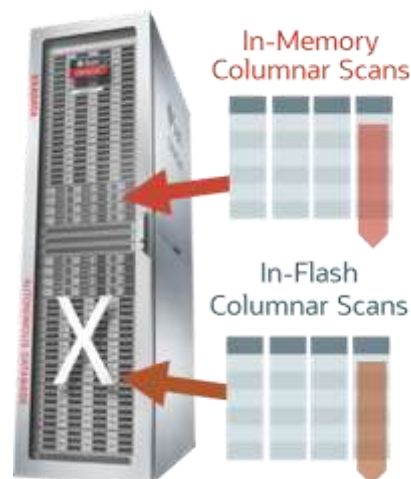
Это позволяет многократно увеличить производительность аналитических операций — аналитика работает в режиме реального времени.

При применении данной возможности в OLTP-приложениях можно избавиться от ненужных индексов, замедляющих операции по модификации данных в таблице.

На машине баз данных Exadata колоночный формат применяется для эффективной компрессии данных. В результате этого экономится физическое пространство, необходимое для хранения данных, больше данных может разместиться в кэше, а также за счет ускорения доступа к данным увеличивается производительность выполнения запросов.

Помимо возможностей поколоночной векторной обработки на серверах СУБД, алгоритмы опции In-Memory перенесены в интеллектуальные серверы хранения Exadata. Это добавляет новые возможности по увеличению производительности.

- Автоматическая идентификация и трансформация данных в колоночный формат на кэше, расположенном на flash-памяти. Большие объемы flash-памяти позволяют разместить на нем гораздо больший объем данных в колоночном формате.
- Агрегация flash-памяти на множестве серверов хранения обеспечивает максимальную производительность, масштабируемость и доступность.
- Производительность использования flash-памяти в Exadata для хранения и обработки данных в колоночном формате сопоставима с производительностью DRAM.



Машинное обучение

СУБД Oracle обеспечивает свыше 30 параллельных алгоритмов машинного обучения, включая глубокое. Помимо высокой производительности, отличительным преимуществом для заказчиков, которые их используют, является возможность запуска механизмов машинного обучения и скорринга непосредственно на данных транзакционной системы без необходимости их перемещения в специализированное хранилище.

- Получение результатов анализа в реальном времени, обнаружение мошенничества, sentiment-анализ и т. д. Запуск алгоритмов машинного обучения на оперативно изменяющихся данных.



- Исключение накладных расходов при копировании данных и поддержке специализированной системы для запуска алгоритмов машинного обучения.

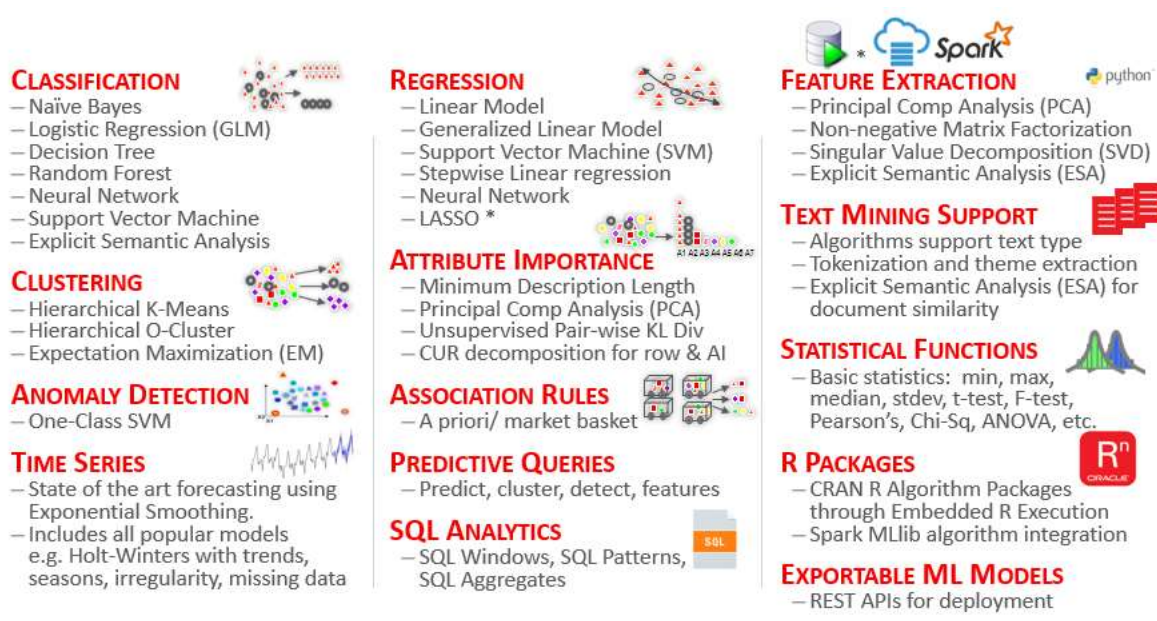


Рис. 1. Алгоритмы машинного обучения СУБД Oracle

Встроенные возможности позволяют для машинного обучения использовать декларативный язык SQL или выполнять вызовы процедур из языков Python или R.

Новые возможности СУБД Oracle версии 20c позволяют использовать машинное обучение специалистами, не обладающими экспертизой в этой области. AutoML порекомендует наиболее подходящие алгоритмы, автоматизирует выбор признаков и произведет настройку гиперпараметров, благодаря чему точность модели значительно повысится.

Ранее платная опция Oracle Advanced Analytics сейчас БЕСПЛАТНА во всех редакциях СУБД Oracle Database.

Оптимизация для приложений Интернета вещей

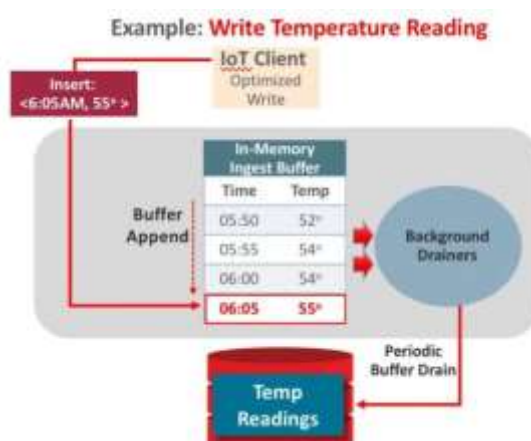
Начиная с СУБД Oracle версии 19c, обеспечена поддержка массивированной потоковой вставки данных. Потребность в высочайшей производительности данной функции характерна для приложений Интернета вещей, связанных с непрерывным приемом параметрической информации от множества различных устройств.

Стандартные операции вставки через INSERT накапливают данные в буфере памяти, а затем асинхронно переносятся на постоянное устройство хранения.

Использование СУБД Oracle в качестве приемника потоковых данных от устройств позволяет применять над поступающей информацией алгоритмы машинного обучения для мгновенного их анализа и, например, обнаружения аномалий в работе оборудования.

Тесты показывают возможности 25 миллионов вставок строк в секунду на обычном 2-процессорном сервере.

Данный механизм поддерживается при работе СУБД Oracle только на платформе Exadata.



Работа с большими объемами неструктурированных данных

Тип данных SecureFiles предназначен для обработки и хранения неструктурированных данных в базе данных Oracle и обеспечивает производительность, аналогичную файловой системе для базовых операций запроса и вставки. Оптимизированные алгоритмы для SecureFiles в 10 раз быстрее, чем алгоритмы обработки типа данных LOB (теперь называется BasicFiles).

Работа приложения базы данных с неструктурированными данными, размещаемыми в SecureFiles, имеет преимущества над использованием обычных файлов в файловых системах.

- Высокий уровень масштабируемости в среде Oracle Real Application Clusters.
- Приложения используют единый прикладной интерфейс для работы с реляционными данными и связанным с ними содержимым файлов.
- При использовании SecureFiles все изменения реляционных данных и содержимого файлов могут быть частью транзакции СУБД, что освобождает приложение от сложности гарантирования атомарности и согласованности.
- С технологией SecureFiles также может использоваться механизм Transparent Data Encryption (TDE). СУБД Oracle поддерживает автоматическое управление ключами для всех LOB-столбцов и прозрачно шифрует или дешифрует данные, резервные копии, журналы транзакций, информацию для отмены изменений.

Для содержимого SecureFiles доступны дедупликация и сжатие.

Дедупликация устраняет множество избыточных копий данных в SecureFiles и полностью прозрачна для приложений.

Содержимое SecureFiles может быть сжато с использованием стандартных алгоритмов компрессии, что приводит к значительной экономии пространства для хранения, а также повышению производительности. База данных Oracle автоматически определяет, является ли содержимое SecureFiles сжимаемым и есть ли смысл сжимать. Может использоваться несколько уровней сжатия, каждый из которых представляет собой компромисс между коэффициентом сжатия и скоростью. Организации могут выбрать уровень сжатия, который наилучшим образом соответствует их потребностям.

Применение параллельного выполнения DML над содержимым больших объектов SecureFiles увеличивает скорость работы с ними.

СУБД Oracle как мультимодельная база данных

Идея использования специальных моделей данных для определенных классов приложений существует десятки лет. Для эффективной работы транзакционной системы (OLTP) необходима модель данных, отличная от той, которая используется для аналитической нагрузки (OLAP). Другими востребованными моделями данных являются базы данных документов, основанных на форматах JSON и XML, графовые базы данных, пространственные базы данных и хранилища записей «ключ-значение». Концепция того, что для определенного типа приложений используются базы данных, разработанные специально для поддержки определенной модели данных, теперь называется Polyglot Persistence (в русскоязычной литературе иногда встречается термин «ситуационно ориентированные базы данных»).



Graph



Spatial



JSON



XML



Text



Relational



В линейке продуктов Oracle, кроме СУБД Oracle Database, есть специализированные базы данных: Berkeley DB (хранилище Key-Value), Oracle NoSQL Database (хранилище Key-Value и шардинг), Oracle TimesTen (обработка данных памяти) и Essbase (аналитическая обработка). На рынке технологий существует множество других продуктов с открытым кодом и проприетарных, поддерживающих единый подход Polyglot Persistence.



Рис. 2.

Поскольку коммерческие промышленные СУБД постоянно развиваются, они постепенно включают новые востребованные модели данных и методы доступа в рамках единой системы. Эта концепция называется Multimodel Polyglot Persistence и позволяет многим приложениям использовать единую СУБД, при этом имея возможность работать со специализированной моделью данных, необходимой для конкретного приложения.

Oracle Database является мультимодельной СУБД, которая поддерживает форматы данных, методы доступа и индексы, а также программные языки, требуемые специфическим приложениям, что обеспечивает интеграцию, транзакционность и согласованность данных в разных форматах и реализует единое администрирование и политики безопасности.

На протяжении десятилетий технология СУБД Oracle использовалась для решения уникальных проблем, возникающих при управлении большими объемами всех форм информации. Специализированные базы данных часто используются только для ведения реестров со ссылками на документы, изображения и медиаконтент, хранящийся в файловой системе. В СУБД Oracle для хранения этих данных в таблицах базы данных используются большие двоичные объекты (или BLOB). Помимо простых BLOB, в СУБД Oracle в качестве бинарных объектов могут быть различные модели, сложные типы и оптимизированные структуры данных, при работе с которыми представляются операторы для анализа и манипулирования документами JSON и XML, мультимедийным контентом, текстом, графами и геопространственной информацией (рис. 2).

Помимо традиционной для СУБД Oracle **реляционной структуры данных**, в ядро СУБД встроена поддержка наиболее востребованных моделей данных.

JSON

Современные разработчики приложений часто предпочитают хранить данные в виде документов, а не использовать реляционную модель. Основным фактором для этого перехода является гибкость, предоставляемая хранением данных на основе JSON или XML. Эта гибкость позволяет разработчику приложений гораздо более оперативно реагировать на потребности бизнеса: по мере изменения модели данных приложения больше не требуется изменений в схеме базы данных. Это позволяет развертывать и обновлять приложения гораздо быстрее. Переход на хранение данных на основе документов привел к развитию баз данных NoSQL.



Новые версии СУБД Oracle Database разрабатываются для обеспечения полной поддержки этого стиля разработки приложений. Спецификация Simple Oracle Document Architecture (SODA), представленная как часть функционала СУБД, описывает чрезвычайно простой API, который позволяет использовать базу данных Oracle в качестве хранилища документов JSON. API SODA обеспечивает возможность по созданию и удалению коллекций документов, а также операций по манипулированию документами внутри коллекций: создания, извлечения, обновления и удаления (CRUD) документов. С помощью API SODA можно выполнять операции получения списка и запросов по примеру (QBE) над коллекциями документов, а также различные вспомогательные операции, такие как массовая вставка и индексирование. SODA позволяет разработчикам приложений создавать и развертывать приложения, которые управляют данными с использованием документов JSON без каких-либо знаний SQL, JDBC и без помощи какого-либо администратора базы данных Oracle. Обеспечение проверки достоверности JSON встроено в саму СУБД. Содержимое JSON можно индексировать по атрибутам, используя эти индексы для оптимизации операций с содержимым JSON.

Выбор SODA позволяет разработчикам приложений получать все преимущества работы на основе JSON, не теряя ни одного из преимуществ платформы управления данными Oracle – можно работать в стиле NoSQL, не сталкиваясь со сложностями управления несколькими типами баз данных. Пользователи могут по-прежнему полагаться на базу данных Oracle, чтобы обеспечить приложениям высокую доступность, масштабируемость, безопасность и восстанавливаемость.

Другим важным преимуществом использования СУБД Oracle в качестве хранилища документов JSON в стиле NoSQL является то, что для доступа к содержимому объектов JSON можно использовать обычный SQL. Разработчики приложений могут создавать и развертывать свои приложения без каких-либо знаний SQL, используя методы Query-by-Example для запроса данных приложения. Однако, когда приходит время использовать данные приложения способами, отличными от предусмотренных разработчиком приложения (специальные запросы), или выполнять отчеты или аналитику данных, содержащейся в документах JSON, СУБД Oracle позволяет для этой цели использовать SQL.

СУБД Oracle имеет расширение языка SQL, позволяя запрашивать документы JSON как часть операций SQL. Эти расширения позволяют применять всю мощь SQL к содержимому документов JSON в простой и понятной манере. Они также позволяют выполнять операции соединения между документами JSON и операции соединения между документами JSON и другим видом контента, управляемым базой данных Oracle, включая реляционные данные, содержимым XML, пространственными, семантическими и текстовыми данными.

СУБД Oracle Database для Oracle Cloud также включает в себя Oracle Data Guide for JSON – полезный функционал, помогающий понять структуру документов JSON, которыми управляет база данных.

Data Guide for JSON динамически отслеживает структуру документов JSON, позволяя легко создавать реляционные представления документов JSON, облегчая работу специалистов, не знающих тонкостей технологии JSON.

СУБД Oracle обеспечивает поддержку JSON в базе данных с помощью следующего функционала.

- Генерация JSON-документа напрямую из реляционных данных в SQL-запросе.
- Операции частичного обновления документов JSON, позволяющие программисту изменять содержимое конкретных частей документа JSON.
- Документы JSON поддерживаются Oracle Database In-Memory и Oracle Exadata, что позволяет ускорить их обработку.
- Использование Oracle Spatial и Graph для запроса объектов GeoJSON, встроенных в документы JSON, и использование объектов JSON для хранения, индексирования географических данных в формате JSON и управления ими.
- Преобразование объекта Oracle Spatial и Graph SDO_GEOMETRY в объект геометрии JSON и объекта JSON геометрии обратно в объект SDO_GEOMETRY.



Технология блокчейн

Нативные блокчейн-таблицы Oracle упрощают использование технологии блокчейн для выявления и предотвращения мошенничества (антифрод). Блокчейн-таблицы Oracle выглядят как стандартные таблицы (см. рис. 3). Они допускают вставки SQL, а добавленные строки криптографически связываются в цепочку. При желании строки могут быть подписаны для защиты от мошенничества с помощью электронной подписи. Цепочка может быть проверена и подтверждена участниками.

Блокчейн-таблицы Oracle, как обычные таблицы, легко интегрируются в приложения. Их можно использовать в транзакциях и запросах наряду с другими таблицами. Кроме того, вставка производится очень быстро по сравнению с децентрализованным блокчейном, поскольку фиксация изменений не требует консенсуса.

```
CREATE Blockchain Table  
Trade_Ledger;
```



Рис. 3.

Хранение и анализ графовых данных

База данных Oracle включает специальные возможности графовой базы данных уже более десяти лет. Сетевые графовые модели данных широко используются государственным сектором, коммунальными службами, энергетикой и телекоммуникациями для моделирования и выполнения анализа соединений в пространственных сетях, таких как дороги, трубопроводы и инфраструктура. Графы RDF на основе стандартов позволяют статистическим бюро, финансовым учреждениям, медико-биологическим и фармацевтическим компаниям и государственным учреждениям использовать возможности семантической сети для публикации и обмена информацией в рамках стандартов связанных данных.

Поскольку корпоративные приложения включают и анализируют данные социальных сетей, данные датчиков и информацию Интернета вещей (IoT) для выявления закономерностей, взаимосвязей и аномалий, базы данных графов свойств общего назначения становятся частью ландшафта информационных технологий. С помощью Oracle Spatial и Graph в базе данных Oracle реализуется высокопроизводительная, масштабируемая графовая база данных. В отличие от других графовых баз данных, она включает в себя десятки встроенных, мощных, параллельных аналитических инструментов для упрощения работы разработчиков и исследователей данных, повышения их эффективности. Эти алгоритмы включают ранжирование, централизацию, рекомендации, обнаружение связей и поиск путей в социальных сетях, обнаружение мошенничества, перепродажи и перекрестные продажи, обнаружение влияния и анализ оттока клиентов.

Как часть Oracle Database, графовая модель находится в таблицах базы данных и может запрашиваться и фильтроваться с использованием SQL или различных поддерживаемых API. Для выполнения расширенного анализа графы загружаются в память, где используются алгоритмы анализа в памяти (PGX). Аналитика может выполняться либо в приложении Java, либо в многопользовательской, многографовой среде аналитического сервера в памяти сервера Oracle WebLogic, Apache Tomcat или Eclipse Jetty. Результатом анализа графа может быть другой граф, такой как двудольный, фильтрованный, ненаправленный, отсортированный или упрощенный граф ребер.

Oracle R Enterprise может вызывать алгоритмы графа свойств (функция Oracle Advanced Analytics для выполнения статистического анализа в базе данных Oracle). Данные графов могут быть проиндексированы с использованием индексов TEXT Oracle, текстовые запросы автоматически переводятся в операторы SQL SELECT с предложением contains. Данные графов могут запрашиваться с помощью SQL, а запросы графов могут включать пространственную фильтрацию (например, поиск результатов на определенном расстоянии от местоположения). Многоуровневая система безопасности может быть обеспечена с помощью контроля доступа на уровне графа, а Oracle Label Security может использоваться для детального контроля доступа к отдельным элементам графа.



Пространственная база данных

Oracle Spatial и Graph предоставляют наиболее полную пространственную базу данных на сегодняшний день, доступную в отрасли. Ее возможности обеспечивают встроенную поддержку самой высокой производительности для операций векторного и растрового анализа, топологических и сетевых моделей, трехмерных данных, геокодирования, маршрутизации и веб-сервисов стандарта OGC. Решение разработано с учетом передовых геопространственных требований крупнейших корпоративных и государственных приложений, таких как бизнес-аналитика, управления земельными ресурсами, телекоммуникаций, коммунальных услуг, обороны и безопасности страны. Оно реализовано с использованием параллельных архитектур, пространственных индексов в памяти и сотен пространственных алгоритмов, операторов и функций. Благодаря открытой, естественной пространственной поддержке Oracle Spatial и Graph исключают затраты и сложность отдельных проприетарных систем, а также позволяют использовать все ведущие инструменты ГИС. Это расширяет лидирующие в отрасли показатели безопасности, производительности, масштабируемости и управляемости Oracle для критически важных пространственных систем. Это выбор самых требовательных ГИС и геоприложений в мире.

Возможности RDF SEMANTIC GRAPH TRIPLE STORE FEATURES

Функция семантического графа RDF в Oracle Spatial and Graph представляет собой специальный граф для связанных данных и семантических веб-приложений стандарта W3C, распространенных в здравоохранении, финансовой области, СМИ. Oracle обеспечивает расширенное управление и анализ данных семантической диаграммы RDF со встроенной в ядро поддержкой стандартов Resource Description Framework (RDF) и Web Ontology Language (OWL) для представления и определения семантических данных и SPARQL, языка запросов, разработанного специально для анализа графов. Разработчики приложений используют ведущую в отрасли открытую масштабируемую платформу графовых данных.

Разработчики приложений могут придавать смысл данным и метаданным, определяя набор терминов и отношения между ними. Эти наборы терминов (онтологии) позволяют выполнять запросы, анализ и обработку данных, основанных на семантическом содержании, а не просто значениях. Онтологии используются для создания приложений, использующих специфичные для предметной области знания. Наборы онтологических данных, часто содержащие сотни миллионов элементов данных и связей, могут храниться в группах по три (triples), с использованием модели данных RDF. Oracle позволяет масштабировать систему до миллиардов троек, чтобы удовлетворить потребности самых требовательных приложений.

Анализ графиков RDF позволяет обнаруживать связи между наборами данных и документами, а также интегрировать и получать доступ приложений к системам с разнородными метаданными.

Oracle XML DB

XML широко применяется практически в каждой отрасли. Стандарты на основе XML можно найти в секторах здравоохранения, производства, финансовых услуг, государственном и издательского дела. Введение стандартов на основе XML, таких как XBRL, привело к тому, что XML стал де-факто механизмом обмена информацией между прикладными системами. Это привело к росту использования XML в качестве модели хранения для критически важных данных.

Чтобы обеспечить эти возможности, Oracle разработала Oracle XML DB. Это встроенная высокопроизводительная технология хранения и извлечения XML, которая поставляется со всеми версиями Oracle Database. Обеспечивает полную поддержку для всех ключевых стандартов, включая XML, пространства имен, DOM, XQuery, SQL/XML и XSLT.

Oracle XML DB является первой платформой, которая предоставляет действительно гибридные реляционные и XML-возможности, позволяя использовать всю мощь языка SQL для работы с содержимым XML и всю мощь парадигмы XML для реляционных данных. Oracle XML DB включает в себя XML Developer's Kit (XDK) —



универсальный набор компонентов, позволяющий создавать и развертывать программы на C, C++ и Java, которые обрабатывают XML. Это позволяет собрать эти компоненты в приложение XML, которое соответствует потребностям бизнеса.

Oracle постоянно расширяет ведущую в отрасли поддержку XML, гарантируя, что Oracle Database остается лучшей платформой для хранения всех возможных типов XML-контента и запросов к ним, а также для управления ими.

Oracle TEXT

Oracle Text — это ведущая система поиска, извлечения текста и управления текстом, интегрированная в среду СУБД Oracle. Она использует стандартный SQL для индексации, поиска и анализа текста и документов, хранящихся в базе данных Oracle, файлах и Интернете. Oracle Text может выполнять лингвистический анализ документов, а также поиск текста с использованием различных стратегий, включая поиск по ключевым словам, контекстные запросы, логические операции, сопоставление с образцом, смешанные тематические запросы, поиск по разделам HTML/XML и т. д.

Он может отображать результаты поиска в различных форматах, включая неформатированный текст, HTML с выделением терминов и оригинальный формат документа. Oracle Text поддерживает несколько языков и использует передовые технологии ранжирования по релевантности для улучшения качества поиска.

Oracle Text также предлагает расширенные функции, такие как классификация, кластеризация и поддержка метафор визуализации информации.

Поддержка современных парадигм разработки

СУБД Oracle имеет встроенные возможности по поддержке самых современных и востребованных технологий, используемых в распределенных системах. Главным образом это обеспечение модульности приложений, их оркестрации и непрерывности изменений.

Построение архитектуры, построенной на событиях. Интеграция с Kafka

СУБД Oracle упрощает создание приложений, основанных на событиях. Встроенные и богатые по возможностям очереди сообщений используются в СУБД Oracle уже десятки лет. Механизм Advanced Queueing обеспечивает ACID транзакционность и SQL-запросы.

Oracle интегрирует поток событий с Kafka. Golden Gate реплицирует изменения БД потоком в Kafka. Коннектор взаимодействия передает события между Kafka и очередями БД.

Начиная с СУБД Oracle версии 20с, новый механизм Kafka обеспечивает приложениям Kafka API, позволяющий управлять очередями в базе данных.

Шардинговая модель базы данных



Шардинг — форма горизонтального разбиения базы данных, которая допускает линейное горизонтальное масштабирование, становится все более популярной для многих веб-приложений, ориентированных на клиента, таких как электронная коммерция, приложения для мобильных устройств и социальные сети. Такие приложения имеют четко определенную модель данных и стратегию распределения данных (по хешу, диапазону, списку или их комбинации) и главным образом обращаются к данным с помощью



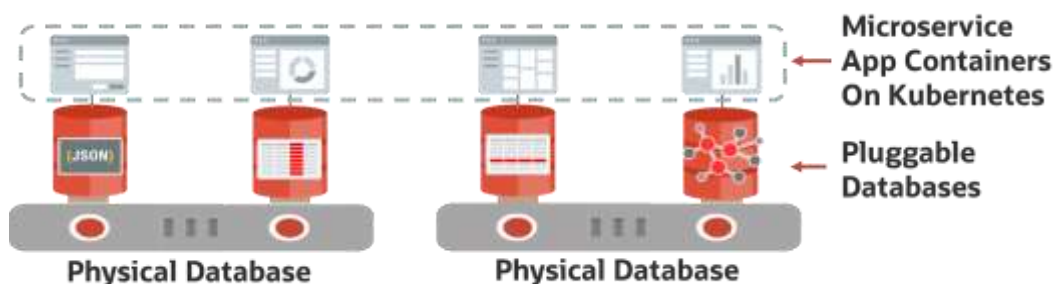
ключа шардинга. Примерами ключей шардинга являются идентификатор клиента, номер счета или идентификатор страны.

Oracle Database поддерживает шардинг. В этой архитектуре независимые базы данных являются отдельными шардами в модели данных. OLTP-транзакции, которые обращаются к данным, связанным с одним значением ключа шардинга, — это основной сценарий использования для шардинговых баз данных. Например, это может быть поиск и обновление записей клиента, документы подписок, финансовые транзакции, транзакции электронной коммерции и тому подобное. Поскольку все строки с одинаковым значением ключа шардинга гарантированно находятся в одном шарде, такие транзакции всегда выполняются в одной базе данных с высочайшей производительностью и обеспечением высочайшего уровня согласованности. Поддерживаются операции с участием нескольких шардов, но с пониженным уровнем производительности и согласованности. Такие транзакции включают в себя простые агрегации, отчеты и т.п., играют незначительную роль в изолированном приложении по сравнению с рабочими нагрузками, в которых преобладают транзакции OLTP с одним шардом.

Распределенные базы данных на основе Oracle Sharding можно применять для облачных и других приложений, которые выигрывают от линейной масштабируемости, локализации сбоев и географического распределения данных. Это может упростить обновление, потому что изменение конфигурации одного шарда не влияет на другие шарды и позволяет администраторам сначала проверить изменение на небольшом подмножестве данных. Шардинг хорошо подходит для размещения в облаке. Размеры шардов могут соответствовать размеру, необходимому для размещения любой облачной инфраструктуры, при этом достигать требуемого уровня обслуживания. Oracle Sharding поддерживает локальные, облачные и гибридные модели развертывания.

Подключаемые БД как основа поддержки для микросервисов

Контейнерные базы данных СУБД Oracle позволяют выделять подключаемые базы данных для отдельных независимых модулей приложения, микросервисов. Pluggable Databases могут физически объединяться в одну контейнерную базу данных для упрощения развертывания или разделяться по разным базам и физическим серверам для повышения изоляции и масштабирования (см. рис. ниже).



Оперативная реорганизация данных и обеспечение CI/CD

СУБД Oracle упрощает обеспечение непрерывности интеграции и доставки (CI/CD) для приложений. Обеспечивается полная поддержка реорганизации схем и данных онлайн.

- Edition Based Redefinition позволяет в базе данных одновременно выполняться нескольким версиям одного приложения во время его обновления.
- Online Table Redefinition обеспечивает непрерывность выполнения приложений при физическом изменении структур хранения, реорганизации таблиц и индексов.
- Встроенные модели JSON и XML позволяют гибко манипулировать структурой данных.



Все вместе обеспечивает механизмы по обновлению приложений без их остановки: не требуются технологические окна для остановки приложений и выполнения больших пакетных изменений.

Построение информационных систем с повышенными требованиями по безопасности и защите информации

Двадцать пять лет назад в основном правительственные и финансовые учреждения были заинтересованы в безопасности (в основном шифровании), в то время как все остальные доверяли администраторам, пользователям и вычислительной среде и считали, что их данные в безопасности. В конце 90-х годов, когда открылись новые возможности для торговли через сеть Интернет, компании начали осознавать необходимость повышения защищенности своих информационных ресурсов. Эта новая перспектива привела к появлению SSL, сетевых брандмауэров и более сильной криптографии.

Сегодня мы живем в совершенно ином мире, где каждый фрагмент данных находится в режиме онлайн и доступен круглосуточно. Чтобы справиться с этой новой реальностью, мы видим множество различных технологий безопасности, защищающих различные элементы информационных систем на всем пути: от клиентских приложений до серверных чипсетов. В то время как расходы на глобальную безопасность возрастают, взломы становятся все более масштабными и смелыми, а также затрагивают все: от клиентских и гражданских баз данных до нефтяных вышек и Wi-Fi-маршрутизаторов.

Хакеры создали сложные инструменты наряду с процветающим подпольным рынком, чтобы получать информацию из всего, что у нас есть, будь то мобильные устройства, ноутбуки, файловые серверы или базы данных. Для большинства хакеров целью выбора является не ноутбук или электронная таблица, чаще всего это база данных с сотнями миллионов записей. Хакеры используют различные методы, включая атаки на сеть, операционную систему, базу данных, приложения и ИТ-инфраструктуру. Они также нацелены на пользователей, которые имеют законный доступ к информационным системам. Иногда именно инсайдеры, обладающие глубокими знаниями о данных и способах защиты систем, атакуют системы ради выгоды.

Почему организации так уязвимы для атак? Многие могут сказать, что они не знают, где находятся их конфиденциальные данные, где находятся уязвимости и какие исправления необходимы. Они могут использовать устаревшие версии ПО с обнаруженными уязвимостями из-за опасений, что исправления могут нарушить работоспособность их приложения. Слишком многие все еще полагаются только на средства обеспечения безопасности периметра, не понимая, как легко хакеры могут обойти периметр сети, добраться до баз данных и спокойно уйти с украденными данными. Неудивительно, что в среднем жертвам требуется шесть месяцев, чтобы просто узнать, что они были взломаны, и также неудивительно, что часто они узнают о нарушении от клиентов или правоохранительных органов.

Многие лидеры в области информационных технологий, баз данных и безопасности теперь понимают, что защита баз данных должна быть одной из их самых важных целей. В конце концов, в большинстве компаний именно базы данных содержат большую часть конфиденциальных данных активов. К тому же специалисты информационной безопасности признают, что они никогда не смогут найти и заблокировать все пути, которые могут использовать хакеры.

За последние двадцать лет произошел значительный сдвиг в том, как хакеры охотятся за базами данных. В ответ на это Oracle создала несколько технологий безопасности для защиты данных в источнике — непосредственно в базе данных. Отраслевые аналитики и специалисты по безопасности признают, что база данных Oracle обеспечивает наиболее полную безопасность в отрасли.

Нарушения происходят быстрее, чем мы можем себе представить, и очень важно, чтобы вы были готовы! Ваши данные — ваш актив, и, если вы не защитите их должным образом, они могут попасть в чужие руки и стать причиной многих неприятностей.



Данные — новая валюта

Организации во всем мире испытывают на себе воздействие утечки данных с беспрецедентной скоростью. Похоже, каждый день приносит новость о том, что поставщик услуг теряет личную информацию абонентов, работодатель теряет кадровые записи сотрудников или государственный подрядчик теряет конфиденциальную интеллектуальную собственность. Данные — новая валюта, и злоумышленники часто могут использовать украденные данные для получения финансовой или политической выгоды в течение многих лет после того, как произошло нарушение.



Рис. 1. Многочисленные требования по обеспечению информационной безопасности

Субъекты угрозы

Чтобы понять, почему важен всесторонний подход к защите базы данных, необходимо уяснить, кто является субъектами, которым нужны ваши данные, и как они пытаются их получить.

Субъекты угроз можно в широком смысле разделить на две группы: аутсайдеры и инсайдеры. Внешние злоумышленники сильно различаются по уровню мастерства и ресурсам. Они включают в себя всех: от одиноких хактивистов и киберпреступников, стремящихся разрушить бизнес или получить финансовую выгоду, до преступных групп, стремящихся совершить мошенничество и создать беспорядки в национальном масштабе. Однако один из классов субъектов угроз, часто упускаемых из виду в сообщениях СМИ, — инсайдеры. Они могут включать в себя действующих или бывших сотрудников, любопытных и настырных, а также клиентов или партнеров, которые используют доверие к ним для кражи данных. Приз для обеих этих групп включает персональные данные, финансовые данные, коммерческую тайну и другие секреты.

Из-за особенностей архитектуры построения современного приложения субъекты угроз могут использовать в поисках уязвимостей множество подходов.

- Кража учетных данных администратора или привилегированного пользователя приложения с помощью фишинга на основе электронной почты и других форм социальной инженерии или с помощью вредоносных программ для поиска учетных данных и данных.
- Использование слабых мест в приложениях с помощью таких методов, как SQL-инъекция и обход безопасности прикладного уровня путем встраивания SQL-кода в кажущийся безобидным ввод данных, предоставляемый конечным пользователям.
- Повышение привилегий за счет использования уязвимых приложений.
- Доступ к файлам базы данных, которые не зашифрованы на физическом носителе информации.
- Кража носителей, содержащих резервные копии баз данных.



- Копирование реальных данных из систем разработки и тестирования, где они обычно не так хорошо защищены, как в производственных системах.
- Избыточный доступ к конфиденциальным данным с помощью приложений, которые непреднамеренно предоставляют информацию в объеме, превышающем тот, который пользователи должны использовать для выполнения своих служебных обязанностей.
- Использование незащищенных систем или неправильно сконфигурированных баз данных для обхода контроля доступа.

Контуры контроля безопасности

Сегодня многие организации переносят свои рабочие нагрузки в облако и внедряют новые гибкие модели развертывания. Хотя облачные развертывания могут обеспечить более высокий уровень безопасности инфраструктуры, они также изменяют границы доверия и могут подвергать данные новым угрозам. В результате элементы управления безопасностью данных должны масштабироваться и беспрепятственно работать в локальных, частных, общедоступных и гибридных облачных средах.

Аутентификация и авторизация

Фундаментальным шагом в обеспечении безопасности системы баз данных является проверка подлинности пользователя, который обращается к базе данных (аутентификация), и контроль того, какие операции он может выполнять (авторизация).

Пользователи базы данных

Весь доступ к базе данных осуществляется через учетные записи пользователей базы данных, будь то административные пользователи, учетные записи приложений или обычные пользователи. Oracle поддерживает различные типы пользователей баз данных, каждый из которых имеет различные права доступа к базе данных.

- **Обычные пользователи баз данных** ограничены своей схемой, содержащей их таблицы, представления, индексы и хранимые процедуры. Если хакеры взломают их учетные записи, они смогут не только просматривать или обновлять данные в пользовательской схеме, но и получать доступ к объектам в других схемах, доступ к которым может быть разрешен пользователю.
- **Учетные записи приложений** — учетные записи баз данных, используемые для запуска приложений, как коммерческих, так и собственной разработки. Эти учетные записи похожи на обычные учетные записи пользователей базы данных, но, поскольку приложения должны работать 24x7, их пароли часто хранятся на нескольких серверах среднего уровня. Компрометация этих учетных записей базы данных может привести к потере данных всего приложения, включая данные о конечных пользователях.
- **Администраторы приложений.** Эти учетные записи используются для исправления и обновления приложения, управления им и, следовательно, имеют полный доступ ко всем данным и хранимым процедурам, используемым приложением.
- **Аналитики приложений или пользователи бизнес-аналитики.** Эти пользователи обычно получают неограниченный доступ на чтение к схеме приложения, не проходя через элементы управления доступом уровня приложения.
- **Администраторы баз данных (DBAs)** отвечают за широкий спектр задач базы данных, включая управление производительностью, диагностику и настройку, обновление и исправление, запуск и завершение работы базы данных, а также резервное ее копирование. Их привилегированный доступ к базе данных также дает им доступ к любым конфиденциальным данным (личные, медицинские, корпоративные финансовые записи и т. д.), даже если этот доступ не требуется для выполнения задач.



DBA. DBA, по существу, владеют ключами к царству данных, поэтому им, как правило, полностью доверяют в организациях. К сожалению, это же делает их главной мишенью хакеров.

- **Администраторы безопасности.** Многие организации имеют специализированные базы данных, выполняющие функции администраторов безопасности, включая управление учетными записями пользователей, управление ключами шифрования и управление аудитом.

Хакеры обычно нацеливаются на пользователей базы данных.

Пользователи — самое слабое звено

Самый простой способ взломать базу данных — выдать себя за авторизованного пользователя этой базы данных. Некоторые из распространенных методов включают в себя следующее.

- **Социальная инженерия для кражи аутентификационных данных учетной записи.** С помощью целенаправленных фишинговых атак хакеры могут нацелиться на конечных пользователей или администраторов БД в организации (информацию о которых легко найти через каналы социальных сетей, таких как LinkedIn) и украсть их учетные данные.
- **Пароли, используемые на других скомпрометированных сайтах.** Многие пользователи используют один и тот же пароль в нескольких приложениях или нескольких веб-сайтах, и, если какой-либо из них будет скомпрометирован, злоумышленники могут попробовать эти пароли для атаки на вашу базу данных.
- **Поиск жестко закодированной информации о подключении к базе данных.** Приложения или пользователи часто подключаются к базе данных с помощью встроенных имен пользователей и паролей базы данных или хранят эти учетные данные в файле конфигурации с открытым текстом. Компрометация этих учетных записей позволяет хакерам просачиваться, изменять или удалять любые данные, к которым эта учетная запись может получить доступ.
- **Использование паролей по умолчанию или опубликованных паролей.** Хакеры пытаются использовать обычные пароли по умолчанию для подключения в качестве пользователя и использовать свои привилегии для доступа к конфиденциальным данным.
- **Атаки подбора паролей.** Пробуя различные комбинации известных паролей и их вариации, хакеры могут взламывать учетные записи баз данных со слабыми паролями, когда нет ограничений на повторные попытки паролей. Без применения сложных паролей некоторые пользователи могут использовать легко угадываемые пароли, такие как password и oracle123.

Эти атаки необязательно изощренные и могут быть выполнены «скриптовыми детишками», но они дают хакерам по крайней мере столько же доступа, сколько и этому конкретному пользователю, а затем еще и еще.

Методы проверки подлинности

СУБД Oracle поддерживает различные способы аутентификации, включая пароли, хранящиеся локально в базе данных или централизованных службах каталогов. Пользователи также могут проходить проверку подлинности с помощью операционной системы, используя предложение IDENTIFIED EXTERNALLY при создании пользователя, или с помощью различных внешних служб проверки подлинности, включая Kerberos, сертификаты открытого ключа (PKI) и RADIUS.

Пароли используются для односторонней аутентификации пользователя в базе данных, в то время как сертификаты Kerberos и открытого ключа поддерживают взаимную аутентификацию, гарантируя, что пользователь действительно подключается к соответствующей базе данных. Хотя пароли удобны в использовании, легче скомпрометировать пароль пользователя по сравнению с его учетными данными Kerberos или PKI.



После проверки подлинности учетной записи пользователя она сопоставляется со схемой базы данных, состоящей из таблиц, представлений, индексов и процедур, а затем получает соответствующую авторизацию с помощью ролей. При проверке подлинности пользователей с помощью службы каталогов пользователи либо получают свою собственную схему базы данных (эксклюзивное сопоставление), либо сопоставляются с общей схемой (общее сопоставление).

Обеспечение устойчивости пользователей к атакам

При аутентификации по паролю пользователи должны запоминать и использовать надежные, длинные и сложные пароли, а также вводить их при необходимости. Предполагается, что для разных баз данных будут использоваться разные пароли и совместное использование паролей баз данных запрещено. Однако и администраторов, и обычных пользователей привлекают определенные сочетания клавиш, а хакеры готовы эксплуатировать такое поведение человека. Необходимо задействовать механизмы, которые управляют таким поведением.

Проблема: хранение простых текстовых паролей

При проверке подлинности на основе пароля пользователи предоставляют пароль при подключении к базе данных, но приложения, системы промежуточного слоя и пакетные задания не должны зависеть от ввода пароля человеком. В прошлом распространенным способом предоставления паролей было встраивание имен пользователей и паролей в код или сценарии. Это автоматизировало процессы аутентификации, но увеличивало риск реализации атаки на базы данных, и люди должны были убедиться, что сценарии не были подсмотрены злоумышленниками. Кроме того, после изменения пароля требовалось внести изменения в скрипты.

Wallet (кошельки) Oracle были разработаны для решения этой проблемы путем хранения различных учетных данных аутентификации, включая пароли, закрытые ключи и сертификаты, в зашифрованном виде. В случае кошельков пользователям необходимо запомнить только пароль для кошелька, который затем разблокирует все оставшиеся учетные данные пользователя для одной или нескольких баз данных. Приложения и серверы баз данных также могут использовать режим автоматического входа в кошелек для доступа к учетным данным 24x7.

При использовании Wallet пароли базы данных больше не отображаются в истории командной строки или в файлах конфигурации с открытым текстом. Кроме того, код приложения не нужно менять всякий раз, когда меняются имена пользователей или пароли. Кошельки также используются для хранения мастер-ключа для прозрачного шифрования данных в колонках или табличных пространствах (ASO: Transparent Data Encryption).

Чтобы настроить использование пароля, хранящегося в Oracle Wallet, необходимо задать значение параметра `WALLET_LOCATION` в клиентском файле `sqlnet.ora`. После этих изменений приложения и пользователи могут подключаться к базе данных, не указывая непосредственно учетных данных для входа в командной строке и не скрывая их в файле конфигурации.

Проблема: совместное использование учетных записей и паролей

Администраторы приложений часто должны подключаться к схеме приложения для обслуживания. Если существует несколько администраторов приложений, то все они обычно используют одно и то же имя пользователя и пароль приложения. Если существует несколько баз данных, они также иногда обмениваются паролями. Совместное использование паролей может быть удобным, но не обеспечивает никакой подотчетности и затрудняет аудит и расследование проблем.

Прокси-аутентификация позволяет во многом решить указанную проблему. При авторизации для проверки подлинности учетной записи приложения администраторы сначала проходят проверку подлинности в базе данных со своими собственными учетными данными, а затем автоматически регистрируются через прокси в схеме приложения без предъявления пароля учетной записи схемы приложения. Например, `nick_appdba`



подключается с помощью пароля личной учетной записи, а затем получает привилегии схемы **hrapp** через прокси-подключение следующим образом:

```
SQL> CONNECT nick_appdba[hrapp]
```

Enter password: <nick_appdba_password>

Записи аудита теперь регистрируют hrapp как USER, в то время как пользователь nick_appdba регистрируется как PROXY_USER_NAME. К прокси-пользователю могут быть применены механизмы контроля доступа, такие как Database Vault.

Проблема: плохие пароли

Иногда пользователи используют очень слабые и короткие пароли, что позволяет злоумышленнику легко угадать пароль и выполнить несанкционированное подключение к базе данных.

Профили пользователей используются для создания общей политики параметров авторизации паролей и ресурсов для учетных записей пользователей.

Каждая учетная запись пользователя может (должна) быть связана с профилем, чтобы упростить управление общими политиками безопасности в организации. Если при создании пользователя политика не задана, то назначается политика по умолчанию. Приведенный ниже пример политики (org_profile) включает в себя как параметры авторизации паролей, так и параметры авторизации ресурсов.

```
SQL> CREATE PROFILE org_profile LIMIT
CONNECT_TIME 90                -- Limit connection time to 90 min
SESSIONS_PER_USER 2            -- Allow two sessions for each user
IDLE_TIME 30                   -- Automatic logout after 30 min idle
FAILED_LOGIN_ATTEMPTS 6        -- Lock the account after 6 attempts
PASSWORD_LIFE_TIME 180         -- Force password change after 180 days
PASSWORD_VERIFY_FUNCTION       -- ora12c_stig_verify_function;
```

Функция ora12c_stig_verify_function устанавливает ряд требований к паролю, которые соответствуют требованиям политики информационной безопасности Министерства обороны США, включая минимальную длину, минимальное количество буквенно-цифровых символов и использование по крайней мере одного специального символа (среди прочих требований).

Каждый пользователь должен иметь соответствующий профиль, чтобы обеспечить единообразное применение общей политики безопасности. Изменения в политике безопасности можно легко выполнить, изменив одну политику вместо изменения каждой учетной записи.

Проблема: управление несколькими учетными записями и паролями

Если у пользователей есть учетные записи в нескольких базах данных, они, как правило, сохраняют один и тот же пароль, но затем должны обновить пароль в каждой базе данных, когда какая-либо база данных требует произвести обновление пароля. Когда сотрудник покидает организацию, администраторы должны удалить его учетную запись из всех баз данных. Однако на практике эти аккаунты живут очень долго, что делает их весьма привлекательной мишенью для хакеров.

Благодаря использованию глобальных пользователей (global users) с помощью Oracle Enterprise User Security (EUS) или Oracle Centrally Managed Users (CMU) администраторы могут централизованно управлять пользователями и ролями в нескольких базах данных в службах каталогов организации. Это также означает, что пользователи должны менять свои пароли только один раз в соответствии с политикой паролей каталога.

Централизованное управление пользователями

В организации с большим количеством пользователей, имеющих доступ к нескольким базам данных, пользователи испытывают трудности в поддержании соответствия пароля политике конкретной базы данных



и запоминании различных паролей для своих различных учетных записей. Кроме того, администраторам трудно управлять уникальными учетными записями для каждого пользователя в каждой базе данных, поскольку каждый пользователь должен быть подготовлен отдельно вместе со своими паролями. Что еще более важно, каждая учетная запись должна быть деактивирована, когда пользователь меняет роль и/или покидает организацию. Хакеры часто используют такие бесхозные учетные записи для получения несанкционированного доступа к данным.

Технология Oracle Enterprise User Security (EUS) позволяет централизованно управлять пользователями и ролями в нескольких базах данных через службы каталога Oracle Directory Services, которые при необходимости могут интегрироваться с другими корпоративными каталогами. EUS позволяет пользователям и администраторам проходить проверку подлинности с помощью Oracle Directory Services с использованием паролей, меток Kerberos и сертификатов. При подключении пользователя база данных обращается к каталогу для получения информации об его авторизации (ролях). Пользователи базы данных, управляемые с помощью службы каталогов, называются корпоративными пользователями, поскольку они охватывают несколько баз данных по всему предприятию. Таким корпоративным пользователям могут быть назначены корпоративные роли или группы, определяющие права доступа к нескольким базам данных. Корпоративная роль сопоставляется с одной или несколькими ролями в отдельных базах данных.

Начиная с версии Oracle Database 18c, реализована технология централизованного управления пользователями (Centrally Managed Users, CMU), которая интегрирует базу данных с Microsoft Active Directory для управления аутентификацией и авторизацией пользователей. Пользователи Active Directory могут иметь свою собственную схему (эксклюзивное сопоставление) или совместно использовать схему через группы Active Directory. Группы Active Directory также могут напрямую сопоставляться с ролями базы данных.

Защита пользователей от взлома

Нельзя ожидать, что обычные пользователи баз данных будут помнить все руководящие принципы безопасности, касающиеся надежности их паролей, управления паролями и предоставленных им ролей или привилегий, и добровольно следовать этим принципам. Эти рекомендации по безопасности необходимо применять на уровне базы данных, чтобы убедиться, что пользователи следуют им. Перечисленные ниже политики помогают свести к минимуму риск взлома пользователей.

- Строгие требования в профиле пароля для всех пользователей, контролирующих сложность пароля, период бездействия, количество повторных попыток ввода пароля и т. д.
- Использование надежных механизмов аутентификации с помощью инфраструктуры открытых ключей (Public Key Infrastructure, PKI) или Kerberos.
- Использование Oracle Wallet вместо того, чтобы записывать пароль где-нибудь.
- Централизованное управление пользователями с помощью каталогов для снижения вероятности появления бесхозных учетных записей, когда сотрудники покидают организацию или меняют свои должности.
- Создание пользователей с общими схемами, если им не нужна собственная схема для объектов.
- Регулярная проверка, были ли пользователям предоставлены дополнительные привилегии для объектов вне их схемы. Внедрите анализ привилегий, чтобы гарантировать, что пользователи имеют только те, которые им действительно нужны для выполнения служебных обязанностей.

Учетные записи администратора баз данных (DBA) обычно имеют полномочия на доступ к базе данных, гораздо больший, чем фактически требуется только для администрирования, поэтому их доступ должен контролироваться с использованием следующих передовых методов.

- Администраторы баз данных должны использовать именованные учетные записи с надежной аутентификацией для обеспечения подотчетности. Использование общих учетных записей



или учетных записей по умолчанию, таких как SYSTEM, должно быть запрещено, вместо этого они должны использовать прокси-аутентификацию и/или усиленные механизмы аутентификации, такие как PKI и Kerberos.

- Адаптирование привилегий для отдельных задач и обязанностей, чтобы уменьшить область возможной атаки, связанной с этими учетными записями, вместо предоставления роли DBA. Если для устранения каких-либо неполадок требуются дополнительные привилегии и роли, их следует отменить после выполнения работ.
- Блокирование доступа к пользовательским схемам с помощью Database Vault.
- Полный аудит всей деятельности DBA для обеспечения подотчетности и контроля.
- Использование системы управления привилегированными учетными записями (PAM), если для доступа во время обновления требуются учетные записи операционной системы root и владельца базы данных SYSDBA.

Учетные записи DBA приложений и служб приложений в базе данных не только имеют все роли и привилегии, необходимые для запуска приложения от имени каждого пользователя приложения, но и для удобства могут иметь роли и привилегии для выполнения установки приложения, обновления, исправления и других действий по обслуживанию. Такие учетные записи должны быть хорошо защищены с использованием следующих передовых методов.

- Использование надежных механизмов аутентификации в базе данных (PKI, Kerberos).
- Oracle Wallet для хранения учетных данных.
- Прокси-аутентификация вместо совместного использования паролей между несколькими администраторами.
- Чтобы свести к минимуму прямой вход в систему от имени владельца схемы приложения, целесообразно создать учетную запись только для схемы для объектов и процедур приложения, а затем использовать отдельную учетную запись службы приложений во время выполнения для доступа к объектам приложения через управляемые процедуры и представления. Функция учетной записи только для схемы была введена в Oracle Database 18c, где учетные записи имеют именованную схему, но не имеют пароля или другой возможности входа в систему. Начиная с Oracle Database 19c, почти все учетные записи Oracle, установленные вместе с базой данных, устанавливаются только как учетные записи схемы, чтобы предотвратить вход в эти привилегированные учетные записи злоумышленниками. В прошлом у этих учетных записей были пароли, которые нужно было бы периодически менять. Другим пользователям по-прежнему может быть предоставлен доступ на чтение и запись к этим схемам.
- Предоставлять администратору приложений (Application DBA) доступ только к объектам схемы приложения, а не к привилегиям всей базы данных, хотя это удобно.

Оценка текущего состояния безопасности

Необходимость оценки безопасности

Современные информационные системы стали очень сложными со множеством конфигурационных параметров. Как показал анализ причин утечек данных, крайне важно иметь правильно настроенные и безопасные системы. Человеческая ошибка потенциально может оставить базу данных открытой для всех, или злоумышленник может злонамеренно использовать уязвимости в конфигурации, чтобы получить несанкционированный доступ к конфиденциальным данным. Пренебрегая элементарными средствами контроля безопасности, организации могут в конечном итоге потерять имена своих клиентов, адреса, даты рождения, информацию об учетных записях и другие личные данные, влияющие на их репутацию. Важно проверить базу данных, чтобы убедиться, что она надежно настроена, и исправить ситуацию, если есть отклонение.



Оценка конфигурации безопасности также стала важной частью многих нормативных актов, таких как GDPR ЕС, PCI DSS, Sarbanes-Oxley, и многочисленных законов, обязывающих уведомлять об утечках. Различные организации, такие как Центр интернет-безопасности (CIS) и Министерство обороны США, также имеют рекомендации по лучшим практикам настройки безопасности. Важность конфигурации невозможно недооценивать, поскольку выпускаются новые правила и развиваются существующие. В результате крайне важно иметь решение, которое может адаптироваться к изменениям таких правил.

Исследование и оценка конфигурации базы данных

Нападающие неторопливо готовятся к атаке и тратят значительное время на разведку. Они используют несколько инструментов, которые автоматизируют обнаружение баз данных, открытых портов и уязвимостей, автоматизируют атаки приложений и SQL-инъекций, а также выполняют атаки с использованием подбора паролей. Как только они заканчивают зондирование и выявляют самые слабые звенья, они определяют последующие шаги. По сути, злоумышленники оценивают ваш текущий статус безопасности, чтобы найти самый простой способ получить доступ к данным.

- Какая версия базы данных используется? Каковы известные уязвимые места? Были ли установлены необходимые обновления системы безопасности?
- Есть ли какие-либо известные пользователи с паролями по умолчанию или легко угадываемыми паролями?
- Кто является привилегированным пользователем этой базы данных? Есть ли способ повысить привилегии у обычных пользователей?
- Включен ли аудит? Для каких пользователей? Каковы условия аудита?
- Зашифрованы ли данные? Если нет, то как получить доступ на физическом уровне к ним или их резервной копии?

Все эти вопросы обдумывает хакер, и ответы на них помогают ему придумать план взлома базы данных и получения данных. Организации, как владельцы данных, должны знать ответы на поставленные вопросы, но с целью улучшения положения в области безопасности.

Основные рекомендации по защите баз данных

- Почти все базы данных содержат конфиденциальные данные, но уровень их важности может отличаться. Например, дата рождения может быть более значимой, чем адрес электронной почты. Важно выяснить, какие базы данных содержат конфиденциальные данные того или иного типа, чтобы можно было соответствующим образом установить контроль.
- Параметры конфигурации безопасности тесно связаны с тем, как ведет себя база данных, и требуют понимания того, что это за параметры, что они делают, влияния их изменения и их зависимости.
- Не все пользователи базы данных равны. Помимо БД, есть еще несколько других субъектов или процессов, которые должны взаимодействовать с вашими данными через учетную запись: само приложение, администраторы приложений, администраторы безопасности и другие, такие как учетные записи служб, пакетные программы и т.д. Четкое определение различных типов пользователей и различных видов действий, которые они должны выполнять в базе данных, поможет правильно управлять привилегиями и ролями и реализовать принцип наименьших достаточных привилегий.
- Не все базы данных создаются одинаково и редко учитывают параметры настройки, необходимые для защиты конфиденциальных данных. В базе данных могут быть определены требования к сложности пароля, но в то же время в ней не предусмотрена защита данных клиентских таблиц с важной информацией. Злоумышленники используют эту уязвимость, которая возникает при установке системы по умолчанию или плохо настроенных баз данных.



Быстрая оценка состояния безопасности баз данных Oracle

Бесплатный инструмент Oracle Database Security Assessment Tool (DBSAT), доступный для использования зарегистрированными пользователями Oracle, определяет местоположение конфиденциальных данных и области, в которых конфигурация, работа или реализация базы данных сопряжена с риском. DBSAT собирает и анализирует различную информацию о базе данных. DBSAT также формирует рекомендации по внесению изменений и способы контроля для снижения этих рисков.

Помимо конфигурации базы данных и Listener-a, DBSAT собирает информацию об учетных записях пользователей, привилегиях и ролях, способах контроля авторизации и разделения обязанностей, детальном контроле доступа, шифровании данных и управлении ключами, политике аудита и разрешениях на доступ к файлам операционной системы. DBSAT применяет правила для быстрой оценки текущего состояния безопасности базы данных и рекомендует рациональные методы устранения недостатков. Обновленные правила на основе лучших практик безопасности периодически поставляются вместе с новыми версиями этого инструмента.

Одна из проблем, возникающих при проведении оценки безопасности баз данных и оценки уязвимости средств защиты, заключается в том, что в большом количестве полученных результатов трудно определить приоритеты и действовать в соответствии с критическими выводами в первую очередь.

DBSAT не только сканирует базу данных на наличие слабых мест и уязвимостей, но и указывает уровень приоритета. Он также предоставляет подробную информацию высокого уровня и конкретные рекомендации по каждому из вопросов, что делает его применение более простым и оперативным. DBSAT также используется консультантами и партнерами Oracle при выполнении оценок безопасности баз данных клиентов.

Компоненты DBSAT

DBSAT состоит из трех компонентов: Collector, Reporter и Discoverer. Первые два из перечисленных используются для получения оценок рисков безопасности базы данных, а Discoverer — для обнаружения конфиденциальных данных в базе данных.

Collector собирает информацию о конфигурации безопасности из базы данных и базовой операционной системы сервера БД. Затем Reporter анализирует собранные данные и формирует отчет с подробными выводами и рекомендациями. Отчеты представлены в форматах HTML, XLS, TXT и JSON. Модуль Discoverer помогает идентифицировать конфиденциальные данные, изучая метаданные таблиц (комментарии и имена столбцов), классифицирует и обобщает полученные результаты в отчетах HTML и XLS.

Отчеты в формате HTML предоставляют подробные результаты оценки в удобном для навигации формате. Формат XLS предоставляет высокоуровневую сводку по каждому обнаружению, чтобы пользователи могли добавлять столбцы для целей отслеживания и приоритизации работ по устранению выявленных недостатков. Отчет в текстовом формате позволяет удобно копировать части выходных данных для иного использования. Вывод JSON удобен для целей агрегации и интеграции данных.

Database Security Assessment Tool



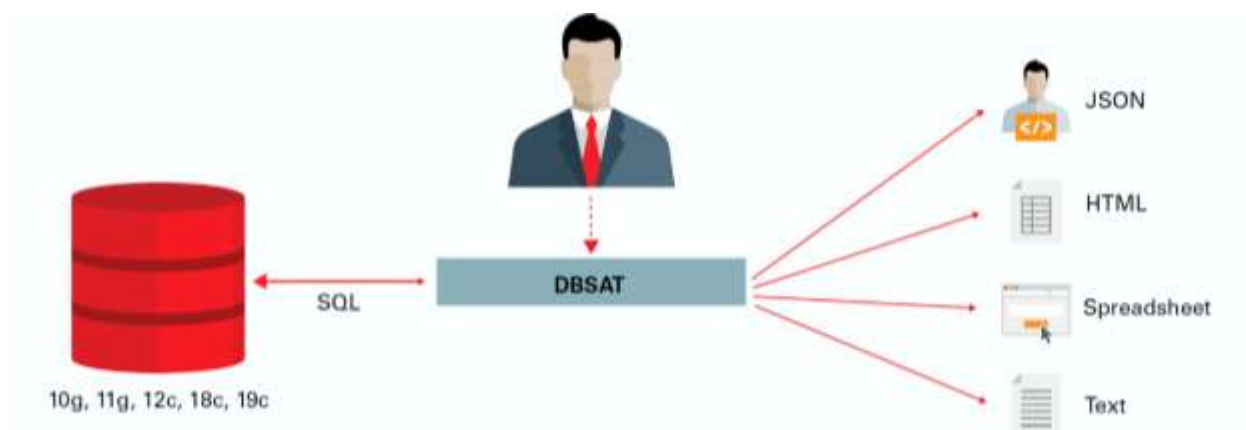


Рис. 2. Схема выполнения запросов и формирования отчетов утилитой DBSAT

Резюме оценки безопасности баз данных (вид итогового отчета в формате HTML)

Assessment Date & Time

Date of Data Collection	Date of Report	Reporter Version
Fri Mar 01 2019 11:43:00	Fri Mar 01 2019 11:44:28	2.1 (March 2019) - 7a38

Database Identity

Name	Container (Type:ID)	Platform	Database Role	Log Mode	Created
ORCLCDB	ORCL (PDB:3)	Linux x86 64-bit	PRIMARY	NOARCHIVELOG	Mon Oct 15 2018 15:36:00

Summary

Section	Pass	Evaluate	Advisory	Low Risk	Medium Risk	High Risk	Total Findings
Basic Information	0	0	0	0	0	1	1
User Accounts	5	0	0	4	2	1	12
Privileges and Roles	5	16	0	0	0	0	21
Authorization Control	0	1	1	0	0	0	2
Fine-Grained Access Control	0	1	4	0	0	0	5
Auditing	0	4	2	0	6	0	12
Encryption	0	1	1	0	0	0	2
Database Configuration	5	3	0	3	2	1	14
Network Configuration	1	1	0	0	3	0	5
Operating System	1	0	0	2	1	1	5
Total	17	27	8	9	14	4	79

Рис. 3. Пример отчета DBSAT в формате HTML

Результаты анализа в отчете представлены блоками, называемыми выводами. Отчет содержит следующие элементы.

- Status указывает на уровень риска, связанного с обнаружением (Pass, Low Risk, Medium Risk, High Risk), или на рекомендацию для улучшения, например информацию о дополнительной функции безопасности, которая в настоящее время не используется. В тех случаях, когда требуется дальнейший анализ, статус отображается как оценка (Evaluate).
- Summary — краткое изложение результатов обследования.



- Details — подробные сведения о результатах, за которыми следуют рекомендации.
- Remarks — причина регистрации и рекомендуемые действия по исправлению положения.
- References — когда это применимо, ссылка на соответствующую рекомендацию CIS Oracle Database 12c benchmark, правило Oracle Database STIG и статью или отчет EC GDPR.

В приведенном ниже отчете DBSAT определил, что существует пять пользователей с ролью DBA и необходим дальнейший анализ (Status = Evaluate). Эти замечания содержат дополнительную информацию о том, почему важно ограничить использование этой роли небольшим числом доверенных администраторов. Ссылки помечают этот вывод как CIS Oracle Database 12c Benchmark, рекомендация 4.4.4.

DBA Role

PRIV.DBA

CIS

Status

Evaluate

Summary

5 grants of DBA role.

Details

Grants of DBA role:

DEBRA <- APP_ROLE: DBA

OUTSRC_DBA: DBA

SCOTT: DBA

SSHADMIN: DBA

SYSTEM: DBA

Remarks

The DBA is a powerful role and can be used to bypass many security controls. It should be granted to a small number of trusted administrators. As a best practice, it is recommended to create custom DBA-like roles with minimum set of privileges that users require to execute their tasks (least privilege principle) and do not grant the DBA role. Privilege Analysis can assist in the task of identifying used/unused privileges and roles. Having different roles with minimum required privileges based on types of operations DBAs execute also helps to achieve Separation of Duties. Furthermore, each trusted user should have an individual account for accountability reasons. Avoid granting the DBA or custom DBA-like powerful roles WITH ADMIN option unless absolutely necessary. Please note that Oracle may add or remove roles and privileges from the DBA role.

References

CIS Oracle Database 12c Benchmark v2.0.0: Recommendation 4.4.4

Рис. 4. Фрагмент отчета DBSAT. Информация о владельцах роли DBA, пояснения и ссылки на рекомендации

Оценка безопасности баз данных в облаке

Пользователь базы данных в Oracle Cloud может использовать Data Safe, облачную службу безопасности данных, которая предоставляет полный набор возможностей по обеспечению безопасности, включая оценку пользователей и безопасности базы данных в целом. Тесно интегрированные возможности оценки помогают эффективно выявлять и устранять уязвимости.

Заключение

Лучший совет заключается в том, что вы должны планировать внедрение набора средств контроля безопасности, соответствующих важности данных для бизнеса, с учетом актуальных угроз безопасности. Предположим, что вы можете назначить различные уровни важности для ваших данных и систем: от бронзы до серебра, золота и платины. Бронзовые системы могут включать внутренние порталы, каталоги сотрудников и вики-сайты. Серебряные системы могут включать в себя системы бизнес-транзакций, информацию о поставщиках и каталоги запчастей. Системы Gold могут включать данные, подлежащие



соблюдению нормативных требований, будь то GDPR EC, CCPA, PII, PCI, HIPAA или SOX. Платиновые системы могут включать в себя конфиденциальные данные и информацию ограниченного доступа, включая квартальные итоги продаж, прогнозы продаж, деятельность по слияниям и поглощениям и интеллектуальную собственность, такую как исходный код программного обеспечения.



Рис. 5. Различные уровни важности информационных систем и рекомендуемые механизмы их защиты

В любом случае вы должны иметь правильную стратегию, которая учитывает общие бизнес-цели, ресурсы и время, имеющиеся в наличии. Таким образом мы можем защитить как зубные щетки, так и бриллианты с соответствующим уровнем безопасности, получая максимальную отдачу за свои усилия по обеспечению безопасности.

Архитектура обеспечения максимальной безопасности баз данных Oracle

Субъекты угроз в поисках уязвимостей могут использовать различные подходы.

- Кража учетных данных администратора или привилегированного пользователя приложения с помощью фишинга на основе электронной почты и других форм социальной инженерии или с помощью вредоносных программ для поиска учетных данных и данных.
- Доступ к файлам базы данных, которые не зашифрованы на физическом носителе информации.
- Использование слабых мест в приложениях с помощью таких методов, как SQL-инъекция и обход безопасности прикладного уровня путем встраивания SQL-кода в кажущийся безобидным ввод данных, предоставляемый конечным пользователям.
- Повышение привилегий за счет использования уязвимых приложений.
- Кража носителей, содержащих резервные копии баз данных.
- Копирование реальных данных из систем разработки и тестирования, где они обычно не так хорошо защищены, как в производственных системах.
- Избыточный доступ к конфиденциальным данным с помощью приложений, которые непреднамеренно предоставляют информацию в объеме, превышающем тот, что пользователи должны использовать для выполнения своих служебных обязанностей.



- Использование незащищенных систем или неправильно сконфигурированных баз данных для обхода контроля доступа.

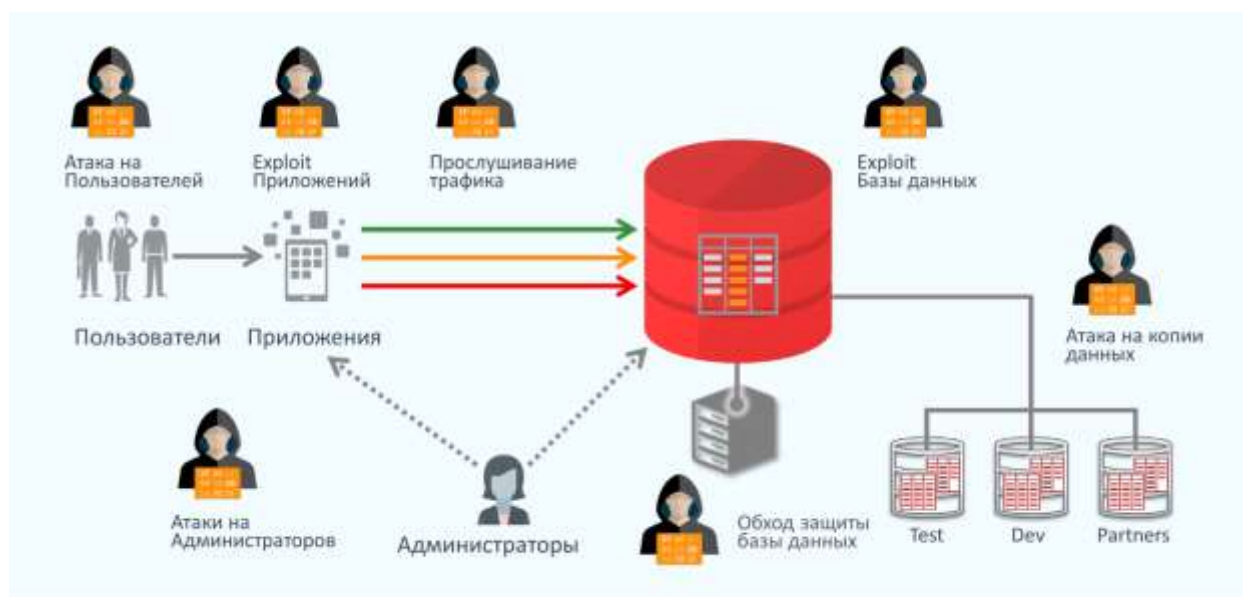


Рис. 1. Потенциальные атаки субъектов угроз

Мы должны защитить данные пользователя и работу приложений от этих угроз. Oracle предоставляет широкий набор средств для обеспечения максимальной безопасности баз данных Oracle. Этот набор средств реализует архитектуру максимальной безопасности баз данных Oracle. Рассмотрим ее элементы подробнее.

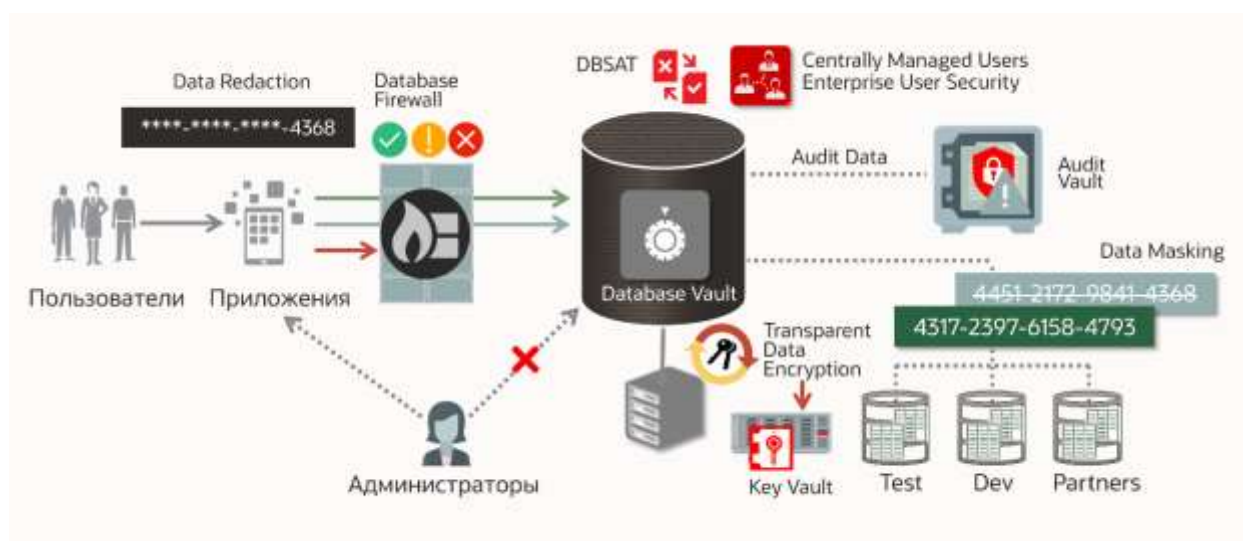


Рис. 1. Архитектура максимальной безопасности баз данных Oracle

Управление привилегированными пользователями, ORACLE DATABASE VAULT

Кража конфиденциальных данных с использованием скомпрометированных привилегированных учетных записей является одним из наиболее распространенных векторов атаки.

Привилегированные пользователи обычно имеют мощные системные привилегии, которые дают им неограниченный доступ к базе данных, поэтому они могут легко управлять базой данных 24x7, будь то для управления производительностью, диагностики и настройки, обновления и исправления, запуска



и остановки базы данных или резервного копирования базы данных. Однако это также дает им полный доступ ко всем конфиденциальным данным в базе данных, таким как зарплата, номера социального страхования, финансовые прогнозы компаний и интеллектуальная собственность.

Поскольку большинство организаций доверяют своим DBA и опытным пользователям, злоумышленники часто пытаются получить доступ к учетным записям привилегированных пользователей, чтобы получить полный доступ ко всем данным.

Обычно доступ к конфиденциальным данным разрешен только владельцу схемы и пользователям с прямыми (на основе соответствующего гранта) правами доступа к объектам. Однако и привилегированные пользователи имеют доступ к этим конфиденциальным данным через, например, системную привилегию `SELECT ANY TABLE`. Сложные приложения с несколькими схемами также часто используют системные привилегии вместо привилегий объектов для удобства, что позволяет злоумышленникам при атаке с использованием SQL-инъекций получать доступ к данным всей базы данных.

Поэтому необходим набор специализированных механизмов безопасности в базе данных для обеспечения ограничений на доступ привилегированных пользователей к схемам с конфиденциальными данными, таблицам и представлениям.

Oracle Database Vault (DBV), впервые появившийся в Oracle Database 9i, ограничивает доступ привилегированных пользователей или DBA с системными привилегиями, такими как `SELECT ANY TABLE`, к конфиденциальным данным. Он вводит понятие области (Realm), которая по существу является списком схем или конкретных объектов, доступ к которым может получить только владелец объекта и те, кто имеет прямые объектные гранты. Таким образом, администраторы баз данных по-прежнему будут управлять базой данных, но не смогут получить доступ к конфиденциальным схемам или объектам, защищенным такими областями.

На следующем рисунке показан пример области, защищающей конфиденциальные данные в схеме отдела кадров (HR). Oracle Database Vault не позволяет DBA получить доступ к данным в области HR, но все же позволяет ему выполнять свои задачи в части администрирования базы данных.



Рис. 2. Oracle Database Vault использует Realm для защиты конфиденциальных данных

Однако есть и другие ситуации, когда нежелательно, чтобы владелец схемы или DBA имел возможность предоставлять доступ к определенным данным. Поэтому можно возложить ответственность за предоставление доступа не на DBA или разработчиков, а на специальных администраторов безопасности. Тогда по умолчанию никому не будет разрешен доступ до тех пор, пока администраторы безопасности не разрешат его. Механизмы защитных областей также полезны в случаях, когда администратору необходимо внести изменения во время обновления приложения. При выполнении обновлений, когда требуется внести изменения только в процедуры и функции приложений, а не в содержащие конфиденциальные данные таблицы. В этом случае защита может быть установлена вокруг защищаемых таблиц и представлений и не запрещать доступа к обновлению процедур.



Для реализации описанных выше вариантов использования Oracle Database Vault поддерживает обязательные области (mandatory realms), которые блокируют доступ к защищенным данным даже владельцу объекта или другим пользователям с прямыми правами доступа к объекту. Единственный способ получить доступ к таким защищенным данным — это авторизоваться в обязательной области. В этом примере администратору приложения предоставляется доступ к обычной области хранилища баз данных Oracle, защищающей всю схему для обновления, но обязательная область помещается вокруг таблиц конфиденциальных данных, чтобы запретить администратору доступ к конфиденциальным данным. Это безопаснее, чем временно отказаться от защиты конфиденциальных объектов данных или разрешить доступ ко всем конфиденциальным объектам.

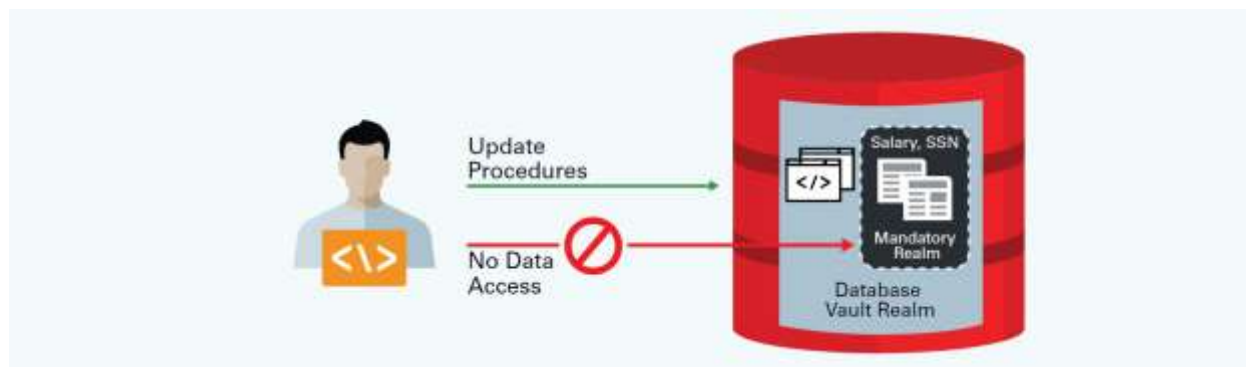


Рис. 3. Обязательные области в Database Vault

Единственный пользователь, который может предоставить доступ к данным, — это сотрудник службы безопасности, а не владелец данных. Это упрощает определение того, какие пользователи имеют доступ, поскольку аудиторам не нужно проходить потенциально сложную цепочку грантов, предоставленных различным пользователям и ролям. Все разрешения на доступ к объектам поддерживаются как разрешения областей, управляемые сотрудником службы безопасности.

При настройке DBV в базе данных создаются дополнительные роли для разделения обязанностей, включая управление учетными записями (Oracle Database Vault Account Manager) и управление безопасностью баз данных Oracle (Oracle Database Vault Owner).

Роли предназначены для создания пользователей в базе данных вместе с их профилями паролей, управления ими и для создания, изменения элементов безопасности DBV, управления ими, включая возможность управлять областями и добавлять участников областей соответственно.

Администратор базы данных (DBA) по умолчанию не имеет возможности использовать перечисленные роли DBV. Это предотвращает возможность реализации атаки, когда злоумышленник крадет учетные данные DBA, а затем использует эту законную учетную запись для создания мошеннической учетной записи пользователя и предоставления ей высоких привилегий, позволяющих красть конфиденциальные данные.

Отдельная роль для управления средствами безопасности базы данных обеспечивает разделение ответственности между администраторами БД и сотрудниками службы безопасности. Это предотвращает угрозу отключения элементов управления безопасностью и кражу конфиденциальных данных с помощью одной учетной записи DBA или администратора безопасности. Поскольку задачи управления учетными записями и контроля безопасности DBV реализуются через соответствующие роли, организации могут включить DBV и реализовать разделение обязанностей с течением времени, первоначально предоставив эти роли администратору БД.

Общие задачи DBA, такие как настройка производительности, управление памятью, резервное копирование и другие, все еще могут выполняться командой DBA (администраторы БД и администраторы приложений). Некоторые задачи DBA, которые могут предоставить DBA возможность доступа к конфиденциальным данным, такие как экспорт и планирование заданий, потребуют дополнительного шага авторизации этого доступа со стороны команды безопасности.



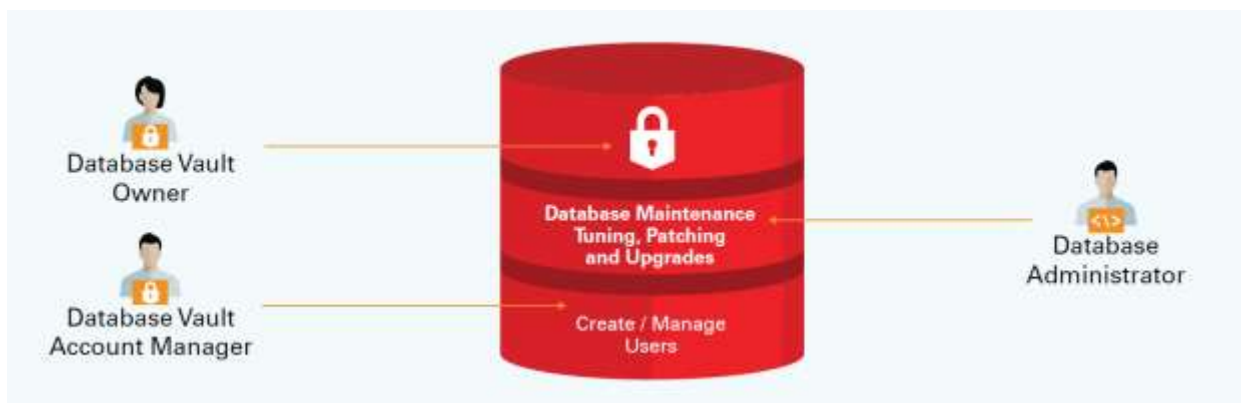


Рис. 4. Разграничение доступа в соответствии с функциональными обязанностями

Привилегированные пользователи в производственных системах не должны удалять таблицы, изменять системные параметры или объекты базы данных вне окон обслуживания. Изменение данных вне контроля приложения также может быть регламентировано. Администраторы БД на своем рабочем месте часто имеют одновременно несколько средств управления (открытые окна терминалов) в различных системах разработки, тестирования и производства и потенциально могут случайно запускать разрушающий SQL в производственных системах.

Oracle Database Vault предоставляет возможность создавать правила выполнения SQL-команд для предотвращения случайных или вредоносных изменений в данных и базе данных. Эти детализированные правила могут запрещать выполнение определенных команд или определять время суток, когда команды могут выполняться вместе с допустимым IP-адресом клиента и другими контекстными данными. Например, организации могут ограничить DBA для выполнения определенных системных команд ALTER только со своих рабочих мест и только в рабочее время, чтобы предотвратить несанкционированный удаленный доступ в нерабочее время. Другой пример — ограничить SQL-команды, такие как CREATE, DROP или ALTER TABLE, окнами обслуживания, требовать, чтобы любое изменение было связано с определенной заявкой на обслуживание, или обеспечить, чтобы только при подключении к системе одновременно двоих (исполнитель и контролер) сотрудников команда может быть успешно выполнена.

При эксплуатации DBV необходимо учитывать три ключевых вопроса:

- Что нужно защищать и при каких условиях?
- Какие SQL-команды должны быть разрешены в вашей БД?
- Какие технологические и организационные изменения необходимы в связи с новым разделением обязанностей?

Прежде чем создавать защитные области DBV, необходимо знать, какие данные являются конфиденциальными. Часто по этому поводу возникают возражения, что все пользовательские данные являются конфиденциальными и что защитные области должны автоматически охватывать каждую пользовательскую схему. Начиная с Oracle Database 19c, управление механизмами DBV по умолчанию блокирует доступ обычных пользователей (например, инфраструктурных DBA) к локальным данным в подключаемых базах данных (PDB) в автономных, облачных или локальных средах. Не требуется для этого создавать специальные защитные области.

Чтобы определить, какие команды SQL разрешить, следует провести анализ соответствующих записей аудита, чтобы обнаружить, какие критические команды SQL (например, ALTER SYSTEM, ALTER DIRECTORY и т. д.) были запущены, и определить контекст, в котором они были запущены (IP-адрес, имя программы, имя пользователя и т. д.). Правила выполнения команд могут быть созданы для ограничения разрешенных команд временем суток, входящим IP-адресом и другими правилами для обеспечения внесения изменений в соответствии с политикой.



Однако особое внимание должно быть уделено разделению обязанностей. Точно так же, как инициатором заказов на покупку не может быть тот же человек, который утверждает заказы на покупку, разделение обязанностей с применением DBV обеспечивает безопасность и снижает вероятность нарушения. Некоторые организации предпринимают несколько шагов, прежде чем полностью внедрить режим разграничения в соответствии со служебными обязанностями (Separation of Duties, SoD). Например, поскольку SoD реализуется новыми ролями DBV, новые роли могут быть предоставлены тем же DBA изначально, а затем могут быть предприняты шаги по разделению SoD. Это позволяет быстрее реализовать некоторые элементы управления в DBV и распределить изменения разделения обязанностей на более длительном временном интервале.

Администраторы БД (DBA) могут не решаться включить превентивный контроль DBV, опасаясь, что приложения перестанут нормально функционировать или какие-либо административные скрипты перестанут работать. Начиная с версии Oracle Database 12c, Release 2, в DBV введен режим моделирования (имитации), так что можно проверить правильность функционирования механизмов защитных областей (Realms) и правила выполнения команд SQL, не оказывая никакого влияния на работу приложений и операции с БД. Режим имитации только регистрирует нарушения в журнале, а не блокирует их выполнение, поэтому полный сквозной регрессионный тест может быть выполнен без остановки системы. Затем настройки политик безопасности DBV можно обновить на основе данных журнала моделирования.

В DBV реализованы механизмы внутреннего аудита. Данные этого аудита имеют высокую ценность, поскольку любое предупреждение указывает либо на первоначальное зондирование злоумышленником базы данных на предмет возможности реализации в ней несанкционированных действий, либо на подготовку любопытных пользователей для получения конфиденциальных данных, доступ к которым у них должен быть ограничен.

Шифрование данных

Самая сильная защита, которую только можно вообразить, бесполезна, если нападающий может просто обойти ее. Нападающие не ограничиваются атакой там, где защита самая сильная, они обходят входную дверь, закрытую на засов, и ищут заднюю дверь без засова или даже открытое окно. Аутентификация и авторизация базы данных гарантируют, что данные доступны только тем, кто авторизован, и не доступны никому другому. Если злоумышленник не может получить доступ через базу данных (то есть входную дверь), то он попытается обойти контроль доступа к базе данных и получить доступ к файлам БД, резервным копиям или просто скопировать данные, когда они перемещаются по сети. Злоумышленники могут получить доступ к системе как привилегированный пользователь операционной системы и непосредственно прочитать файлы базы данных (.dbf) в обход механизмов защиты системы управления базами данных.

Одним из простых способов получения доступа к данным злоумышленниками является перехват данных при их перемещении по сети, например между клиентом и сервером баз данных или между серверами баз данных. Во многих сетях злоумышленнику относительно легко перехватить сетевой трафик, а затем извлечь любую информацию, переданную между двумя системами, поскольку многие внутренние сетевые соединения остаются незашифрованными.

Шифрование данных при передаче

Возможность шифрования данных в сети с использованием протокола Transport Layer Security (TLS) или собственного шифрования Oracle (Native Network Encryption, NNE) является стандартной функцией базы данных Oracle. Эта функция может быть сконфигурирована для обеспечения как конфиденциальности (чтобы другие не могли читать данные, передаваемые по сети), так и защиты целостности (чтобы другие не могли изменять или воспроизводить данные). Кроме того, протокол TLS может обеспечить аутентификацию клиента и сервера. Параметры защиты сетевых данных для каждого клиента и сервера Oracle настраиваются в соответствующем файле sqlnet.ora. Применимые настройки для каждой системы включают список криптографических алгоритмов, которые она поддерживает, а также параметры согласования соединений. Когда параметры шифрования и контрольного суммирования включены на сервере базы данных, все клиенты автоматически следуют этим настройкам.



Шифрование данных в БД

Когда информация записывается в базу данных, она хранится в файлах, локальной файловой системе или какой-либо другой форме хранения (например, ASM). Шифрование этих данных гарантирует, что злоумышленник не сможет прочитать информацию непосредственно из этих файлов.

Существует несколько способов шифрования данных: на уровне приложения, на уровне файла или тома или на уровне базы данных. Независимо от уровня, на котором выполняется шифрование, ключевые соображения включают безопасность, производительность, простоту установки и использования, прозрачность для существующих приложений, миграцию данных из открытого текста в зашифрованную форму, патчирование и управление ключами.

Шифрование на уровне приложения осуществляется из кода приложения, где оно зашифровывает данные перед сохранением их в базе данных. Это требует не только, чтобы каждое приложение знало, как зашифровывать и расшифровывать в различных частях потока процесса, но также должно управлять ключами шифрования и надежно хранить их где-то. Если несколько приложений совместно используют информацию в одной и той же базе данных, они должны будут совместно зашифровывать и расшифровывать одни и те же данные одним и тем же способом. Шифрование на уровне приложений ограничивает типы запросов, которые могут выполняться по зашифрованным столбцам. Например, обычные аналитические запросы, соответствующие диапазонам данных или вычисленным значениям, не будут работать. Кроме того, шифрование уровня приложений не выигрывает от использования Oracle Database In-Memory и высокопроизводительной архитектуры Exadata. По существу, шифрование данных на самом высоком прикладном уровне накладывает значительную нагрузку на разработку и управление, поскольку ограничивает выполнение значимых реляционных вычислений над данными.

Шифрование файлов или томов, хотя и удобно, имеет ряд недостатков при использовании для защиты табличных данных в базах данных Oracle. Прежде всего программное обеспечение для шифрования томов не может отличить законного пользователя от нелегального. Это означает, что база данных Oracle, работающая как пользователь Oracle, и пользователь человек (злонамеренный), который также входит в систему как Oracle, будут иметь доступ к расшифрованным данным. Другим большим недостатком является общая неэффективность системы. Например, чтобы перенести redo log из основной базы данных в резервную, сначала данные должны быть расшифрованы в основной базе данных, зашифрованы при транспортировке по сети, расшифрованы в пункте назначения, а затем повторно зашифрованы для хранения в резервной системе. Данные не могут быть просто скопированы в принимающую систему, поскольку и источник и цель зашифрованы с помощью различных собственных ключей шифрования. Наконец, в случае сторонних решений шифрования существует потенциал для введения нестабильности системы и проблем обновления с помощью инвазивных модулей операционной системы и/или файловой системы. Такое стороннее программное обеспечение также может нарушить политику патчирования, не позволяя применять исправления операционной системы до тех пор, пока не будет доступен зависимый патч.

Эти два метода не оптимизированы для шифрования базы данных. Нужен механизм, повышающий безопасность и надежность при минимизации управленческой нагрузки.

TRANSPARENT DATA ENCRYPTION в базах данных Oracle

Transparent Data Encryption (TDE) для баз данных Oracle зашифровывает данные на уровне базы данных. Шифрование прозрачно для авторизованных приложений и пользователей, поскольку база данных автоматически зашифровывает данные перед их записью в БД и автоматически расшифровывает их при чтении. Авторизованные приложения, которые хранят и извлекают данные в базе данных, видят только расшифрованные (или открытые) данные. Поскольку зашифрование и расшифрование происходит автоматически, это не механизм контроля доступа для пользователей базы данных, а скорее способ предотвратить возможность обхода элементов управления базой данных для прямого доступа к данным для привилегированных пользователей операционной системы, администраторов сети и системы хранения или кого-то маскирующегося под них. Авторизованным пользователям базы данных и приложениям не нужно представлять ключ при обработке зашифрованных данных. Вместо этого база данных применяет правила



контроля доступа и запрещает доступ, если пользователь не авторизован для просмотра данных. TDE может шифровать как столбцы таблиц приложений, так и табличные пространства со всеми данными приложения. При шифровании табличных пространств не нужно контролировать то, какие столбцы следует зашифровать, а также их характеристики, такие как индексы и ограничения.

Создание зашифрованного табличного пространства очень просто, как показано ниже.

```
CREATE TABLESPACE investigations  
DATAFILE '$ORACLE_HOME/dbs/investigations.dbf'  
ENCRYPTION USING 'AES256' DEFAULT STORAGE (ENCRYPT);
```

TDE автоматически использует специальные инструкции в процессорах Intel (AES-NI) и SPARC для ускорения криптографических операций. Шифрование табличного пространства TDE легко интегрируется с оптимизацией производительности, встроенной в инженерные системы Oracle. Все системы Oracle под ключ используют процессоры с аппаратным ускорением шифрования, а также прозрачного шифрования данных, шифрование табличного пространства не мешает сжатию Exadata (EHCC) и технологии интеллектуального сканирования.

С помощью TDE конфиденциальные данные остаются зашифрованными во всей базе данных, будь то файлы хранения табличных пространств, временные табличные пространства или табличные пространства undo tablespaces, а также другие файлы, такие как redo logs. Кроме того, TDE может шифровать все резервные копии баз данных и экспорт Data Pump. Oracle Recovery Manager (RMAN) и Data Pump Export или Import интегрированы с TDE.

Миграция незашифрованных данных

В настоящее время TDE стал одним из наиболее хорошо зарекомендовавших себя способов защиты, чтобы данные при хранении оставались зашифрованными. Однако большинство клиентов имеют терабайты незашифрованных критичных данных в своих БД, и им нужен способ быстро зашифровать их без каких-либо простоев.

Онлайн-шифрование табличных пространств, введенное в Oracle Database 12c, Release 2, обеспечивает нулевое время простоя при переходе к зашифрованным данным или при преобразовании из одного алгоритма шифрования в другой. Однако он временно требует дополнительного хранилища того же размера, что и обрабатываемое табличное пространство. Существующее табличное пространство можно легко зашифровать в режиме онлайн с помощью команды, аналогичной следующей:

```
ALTER TABLESPACE online_accounts ENCRYPTION ONLINE using 'AES256' ENCRYPT;
```

Существует два автономных метода шифрования, которые не требуют дополнительного места хранилища. Оба автономных метода шифрования зашифровывают данные с помощью AES128. Они требуют, чтобы табличное пространство было автономным или база данных находилась в режиме монтирования. В Oracle Database 12.2.0.1 и более поздних версиях одно табличное пространство может быть зашифровано с помощью команды, аналогичной следующей:

```
ALTER TABLESPACE court_documents ENCRYPTION OFFLINE ENCRYPT;
```

Кроме того, несколько файлов данных, принадлежащих любому числу табличных пространств приложений, могут быть параллельно зашифрованы в автономном режиме. Файл(ы) данных существующего табличного пространства в Oracle Database 11.2.0.4, 12.1.0.2 и более поздних версиях может быть легко преобразован в автономном режиме с помощью команды, аналогичной следующей:

```
ALTER DATABASE DATAFILE '$ORACLE_BASE/oradata/investigations.dbf' ENCRYPT;
```

В средах Data Guard, где резервные базы данных уже находятся в режиме монтирования, для минимизации времени простоя можно использовать автономное шифрование. Процедура состоит в том, чтобы отключить управляемый процесс восстановления, затем зашифровать файлы данных резервной базы данных, выполнить



переключение и зашифровать бывшую основную базу данных. В этом случае время простоя для шифрования базы данных любого размера равно длительности одного или двух переключений.

Чтобы гарантировать, что все новые табличные пространства зашифрованы по умолчанию, установите системный параметр `encrypt_new_tablespaces` в ALWAYS. Обратите внимание, что такие табличные пространства будут зашифрованы ключом AES 128, но позже они могут быть зашифрованы с ключом AES 256 или с использованием любого другого поддерживаемого алгоритма шифрования с помощью онлайн-шифрования табличных пространств. Системный параметр `encrypt_new_tablespaces` доступен из баз данных Oracle 11.2.0.4, 12.1.0.2 и более поздних версий.

Двухуровневая ключевая архитектура Oracle TDE

Безопасность зашифрованных данных зависит от сохранения секретности ключей, используемых для шифрования. Правильное управление ключами необходимо для того, чтобы злоумышленник не обнаружил ключ шифрования и не получил доступа к данным. Также крайне важно обеспечить безопасное хранение активных ключей, а также всех ранее использованных ключей, периодическую ротацию ключей и надежное архивирование старых ключей для обеспечения постоянного доступа к резервным наборам данных. Многие правила, такие как правила, разработанные индустрией платежных карт (PCI), требуют физического разделения между зашифрованными данными и ключами шифрования, а также периодической ротации ключей шифрования.

Прозрачное шифрование данных (TDE) использует двухуровневую архитектуру ключей для создания ключей, используемых для шифрования, и управления ими. Главный ключ шифрования (MEK) для шифрования внутренних ключей шифрования данных (DEK), которые используются для шифрования столбцов и табличных пространств. Внутренние ключи формируются базой данных. Двухуровневая архитектура ключей упрощает процедуру ротации ключей и таким образом делает более практичной более частую ротацию мастер-ключей. Когда MEK заменяется новым MEK, нет необходимости повторно зашифровывать все данные, только гораздо меньший набор внутренних ключей. Если требуется произвести ротацию ключа DEK в дополнение к MEK, используется онлайн-шифрование табличного пространства.

Администраторы выполняют операции управления ключами (создание, открытие, ротацию, резервное копирование и т. д.) либо с помощью серии команд SQL, либо через Oracle Enterprise Manager. Начиная с Oracle Database 12c, пользователи могут подключаться к базе данных для выполнения операций управления ключами как SYSKM вместо SYSDBA. Административная привилегия SYSKM поддерживает разделение обязанностей, предоставляя возможность выполнять все ключевые задачи управления, но без уровня широкого доступа, предоставляемого SYSDBA.

Главный ключ шифрования отделен от зашифрованных данных, хранится вне базы данных и непосредственно управляется администратором безопасности базы данных в хранилище ключей. Хранилище может быть локальным, таким как Oracle wallet, или централизованным, например Oracle Key Vault.

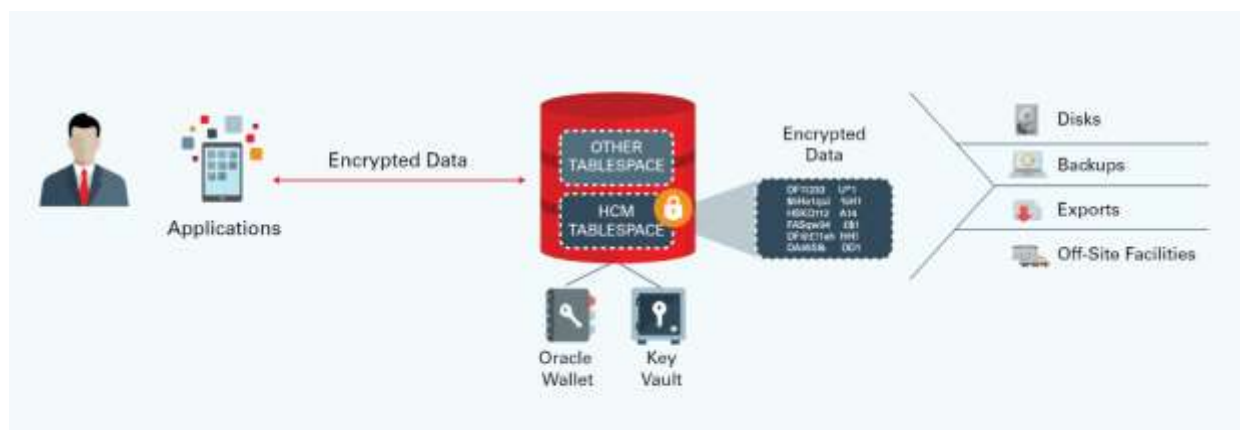


Рис. 5. Oracle Transparent Data Encryption



TDE запрещает привилегированным пользователям операционной системы напрямую получать доступ к конфиденциальной информации, обходя элементы управления и защиты БД, непосредственно читать содержимое файлов базы данных. Он также защищает от кражи, потери или неправильного вывода из эксплуатации носителей данных и резервных копий. Oracle Key Vault облегчает управление накладными расходами при массовом развертывании TDE, предоставляя централизованную платформу управления ключами. Шифрование и централизованное управление ключами имеют решающее значение для соблюдения правил конфиденциальности и стандартов, таких как PCI-DSS, HIPAA и EU GDPR.

Маскирование статическое и динамическое

Контроль доступа к конфиденциальным данным является проблемой, с которой сталкиваются многие организации. Например, копии текущих производственных данных обычно совместно используются средами разработки и тестирования, чтобы эти команды имели доступ к реалистичным данным для поддержки своей деятельности. Эти данные могут включать номера кредитных карт, номера социального страхования, данные о продажах и другую личную или служебную информацию. Злоумышленники знают, что такого рода информация часто оказывается в системах тестирования и разработки, где она обычно менее защищена, чем в производственных системах, что делает эти системы привлекательными объектами для атак.

Даже в производственных системах организации должны ограничить доступ к конфиденциальным данным таким образом, чтобы информация, отображаемая различным пользователям, ограничивалась минимумом, необходимым им для выполнения своей работы. Например, в случае заявки на обслуживание клиентов представителю колл-центра может потребоваться доступ только к последним четырем цифрам, например, номера социального страхования клиента для проверки его личности. Хотя система может потребовать номер социального страхования для других целей, таких как кредитная отчетность, отображение полного номера социального страхования сотруднику колл-центра через пользовательский интерфейс приложения нарушает принцип наименьших привилегий и создает ненужный путь для компрометации этой информации.

В обоих случаях организациям необходимо маскировать конфиденциальные данные, чтобы свести к минимуму воздействие и связанный с этим риск. Oracle позволяет использовать 2 способа маскирования данных: статическое маскирование (Data Masking) и динамическое маскирование (редактирование данных — Data Redaction).

Data Masking производит постоянное изменение данных, чтобы они больше не были конфиденциальными. Маскирование конфиденциальных данных перед их передачей командам тестирования и разработки устраняет риск утечки данных в непроизводственных средах, необратимо заменяя исходные конфиденциальные данные реалистичными, но вымышленными данными. Напротив, редактирование данных подменяет данные на лету, когда они извлекаются сеансом базы данных, но фактически не изменяет хранимое в БД значение данных. Data Redaction ограничивает раскрытие конфиденциальных данных в пользовательских интерфейсах приложений и помогает снизить риски раскрытия информации.

Другим связанным требованием является извлечение и совместное использование части или подмножества данных вместо совместного использования всего производственного набора данных. Например, команде разработчиков может потребоваться только 20 % производственных данных: данные, собранные только за последний год, или данные, относящиеся к конкретному региону. Подмножество данных снижает риск безопасности и минимизирует затраты на хранение, удаляя ненужные данные из тестовой базы данных.

Маскирование конфиденциальных данных (статическое)

Oracle Data Masking and Subsetting, пакет Oracle Enterprise Manager, обеспечивает маскирование и создание подмножества данных для задач разработки и тестирования. После создания модели данных приложения, состоящей из списка столбцов критичных данных и ссылочных связей, Oracle Data Masking используется для задания способа маскирования, определяющего правила маскирования конфиденциальных данных. Например, номера кредитных карт и номера социального страхования могут быть заменены вымышленными значениями, имена и даты рождения могут быть перемешаны, а зарплаты могут быть рандомизированы



в пределах заданного диапазона. Полученный после выполнения процедуры маскирования по заданным правилам набор данных может быть использован для непроизводственных целей, таких как разработка, тестирование и аналитика. Маскирование данных помогает свести к минимуму распространение конфиденциальных данных, сохраняя при этом удобство использования данных.

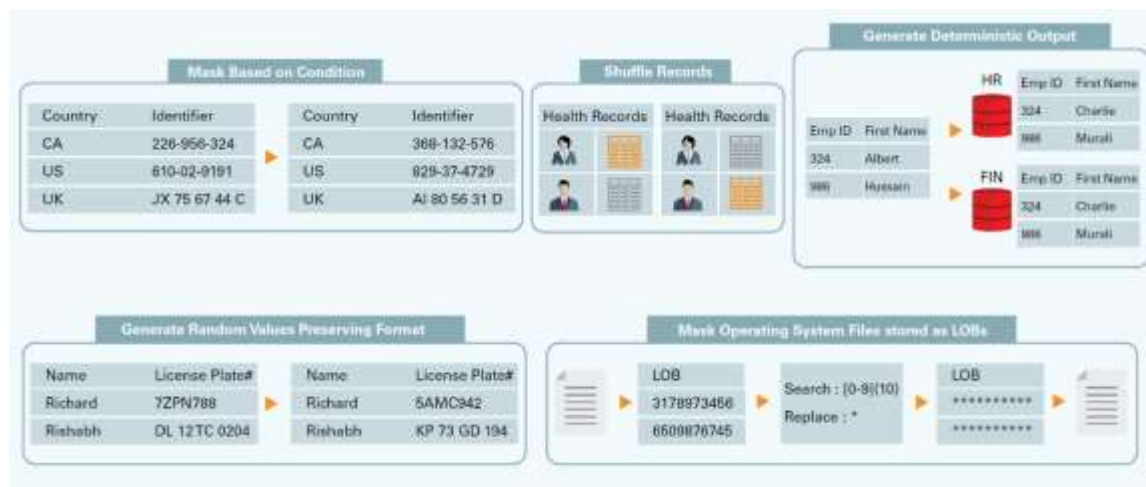


Рис. 6. Варианты маскирования данных

Компонент Data Subsetting Oracle Data Masking and Subsetting также использует модель данных приложения для создания сокращенного набора исходных данных с сохранением ссылочной целостности. Например, большой (>2 ТБ) набор данных может быть источником для создания меньшего (100 ГБ) набора данных для тестирования и разработки, что позволяет эффективно выполнять эти задачи при меньшем потреблении ресурсов. При урезании учитываются описанные в модели и в декларативных ограничениях БД связи между данными. Как показано на рис. 7, Data Subsetting поддерживает различные варианты использования, такие как извлечение части большой таблицы для разработки приложений, извлечение данных, принадлежащих определенному региону для аналитики, или извлечение строк, принадлежащих определенному разделу или подразделу таблицы.

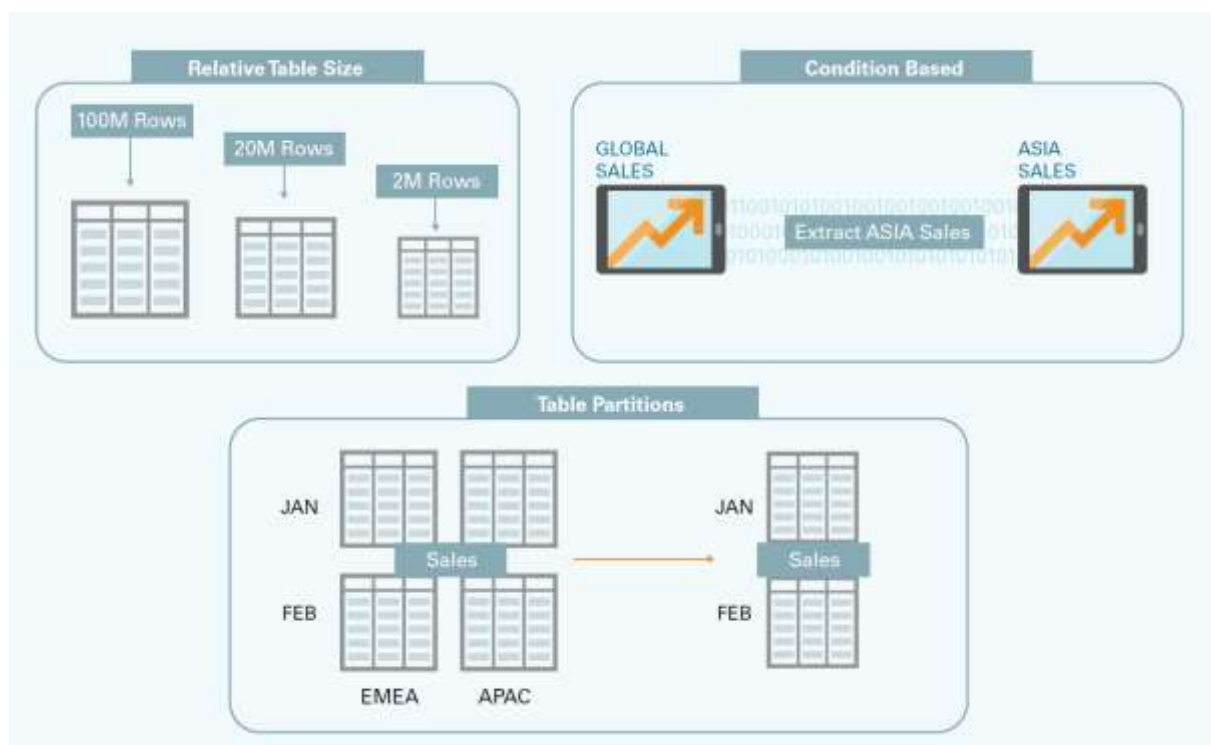


Рис. 7. Примеры создания сокращенных наборов данных



Пользователи также могут выполнять одновременное маскирование и урезание данных для извлечения подмножества производственных данных, а затем маскировать конфиденциальные данные в этом подмножестве. Это максимизирует деловую ценность производственных данных без ущерба для конфиденциальной информации или потери ресурсов.

Редактирование конфиденциальных данных (динамическое маскирование)

Растет потребность в контроле за отображением конфиденциальных данных, содержащихся в приложениях. Например, приложение колл-центра может иметь страницу, которая предоставляет операторам колл-центра кредитную карту клиента и другую личную информацию. Раскрытие этой информации даже законным пользователям приложений может нарушить правила конфиденциальности и подвергнуть данные ненужному риску, открыв их тем, кому это не нужно знать.

Одним из распространенных способов ограничения конфиденциальных данных, отображаемых в приложении, является их редактирование. При редактировании отображаемые данные могут быть полностью скрыты, заменены случайными данными или специальные символы могут быть применены к части извлеченного элемента данных, например к первым двенадцати цифрам номера кредитной карты или части имени пользователя адреса электронной почты.

Организации редактируют конфиденциальные данные, отображаемые на экранах приложений, чтобы лучше защитить личную информацию и соблюдать правила. Однако реализация такого способа защиты внутри кода приложения может быть трудной для поддержания. Кроме того, могут возникать разночтения в данных в масштабах предприятия, особенно в консолидированных средах, где несколько приложений обращаются к одним и тем же данным.

Наилучший подход для контроля доступа к конфиденциальным данным в приложениях заключается в применении политик редактирования внутри производственной базы данных по мере того, как данные передаются приложению в реальном времени.

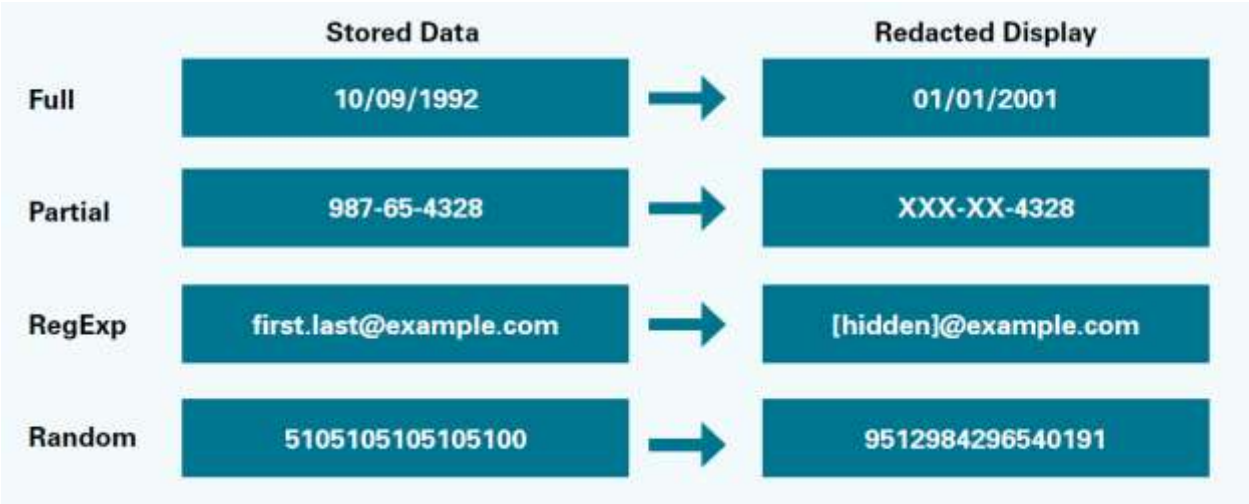


Рис. 8. Варианты динамического маскирования

В отличие от традиционных подходов, основанных на внесении изменений в алгоритмы приложений, политики редактирования данных применяются непосредственно в ядре базы данных Oracle, обеспечивая согласованность между модулями приложений и повышая безопасность. Политики редактирования хранятся и управляются внутри базы данных, и они вступают в силу сразу же после включения. Они могут применяться ко всем пользователям базы данных или выборочно в зависимости от среды пользователя или других факторов.



Аудит за действиями пользователей

Если предположить, что средства контроля доступа правильно настроены и предоставление привилегий сведено к минимуму, все еще остаются два важных риска. Первый риск заключается в том, что пользователи, которым требуются значительные привилегии для выполнения своей работы, могут злоупотреблять этими привилегиями. Второй риск заключается в том, что хакер может завладеть этими учетными записями пользователей и злоупотреблять ими. Поэтому очень важно проводить аудит деятельности этих пользователей, чтобы знать, кто что делал и когда, что позволяет организациям соблюдать строгий контроль безопасности и требования соответствия. Аудит является основным инструментом для отслеживания этих действий пользователей.

Для того чтобы данные аудита использовались в целях обеспечения безопасности и соответствия требованиям, они должны представлять собой надежную запись событий, которую нельзя удалить или которой нельзя манипулировать для сокрытия подозрительных действий. Поскольку для обнаружения многих нарушений в базах данных требуются месяцы, важно, чтобы данные аудита хранились как можно дольше для анализа прошлых событий. Наконец, нужны простые способы поиска собранных данных аудита для поиска конкретной информации для экспертизы или обнаружения аномалий.

Чтобы обеспечить разделение обязанностей, Oracle предоставляет две отдельные роли, связанные с аудитом: `AUDIT_ADMIN` и `AUDIT_VIEWER`. Роль `AUDIT_ADMIN` позволяет пользователям создавать политики аудита, использовать инструкции `SQL AUDIT` и `NOAUDIT`, просматривать данные аудита и управлять журналом аудита. Роль `AUDIT_VIEWER` ограничена просмотром и анализом аудиторских данных.

Данными аудита можно управлять только с помощью встроенного пакета управления данными аудита в базе данных, который не может быть непосредственно обновлен или удален с помощью команды `SQL UPDATE` или `DELETE`.

Существует много случаев, когда клиенты имеют большое количество баз данных Oracle и других баз данных, которые генерируют большой объем данных аудита. Требования регуляторов и лучшие практики обеспечения безопасности диктуют, что данные аудита должны направляться в удаленное централизованное хранилище, где они защищены от вмешательства со стороны лиц, деятельность которых подвергается аудиту. Кроме того, важно иметь возможность эффективно отслеживать текущий поток данных аудита, чтобы находить конкретные события, имеющие последствия для безопасности, и выявлять проблемы, требующие немедленного реагирования.

Продукт Oracle Audit Vault and Database Firewall (AVDF) упрощает сбор записей аудита в защищенном хранилище. AVDF собирает данные аудита не только из баз данных Oracle, но и из баз данных Oracle MySQL, Microsoft SQL Server, SAP Sybase, PostgreSQL, MongoDB и IBM DB2 for LUW, а также из широко используемых платформ операционных систем, таких как Linux, Solaris, Windows и AIX. AVDF также планирует расширить поддержку сбора данных аудита для Hadoop и NoSQL.

Сетевой мониторинг активности БД

Широкое использование приложений сделало их точкой входа номер один для хакеров. Различные исследования пришли к выводу, что значительный процент утечек данных вызван неправильным использованием учетных данных инсайдеров, скомпрометированных клиентов или с помощью SQL-инъекций. SQL-инъекция уже несколько лет является угрозой безопасности приложений номер один в топ-10.

Трехуровневая архитектура приложений в сочетании с растущим числом атак на базы данных с помощью SQL-инъекций или инсайдеров с доступом к привилегированным учетным записям сделала сетевой мониторинг активности баз данных важным компонентом общей архитектуры безопасности. При почти нулевых накладных расходах на отслеживаемые базы данных сетевой мониторинг активности баз данных может осуществляться независимо от платформы, не требует модификации баз данных и может одновременно контролировать несколько гетерогенных систем баз данных. Сетевой мониторинг активности базы данных должен использоваться в сочетании с аудитом базы данных, который может захватывать



не только локальную активность в базе данных, но и любую активность базы данных, которая не пересекает сеть как SQL, например протоколирование локальных или удаленных консольных подключений для внесения изменений в базу данных и пользователей. Эффективное решение для мониторинга активности баз данных должно сочетать локальный аудит баз данных с сетевым мониторингом активности баз данных для обнаружения вторжений.

Oracle Database Firewall предоставляет комплексное, масштабируемое и гибкое решение для мониторинга и защиты систем баз данных. Database Firewall, компонент Oracle Audit Vault and Database Firewall, выступает в качестве первой линии сетевой обороны базы данных, помогая предотвратить SQL-инъекции и другую вредоносную активность. Database Firewall может блокировать определенные типы инструкций SQL и заменять их безопасными инструкциями, которые предписаны политикой безопасности. На основе этой политики Database Firewall также может прервать подключение приложения к базе данных.

Реализация систем высокой доступности средствами Oracle

Сегодня ИТ-инфраструктура стала для предприятий средством достижения конкурентных преимуществ, повышения производительности и инструментом, позволяющим пользователям принимать более оперативные и обоснованные решения. Однако со всеми этими преимуществами пришла и постоянно растущая зависимость предприятий от ИТ-инфраструктуры. При нарушении доступности ключевого приложения, сервера или данных под угрозой оказывается весь бизнес. Компания может потерять доходы и клиентов, подвергнуться штрафам, а негативная реакция прессы может привести к отрицательным долгосрочным последствиям для клиентов и репутации компании. Построение ИТ-инфраструктуры высокой готовности является важнейшим условием успеха любого предприятия в современной динамичной экономике.

Причины простоев

Одной из задач при разработке ИТ-инфраструктуры высокой готовности является исследование и анализ всех возможных причин простоев. На рисунке ниже показана классификация простоев, разделенных на две основные категории: плановые и внеплановые. Для разработки отказоустойчивой и надежной ИТ-инфраструктуры важно учитывать причины обоих типов простоев.

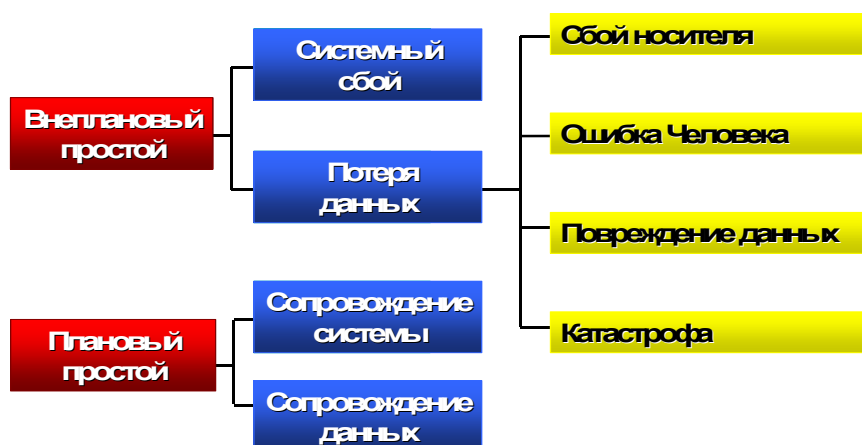


Рис. 1. Причины простоя БД



Внеплановые простои в основном являются результатом отказов компьютеров, сбоев операционных систем или другого программного обеспечения, а также потери данных из-за повреждений носителей или человеческих ошибок.

Плановые простои, как правило, вызываются необходимостью выполнения сервисных работ на эксплуатируемых системах.

В следующих разделах будут последовательно рассмотрены возможные причины простоев и приведены технологии Oracle, используемые для исключения или минимизации простоя, если данная причина имела место.

Защита от системного сбоя

Одна из наиболее распространенных причин незапланированных простоев — сбой или отказ вычислительной системы. Отказы систем являются результатом аппаратных неисправностей, сбоев электропитания, сбоев операционных систем или серверов. С такого рода простоями лучше всего бороться с помощью средств быстрого аварийного восстановления баз данных и кластерных технологий.

Oracle Real Application Clusters

Real Application Clusters (RAC) — это самая первая технология кластеризации баз данных, которая позволяет объединить несколько независимых аппаратных серверов (узлов кластера) в пул серверов базы данных для одновременного доступа к единой, доступной каждому узлу кластера базе данных. Экземпляры СУБД Oracle работают каждый на отдельном узле кластера, но с точки зрения приложения это единая СУБД. Такая архитектура расширяет возможности высокой готовности и масштабирования приложений.

- Отказоустойчивость приложений при аппаратных сбоях.
- Эластичность и экономическая эффективность при планировании мощностей, чтобы производительность систем могла соответствовать требованиям бизнеса.

Ключевым преимуществом Real Application Clusters является отказоустойчивость за счет использования нескольких узлов в кластере. Каждый физический узел работает независимо, выход из одного или нескольких узлов не влияет на другие узлы. Эта архитектура позволяет включать или выключать узлы кластера, в то время как остальная часть пула серверов продолжает предоставлять сервис базы данных.

Также Real Application Clusters дает пользователям эластичность при добавлении новых узлов в кластер, тем самым снижая затраты на оборудование, так как при возрастании требований к ресурсам нет необходимости полностью заменять существующую систему новой, более мощной, но при этом и более дорогой.

Oracle Clusterware

Программное обеспечение Oracle Database на всех платформах содержит Oracle Clusterware — полное интегрированное решение управления ресурсами кластера. Oracle Clusterware обеспечивает необходимую кластерную инфраструктуру для любых типов приложений, не только для Oracle Database или Real Application Clusters. С помощью Oracle Clusterware High Availability API можно обеспечить высокую готовность любого приложения, обеспечивая мониторинг, автоматическое перемещение на другой узел кластера при необходимости и перезапуск при сбоях приложения.

Защита приложений

Oracle Database обеспечивает встроенную интеграцию с Oracle Fusion Middleware и Oracle Clients при сбоях связи. Технология Transparent Application Failover (TAF) обеспечивает прозрачность восстановления соединения с базой данных при потере связи с БД или сбое узла кластера. Технология Single client access name (SCAN) позволяет клиентам использовать единый адрес для доступа к базе данных, не зависящий



от процессов обработки сбоев или балансировки нагрузки в кластере. Технология Application Continuity маскирует сбои для конечных пользователей и приложений, выполняя прозрачно для приложения повтор транзакции, поэтому сбой воспринимается приложением как слегка отложенное выполнение.

Защита от сбоев устройств хранения

Защита от сбоя дисковых устройств хранения является одной из важнейших задач обеспечения высокой доступности. Традиционным способом исключения простоев в случае выхода из строя компонент дисковых систем является оперативное хранение избыточной информации.

Automatic Storage Management

Oracle Database предлагает интегрированное в ядро СУБД средство для обеспечения защиты данных от сбоя дисковых подсистем. Automatic Storage Management (ASM) может осуществлять двукратное или даже трехкратное зеркалирование данных как внутри одной группы физических дисков, так и между группами дисков. Группы дисков администратор может определить самостоятельно, так что зеркалирование данных может осуществляться между дисками, подключенными к разным контроллерам или даже между дисковыми массивами, находящимися на значительном расстоянии друг от друга.

ASM оптимально с точки зрения производительности распределяет данные по всем имеющимся дисковым ресурсам, одновременно обеспечивая защиту данных от сбоя дискового устройства хранения, осуществляя зеркалирование данных на уровне блоков базы данных, а не на уровне целых дисков. Технология ASM Block Repair также позволяет защитить данные на уровне блока базы данных от некорректной записи на диск. При выявлении несоответствующей контрольной суммы блока на диске ASM автоматически восстанавливает блок с другого диска из корректной зеркальной копии блока. Кроме того, ASM обеспечивает автоматизацию именования и размещения файлов базы данных, что экономит время администратора, защищает его от ошибок и гарантирует соблюдение стандартов администрирования базы данных.

Технология Oracle ASM Flex Disk Groups дает возможность более гибко настраивать зеркалирование данных при хранении на ASM файлов сразу нескольких баз данных, определять квоты на использование дискового пространства для каждой БД и даже создавать клоны баз данных.

Очевидным преимуществом использования механизма Automatic Storage Management является то, что он оптимизирован и тесно интегрирован с ядром СУБД Oracle Database и не требует дополнительного лицензирования даже при наличии лицензии только на Oracle Database Standard Edition. Обеспечивая высокую готовность данных, ASM позволяет избавиться от затрат на приобретение, установку и сопровождение специализированных продуктов управления дисковой подсистемой.

Технология ASM Cluster File System (ACFS) на операционных системах Linux, Solaris, AIX и Windows расширяет функционал ASM для возможности хранения на дисковых группах ASM любых файлов, как бинарных файлов приложений Oracle, так и файлов приложений сторонних производителей, включая бизнес-данные, не хранящиеся в Oracle Database. ASM Cluster File System (ACFS) — это кластерная файловая система, файлы которой доступны с любого узла кластера как на чтение, так и на запись. Также поддерживаются снимки файловой системы в режиме «только для чтения».

Кроме того, технология ASM Dynamic Volume Manager (ADVM) позволяет использовать дисковые группы ASM не только для хранения данных файловой системы ACFS, но и для создания виртуальных дисковых томов и создания на них файловых систем, таких как ext3 для Linux. ADVM — это загружаемый в ядро операционной системы модуль, который обеспечивает основные функции управления дисковыми томами.



Защита от повреждения данных

Программная или аппаратная неисправность любой составляющей стека ввода-вывода может привести к появлению ошибок в данных и записи на диск испорченной информации. Подобное повреждение данных может затронуть как управляющую информацию базы данных, так и данные пользователей.

Для того чтобы предотвратить запись поврежденных данных на диск, Oracle Database осуществляет проверки данных перед записью на носитель. Подобная проверка выполняется сверху вниз, от программного до аппаратного уровня, добавляя в блоки данных контрольную информацию. Подобная проверка осуществляется при установке специальных параметров экземпляра БД, при хранении данных на дисках, управляемых Oracle Automatic Storage Management (ASM), и при передаче журналов транзакций на резервную базу данных в конфигурации Oracle Data Guard. Если же при чтении блока данных обнаружена несогласованность данных, блок может быть восстановлен из резервной копии базы данных, или из зеркальной копии ASM, или из копии блока из резервной базы данных при использовании опции Active Data Guard.

Резервное копирование данных является незаменимым способом защиты. Oracle Database предоставляет оперативные инструментальные средства для резервного копирования данных, восстановления данных из предыдущей резервной копии и последующего восстановления их состояния на момент, предшествовавший сбою.

Утилита *Recovery Manager (RMAN)* является мощным и чрезвычайно гибким инструментом для создания резервных копий и осуществления эффективного восстановления базы данных. Эта утилита поддерживает различные политики резервного копирования (как встроенные, так создаваемые пользователем), производит автоматическое оперативное резервное копирование всех необходимых для восстановления данных, при резервном копировании на ленту *RMAN* производит взаимодействие с Oracle Secure Backup или другими программными менеджерами ленточных накопителей, а при резервировании на диск — *Flash Recovery Area*. Восстановление данных пользователей может осуществляться как на уровне файлов, так и на уровне отдельных блоков — в обоих случаях это не прерывает доступ пользователей к остальной части базы данных. Oracle Database также имеет интеллектуальное средство идентификации проблем базы данных *Data Recovery Advisor*, которое сокращает время для определения необходимых шагов по восстановлению работоспособности базы данных в кратчайшие сроки, используя для восстановления резервные копии базы данных как на ленте, так и на диске, а также данные резервной базы данных, созданной с помощью Oracle Data Guard.

В наше время базы данных достигают огромного размера и полное резервное копирование таких баз данных требует продолжительного времени. Эту проблему компания Oracle решает с помощью программно-аппаратного комплекса *Zero Data Loss Recovery Appliance*. Ключевой термин при работе с *Recovery Appliance* — «виртуальная полная копия БД». Сначала только один раз делается полная резервная копия, а затем благодаря инкрементальным копиям *Recovery Appliance* может на лету воссоздать виртуальную полную копию БД. При выполнении инкрементального резервного копирования никогда не выполняется резервная копия дубликатов блоков, блоков Undo для завершенных транзакций или неиспользованных блоков. *Recovery Appliance* может быстро воссоздать полную копию БД на любой момент времени, быстро сложив все «дельты», таким образом минимизируется объем передаваемой информации при необходимости восстановления базы данных в случае сбоя. Чтобы обеспечить надежное восстановление баз данных, *Recovery Appliance* постоянно проверяет целостность резервных копий. Сжатие данных и резервное копирование на ленты также может выполнять *Recovery Appliance*, высвобождая промышленные серверы для основной работы.

Защита от катастроф

СУБД Oracle предоставляет широкий выбор решений для защиты от катастрофических событий, нарушающих на продолжительное время работу центра обработки данных (ЦОД). В качестве примера таких событий можно привести перебои в работе сетей энергоснабжения и связи, природные катастрофы и даже террористические



атаки. Простейшая форма защиты данных — резервные базы данных в резервных ЦОД. В случае если центр обработки данных не способен восстановить обслуживание в разумные сроки, резервные БД можно использовать для восстановления системы на другой территории. Для ускорения процесса аварийного восстановления Oracle предлагает технологию Data Guard. Oracle Data Guard позволяет создавать резервные базы данных (standby) и поддерживать их в синхронном состоянии с основной базой данных (primary). В случае повреждения данных или катастрофы в основном ЦОД пользователи могут продолжить работу, используя базу данных в резервном ЦОД.

Технология Data Guard является основой стратегии обеспечения катастрофоустойчивости баз данных Oracle Database. Data Guard позволяет создать и поддерживать копию рабочей базы данных, функционирующую в режиме горячего резерва. Эта резервная база данных может находиться где угодно: в том же центре обработки данных или на другой стороне земного шара. Data Guard дает возможность автоматизировать сложные задачи создания, синхронизации, мониторинга, управления, штатного и аварийного переключения. Это решение позволяет сохранить базу данных даже в случае катастрофы, затронувшей основной ЦОД.

Передача изменений от основной к резервной базе данных производится автоматически с помощью пересылки журналов транзакций, обязательной структуры любой базы данных Oracle Database. Используя Data Guard, можно поддерживать два типа резервных баз данных.

- **Физическая резервная база данных** является бинарной копией основной базы данных. Синхронизация резервной базы данных осуществляется с помощью журналов транзакций с применением стандартной технологии восстановления базы данных Oracle. Это гарантирует быструю и бесперебойную синхронизацию. Резервную базу данных можно открывать в режиме чтения, что позволяет выполнять на ней сложные запросы и отчеты, тем самым снимая ненужную загрузку основной базы данных. Резервные копии, выполненные на резервной базе данных, можно использовать для восстановления основной базы данных. Опция Active Data Guard позволяет активно использовать базу данных standby без остановки синхронизации с базой данных primary. Режим snapshot standby позволяет использовать резервную базу данных не только для чтения, но и для записи, например для тестирования, по окончании которого можно быстро синхронизировать с primary и перевести ее обратно в режим standby.
- **Логическая резервная база данных** по своей сути не является копией основной базы данных, однако содержит идентичные пользовательские данные. Это достигается тем, что поток журнальной информации основной базы данных преобразуется в SQL-команды, которые затем выполняются на резервной базе данных в обычном режиме. В отличие от физической резервной базы данных логическая резервная база данных находится в открытом режиме всегда. Это позволяет производить на ней без ограничений операции, не предусматривающие изменения данных, тем самым снимая нагрузку с основной базы данных. Для этих целей логическую резервную базу данных можно даже оптимизировать с помощью дополнительных индексов и материализованных представлений, отсутствующих в основной базе данных. Также логическую резервную БД можно использовать для уменьшения простоев при обновлении программного обеспечения Oracle Database, для автоматизации такого обновления разработан специальный пакет DBMS_ROLLING.

Data Guard обеспечивает передачу журналов транзакций как в асинхронном режиме, так и в синхронном режиме, при котором завершение транзакции на основной базе данных происходит только при успешной передаче журнальной информации на резервную базу данных. Синхронный режим передачи журналов транзакций позволяет построить решение для аварийного восстановления с нулевой потерей данных. Data Guard в режиме Real Time Apply позволяет применять изменения к резервной базе данных сразу же после их получения с основной БД, не дожидаясь архивирования текущего журнала транзакций на резервной БД. Это обеспечивает оперативную синхронизацию резервной и основной баз данных и дает возможность получения актуальных отчетов в реальном времени. Кроме того, Real Time Apply сокращает время штатного (switchover) и аварийного (failover) переключения, минимизируя время плановых и внеплановых простоев системы. Data Guard в режиме Fast-Start Failover позволяет организовать мониторинг работы и аварийное автоматическое переключение на резервную базу данных в случае проблем с основной. При этом прежняя



основная база данных, после того как становится доступной вновь, автоматически восстанавливается и преобразуется в резервную базу данных, что предотвращает необходимость пересоздания основной базы данных и очень быстро восстанавливает требуемый уровень защиты системы.

Мониторинг, управление и конфигурация Data Guard могут осуществляться как с помощью графического интерфейса Enterprise Manager, так и с помощью интерфейса командной строки.

Опция Oracle Active Data Guard позволяет открыть резервную БД для чтения и небольшого обновления, то есть эта БД служит и как актуальная копия промышленной БД, и как инструмент разгрузки промышленной БД от операций чтения, бэкапирования, построения отчетов. Она не только позволяет использовать физические резервные базы данных для чтения, но и значительно повышает отказоустойчивость БД благодаря таким технологиям, как Automatic Block Repair, Far Sync и Application Continuity.

Защита от человеческих ошибок

Почти каждое исследование, посвященное причинам простоев, указывает ошибки персонала как наиболее существенную причину простоев. Такие человеческие ошибки, как непреднамеренное удаление важных данных или ошибочное условие WHERE в операторе UPDATE, меняющее значительно больше строк, чем предполагал автор, по возможности должны быть предотвращены. Если же предосторожности не помогают, необходимы эффективные механизмы восстановления данных к прежнему состоянию. СУБД Oracle Database предоставляет мощные, но простые в использовании средства, позволяющие администратору быстро диагностировать подобные ошибки и устранять их последствия. Кроме того, предусмотрены средства, дающие возможность конечным пользователям устранять последствия ошибок без вмешательства администратора, тем самым облегчая его работу и ускоряя восстановление потерянных и поврежденных данных.

Лучший способ предотвратить ошибки — ограничить доступ пользователей, предоставляя доступ только к тем данным и сервисам, которые им действительно необходимы для выполнения служебных обязанностей. СУБД Oracle предоставляет широкий выбор средств безопасности, позволяющих управлять доступом пользователей к данным приложений с помощью аутентификации пользователей и предоставления пользователям права доступа только к объектам, необходимым им для выполнения их функциональных обязанностей. В дополнение к этой модели безопасности Oracle Database позволяет ограничить доступ к данным на уровне строк с помощью механизма *Virtual Private Database (VPD)* и опции *Label Security*.

Если ошибки совершаются авторизованными пользователями, необходим механизм устранения их последствий. В СУБД Oracle Database предусмотрено целое семейство технологий исправления человеческих ошибок, получившее название *Flashback* (*Flashback Query*, *Flashback Versions Query*, *Flashback Transaction Query*, *Flashback Transaction*, *Flashback Table*, *Flashback Drop*, *Flashback Database*). Технологии *Flashback*, знаменующие собой революцию в восстановлении данных, позволяют очень быстро отменить изменения данных любых масштабов: от одиночной строки до целиком всей базы данных. Как правило, повреждение базы данных происходит за минуты, а в прошлом для ее восстановления требовались часы и даже дни. Благодаря механизму *Flashback Database* время исправления ошибки сопоставимо со временем ее совершения. Эта возможность исключительно проста в использовании: чтобы восстановить всю базу данных на момент времени в прошлом, достаточно одной короткой команды и никаких сложных процедур. *Flashback Query* имеет SQL-интерфейс для быстрого поиска и устранения человеческих ошибок. Пользователь на работающей БД выполняет запрос в прошлое, на момент, когда ошибка еще не была совершена, и *Flashback Transaction* предоставляет средства для тонкого анализа и восстановления локальных повреждений (например, ошибочного удаления одной строки таблицы), а также возможность без длительных простоев исправить более широкомасштабные ошибки (например, в случае удаления нескольких строк нескольких таблиц в рамках одной транзакции). Механизмы *Flashback* являются уникальной особенностью СУБД Oracle Database. Они обеспечивают быстрое восстановление на любых уровнях, включая строки, транзакции, таблицы, табличные пространства и базу данных в целом.



Предотвращение плановых простоев

Плановые простои могут наносить не меньший ущерб для бизнеса, чем внеплановые. При проектировании системы чрезвычайно важно максимально сократить предполагаемые плановые простои. Плановые простои связаны с текущими операциями по обслуживанию системы, периодическими сервисными работами, развертыванием новых мощностей.

В условиях, когда пользователи по всему миру обращаются к данным 24 часа в сутки, окна технического обслуживания сокращаются практически до нуля. Временных окон, когда плановые простои не влияют на работу пользователей, для таких систем больше не существует. Если объем данных в базах данных очень велик, операции обслуживания становятся весьма длительными. Важно, чтобы проведение этих операций не влияло на работу пользователей.

Технологии Oracle устраняют необходимость планирования окон технического обслуживания, которые администраторы БД выделяют для административных операций, и создают условия для круглосуточного функционирования корпоративных систем.

Предотвращение простоев при изменении данных

Oracle Database позволяет проводить множество сервисных операций без нарушения функционирования базы данных и работы пользователей. Добавление, перестройка и дефрагментация таблиц и индексов могут осуществляться одновременно с чтением и изменением этих данных пользователями. Точно так же возможно осуществлять и изменение структуры таблиц, добавление, удаление и переименование столбцов, изменение их типов, создание и изменение партиций и других параметров хранения данных без нарушения работы конечных пользователей. Хранимые процедуры на Java™ и PL/SQL™ можно обновлять оперативно, при этом Oracle управляет всеми взаимозависимостями, обеспечивая правильную интеграцию в БД новых процедур без нарушения работы конечных пользователей, возможен ряд изменений таблицы без перекомпиляции хранимых процедур, связанных с данной таблицей.

Если же пользовательское приложение чувствительно к изменениям структуры данных в БД, то для обновления приложения необходимо синхронно произвести изменения как в БД, так и в приложении, что приводит к значительному увеличению времени планового простоя. Для таких задач можно использовать технологию Edition-based Redefinition, которая позволяет создавать редакции объектов БД для приложений, которые могут отличаться от редакции, находящейся в промышленной эксплуатации, как структурой хранения данных в базе данных, так и спецификацией хранимых процедур. Установка новой редакции может быть выполнена без нарушения работы пользователей предыдущей редакции. Новые пользователи могут работать с новой редакцией, при этом имея доступ к данным тех же таблиц и к тем же хранимым процедурам, что и пользователи предыдущих редакций.

С ростом баз данных управление ими становится весьма трудоемким. Возможность секционирования таблиц и индексов базы данных при использовании опции Partitioning позволяет администраторам делить большие таблицы на более управляемые части меньшего размера и выполнять операции обслуживания по отдельным секциям.

Предотвращение простоев при изменениях в системах

Каждая новая версия СУБД Oracle еще более расширяет поддержку динамического реконфигурирования, позволяя реализовать адаптацию к изменениям нагрузки и аппаратных средств без прерывания обслуживания. СУБД Oracle динамически адаптируется к таким изменениям в аппаратных конфигурациях, как:

- добавление и удаление процессоров сервера;
- добавление и удаление узлов кластера RAC;



- динамическое наращивание и сокращение выделенных ресурсов памяти, автоматическая настройка распределения памяти в процессе работы;
- добавление и удаление дисков базы данных без нарушения работы пользователей БД;
- изменение параметров хранения таблиц и индексов;
- перемещение или переименование файлов данных.

Своевременная установка обновлений программного обеспечения (patches) является одной из основных задач администраторов баз данных. Технология RDBMS Online Patching — это возможность устанавливать специальные предварительные патчи, которые могут быть установлены, пока экземпляр базы данных находится онлайн. Установка патча онлайн рекомендуется для того, чтобы избежать лишней остановки работы пользователей базы данных.

Oracle Database поддерживает установку программных обновлений на узлы системы Real Application Clusters (RAC) в чередующемся режиме без перерывов в обслуживании конечных пользователей. В этом случае обновления устанавливаются последовательно на каждый узел кластера, в то время как другие узлы продолжают выполнять обслуживание пользователей.

Кроме того, при использовании логических резервных баз данных Oracle Data Guard разработана методология Transient Logical Standby Rolling Upgrades и специальный пакет DBMS_ROLLING для автоматизации этого процесса, которые позволяют производить существенные изменения в системе практически без остановки в обслуживании. Под существенными изменениями подразумеваются работы по изменению состава и конфигурации аппаратного и программного обеспечения, а также модернизации версий Oracle Database, требующих значительных изменений в словаре базы данных.

Оптимальные методики архитектуры максимальной готовности (Maximum Availability Architecture, MAA)

Оптимальные методики — ключевой фактор для успешного развертывания ИТ-инфраструктуры. Одной лишь технологии недостаточно. Архитектура Oracle Maximum Availability Architecture (MAA) представляет собой проверенную архитектуру для построения систем высокой готовности. Предприятия, построившие архитектуры своих систем, руководствуясь методиками MAA, убедились, что это дает возможность быстро и эффективно создавать и развертывать приложения, отвечающие требованиям бизнеса к уровню готовности. Методики MAA включают конкретные рекомендации по проектированию и конфигурации, прошедшие тщательный анализ и проверку на обеспечение оптимальной готовности и надежности системы. В методиках MAA рассматриваются и подробно излагаются вопросы совместного использования ключевых возможностей СУБД Oracle для обеспечения высокой готовности, включая Real Application Clusters, Data Guard, Recovery Manager и Enterprise Manager. В них также затрагиваются вопросы настройки и интеграции других, критически важных компонентов систем высокой готовности, включая серверы, устройства хранения данных, сетевое оборудование и серверы приложений.

Руководство по оптимальным методикам MAA постоянно совершенствуется, в комплекте документации Oracle Database есть ссылки на соответствующие документы с описанием методик для обеспечения высокой готовности жизненно важных приложений.



Oracle Maximum Availability Architecture (MAA)

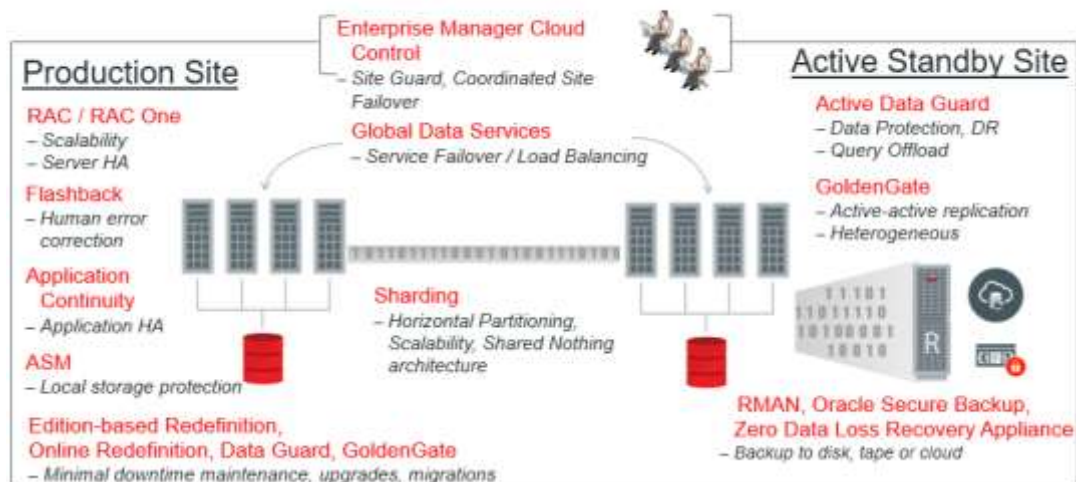


Рис 2. Архитектура максимальной доступности

Автономные базы данных

Новое направление в области СУБД. Автономные базы данных призваны совместить простоту создания и эксплуатации БД с мощностью современных коммерческих СУБД. В какой-то степени идея автономных БД перекликается с модной сегодня темой беспилотного или автономного автомобиля, где человеку достаточно сесть в автомобиль и сообщить, куда нужно ехать. Все остальное выполнит система управления и поддержки автономного автомобиля.

Та же идея заложена и в концепцию автономных БД. Пользователь при запросе БД определяет политики и уровень сервиса, который ему нужен, то есть ставит цели, а система сама создает БД нужной архитектуры, обеспечивающей заданный уровень сервиса (надежность, безопасность, производительность и т. д.), и далее сама его поддерживает. Что же должна делать за администратора автономная БД? Если проанализировать наиболее трудоемкие задачи, выполняемые администратором, то они сведутся к следующим:

- Конфигурирование аппаратной и программной инфраструктуры (серверов, операционной системы, сети, подсистемы хранения данных).
- Установка СУБД Oracle, установка новых версий, установка патчей.
- Резервное копирование, восстановление.
- Построение отказоустойчивого решения.
- Защита данных, устранение дыр в безопасности.
- Оптимизация работы БД, настройка производительности SQL-запросов.
- Управление жизненным циклом данных.
- Выявление ошибок в ПО и их устранение.
- Выделение дополнительных вычислительных ресурсов при увеличении нагрузки (обеспечение эластичности сервиса БД).
- Развертывание новых сред (для тестирования, разработки и т. п.).



Не забудем еще, что до начала работы нужно получить ИТ-инфраструктуру для установки БД и приложений (то есть компьютеры, сетевую инфраструктуру, систему хранения и т. д.). В случае автономной БД все эти задачи снимаются с администратора, их выполняет сама автономная БД.

Итак, автономная БД обеспечивает три основные функции.

- **Самоуправляемость** (автоматически создает базу данных, устанавливает обновления, выполняет резервное копирование, настройку производительности, обеспечивает эластичность сервиса в зависимости от нагрузки).
- **Самозащита** (защита от внешних и внутренних атак, автоматическая установка обновлений безопасности, шифрование данных как при хранении, так и при передаче по сети).
- **Самовосстановление** (автоматическое восстановление после сбоев, защита от простоев, обеспечение заданного уровня сервиса, гарантия нужного уровня SLA).

Преимущества такой автономной БД очевидны. Она облегчает работу администратора, снижает риски, повышает качество работы СУБД и приложений. Она снижает ТСО и время развертывания систем, обеспечивает более высокую надежность и безопасность, чем большинство традиционных СУБД. Пользователь может просто заказать БД, получить ее за несколько минут, создать таблицы, загрузить данные и начать с ними работать (анализировать, писать и отлаживать приложения, строить отчеты и т. д.). При этом создавать индексы, партиции, материализованные представления, табличные пространства и т. д. необязательно. Это кардинально меняет подход пользователей к решению бизнес-задач, сроки создания приложений, позволяет сосредоточиться на развитии бизнеса и выводе новых услуг и продуктов на рынок.

Архитектура автономной БД

Какова же архитектура автономной БД Oracle? Конечно, в ее основе лежит СУБД Oracle, начиная с версии 18с, но было бы ошибкой говорить, что Oracle 18с — автономная БД. Автономная БД состоит из трех компонентов, каждый из которых отвечает за разные аспекты автономности. Это облачная инфраструктура Oracle Cloud, СУБД Oracle и некоторое дополнительное программное обеспечение, дополняющее работу СУБД и облака и управляющее всей конструкцией (рис. 1).



Рис. 1. Компоненты автономной базы данных

Из сказанного выше уже понятно, что автономная БД Oracle может работать только в облачной инфраструктуре Oracle Cloud, так как эта инфраструктура обеспечивает ИТ-ресурсы, быстрое развертывание, бэкап и патчирование. Конечно, заказчик может установить в своем ЦОД просто СУБД Oracle или заказать сервис DBaaS в облаке, но это не будет автономной БД. Поэтому очевидно, что автономная БД Oracle — это специальный сервис, который может работать только в облаке Oracle или на Oracle Cloud@Customer (Cloud-машина Oracle, установленная в ЦОД заказчика и администрируемая удаленно сотрудниками Oracle). Более того, автономная БД работает только на платформе Exadata (машина баз данных Oracle), которая и обеспечивает высокую производительность и надежность работы СУБД. Exadata позволяет автономной БД



использовать такие уникальные механизмы, как smart scan на ячейках, storage index, гибридное колоночное сжатие, операции в памяти, кэширование данных на флэш для OLTP, шифрование всех данных БД, разделение и приоритизация ресурсов ввода-вывода, параллельная обработка, Real Application Cluster (RAC).

СУБД Oracle — универсальная СУБД. Она может использоваться для создания разных систем: OLTP, DSS, хранилищ данных, геоинформационных систем, работы с графами, документами и т. д. Эффективность работы СУБД для конкретной задачи обеспечивается конфигурированием и настройкой СУБД. Для того чтобы автономная БД эффективно поддерживала конкретные задачи, пользователь должен заказать не просто БД, а конкретный специализированный сервис, тогда при создании БД она будет автоматически сконфигурирована и настроена под его задачу. Сейчас можно заказать автономный сервис хранилищ данных (Autonomous Data Warehouse service, ADW), который удобен для создания витрин и небольших хранилищ данных, автономный сервис для OLTP нагрузки и смешанной нагрузки (Autonomous Transaction Processing), который предназначен для выполнения высокопроизводительной обработки транзакций, генерации отчетов, пакетной обработки данных, решать задачи IoT, задачи с использованием машинного обучения, а также сейчас доступен автономный сервис для работы с документами в формате JSON (Autonomous JSON Database), который оптимизирован для быстрой обработки формата данных JSON.

В состав автономной БД входит популярное средство быстрой разработки приложений Oracle Application Express (APEX). Оно позволяет упростить разработку и развертывание приложений, реализовать аналитику в реальном времени, использовать возможности персонализации и функции обнаружения мошенничества в применении к реальным транзакционным данным.

Выполнять SQL-запросы и разрабатывать хранимые процедуры в Oracle Autonomous Database можно как через SQL Developer, так с помощью любого другого инструмента разработки, который поддерживает Oracle Database. Интегрировать другие системы с данными в Autonomous Database можно через различные интеграционные сервисы от Oracle или других производителей, причем эти интеграционные сервисы могут работать как в облаке, так и в вашем ЦОД. Для загрузки данных в таблицы Autonomous Database можно использовать Object Storage как в Oracle Cloud, так и на Amazon S3 и Azure Blob Storage. В базе данных есть удобная процедура для загрузки этих данных или для работы с ними как с внешними таблицами. Для анализа загруженных в таблицы данных можно использовать любые BI-инструменты, которые умеют выполнять запросы к БД по протоколу SQL*Net, работающие как в облаке, так и в вашем ЦОД. Oracle Autonomous Database основана на ядре Oracle Database, поэтому любые приложения, работающие с Oracle Database, совместимы и с Oracle Autonomous Database.

Еще одной особенностью автономной БД является ее эластичность. Поскольку она работает на ИТ-инфраструктуре, скрытой в облаке, пользователь может по мере необходимости увеличивать или уменьшать количество процессоров и дискового пространства, необходимого для решения его задач. Это делается без остановки работы БД через простой веб-интерфейс. Поскольку оплата сервиса осуществляется по факту его использования, можно, например, на нерабочие дни снизить число процессоров до нуля, а во время пиковой нагрузки увеличить его до нужного уровня. При увеличении числа используемых процессоров также увеличивается объем оперативной памяти для автономной БД и пропускная способность ввода-вывода. Кроме того, автономия БД имеет возможность автоматически увеличивать эти ресурсы в несколько раз при увеличении нагрузки со стороны пользователей, и снова возвращать в исходное состояние, когда нагрузка уменьшится. Это делает автономную БД еще более экономным сервисом для приложений, которые имеют неравномерную нагрузку.

Из концепции автономных БД понятно, что администрированием всего стека (от оборудования до ПО) занимается Oracle, отвечает за поддержку оборудования и операционной системы, выпуск патчей, исправление ошибок. Oracle постоянно мониторит систему. Для облегчения управления и сопровождения огромного количества автономных БД они реализуются в виде PDB (Pluggable Database).

Для обеспечения самовосстановления автономная база данных использует архитектуру максимальной доступности, необходимую для обеспечения уровня сервиса, заданного пользователем. В зависимости от этих требований она может использовать Real Application Clusters для защиты от сбоя узла, Active Data



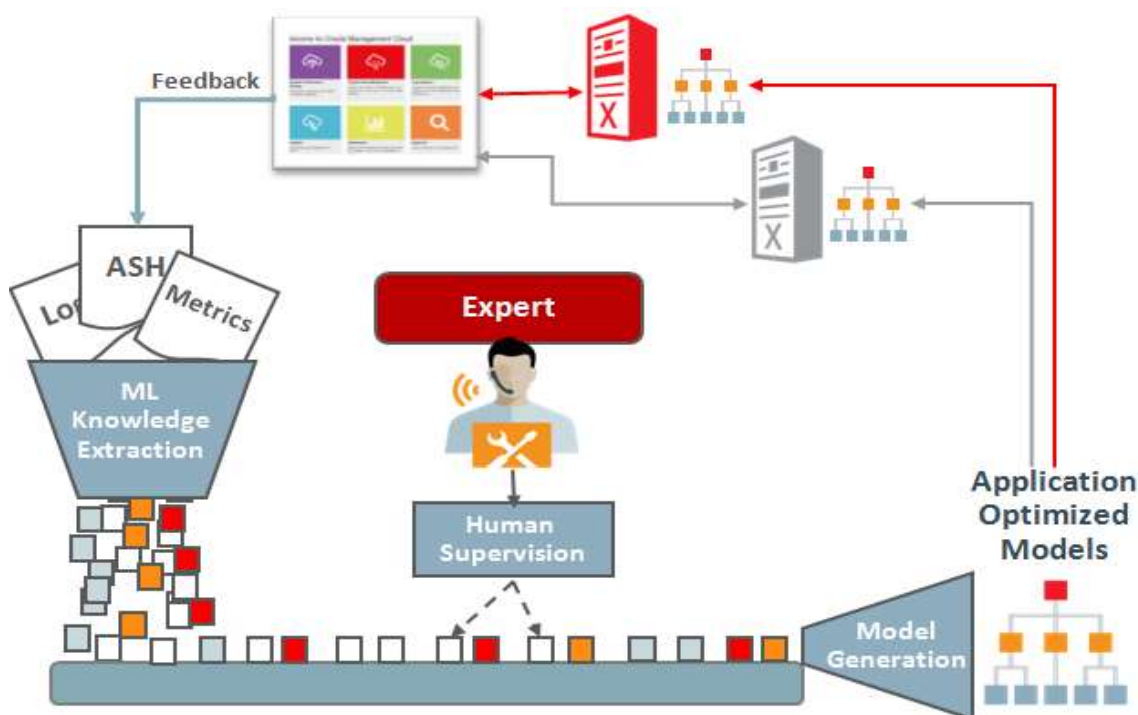
Guard с резервной базой данных (Standby) для защиты от катастрофических сбоев, ASM для защиты от сбоя дисковой подсистемы. Патчи и апгрейды не приводят к длительным простоям, так как используют стандартные механизмы Oracle, такие как Rolling RAC Upgrade (последовательное патчирование или апгрейд узлов кластера), краткосрочный логический Standby, создание нового Oracle Home из Gold Image и быстрое переключение на него, Flashback (защита от человеческих ошибок). Максимальный уровень надежности, который может обеспечить Oracle для автономной базы данных, — 99,995. Это 2,5 минуты простоя в месяц или 30 минут простоя в год, включая регламентные работы.

Автономную БД можно заказать в одном из 2 режимов: разделяемом (Shared) или выделенном (Dedicated). В первом случае ваша БД (pdb) будет размещаться совместно с БД других заказчиков, то есть возможна конкуренция за ресурсы. Во втором случае для заказчика в ЦОД Oracle выделяется пул оборудования, на котором размещаются только БД этого заказчика. Заказчик сам решает, на какой Exadata и в какой CDB будет развертываться его БД, выбирает время патчирования и т. д. Этот режим хорош для промышленной эксплуатации.

Самоуправляемость автономной базы данных реализована на основе множества советчиков (advisors) и механизмов автоматического управления и мониторинга, встроенных в Oracle Database, таких как Tuning advisor, In-Memory advisor, Automatic Indexing, Compression advisor, Real-time Statistics, Repair advisor, Autonomous Health Framework. Пользователи могут очень быстро создавать новые автономные базы данных и легко переносить в облако существующие БД, что значительно снижает затраты и время выхода на рынок. Большинство задач по администрированию базы данных система выполняет автоматически.

Для обеспечения самозащиты автономная БД обязательно шифрует все данные БД, автоматически применяет патчи безопасности по мере их появления, механизмы машинного обучения и аудит позволяют выявлять внешние атаки и внутренние злонамеренные действия. Можно применять стандартные средства защиты Oracle, такие как шифрование, маскирование данных, Database Vault (сокрытие данных от привилегированных пользователей).

Для постоянного мониторинга и поддержки инфраструктуры автономных БД, выявления самых важных и приоритетных для исправления ошибок служба технической поддержки Oracle использует инструмент Autonomous Health Framework (Trace File Analyzer, Cluster Health Advisor, Hang Manager). Он работает по следующему принципу (см. рисунок ниже):



Автоматически строится базовая модель нормального поведения системы. Далее при работе системы выявляются отклонения от базовой модели, производится анализ этих отклонений и на основе результатов анализа либо выявляются тенденции ухудшения работы, либо выявляются причины отклонений. После чего, если необходимо, автоматически выполняются корректирующие действия. Модели, используемые для работы этих компонент, генерируются с помощью алгоритмов машинного обучения. В составлении этих алгоритмов участвуют эксперты по машинному обучению. Далее эти модели загружаются в систему и используются для анализа работы подсистем. Модели постоянно совершенствуются и обновляются.

Oracle Autonomous Database — это семейство облачных сервисов, которые самостоятельно обеспечивают свою работоспособность, безопасность и восстановление. Первая в мире автономная база данных меняет подход к управлению благодаря использованию средств машинного обучения и автоматизации. Они дают возможность устранить потребность в ручном администрировании и настройке, избежать ошибок, вызванных человеческим фактором, сократить расходы, упростить поддержку и настройку, обеспечить большую надежность, безопасность и операционную эффективность по сравнению с традиционными решениями.

Автономные БД можно использовать в финансовой отрасли, ритейле, на производстве и в государственных организациях, они поддерживают сложное сочетание высокопроизводительной обработки транзакций, генерации отчетов, пакетных и аналитических нагрузок.

Oracle Sharding

Oracle Sharding — это технология для масштабирования, обеспечения доступности и геораспределения данных для OLTP-приложений, которая распределяет данные между отдельными базами данных Oracle, находящимися в пуле. Каждая база данных в таком эластичном пуле называется шардом (shard). Шардинг (sharding) основан на архитектуре Shared-Nothing, где нет никаких разделяемых компонент между шардами — нет общей системы хранения и кластерного ПО. На рис. 1 показано, как крупная база данных разбивается на пул шардов.

Одна большая база данных, разделенная
на множество небольших баз данных (шардов)

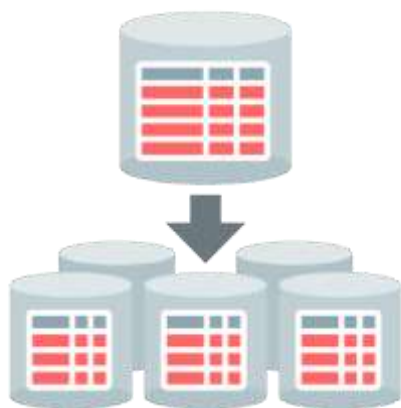


Рис. 1. Одна большая база данных, разделенная на множество небольших баз данных (шардов)

Oracle Sharding распределяет данные по шардам методом горизонтального секционирования. Горизонтальное секционирование распределяет таблицу базы данных по шардам таким образом, что каждый шард содержит таблицу одинаковой структуры, но с другим набором строк.

Такое разделение основано на ключе шардинга (sharding key). Примеры ключей шардинга: customer_id, account_no, email_id, country_id и т.д. Каждый шард может реплицироваться для производительности, высокой доступности и аварийного восстановления. Пул шардов представляется приложению как единая логическая база данных Oracle (шардированная база данных, sharded database, SDB). Приложения,



работающие в архитектуре шардированных баз данных, получают преимущества линейного масштабирования, высокой доступности и геораспределения данных.

С точки зрения администратора, шардированная база данных состоит из нескольких баз данных, которыми можно управлять всеми вместе или по отдельности. Однако для разработчика приложений шардированная база данных является единой логической базой данных: количество шардов и распределение данных по ним полностью прозрачны для приложений баз данных. Операторы SQL в приложении не должны ссылаться на конкретные шарды и не зависят от количества шардов и их конфигурации. Oracle Sharding поддерживает весь жизненный цикл шардированной базы данных, включая развертывание, создание схемы, маршрутизацию на основе данных, эластичное масштабирование, мониторинг, внесение исправлений и т. д.

Компоненты и архитектура Oracle Sharding

В следующем разделе описаны основные компоненты инфраструктуры SDB, а именно шарды, Shard Catalog, Shard Director (GSM), глобальный сервис (Global Service) и пулы соединений. На рис. 2 показаны архитектурные компоненты Oracle Sharding.

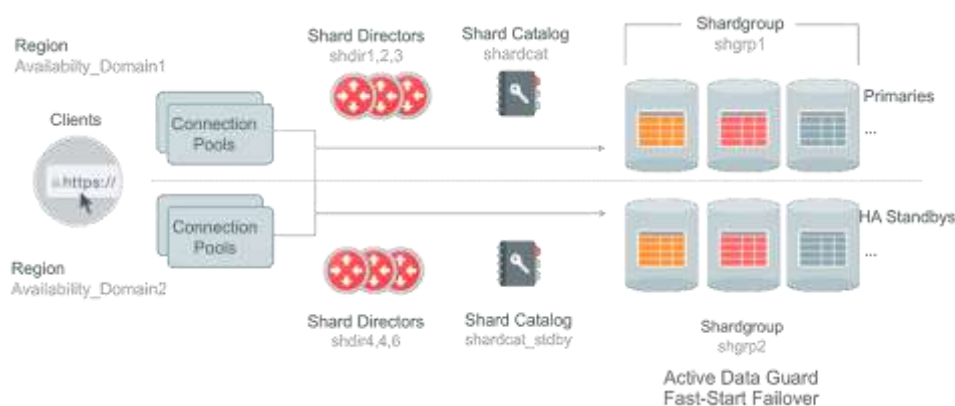


Рис. 2. Архитектура Oracle Sharding без единой точки отказа

Шардированная база данных и шарды

Шарды (shard) — это независимые базы данных Oracle, размещенные на серверах баз данных, имеющих собственные локальные ресурсы (ЦП, оперативную память и диск). Никакая общая система хранения для шардов не требуется. Шардированная база данных представляет собой набор шардов. Шарды могут быть размещены в одном регионе (одном или нескольких ЦОД) или в разных. Регион в контексте Oracle Sharding — это один или несколько ЦОД, расположенных в одной общей и быстрой сетевой инфраструктуре.

Шарды могут реплицироваться для высокой доступности (High Availability, HA) и аварийного восстановления (Disaster Recovery, DR) с помощью таких технологий репликации Oracle, как Active Data Guard или GoldenGate. Для высокой доступности резервные шарды могут быть размещены в том же регионе, что и основные. Для аварийного восстановления резервные шарды размещаются в другом регионе.

Shard Catalog

Shard Catalog (Каталог шардов) — это специальная СУБД Oracle Database, где постоянно хранятся данные о конфигурации шардированной базы данных. Этот компонент играет ключевую роль в автоматическом развертывании шардированной базы данных и централизованном управлении ею. Здесь также хранится определение схемы данных (структуры шардированных таблиц) и содержимое общих справочных данных (duplicated tables – таблицы дублируются на всех шардах). Shard Catalog также выступает в качестве координатора запросов для обработки мультишардовых (multi-shard) запросов и запросов, не имеющих ключа шарда (sharding key).



Любые изменения в конфигурации, такие как добавление и удаление шардов и глобальных сервисов, инициируются в БД Shard Catalog. Все операторы DDL в шардированной базе данных выполняются путем подключения к каталогу шардов.

Аварийное отключение каталога шардов не влияет на доступность шардированной базы данных. Отключение каталога шардов влияет только на возможность выполнять операции техобслуживания или многшардные запросы в течение короткого периода времени, необходимого для переключения на резервный каталог шардов. Транзакции OLTP не затрагиваются, по-прежнему маршрутизируются и выполняются базами данных SDB. В архитектуре максимальной доступности Oracle (Oracle MAA) рекомендуется настраивать локальную резервную базу данных на режим защиты Maximum Availability с Fast-Start Failover и удаленной резервной физической базой данных.

Shard Director

Shard Director — это сетевой прослушивающий процесс для клиентов, подключающихся к SDB. Он поддерживает актуальную топологию шардированной базы данных. Shard Director определяет шард, к которому нужно подключить соединение по ключу шарда, переданному во время запроса на соединение. Shard Director основан на ПО Oracle Global Service Manager (GSM). Для Oracle Sharding программное обеспечение GSM было расширено для управления шардированными базами данных и маршрутизации с соединений на основе ключа шарда.

Для типичной шардированной базы данных экземпляры GSM (они же shard directors) устанавливаются на выделенных низкопроизводительных стандартных серверах в каждом регионе. Для высокой доступности можно использовать несколько Shard Directors. В архитектуре Oracle MAA рекомендуется использовать по три директора шардов на регион для обеспечения доступности и масштабируемости.

Shard Director включают следующий набор функций.

- Сохранение данных о конфигурации SDB и доступности шардов.
- Измеряют задержки в сети между своим регионом и другими.
- Листенер для клиентов, подключающихся к SDB.
- Управление глобальными сервисами.
- Выполняют соединение и предоставляют информацию для выравнивания нагрузки пулам соединений во время выполнения.
- Мониторинг доступности экземпляров баз данных и глобальных сервисов, извещение клиентов об отказах через события FAN (Fast Application Notification).

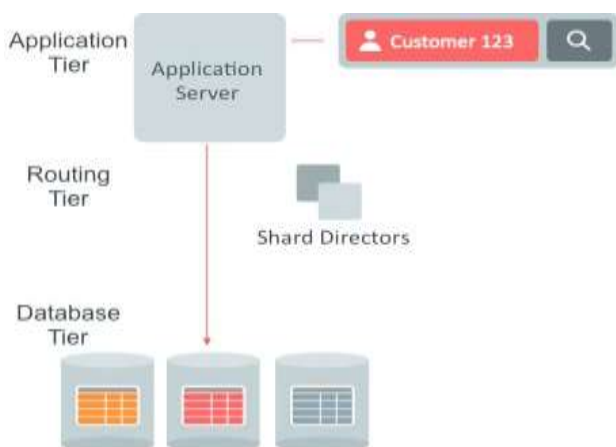


Рис. 3. Логический поток маршрутизации сессий



Global Service

Global Service (Глобальный сервис) — это сервис баз данных, который может работать во множестве баз данных. Глобальный сервис позволяет клиентам получать доступ к данным в любом шарде SDB. Глобальные сервисы предлагают дополнительные свойства для шардированных баз данных, например роль базы данных, допустимые задержки репликации, привязка к регионам и т. д. Вы можете создавать глобальные сервисы на основе ролей: создать глобальный сервис чтения и записи с доступом к данным из основных шардов и отдельный глобальный сервис только для чтения для шардов Active Data Guard.

Пулы соединений

Во время выполнения пулы соединений действуют как Shard Director, направляя запросы к базе данных, к соответствующим шардам через соединения в пуле. Oracle поддерживает пулы соединений в различных драйверах доступа к данным, таких как OCI, JDBC, ODP.NET и т. д.

Преимущества Oracle Sharding

Шардирование с СУБД Oracle Database дает ряд преимуществ приложениям, которым нужно обрабатывать большие объемы данных.

- **Линейная масштабируемость.** Приложения OLTP, разработанные для Oracle Sharding, могут эластично масштабироваться до любого уровня и на любой платформе, просто развертывая новые шарды на дополнительных независимых серверах. Производительность масштабируется линейно по мере добавления новых шардов в пул, так как каждый шард полностью независим от других.
- **Высокая доступность данных.** Oracle Sharding устраняет единую точку отказа (single point of failure, SPOF), отказ в которой приводит к отказу всей системы (совместно используемый диск, SAN, сеть и т. д.), и обеспечивает надежную изоляцию отказов. Недоступность или замедление работы шарда из-за аварийного отключения или планового обслуживания затрагивает только пользователей соответствующего шарда и не влияет на доступность или работу приложения для пользователей других шардов. Каждый шард может использовать свою версию СУБД Oracle Database при условии, что приложение обратно совместимо с самой старой работающей версией (начиная с версии Oracle Database 12.2). Это позволяет сохранять доступность приложения во время планового обслуживания базы данных.
- **Размещение данных в соответствии с законодательством и территориальная близость данных благодаря их географическому распределению.** Шардинг позволяет размещать различные части данных в разных странах или регионах и таким образом соответствовать законодательству, требующему размещения данных в области действия определенной юрисдикции. Поддерживается также хранение определенных данных ближе к потребителям — в соответствующем ЦОД, находящемся в локальной близости к пользователям.
- **Развертывание в облаке.** Поскольку размер шарда может быть произвольно мал, шардированная база данных легко развертывается в облаке, состоящем из низкопроизводительных серверов. Oracle Sharding поддерживает развертывание в облаке, в ЦОД заказчика и в гибридном облаке.



Рис. 4. Oracle Sharding поддерживает все три модели развертывания



- **Простота** благодаря автоматизации многих задач управления жизненным циклом, включая автоматическое создание и репликацию шардов, автоматическое хеш-секционирование, развертывание нового шарда несколькими командами и автоматическую ребалансировку данных в шардах.
- **Высокая производительность** за счет маршрутизации сессий на шард, содержащий нужные данные.
- **Возможности РСУБД корпоративного уровня**, включая реляционную схему, SQL и другие программные интерфейсы, поддержку сложных типов данных, изменение схемы без остановки базы данных, масштабируемость на большое количество ЦПУ, дополнительные средства безопасности, сжатие, высокую доступность, свойства ACID, многоверсионность (MVCC), консистентное чтение, поддержку формата JSON и многое другое.

Database Fleet Maintenance

Обслуживание базы данных требует времени и концентрации. Администратор БД часто слышит жалобы на то, что слишком много дней запланировано на установку обновлений или развертывание баз данных. Или, что еще хуже, незапланированные простои вызваны тем, что производительность работы приложения ухудшилась после обновления базы данных. Это может повлиять на бизнес, в том числе негативно повлиять на доходы компании напрямую. И это приводит к тому, что велика цена ошибки администратора при обновлении и администраторы пропускают установку некоторых обновлений, что вызывает проблему безопасности.

Преимущества автоматизации обслуживания всех баз данных как единого целого становятся очевидными. Database Fleet Maintenance позволяет администраторам поддерживать большое количество серверов с установленным программным обеспечением Oracle Database (Oracle Homes) и связанные с ними базы данных. Администраторы могут сразу для нескольких баз данных устанавливать обновления, которые могут состоять как из промежуточных одноразовых исправлений (Interim one-off patches), так и из ежеквартальных наборов исправлений и обновлений безопасности.

При появлении новых исправлений и обновлений БД администратор устанавливает их на копию существующего Oracle Home и на этой основе создает новый образ Oracle Home (также называемый золотым образом или эталонным состоянием программного обеспечения Oracle Database).

Иногда золотой образ называют также стандартной конфигурацией. Это образ программного обеспечения Oracle Database, которое содержит базовую версию программного обеспечения плюс дополнительные исправления. Чтобы добиться стандартизации, каждый Oracle Home должен быть подписан на золотой образ. Применение стандартной конфигурации может быть выполнено за короткий промежуток времени, потому что золотой образ копируется в параллельный Oracle Home (deploy), который затем превращается в основной Oracle Home (update). Автоматически используется текущая версия подписанного образа при создании нового Oracle Home. При необходимости администратор может выполнить откат применения новой конфигурации (rollback). Для кластерных конфигураций по умолчанию update и rollback выполняются с минимальным остановом с помощью переключения на новый Oracle Home на узлах кластера поочередно (Rolling Upgrade). Поскольку старые Oracle Home занимают место на диске, их можно автоматически очищать, чтобы освободить пространство, когда они больше не используются. Все вызовы Database Fleet Maintenance могут выполняться через утилиту emcli, а также и через RESTful API.

Database Fleet Maintenance может использоваться для патчирования и апгрейда:

- Single Instance Oracle Homes и связанных с ними баз данных;
- RAC Oracle Homes и связанных с ними баз данных;
- Grid Infrastructure Homes;



- Oracle Restart Homes;
- резервных баз данных Standby (Single Instance или RAC).

Database Fleet Maintenance включает в себя следующие этапы.

- Выявление несоответствующих конфигураций с помощью Software Standard Advisor.
- Создание эталонных сред.
- Создание золотых образов с использованием эталонных сред и создание версий для дополнительных исправлений.
- Подписка баз данных и кластеров на золотой образ, к которому были применены соответствующие исправления.
- Развертывание образа и переключение баз данных со старого Oracle Home на новый Oracle Home.

С помощью Enterprise Manager администраторы могут использовать Software Standard Advisor, который сканирует конфигурацию баз данных и генерирует отчет о текущих конфигурациях ПО в существующей среде. В отчете указывается количество уникальных конфигураций (Platform + Version + Database-Type + Patches) и перечисляются Oracle Homes и базы данных для каждой конфигурации. Software Standard Advisor также рекомендует список стандартизированных конфигураций и указывает для каждого Oracle Home, что нужно сделать, чтобы соответствовать стандартной конфигурации. Этот список становится основой Database Fleet Maintenance. Oracle рекомендует администраторам запускать отчет Software Standard Advisor ежеквартально, чтобы понять состояние своих конфигураций базы данных.

Текущие конфигурации ПО (295)



Рекомендованные конфигурации ПО (8)

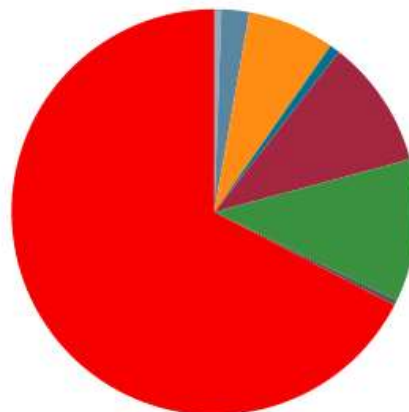


Рис. 5. Пример отчета Software Standard Advisor

Edition Based Redefinition (EBR)

EBR — это возможность обновления приложений на лету.

Обновление приложений без остановки

Эта возможность появилась в СУБД Oracle версии 11.2, по-английски ее называют killer feature — выдающейся возможностью СУБД Oracle, аналогов которой нет в других СУБД. [Edition Based Redefinition \(EBR\)](#) — это возможность обновления кода приложений в СУБД Oracle на лету, без остановки работы приложения.



Приложение, работающее в базе данных, состоит из кода: хранимых процедур, функций, PL/SQL-пакетов, представлений, пользовательских типов и т. д. Внесение изменений в код, хранящийся в СУБД, означает замену одних процедур, функций, пакетов, представлений, синонимов другими. При этом старый код теряется, а его место занимает новый код.

Процедура обновления кода приложения требует даунтайма, потому что вначале новые объекты в СУБД Oracle необходимо создать, а затем откомпилировать. Если обновляемых объектов много, процедура обновления кода и его компиляции может продолжаться несколько часов. Время зависит от количества строк кода и сложности зависимостей между объектами.

При удалении объектов также теряются и привилегии пользователей на пользование этими объектами: пакетами, процедурами, функциями и т. д. Поэтому привилегии (grants, права пользователей на работу с объектами) требуется заново раздать. И очень часто на это уходит много времени. Если какому-то пользователю не хватает прав на работу с какими-то объектами, код может не скомпилироваться или неправильно работать.

Если после перехода на новую версию что-то пошло не так, то заказчику требуется возможность отката к прежним версиям процедур, функций, пакетов. Поэтому перед обновлением всегда возникает вопрос о том, каким образом в случае необходимости можно откатиться на предыдущее состояние (к старому коду), ведь предыдущие версии объектов уже отсутствуют, их место заняли новые процедуры и функции. То есть процедура отката версии приложения есть фактически процедура наката предыдущей версии кода + раздача грантов + компиляция.

Длительные даунтаймы очень болезненны для телекомов, банков, продаж через Интернет. Например, пользователи, не сумевшие расплатиться банковской картой, будут очень недовольны.

Учитывая сложность процесса обновления приложений, в идеале хотелось бы иметь возможность собрать, откомпилировать, раздать необходимые права и протестировать новую версию кода, не удаляя предыдущую версию, а затем просто переключить пользователей на новую версию ПО.

Тестирование на боевой БД дает наилучшую возможность убедиться, что новая версия работает правильно, корректно, не хуже, чем предыдущая версия. Но у кого есть возможность протестировать приложение на боевой БД и боевом сервере?

EBR позволяет избежать этих проблем. EBR — это возможность создать и отладить новую версию приложения одновременно с работающей версией, не останавливая промышленной эксплуатации. Новую версию приложения (новый слой) можно собрать заранее, заранее скомпилировать и даже частично выполнить — протестировать.

EBR базируется на понятии Edition — это новый слой, новая версия ПО. Если раньше каждый объект уникально идентифицировался именем схемы и именем объекта (например, в одной схеме мог быть только один пакет с уникальным именем), то, начиная с версии 11.2, в СУБД Oracle появилось еще одно измерение — Edition. И в одной схеме могут быть несколько пакетов с одним и тем же именем, но в разных Edition. Поэтому технология EBR заключается в том, чтобы создать на боевой БД новый Edition, собрать в этом Edition новую версию приложения (новый слой ПО), раздать гранты (права), откомпилировать код и протестировать его, а после успешного тестирования просто переключить пользователей на новый слой приложения.

При этом предыдущий Edition (слой кода) сохраняется, а заказчик может свободно переключиться на него при необходимости (решение проблемы отката). Предыдущий слой можно будет удалить позже, когда в нем отпадет необходимость (через несколько недель или месяцев).

EBR не является опцией базы данных и не входит ни в какие опции, от нее нельзя отказаться. EBR является частью ядра СУБД Oracle Enterprise Edition и не требует приобретения дополнительных опций и явного включения каких-либо параметров базы данных.



Обновление структур данных: DBMS_REDEFINITION

Не все объекты БД могут быть многоверсионными, не все объекты могут иметь Edition. Объекты БД, которые могут быть версионными:

- Synonym
- View
- Все PL/SQL-объекты:
 - FUNCTION
 - LIBRARY
 - PACKAGE и PACKAGE BODY
 - PROCEDURE
 - TRIGGER
 - TYPE и TYPE BODY

Все остальные объекты базы данных: таблицы, индексы, материализованные представления и т.д. — не имеют версий.

Но что делать, если изменение кода приложения требует одновременного изменения структур данных? Например, изменения типов столбцов в таблице, изменения названий столбцов, переход на новую схему секционирования. В этом случае для изменения структуры таблиц (без остановки работы приложения) используется пакет [DBMS_REDEFINITION](#). Он выполняет большое количество различных трансформаций таблиц и индексов: добавление и удаление столбцов, изменение типа данных столбцов, преобразование простой таблицы в секционированную, изменение типа секционирования, перенос таблицы в другое табличное пространство и т.д. — почти без остановки приложения. Изменяемый объект в процессе изменений остается доступным для запросов Select и DML-операций почти все время. Короткая блокировка таблицы на несколько секунд или десятков секунд выполняется в завершающий момент завершения операции преобразования. Для уменьшения даунтайма рекомендуется выполнять завершающее преобразование таблиц в периоды минимальной нагрузки.

Например, допустим, у вас есть приложение на PL/SQL, где вводится имя, отчество, фамилия человека как единое поле и сохраняется в 1 колонке таблицы. Вы хотите сделать новое приложение, где имя, отчество, фамилия вводятся отдельно и хранятся в 3 разных колонках таблицы. Для этого в таблицу добавляются эти 3 новых поля и делаются 2 View. Одно видит старый набор колонок таблицы, другое — новый набор колонок таблицы. Старое приложение работает на чтение с первым view, а новое — со вторым.

А если вы пытаетесь в старом или новом приложении изменить или добавить значения в упомянутые выше колонки, срабатывают триггеры CrossEdition (их должен написать разработчик), которые синхронизируют старые и новые поля записи (в нашем примере слепляют или разделяют значения «Имя», «Отчество», «Фамилия»).

Какое-то время старая и новая версии приложения сосуществуют в БД и работают одновременно. После того как новый код протестирован, новая редакция (Edition) активизируется для эксплуатационной системы.



Глобальные распределенные сервисы (Global Data Services, GDS)

Введение

Концепция Database Services (сервисы базы данных) представляет собой механизм, используемый для управления нагрузкой в Oracle Real Application Clusters (RAC). Сервис базы данных (Database Service) — это логические абстракции для управления нагрузкой в базе данных Oracle. Каждый сервис представляет рабочую нагрузку с общими атрибутами, порогами уровня сервиса и приоритетами. Группировка может быть основана на атрибутах работы, которые могут включать в себя определенную функциональность приложения, которая будет использоваться. Так, пакет бизнес-приложений Oracle E-Business определяет сервис базы данных для каждого своего модуля, например: «Главная книга», «Дебиторская задолженность», «Ввод заказа» и т. д.

Концепция глобальных сервисов

Технология Global Data Services (GDS) расширяет концепцию Database Services для поддержки реплицируемых баз данных, которая включает в себя любую комбинацию из основной БД и ее реплик, синхронизируемых с помощью технологий Oracle Active Data Guard или Oracle GoldenGate. Это позволяет развертывать сервисы БД в любом месте этой глобальной конфигурации, поддерживая балансировку нагрузки, высокую доступность, региональную привязку и т. д. GDS построен на проверенных временем технологиях, таких как Database Services, Oracle Active Data Guard, Oracle Golden Gate и Oracle Net Listener.

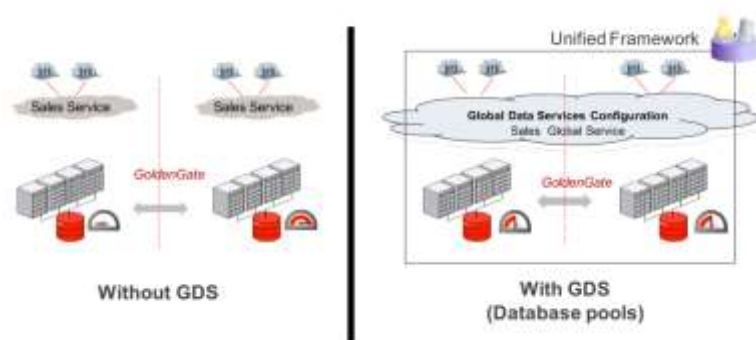


Рис. 1. Концепция глобальных сервисов

Ключевые особенности глобальных сервисов:

- Могут быть read-only или read-write.
- Соединение с GDS осуществляют специализированные листенеры (GDS-листеры).
- На каждый регион (площадку) имеется свой GDS-листер.
- Формат дескриптора TNS расширен для указания региона, к которому относится клиент.
- GDS-листеры осуществляют балансировку нагрузки между БД.
- В процессе балансировки GDS-листер учитывает не только степень загрузки БД, но и сетевую задержку до БД и время отставания (все эти параметры задаются в виде атрибутов gds-сервиса).
- Информация о gds-сервисах хранится в специальном каталоге, который реплицируется между регионами.
- Для клиента использование GDS-сервиса прозрачно. В частности, точно так же отрабатывает технология Transparent Application Failover при потере текущей БД или узла (если это кластерная БД).



Преимущества Global Data Services

GDS позволяет разворачивать и централизованно управлять сервисами баз данных (через набор реплицируемых баз данных), которые отказоустойчивы. Структура GDS обеспечивает балансировку рабочей нагрузки между этими базами данных.

В частности, решение, интегрированное с GDS, обладает следующими преимуществами:

- Более высокая доступность и глобальная масштабируемость позволяют динамически и прозрачно добавлять реплицированные базы данных в инфраструктуру GDS для получения дополнительных возможностей ресурсов для масштабирования нагрузки. GDS позволяет сделать это без изменений в конфигурации приложения или клиента.
- Повышение производительности и эластичности. Благодаря встроенной балансировке нагрузки между несколькими базами данных GDS устраняет проблему фрагментации ресурсов между регионами. Недостаточно используемые ресурсы в одном регионе теперь могут быть использованы для того, чтобы снизить нагрузку на перегруженные ресурсы другого региона и таким образом достичь оптимального использования ресурсов.
- В пуле GDS, содержащем реплицированные базы данных, работающие на серверах баз данных разных поколений процессоров и разных ресурсов (ЦП, память, ввод/вывод), GDS отправляет запросы на работу в более производительные базы данных, но до тех пор, пока они не будут перегружены. Таким образом, обеспечивается выравнивание времени отклика.

GDS и архитектура максимальной доступности

В архитектуре MAA до появления GDS, чтобы оптимально распределить рабочую нагрузку по нескольким синхронизированным копиям БД, заказчики использовали балансировщики нагрузки сторонних производителей либо специально разрабатывали диспетчеры соединений на уровне приложения. Решения на основе сторонних балансировщиков нагрузки требовали значительных затрат на интеграцию и лицензирование, а для самостоятельной разработки на уровне приложения требуются высокие первоначальные затраты и время, а также требуются накладные расходы на обслуживание и поддержку.

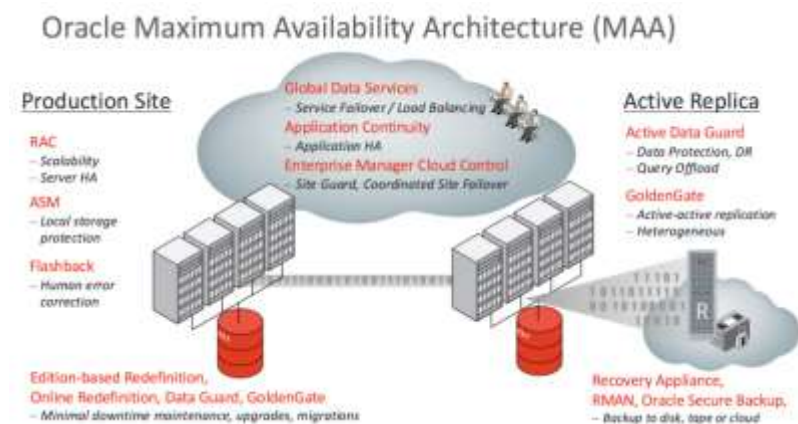


Рис. 2. GDS как компонент Maximum Availability Architecture

С появлением Global Data Services заказчики Oracle теперь могут объединять ресурсы реплицируемых баз данных в единую платформу, что исключает необходимость дополнительной разработки или сторонней интеграции для балансировки нагрузки. Global Data Services — это новый стратегический компонент MAA, доступный в Oracle Database. GDS хорошо интегрируется с экосистемой Oracle, что обеспечивает маршрутизацию соединений, балансировку нагрузки и аварийное переключение сервисов БД между реплицированными базами данных, расположенными как внутри центров обработки данных, так и между ними. Таким образом, GDS является балансировщиком нагрузки для реплицируемых баз данных,



в дополнение к обеспечению высокой доступности через функцию переключения соединения между базами данных после сбоя.

Облачные вычисления. Публичное облако Oracle (Public Cloud)

Марк Хёрд, CEO Oracle:
«К 2025 году 80 % ИТ-бюджетов «уйдут в облака»

Облачные вычисления, облачные технологии, публичное облако (cloud computing, public cloud) — это современная бизнес-модель предоставления в аренду ПО и оборудования, в соответствии с которой пользователь получает ресурсы в виде интернет-сервисов. Иными словами, облачные вычисления – это **способ удаленного предоставления пользователям вычислительных ресурсов во временное пользование за арендную плату**.

В этой схеме с одной стороны существует **владелец** облачных ресурсов (**облачный провайдер**), который на профессиональной основе занимается предоставлением своих ресурсов потребителям и несет все расходы по их поддержанию в рабочем состоянии. С другой стороны, имеется пользователь — **арендатор** ресурсов, который берет эти ресурсы во временное пользование - аренду - для решения своих бизнес-задач. Ресурсы публичного облака предоставляются арендатору на основе подписки или периодической оплаты за потребленный объем.

Институт инженеров электротехники и электроники (Institute of Electrical and Electronics Engineers, IEEE) дает определение облачных технологий как «парадигму постоянного хранения пользовательской информации на интернет-серверах, которая лишь временно кэшируется на стороне пользователя». Слово «облако» здесь присутствует как метафора, олицетворяющая сложную инфраструктуру, скрывающую за собой все технические детали.

Публичность облака означает совместное использование одних и тех же ресурсов всеми арендаторами этого облака. Т.е. процессоры, память, сеть — это совместно используемые ресурсы. Дисковые ресурсы принадлежат одному пользователю, пока он не освободит их. Данные одних пользователей могут храниться на одном физическом сервере с данными других компаний, однако доступ к чужим данным при этом отсутствует. **Облачность** означает виртуальное пространство; это значит, что нельзя однозначно сказать, на каком именно физическом оборудовании будет выполняться обработка данных, поскольку виртуальные машины могут перемещаться между серверами для балансировки нагрузки и повышения отказоустойчивости.

Экономические аспекты

Обычно интерес к облаку возникает в периоды, когда на предприятии заканчивается срок жизни предыдущего оборудования. Предприятию предстоит обновить оборудование и встает вопрос выбора нового оснащения с запасом на 3–5 лет вперед. Возникает потребность проанализировать профиль нагрузки и дать прогноз на будущее.

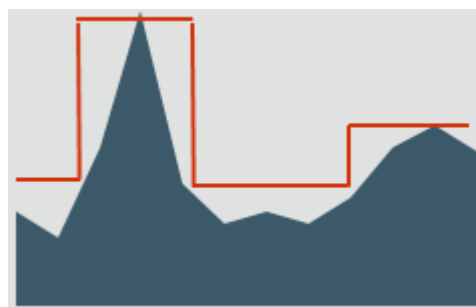
Типичный профиль нагрузки складывается из короткого периода высокой нагрузки и длительных периодов средней и низкой нагрузки. Если период пиковой нагрузки длится 2–6 часов в сутки, то его длительность составляет 8–25 % от 24 часов. В остальное время ресурсы оборудования не используются на полную



мощность, т. е. 75-92 % времени значительная часть ресурсов не используется. Если учесть отсутствие нагрузки в выходные и праздничные дни, то годовая эффективность будет еще ниже.

Как следствие, предприятия с большим запасом оплачивают как ресурс самого оборудования, так и лицензии на программное обеспечение.

Для сравнения: в облаке оплата за использованные ресурсы выглядит примерно так, как показывает красная черта на рисунке справа. Оплата в облаке максимально приближена к реальной нагрузке, ничего лишнего.



Ресурсы в облаке, не использованные одним потребителем, остаются другим потребителям. Лицензии на ПО приобретает провайдер. За счет этого стоимость облака для потребителя в несколько раз ниже, чем стоимость владения собственными ресурсами и лицензиями.

Поскольку арендатору не требуется поддерживать в рабочем состоянии оборудование и ПО (эту работу берет на себя облачный провайдер), то потребители облака могут сконцентрироваться на решении своих бизнес-задач. Использование облачных вычислений позволяет потребителям снизить капитальные расходы на построение центров обработки данных, не приобретать серверное и сетевое оборудование, исключить затраты на приобретение программного обеспечения.

Длительные сроки строительства и ввода в эксплуатацию крупных объектов инфраструктуры информационных технологий и их высокая начальная стоимость ограничивают способность потребителей гибко реагировать на требования рынка, тогда как облачные технологии обеспечивают возможность практически мгновенно реагировать на увеличение спроса на вычислительные мощности. При использовании облачных вычислений затраты потребителя смещаются от капитальных затрат CAPEX в сторону операционных OPEX.

Облачный провайдер использует автоматический механизм, оценивающий объем потребленных пользователем ресурсов и выставяющий счета за их использование — **биллинг**.

Основные признаки облака

Национальный институт стандартов и технологий США (National Institute of Standards and Technology, NIST, <https://www.nist.gov/>) определяет облачные вычисления как «модель обеспечения сетевого **доступа по требованию** к некоторому общему **фонду** конфигурируемых **вычислительных ресурсов и ПО** (сетям передачи данных, серверам, устройствам хранения данных, приложениям и сервисам), которые могут быть оперативно **предоставлены** и **освобождены** пользователем самостоятельно с минимальным обращением к провайдеру».

<https://csrc.nist.gov/Projects/Cloud-Computing>

<https://csrc.nist.gov/publications/detail/sp/800-145/final>

Это определение выделяет 5 основных признаков облака:

- Наличие пула совместно используемых ресурсов (resource pooling)
- Самообслуживание по требованию (on demand self service)
- Эластичность предоставления ресурсов (rapid elasticity)
- Оплата по мере использования (measured service, pay as you go)
- Доступ по сети из любой точки мира (broad network access)



Наличие пула ресурсов (*resource pooling*) — поставщик услуг объединяет ресурсы для обслуживания большого числа потребителей в единый пул для динамического перераспределения мощностей между потребителями в условиях постоянного изменения спроса на мощности.

Совместное использование ресурсов — множество пользователей осуществляют совместный доступ к одним и тем же ресурсам. Это означает, что каждый потребитель контролирует свои параметры ресурсов и услуг (например, объем данных, скорость доступа), а фактическое распределение ресурсов, предоставляемых потребителю, осуществляет провайдер. В ряде случаев потребители все-таки могут управлять некоторыми физическими параметрами распределения ресурсов, например, выбрать желаемый регион или страну, в которой будут размещаться ресурсы.

Самообслуживание по требованию (*on demand self service*) — потребитель самостоятельно определяет перечень необходимых ему вычислительных ресурсов, таких как количество ЦПУ, объем оперативной памяти, дисковые объемы, перечень используемого ПО, сервисов и приложений, самостоятельно заказывает и изменяет объемы потребляемых ресурсов. **Портал самообслуживания** является инструментом, посредством которого пользователь может заказать для себя заранее предопределенный сервис с уточнением деталей конфигурации (например, указать объем оперативной памяти, число процессорных ядер, размер дискового пространства и т. п.), изменить параметры ранее заказанного сервиса и отказаться от него. Портал самообслуживания также содержит:

- **Каталог сервисов** — список сервисов и связанные с каждым из них шаблоны их создания (правила, по которым средства автоматизации будут конфигурировать данный сервис на реальном оборудовании и программном обеспечении).
- **Оркестратор** — механизм, выполняющий последовательность операций, определенных в шаблоне для каждого сервиса.
- Финансовый раздел, в котором пользователь может получить отчет о потребленных ресурсах, увидеть выставленные счета и оплатить их (**система тарификации и выставления счетов**).

Эластичность (*rapid elasticity*) — пользователь может в любой момент изменить выделенный ему объем ресурсов и набор услуг. Услуги могут быть расширены или сужены пользователем в любой момент времени самостоятельно, без взаимодействия с облачным провайдером, как правило, в автоматическом режиме.

Учет потребления — облачный провайдер автоматически регистрирует объемы потребленных ресурсов, и на основе этих данных выставляет счет потребителю. Обычно облачные провайдеры предлагают модель оплаты «за использованные ресурсы» (*pay-as-you-go*), что снижает издержки пользователей по сравнению с затратами на построение собственных ЦОД.

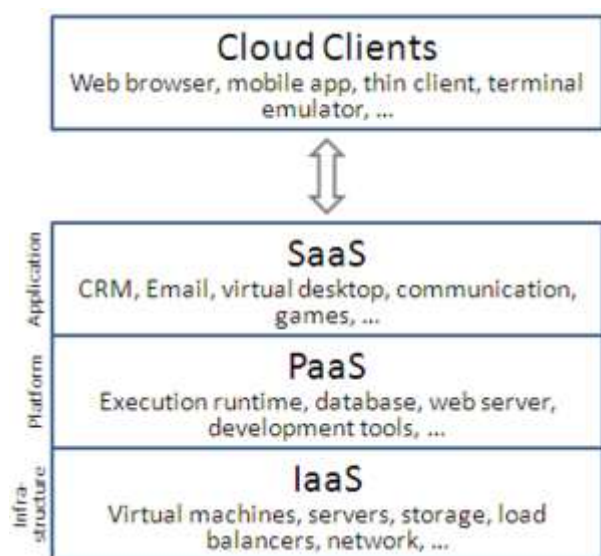
Доступ по сети с любого устройства — облачные услуги доступны потребителям по сети передачи данных вне зависимости от используемого терминального устройства. В большинстве случаев все, что необходимо пользователю для работы с облаком, — это доступ в Интернет и браузер.

Облачные сервисы являются комбинацией существующих технологических решений, которые взаимно интегрированы для обеспечения максимального автоматизма и минимизации участия человека в работе комплекса.



Сервисные уровни

Главный принцип сервис-ориентированной архитектуры — «все есть сервис». NIST выделяет три основных сервисных уровня: **IaaS** (*Infrastructure-as-a-Service*), **PaaS** (*Platform-as-a-Service*), **SaaS** (*Software-as-a-Service*):



Инфраструктура как услуга (IaaS, Infrastructure-as-a-Service, Hardware as a Service) — пользователю предоставляется оборудование или виртуальные машины. В соответствии с определением NIST IaaS, пользователю предоставляется возможность поставить ОС и приложения. Пользователю не дается контроль над оборудованием, но дается контроль над программно-аппаратным стеком от уровня VMC и выше (гипервизор, виртуальные машины, ОС, СУБД, веб-серверы, приложения).

Прикладную настройку пользователю предстоит создать самостоятельно. Например, потребитель может установить произвольное программное обеспечение, которое может включать операционные системы, виртуальные машины, различные СУБД и приложения. Пользователь в этом случае имеет полный контроль и несет полную ответственность за ПО, которое он установил, сам выполняет установку обновлений и контролирует совместимость версий.



Пользователь самостоятельно решает вопрос с лицензиями на установленное им ПО, т. е. обязан принести свои собственные лицензии на СУБД, на приложения (BYOL — Bring Your Own Licenses) или арендовать необходимые лицензии у провайдера. Контроль и управление нижележащей физической и виртуальной инфраструктурой облака (сеть, серверами, системами хранения) осуществляется облачным провайдером.

Достоинством IaaS является гибкость и быстрота выделения новых ресурсов. IaaS-провайдер предоставляет эти ресурсы по требованию из пула своих ресурсов. Стоимость в данном случае считается на основе выделенных и потребленных ресурсов.

Платформа как услуга (PaaS, Platform-as-a-Service) — модель, когда потребителю предоставляется облачная инфраструктура для размещения и создания программного обеспечения. NIST определяет PaaS как возможность для пользователя «приземлить» свои приложения. Т. е. PaaS-провайдер предоставляет пользователю ОС, СУБД, языки программирования (компиляторы), библиотеки, веб-серверы, балансировщики нагрузки, средства интеграции, ETL, инструменты для анализа данных, blockchain, IoT, AI и т. д. Пользователь в этом случае не думает об операционной системе, файловых системах, а сосредоточен на работоспособности своего приложения и его настройках.



Иными словами, PaaS-провайдер предоставляет среду для работы приложения (PaaS-платформу), а разработчик приложения пользуется программным обеспечением в облаке, не обременяя себя заботами о создании и эксплуатации ЦОД. Такая модель подходит разработчикам программного обеспечения и может быть интересной компаниям-интеграторам, чтобы не разворачивать свой ЦОД, а получить готовую платформу, сопровождать которую будет PaaS-провайдер.

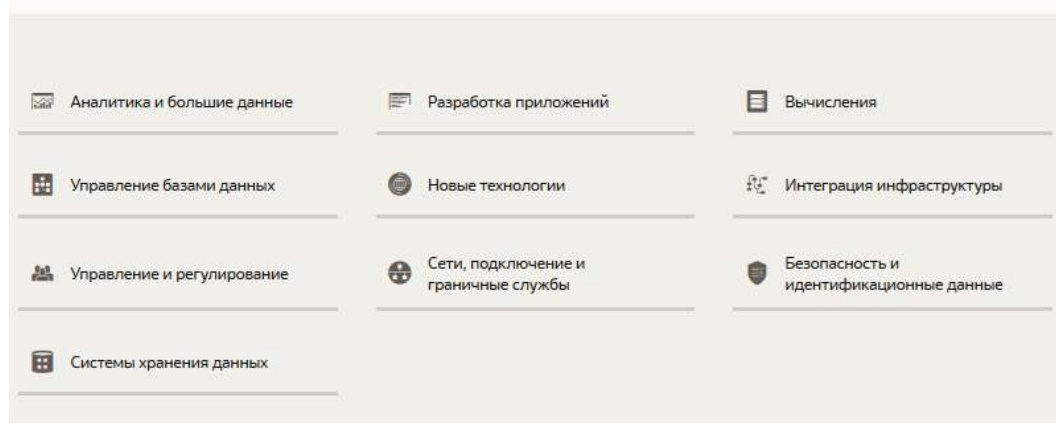


Программное обеспечение как услуга (Software-as-a-Service, SaaS) — в этом случае потребитель берет в аренду готовые приложения, находящиеся в собственности облачного провайдера. При этом ПО работает в облачной инфраструктуре на оборудовании провайдера и доступно пользователю из различных клиентских устройств и браузера. За работоспособность приложения, за его обновление и набор индивидуальных возможностей приложения для конкретного пользователя отвечает облачный провайдер. В модели SaaS у пользователей отсутствует доступ к оборудованию и платформе, есть только доступ к интерфейсу приложения.

SaaS — это фактически «ПО как сервис», поэтому оно оплачивается по принципу pay-per-use (по мере использования) или по принципу абонентской платы за подписку (subscription fee). Оплата для модели SaaS обычно ежемесячная или по количеству зарегистрированных пользователей. Типичными приложениями, предлагаемыми по модели SaaS, являются игры.

Продукты Oracle Cloud Infrastructure по категориям

Oracle Cloud Infrastructure включает большой ассортимент продуктов, которые помогают создавать, переносить, использовать и расширять приложения и рабочие нагрузки. В него входят основные службы вычислений, хранения, баз данных, сетей, безопасности, управления, интеграции и аналитики, а также службы для разработчиков. Все эти продукты используют преимущества архитектуры Oracle Cloud, где первостепенное внимание уделяется производительности, безопасности и поддержке как существующих, так и будущих рабочих нагрузок.



<https://www.oracle.com/ru/cloud/products.html>

Достоинства и недостатки публичного облака

Достоинства публичного облака:

- Экономия на строительстве и обслуживании ЦОД: здание, охрана, электроснабжение, охлаждение и т. д.



- Экономия на приобретении и обслуживании оборудования: пользователям нет необходимости приобретать в собственность дорогие компьютеры с большим объемом памяти и дисков.
- Экономия на приобретении программного обеспечения, экономия на обслуживании ПО. Установку ПО, установку обновлений, диагностику работоспособности и устранение ошибок выполняет облачный провайдер.
- Модель оплаты pay-as-you-go позволяет оплачивать только использованный объем ресурсов.
- Обслуживанием облака занимается специализированная организация — облачный провайдер.
- Доступ к ресурсам облака возможен из любой точки мира, где есть Интернет. Если документы хранятся в облаке, они могут быть доступны пользователям в любое время и в любом месте.
- Простота совместной работы группы пользователей.
- Безопасность и надежность. В облаке настроены автоматические процедуры резервного копирования, чтобы гарантировать надежное хранение данных. Для надежности копии автоматически размещаются на разных системах хранения, возможно даже находящимся на разных континентах.

Недостатки публичного облака:

- Необходимость выхода в Интернет для доступа к данным. Cloud Computing всегда требует соединения с сетью Интернет.
- Скорость доступа в Интернет также имеет значение, некоторые облачные программы требуют быстрого интернет-соединения с большой пропускной способностью.
- Отсутствие контроля со стороны заказчика: работоспособность услуг полностью подчинена провайдеру.
- Вложения в облако не являются капитальными вложениями, т. к. материальные активы не приобретаются.
- Безопасность данных в значительной степени зависит от облачного провайдера: проверенный персонал, соблюдение технологических правил, правильное использование алгоритмов шифрования и условий доступа к данным. Безопасность данных в облаке также в значительной степени зависит от самого пользователя, от соблюдения сотрудниками пользователя правил безопасности. Утечки в большинстве случаев происходят по вине пользователя.
- Надежность. Если Ваши данные в облаке потеряны, они потеряны навсегда. Поэтому наличие копии данных в облаке не отменяет необходимости иметь локальную копию.

Программы, в которых требуется передача значительного количества информации, будут работать на локальном компьютере быстрее не только из-за ограничений скорости доступа в Интернет, но и из-за загруженности удаленных серверов и проблем на пути между пользователем и облаком.

Виды облаков: публичное, частное, гибрид

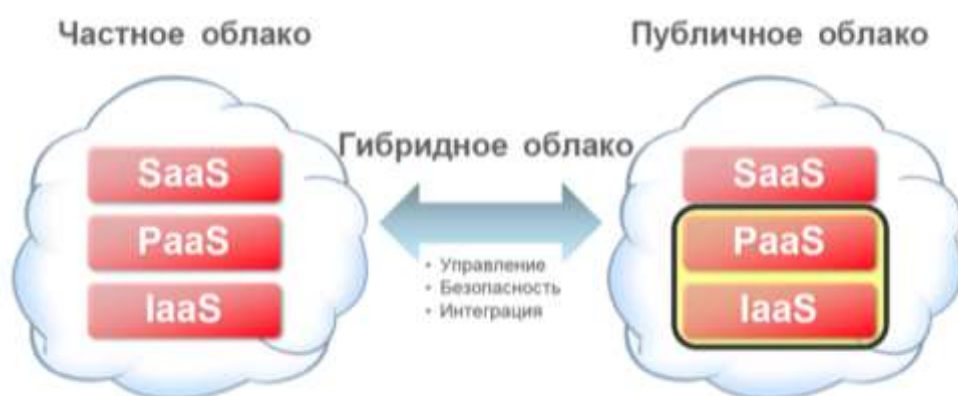
Облака бывают публичными или частными. По определению, «публичное облако» находится где-то далеко «за горизонтом», и не все компании могут его использовать. Для этого есть несколько причин:

- Законодательные ограничения, установленные государственными органами и регуляторами;
- Политики безопасности организаций;
- Технические требования приложений, которым требуются низкие сетевые задержки как в локальной сети; приложения или базы данных, которые тесно переплетены между собой и которые сложно разделить.



Перечисленные выше проблемы решает **частное облако** - облачная инфраструктура, развернутая в ЦОД заказчика на оборудовании заказчика и обслуживаемая специалистами заказчика. В этом случае заказчики могут воспользоваться самыми ценными преимуществами облака — его гибкой, масштабируемой и эффективной инфраструктурой, и при этом получать более широкие возможности управления и защиты, чем при использовании публичного облака.

Таким образом, частное облако решает проблему медленной сети, так как доступ к услугам осуществляется через внутренние сети предприятия, а не через Интернет. Частное облако решает проблему законодательных ограничений и удовлетворяет внутренним политикам безопасности предприятий. Частное облако — это лучший выбор для заказчиков, которым подходят технологии публичного облака, но для работы в публичном облаке существуют ограничения. С учетом этих факторов частное облако больше подходит для больших и сложных приложений или организаций, а также для предприятий, на которых действуют очень строгие правила безопасности в отношении обработки данных.



Гибридное облако — это сочетание публичного и частного облаков. В ряде случаев заказчикам приходится делать выбор между публичным или частным облаком, например, приходится принимать решение каждый раз отдельно для каждого приложения. Однако при переходе к гибридной облачной модели заказчикам больше не требуется принимать решения «или/или»: **гибридное облако** позволяет компаниям использовать преимущества обоих облаков. Гибридное облако обеспечивает такую же инфраструктуру, политики безопасности и наборы инструментов, как публичное облако.

Одно из основных преимуществ гибридного облака заключается в том, что оно позволяет заказчикам перемещать приложения и рабочие нагрузки между средами в зависимости от спроса, потребностей бизнеса и других факторов. Такой смешанный подход дает предприятиям возможность быстро реагировать при эксплуатационной доработке приложений. Например, можно использовать публичное облако для быстрого и экономически эффективного развертывания и тестирования новых приложений, а потом переносить их за брандмауэр и запускать в эксплуатацию.

Это означает, что организации могут решать конкретные бизнес-задачи или удовлетворять определенные нормативно-правовые требования, например, связанные с местонахождением данных и управлением ими, но при этом пользоваться очевидными преимуществами публичного облака: его гибкостью и моделью оплаты по факту использования.

Построив гибридный «мостик» между разными типами облаков, компании смогут приобрести необходимую гибкость облачной инфраструктуры, работать оперативнее своих конкурентов и быстро отвечать на постоянно меняющиеся требования в условиях роста конкуренции.



Достоинства и недостатки частного облака

Плюсы частного облака

- Больше возможностей для контроля по сравнению с публичным облаком, поскольку все компоненты ИТ-инфраструктуры размещаются в ЦОД заказчика.
- Высокий уровень доверия к процедурам безопасности за счет того, что владельцем и потребителем сервиса является одна единственная организация.
- Инфраструктура может быть оптимально сконфигурирована под одного заказчика.
- Возможность обработки персональных данных.

Недостатки частных облаков

- Меньшая экономическая эффективность. При внедрении частного облака заказчику потребуется помещение для размещения оборудования (помещение, ЦОД), электричество, охлаждение, охрана. Это приводит к значительному увеличению расходов на всех этапах жизненного цикла облака по сравнению с публичным облаком. У публичных облаков более высокий коэффициент использования. При использовании частного облака оборудование придется приобретать исходя из пиковой нагрузки (с большим запасом).
- Частным облаком необходимо управлять, что влечет за собой затраты на администрирование и привлечение компетентных ИТ-специалистов.
- По сравнению с публичным облаком риски потери работоспособности сервисов и/или потери данных из-за физических угроз для инфраструктуры намного более существенны.
- Более близкий (по сравнению с публичным облаком) «потолок» ресурсов, когда ресурсов созданной под частное облако инфраструктуры может оказаться недостаточно. Развернуть новый сервис быстро можно только тогда, когда для этого соответствующие ресурсы, а увеличение ресурсов потребует времени.
- Большие, по сравнению с публичным облаком, затраты человеческих и материальных ресурсов для его создания и дальнейшей работы.

Публичное облако	Частное облако
ОРЕХ	CAPEX+ОРЕХ
Дешевле	Дороже
Затраты на инфраструктуру минимальные. ЦОД — это расходы облачного провайдера, которые распределяются на всех пользователей облака	Необходимость выстраивать ИТ-инфраструктуру и ЦОД (помещение, электричество, охлаждение, охрана). Необходимость вникать в функционирование ЦОД
Масштабирование лучше (быстрее, больше)	Хуже масштабирование

Oracle Infrastructure as a Service (OCI), DBaaS

Oracle Cloud — это облачный сервис компании Oracle, который позволяет получить доступ через интернет к разнообразным ресурсам, оборудованию и ПО, которые установлены в дата-центрах компании Oracle. Oracle Cloud предоставляет сервисы **IaaS** (Infrastructure as a Service), **PaaS** (Platform as a Service), **SaaS** (Software as a Service), **Daas** (Data as a Service). Oracle Cloud поддерживает:



- большинство открытых стандартов на SQL, HTML5, REST...
- приложения с открытым исходным кодом Kubernetes, Hadoop, Kafka...
- языки программирования SQL, Java, Python, Ruby...

История облака Oracle началась в 2013 году, когда компания Oracle объявила о построении своего облака. Для облака первого поколения ставилась задача научиться быстро выделять ресурсы по требованию, дать пользователям портал самообслуживания, отладить биллинг и научиться получать плату за использование облачных сервисов.

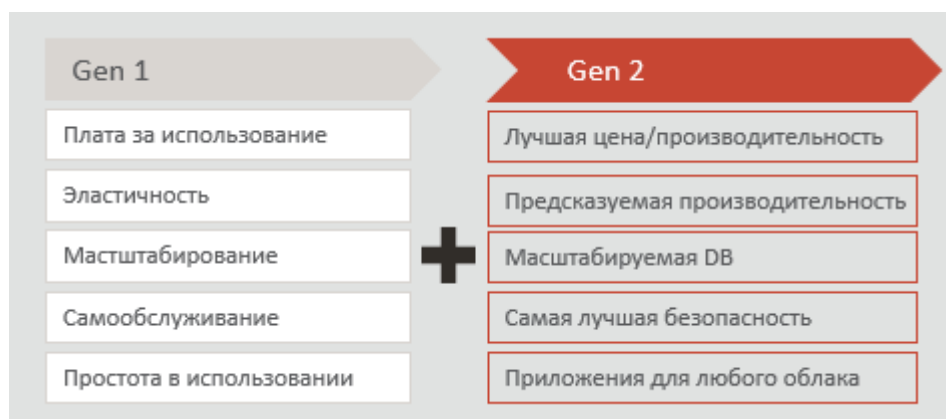


Рис.1. Сравнение Oracle Cloud поколений 1 и 2

С течением времени накопился определенный опыт эксплуатации облаков, и в 2015 году компания Oracle решила пересмотреть принципы, на которых они строят облачные технологии. К этому моменту в компании определились, что облако Oracle предназначено не только для частных лиц, желающих иметь одну виртуальную машину на одном ядре, но и в основном облако Oracle предназначено для корпоративных клиентов, для бизнеса, для предприятий. Для этого необходимо предоставить предприятиям то же самое оборудование, к которому они привыкли, обеспечить плавный и безболезненный перенос приложений из ЦОД заказчика в облако Oracle и обратно без каких-либо изменений приложения.

Таким образом, в 2018 году компания Oracle приступила к построению Oracle Cloud поколения 2.

РЕГИОНЫ

В мире были выбраны 36 наиболее перспективных географических регионов, в которых ожидается наибольший рост спроса на ИТ-услуги. В этих регионах планируется развернуть ЦОД Oracle. Регионы предоставляют собой точки присутствия, место нахождения ЦОД:





Рис.2. Карта размещения ЦОД Oracle

По состоянию на конец 2020 года завершено строительство ЦОД в 26 регионах, до к середине 2021 года планируется завершить строительство во всех 36 регионах. Регионы соединяются друг с другом посредством высокоскоростной сетевой магистрали, быстрой сети передачи данных, которая служит основой для катастрофоустойчивых решений — Disaster Recovery.

Центры обработки данных в облаке Oracle бывают трех типов: коммерческие, правительственные (уровня штата), правительственные федеральные. Центры обработки данных разного типа не имеют между собой ни общей инфраструктуры, ни доступа по сети, это полностью независимые системы ЦОД. Всем ЦОД присваивается realm (логическая группа ЦОД). Когда заказчик подписывается на облачные услуги, то учетной записи его организации (tenancy) присваивается некоторый realm, в соответствии с которым заказчик имеет доступ ко всем ЦОД с данным realm и не имеет доступа к регионам из других realm. На сегодняшний день в OCI имеется 3 realm: коммерческий (oc1) и два правительственных: oc2 для правительственных Government Cloud или oc3 для Federal Government Cloud. ЦОД коммерческого realm oc1 входят в домен oraclecloud.com. ЦОД Government Cloud (oc2) входят в документ oraclegovcloud.com.

OCI регионы состоят из нескольких ЦОД (Availability Domains), объединенных высокоскоростной магистралью. Availability Domain — это одно или три здания, расположенные на расстоянии 30–40 км друг от друга. Каждое здание (один AD) — это отдельный ЦОД со своей независимой системой электропитания, охлаждения, сетевой инфраструктурой и т. д. AD являются основой для построения высоконадежных систем; очень маловероятно, что они выйдут из строя одновременно. Один регион — это один или три Availability Domain (май 2020 г.).

Поскольку масштабное строительство сразу в 36 регионах мира требует значительных ресурсов, то для того, чтобы быстро запустить облачные сервисы, в наиболее важных регионах строятся по 3 AD на регион, а в менее значимых пока по одному. Со временем, по мере роста потребностей в каждом регионе будут построены дополнительные AD, а также, возможно, выбраны новые регионы, в которых запустят строительство новых ЦОД площадок.

Availability Domains

AD в разных регионах разделены большими расстояниями, что позволяет обеспечить защиту от катастроф (disaster recovery). Например, развернуть приложение и БД в одном регионе, а горячий резерв — в другом (в другой стране или даже на другом континенте).



AD внутри одного региона соединены друг с другом высокоскоростной сетью с низкими задержками: пропускная способность 1Тбит, задержки (latency) до 500 мкс между AD, до 100 мкс внутри AD. Дополнительно между AD выполняется шифрование трафика.

Availability Domains обеспечивают основу для высокой доступности внутри региона и содержат несколько fault domains. Рисунок показывает взаимосвязь регионов, дата-центров, Availability Domains, Fault Domains:

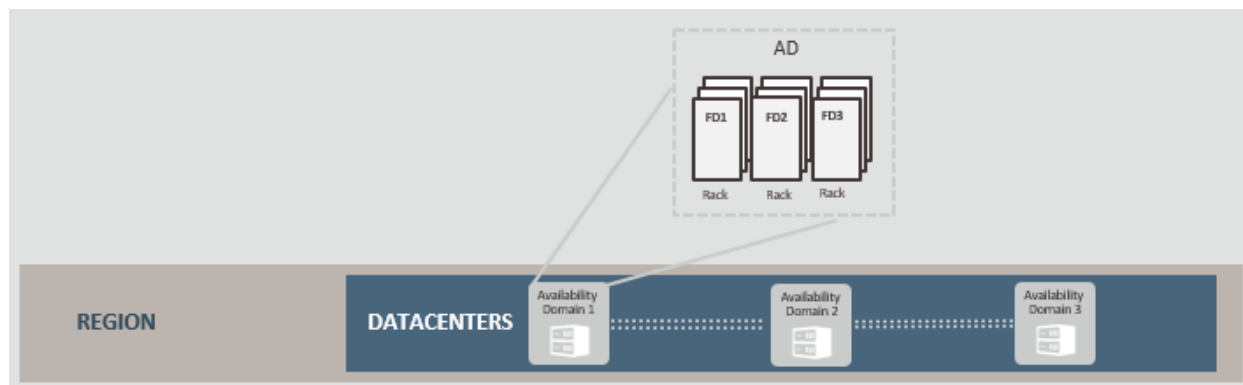


Рис. 3. Регионы, дата-центры, AD, FD

Fault Domains

Fault Domain (FD) — изолированные помещения внутри AD, обеспечивающие изоляцию групп оборудования. Каждый FD имеет свои независимые дублированные источники питания, которые защищают от отключений электричества.

Каждый AD состоит из трех fault domain. FD предназначены для создания систем высокой надежности, для того чтобы пользователь размещал разные группы оборудования в разных FD. Например, если серверы БД+СХД размещаются в первом FD, то систему резервного копирования лучше установить в другом FD, чтобы нештатные ситуации в первом FD не затронули систему резервного копирования. Если у пользователя RAC из 3 узлов, то каждый узел кластера можно разместить в своем FD. Таким образом, в случае плановых работ или нештатных ситуаций недоступным станет только оборудование, находящееся в одном FD, а остальные FD будут продолжать работать. Использование FD защищает от непредвиденных сбоев по электропитанию и от простоев при производстве плановых работ.

При выборе расположения Ваших компьютерных ресурсов рекомендуется явно указывать FD. Если FD явно не указан, то система автоматически разместит все заказы в один FD (оптимизирует заполнение свободного пространства). Для того чтобы изменить FD, остановите инстанс и запустите его в новом FD, указав для него предпочитаемый FD.

Сеть

Сетевой дизайн, задержки и пропускная способность имеют важное значение для работы предприятия. Серверы внутри AD взаимодействуют друг с другом и с СХД, и от этого может зависеть производительность приложений. Физическая сеть, на которую опирается OCI, спланирована так, чтобы поддерживать все сервера, используя их полную пропускную способность одновременно. Все коммутаторы поддерживают скорость 25 Гбит/с, плоская без переподписки сеть разработана так, чтобы позволить приложениям использовать максимальную пропускную способность и обеспечить отсутствие «шумных соседей». Сеть OCI гарантирует, что всем Вашим задачам будет предоставлено достаточно ресурсов, независимо от задач других пользователей. AD проектируется из расчета примерно 1млн портов в одном AD. Предсказуемо низкие задержки менее 100 мкс внутри AD, не более двух hop между любыми точками сети.

Сеть с виртуализацией — переносит виртуализацию V/B из гипервизора в сеть, что позволяет уменьшить задержки, привносимые гипервизором и виртуализацией на уровне ОС. В то время как другие облачные



провайдеры виртуализируют IO с помощью гипервизора, OCI делает это сетью, предоставляя лучшую инкапсуляцию и изоляцию, а также уникальные вычислительные опции.

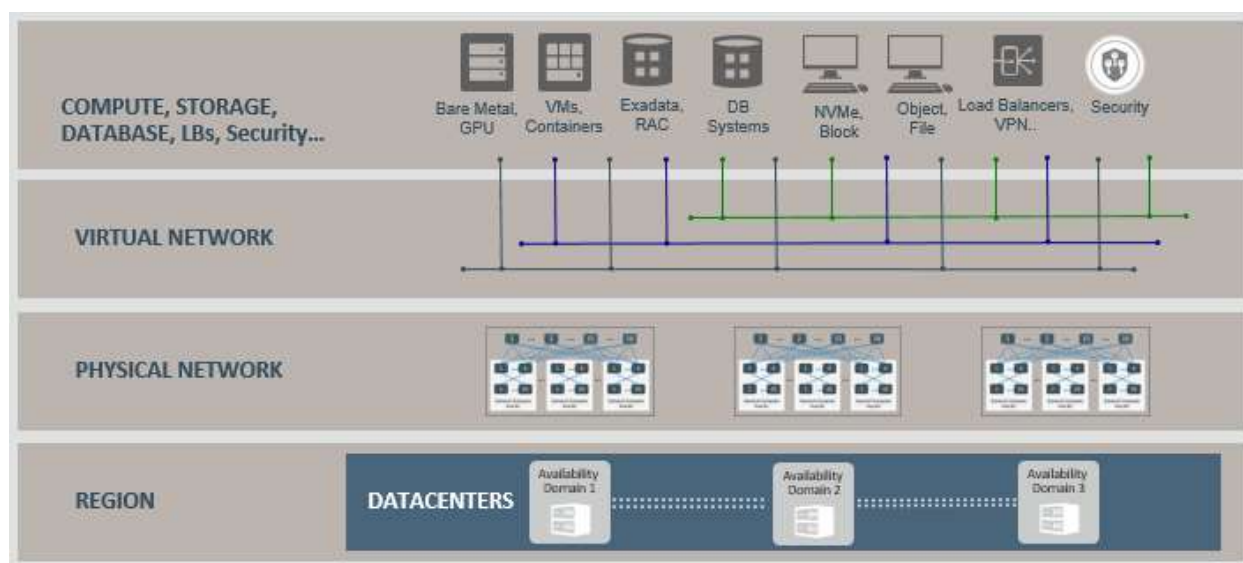


Рис. 4. Регионы, сети, ресурсы

Сеть с виртуализацией означает, что пользовательские VM не будут нагружаться виртуализацией ввода-вывода. Сеть спроектирована с расчетом на использование NVMe, чтобы обеспечить производительность В/В на максимальном уровне, который необходим заказчикам. Виртуальный сетевой дизайн обеспечивает базам данных миллионы транзакций в секунду, а DB масштабируются до сотен терабайт при высочайшем уровне доступности.

Сеть для High Performance Computing (HPC) нагрузок

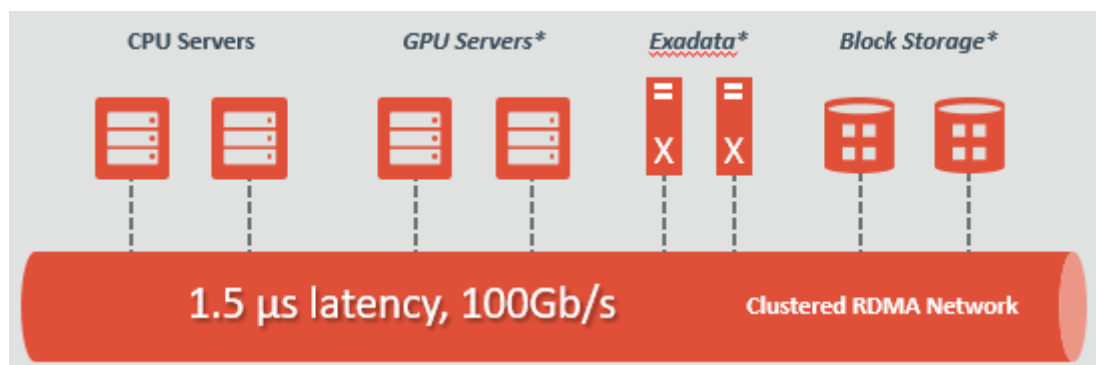


Рис. 5. Сеть для HPC-нагрузок

Сеть для HPC-нагрузок — это первая в облаках промышленная сеть с поддержкой RDMA, позволяет обслуживать тесно связанные серверы, предоставляя для них интерконнект. Кластерная сеть — это сеть на основе RDMA, которая позволяет заказчикам строить кластеры, которые используют сеть с низкими задержками между узлами кластера.

Управление пользователями

Структурирование пользователей на организации-департаменты-отделы является основой управления пользователями. Каждая организация — это отдельная **Tenancy** (объединяет все ресурсы данной организации и управляет ими). Tenancy делится на **Compartments**. Compartments — это подразделения внутри Tenancy, которые отражают структуру данной организации: департаменты, управления, отделы. Compartments могут



иметь иерархическую структуру и наследуют права родительских Compartments. **Пользователь** — это учетная запись с назначенными ей правами доступа к ресурсам некоторого Compartment. Пользователей можно группировать в группы и группам давать права на доступ к облачным ресурсам. Таким образом, Tenancy и Compartments организуют облачные ресурсы в контролируруемую схему. Каждый облачный ресурс принадлежит одному и только одному Compartment.

OCI позволяет назначить каждому Compartment свои собственные ресурсы (виртуальные машины, серверы, блочные устройства, приложения и т. д.), которые будут недоступны для других Compartments. Биллинг OCI позволяет учитывать потребление ресурсов в разрезе Compartments, позволяет анализировать стоимость ресурсов, потребленных разными подразделениями организации.

Безопасность

В многопользовательской среде безопасность является краеугольным камнем. За безопасность несут ответственность обе стороны: как заказчик, так и Oracle. Заказчик отвечает за безопасное использование учетных записей, безопасную генерацию, хранение ключей и паролей, настройку access list, настройку IAM-политик, настройку сетевых файрволов и т. д. Oracle со своей стороны отвечает за физическую безопасность ЦОД, за выстраивание безопасной архитектуры, за своевременное закрытие брешей в защите, установку исправлений и т. д.

В предыдущей версии облака (Gen 1) управление облаком осуществлялось из пользовательского окружения, пользовательский код и код управления были недостаточно хорошо изолированы друг от друга. Поэтому в поколении OCI Gen 2 была добавлена сеть управления, изолированная от всех остальных сетей. В Облаке Oracle Gen2 управление осуществляется отдельными компьютерами на не-Intel процессорной архитектуре, т. е. система сконфигурирована так, что управляющая сеть физически отделена от клиентской сети, по которой идут пользовательские данные, и Oracle не видит пользовательские данные, а заказчикам недоступны данные друг друга, передающиеся по сети. Таким образом, пользователи не имеют доступа к управляющим компьютерам и не могут взломать систему управления. Контролирующие компьютеры также контролируют периметр каждого пользователя, и таким образом обеспечивается изоляция как заказчиков между собой, так и изоляция провайдера от заказчиков и заказчиков от провайдера.

Компания Oracle придерживается следующих принципов по защите данных и защите персональных данных:

- Политика защиты данных **Oracle's Services Privacy Policy**: <http://www.oracle.com/us/legal/privacy/services-privacy-policy/index.html>
- Политика **Data Processing Agreement** для сервисов в облаке Oracle: <http://www.oracle.com/us/corporate/contracts/cloud-dpa-1014-2346862.pdf>
- Дополнительно персональные данные и другая критичная информация в публичном облаке Oracle могут быть защищены с шифрованием БД и/или опцией DB Vault
- Для сред тестирования и разработки Oracle предоставляет **Data Masking Pack**
- Для хранения персональных данных граждан РФ с 1 сентября 2015 г. может быть использован ЦОД в Амстердаме, где при выполнении требований сбора и обработки информации допустимо хранение персональных данных
- Сертификация **Safe Harbour** (является обязательной для юрисдикции Евросоюза, например EU Directive 95/46/EC on Personal Data Protection, и т. д.)

Infrastructure as a Service (IAAS) — OCI

Все OCI-регионы обеспечивают базовый набор инфраструктурных сервисов. Основные:

- **Compute**: серверы bare metal с процессорами Intel или AMD, Серверы с GPU на базе NVidia, виртуальные машины в формате KVM, Xen, VMWare, Container Engine for Kubernetes, Docker



- **Storage:** локальные NVMe SSD, блочные устройства на базе NVMe SSD или HDD, файловые системы общего назначения и файловые системы для архивного хранения данных
- **Networking:** Virtual Cloud Network, DNS, балансировщик нагрузки Load Balancing
- **Edge:** DNS

Глобально доступные облачные сервисы, не перечисленные выше, доступны по запросу. Низкие сетевые задержки делают возможным использовать облачные сервисы в разных регионах с отличным результатом, т. е. часть сервисов запускать в одном регионе, а часть — в другом.

Database Cloud Service (DBAAS)

Database Cloud Service — это платформа для создания инстансов БД в облаке. Создание инстансов БД возможно как на базе виртуальных машин, так и на bare metal с NVMe или SSD-дисками. Пользователь DBaaS имеет полный контроль над конфигурацией и действиями этой БД. Пользователь имеет доступ с учетной записью root на уровне ОС и имеет учетную запись SYS для доступа к базе данных. Таким образом, пользователь имеет полный контроль над ОС и базой данных. Доступ к этому сервису: <https://cloud.oracle.com/database>. В качестве базы данных пользователь может выбрать: Database service, Exadata service, Autonomous Data Warehouse, Autonomous Transaction Processing.

Преимущества DBaaS по сравнению с другими облачными провайдерами:

- Высокое качество сервиса. Высокая производительность, надежность, безопасность
- Быстрота. Новая БД создается за минуты
- Эластичность ресурсов. Возможность увеличивать и уменьшать «на лету» объем используемых ресурсов — это стандартная возможность облака Oracle, заложенная в него от рождения
- Уменьшает затраты пользователей облака на содержание собственных администраторов баз данных

Оплата производится по модели pay-as-you-go, стоимость сервиса зависит от редакции СУБД Oracle, которую использует заказчик, от количества заказанных ядер. Гибкие возможности оплаты позволяют заказчику выбирать между опциями license included или bring your own license.

Oracle Database Cloud Service поддерживает множество последних версий СУБД Oracle.

Редакции СУБД Oracle возможные к заказу как Oracle Database Cloud Service (в дополнение к перечисленным ниже опциям в Oracle бесплатно добавляет опцию Transparent Data Encryption (TDE)):

Редакция	Опции базы данных
Standard Edition	Все опции Standard Edition
Enterprise Edition	Опции Enterprise Edition плюс: Data Masking Pack, Subsetting Pack, Diagnostics Pack, Tuning Packs, Real Application Testing
Enterprise Edition High Performance	Опции Enterprise Edition плюс опции: Data Masking Pack, Subsetting Pack, Diagnostics Pack, Tuning Packs, Real Application Testing, Multitenant, Partitioning, Advanced Compression, Advanced Security, Label Security, Database Vault, OLAP, Advanced Analytics, Spatial & Graph, Database Lifecycle Management Pack, Cloud Management Pack for Oracle Database
Enterprise Edition Extreme Performance	Все опции Enterprise Edition High Performance плюс опции: RAC, In-Memory Database и Active Data Guard.



ОПЛАТА И ФИНАНСЫ

Сервисы в OCI бывают двух видов: с предоплатой и с постоплатой.

Сервисы с предоплатой (**Unmetered, Monthly Flex, Universal Credit**) — заказчик платит фиксированный платеж в месяц в течение заранее определенного периода. За это ему предоставляется заранее оговоренный объем ресурсов с заранее оговоренными параметрами производительности. Вносится депозит, заключается договор, в котором оговаривается объем предоставляемых ресурсов: минимум 1000 долларов в месяц на протяжении минимум 12 месяцев, оплата помесечная, вносится на счет Oracle до начала каждого месяца. Ресурсы предоставляются помесечно в объеме заключенного договора. В этом случае заказчик платит примерно на 30–60 % меньше, чем в случае PAYG. Возможны скидки в зависимости от объема заказываемых ресурсов.

Сервисы с постоплатой (**Pay-as-you-go, PAYG**) — оплата за объем ресурсов, реально потребленных заказчиком. В данном случае с обеих сторон нет ожиданий и обещаний на будущее, нет минимального периода обслуживания:

- Затраты измеряются и списываются на базе реального потребления услуги
- Фиксированный повторяющийся платеж может быть, а может отсутствовать
- Фиксированного срока пользования сервисом нет
- Объем использования ресурсов измеряется на почасовой основе

Bring Your Own License — дополнительная опция к предыдущим двум моделям оплаты.

Заказчику разрешается прийти в облако Oracle со своими лицензиями на ПО Oracle. Т. е. можно принести на оборудование OCI свои лицензии on-prem. В этом случае данные лицензии уже нельзя использовать на оборудовании заказчика в период, когда данные лицензии используются в облаке. Т. е. лицензии используются либо в ЦОД заказчика, либо в OCI. Позже, если заказчик решит уйти из облака, то эти лицензии можно будет вернуть обратно на свое оборудование. Опция BYOL позволяет снизить ежемесячный платеж.

Какие факторы влияют на цену в облаке?

- Объем потребляемых ресурсов: чем больше потребляете, тем больше платите
- Тип используемых ресурсов (виртуальные машины стоят дешевле, чем bare metal)
- Объем передаваемых из облака данных. Данные, передающиеся в облако, не тарифицируются
- Все OCI регионы имеют одинаковую цену, цена на предоставляемые услуги не зависит от региона



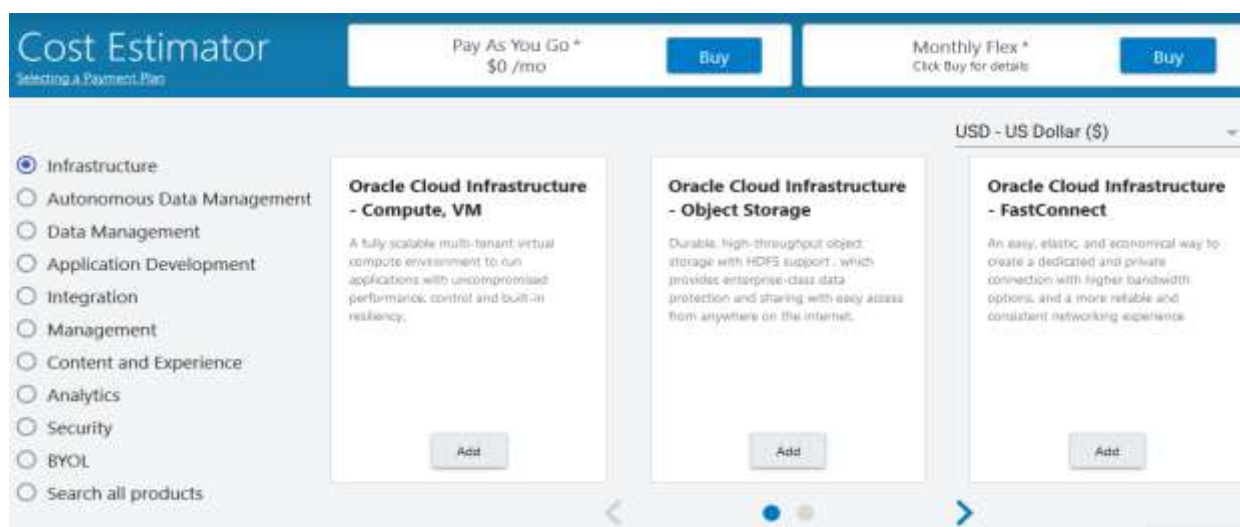


Рис. 6. Облачный калькулятор OCI (<https://www.oracle.com/cloud/ucpricing.html>)

С помощью данного калькулятора любой желающий может выбрать интересующие его ресурсы, определить способ оплаты, узнать текущие цены за различные ресурсы облака Oracle.

Oracle Always Free Cloud

Для разработчиков, студентов и всех желающих Oracle предлагает Always Free Cloud. Это бесплатный и не ограниченный по времени доступ к ограниченным ресурсам Oracle Cloud, в рамках которого предлагаются:

- Две Oracle Autonomous Database + Oracle Apex + SQL Developer, каждой базе данных выделяется по 1 виртуальному ЦПУ и 20 ГБ дискового пространства.
- Две виртуальных машины, которым выделяется 1/8 ЦПУ и 1 ГБ оперативной памяти
- Два блочных устройства до 100 ГБ в сумме (их можно использовать для размещения файлов БД либо как файловую систему для хранения любых файлов)
- 10 ГБ object storage (для хранения данных, к которым необходим быстрый и частый доступ)
- 10 ГБ archive storage (для хранения редко используемых данных, но которые могут понадобиться в течение длительного периода времени)
- Один балансировщик нагрузки Load Balancer
- Сеть с пропускной способностью 10 Мбит/с и объемом трафика до 10 ТБ в месяц
- 1000 e-mail в месяц

Oracle Always Free Cloud доступен во всех странах. Для подключения к Free Cloud потребуется зарегистрироваться на сайте [Cloud.oracle.com](https://cloud.oracle.com)

На первые 30 дней после регистрации новому пользователю дается возможность использовать дополнительные сервисы на сумму 300 долларов: заказать до 8 баз данных, заказать инфраструктуру bare metal или VM, заказать блочные устройства до 5 ТБ и т. д. Через 30 дней пользователю будет предложено расширить свое присутствие в Oracle Cloud и оформить платную подписку. Если пользователь не предпринимает действий по расширению своей подписки, то все избыточные ресурсы будут удалены.



и останутся только ресурсы Always Free Cloud. В последующем пользователь в любой момент может перевести свою бесплатную подписку в платную, если ему требуется заказать дополнительный объем ресурсов.

Подробнее: <https://www.oracle.com/cloud/free>

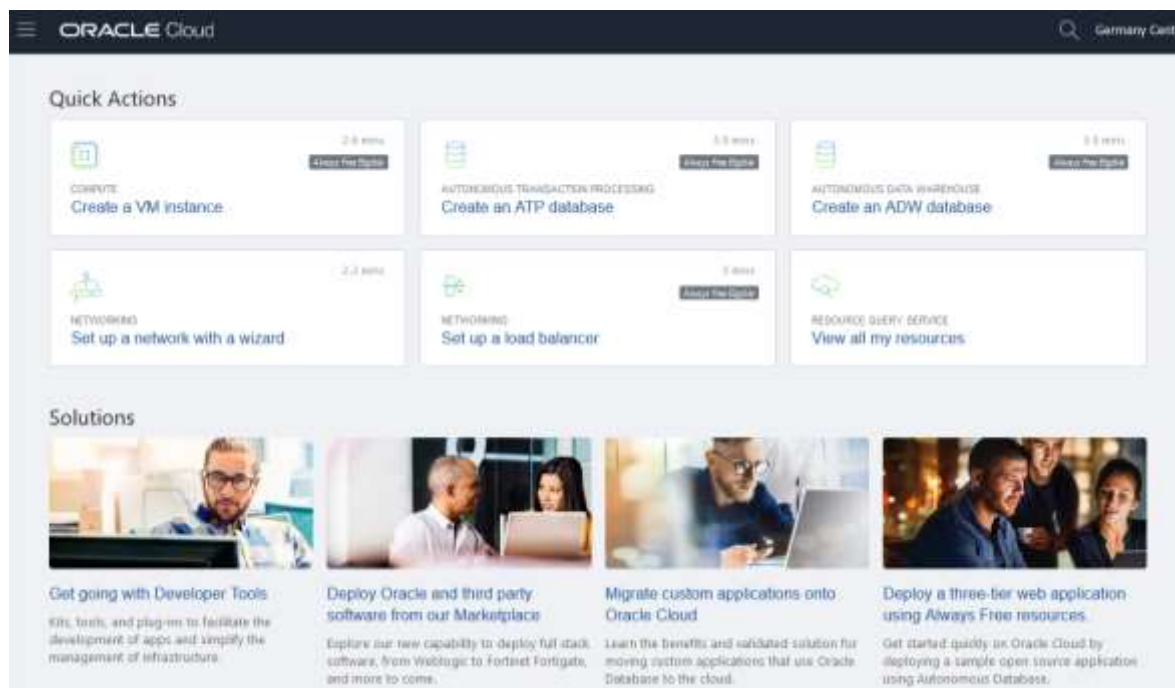


Рис. 7. Пример интерфейса Oracle Free Cloud

Как создать частное облако от Oracle

Облачные вычисления обеспечивают большую адаптивность, масштабируемость и гибкость для клиентов. Вместо траты средств и ресурсов на поддержание ранее разработанных ИТ-систем клиенты могут полностью сконцентрировать свое внимание на решении более стратегических задач. Компании могут быстро получить доступ к необходимым вычислительным ресурсам, не делая крупных первоначальных инвестиций, и оплачивать использование только требующихся ресурсов.

Когда компания решает «перейти в облако», это означает, что ИТ-инфраструктура в центре обработки данных управляется поставщиком облачных вычислений. Поставщик облачных решений несет ответственность за управление ИТ-инфраструктурой клиента, интеграцию приложений и разработку новых возможностей и функций, позволяющих соответствовать меняющимся потребностям рынка.

Существует несколько тенденций, мотивирующих предприятия в различных отраслях переходить в облако. Для большинства организаций существующая практика ведения бизнеса может быть недостаточно адаптивной для роста предприятия или не обеспечивать конкурентоспособную платформу или уровень гибкости. Взрывной рост данных приводит к новому повышению стоимости и сложности хранения данных в ЦОД — и требует от ИТ-специалистов новых навыков и средств аналитики. Вместо управления ИТ-ресурсами организации получают возможность быстро реагировать на стремительно меняющиеся, комплексные условия бизнеса.

Проще говоря, облачные вычисления — это аренда информационных технологий вместо их покупки. Вместо того чтобы вкладывать значительные средства в базы данных, программное обеспечение и оборудование, можно получать доступ к вычислительным ресурсам и платить за их использование. Можно иметь целый



комплекс программного обеспечения в виде сервисов «ПО как услуга» (SaaS), «платформа как услуга» (PaaS), «инфраструктура как услуга» (IaaS).

Существует три различных типа предоставления облачных сервисов: публичное, частное и гибридное облако. Все они различаются по объему управления, необходимому со стороны клиента, а также обеспечиваемому уровню безопасности.

Публичное облако	В публичном облаке вся вычислительная инфраструктура расположена на территории поставщика облачных услуг, который предоставляет эти услуги клиенту через Интернет. Клиенту не нужно поддерживать собственную ИТ-инфраструктуру и можно быстро добавлять пользователей или вычислительные мощности при необходимости. Поставщик облачных служб обслуживает множество клиентов, совместно использующих его ИТ-инфраструктуру.
Частное облако	Частное облако используется только одной организацией. Оно может размещаться локально или в центре обработки данных поставщика облачных служб. Частное облако обеспечивает наивысший уровень безопасности и контроля ресурсов.
Гибридное облако	Согласно названию, гибридное облако представляет собой комбинацию из публичных и частных облачных структур. Как правило, клиенты размещают критически важные для жизнедеятельности организации приложения на собственных серверах для обеспечения большей безопасности и контроля ресурсов, в то время как остальные приложения размещаются у поставщика облачных служб.

Также стоит отметить возможность использования многооблачной среды — многочисленных облачных вычислительных мощностей и ресурсов хранения данных (в том числе и от разных поставщиков облачных сервисов) в единой архитектуре.

Большинство заказчиков интересует вопрос, как создать облако, чтобы получать преимущества облачного подхода. Еще в 2011 году у компании Oracle появился набор продуктов, позволяющий реализовать полный жизненный цикл облачных вычислений — от создания облачной инфраструктуры частного облака, используя собственное оборудование в собственном центре обработки данных, до эксплуатации и мониторинга всех компонент облака. Эти продукты позволяют достаточно быстро и просто создать свое частное облако и начать перенос в него своих приложений. Для сотрудников частное облако выглядит как публичное и обеспечивает все преимущества публичного облака, но реально его развертывание и обслуживание происходит в ЦОД организации. Персонал организации, поддерживающей частное облако, проще призвать к ответу для обеспечения качества сервиса (SLA) или решения проблем обеспечения безопасности. Программные продукты Oracle для решения этих задач могут применяться как в традиционной архитектуре, так и в облаке. Практически перенос приложений и баз данных в частное облако не снижает их защищенности.

Oracle предлагает набор продуктов и технологий для поддержки всего жизненного цикла облака — от планирования и реализации облака до его эксплуатации и мониторинга. При этом используются стандартные элементы, такие как виртуализация и средства управления виртуализацией Oracle VM, полноценная СУБД Oracle Database для DBaaS, стандартные средства управления облаком и элементами инфраструктуры. Oracle также имеет самые мощные и развитые средства для учета и тарификации использования вычислительных ресурсов в облаке. Можно строить очень сложные планы биллинга, учитывающие десятки характеристик использования оборудования и ПО.

Управление всем технологическим стеком облака от железа до приложений осуществляется через графический интерфейс Oracle Enterprise Manager, хорошо знакомого всем пользователям Oracle.



Он позволяет контролировать все этапы жизненного цикла облака от быстрого создания частного облака до переноса в облако приложений, а также перемещение приложения из частного облака в публичное и обратно.

Как создать частное облако?

По методологии полный жизненный цикл облака состоит из следующих этапов:

- Планирование структуры облака и вариантов консолидации приложений в облако
- Создание облачной инфраструктуры
- Подготовка сервисов, образцов машин, БД, серверов приложений, шаблонов, сборок для пользователей облака, выдача привилегий пользователям
- Тестирование созданных сервисов и публикация их для пользователей
- Заказ и использование сервисов конечными пользователями с помощью портала самообслуживания
- Мониторинг использования облачных сервисов
- Управление облачной инфраструктурой
- Тарификация и биллинг используемых ресурсов
- Оптимизация использования ресурсов

Но прежде, чем подробно рассмотреть все эти этапы и ПО для их реализации, посмотрим, как теперь изменится жизнь конечного пользователя. Допустим, для целей изучения или тестирования нового приложения или БД пользователю понадобилось установить и настроить какое-то ПО. При традиционном подходе он должен:

1. Определить характеристики компьютеров, необходимых для установки этого ПО, найти бюджет и купить компьютер (или несколько компьютеров), систему хранения и т. д.
2. Подключить и сконфигурировать все это оборудование, протестировать его.
3. Установить, протестировать операционную систему, установить исправления.
4. Скачать, установить, настроить требуемое ПО и СУБД, установить исправления.
5. При недостатке мощности оборудования докупить новое оборудование и повторить процесс.

Это занимает много времени. При новом облачном подходе пользователь может зайти на портал самообслуживания и создать заявку на требуемый компьютер, или БД, или сервер приложений, или их связку, и через несколько минут получит то, что ему необходимо для работы. Он может также просто увеличить мощность своего компьютера или размер и мощность своей БД. При этом он ничего не знает об инфраструктуре облака, конфигурировании ОС и ПО и т. д.

Интерфейс портала самообслуживания достаточно прост. На нем можно создать заявку на выделение машины с ПО (IaaS), базы данных (DBaaS) или сервера приложений WebLogic. На портале можно видеть, какие машины, базы данных, сервера приложений созданы для этого пользователя, и работать с ними. Можно удалять их после использования (если они созданы на неограниченный срок). Также для каждого пользователя установлены ограничения на использование ресурсов облака (диски, память, процессоры, количество баз данных и виртуальных машин и т. д.) и на портале он может контролировать использование ресурсов. Можно, используя облачный API (emcli или REST API), создать собственный портал



самообслуживания или интегрировать его с существующим. Можно также встроить вызовы через API в свои приложения.

При создании заявки требуется ответить всего на несколько вопросов (например, количество процессоров и памяти для запрашиваемой машины, имя и пароль пользователя для создаваемой БД и т. д.) и выбрать из предлагаемого списка шаблон машины, БД, сервера приложений, кластера, сборки, которую планируется создать.

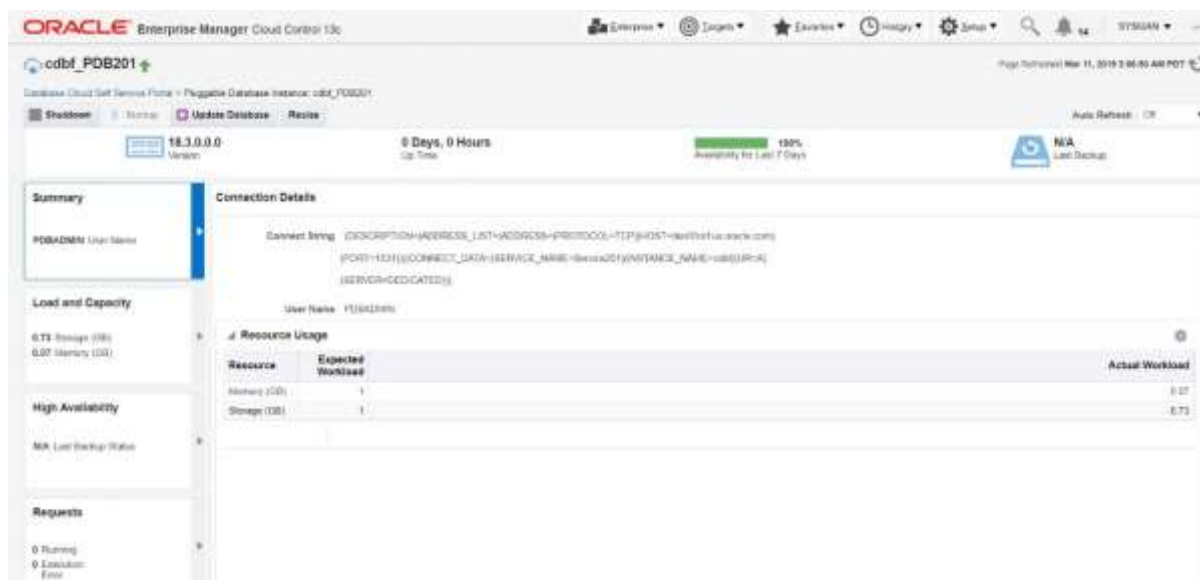


Рис. 1. Портал самообслуживания

Шаг 1. Планирование и создание облачной инфраструктуры, создание пользователей

При организации облака у Вас появляются две новые роли: администратор облака и администратор самообслуживания. Администратор облака создает, контролирует и отвечает за инфраструктуру облака. Он устанавливает ПО гипервизора или другое инфраструктурное ПО (например, Oracle Database Home), конфигурирует систему хранения и сеть, объединяет компьютеры в пулы серверов, а затем пулы в зоны, создает пользователей облака, присваивает им роли и привилегии. Для IaaS, DBaaS, PaaS создаются разные инфраструктуры. Но в каждой из них группа физических серверов объединяется в серверный пул. А пулы объединяются в зоны. Сервера серверного пула имеют доступ к единой системе хранения. Это позволяет, например, виртуальным машинам IaaS без остановки продолжить работу на другом сервере пула при сбое или перегрузке текущего сервера. В случае DBaaS на серверах одного пула могут быть развернуты узлы кластера. На всех серверах пула должно быть установлено одинаковое ПО. Для IaaS — это гипервизор Oracle VM, для DBaaS — это ПО Oracle Database (Oracle Home). Зона объединяет несколько серверных пулов, например, по географическому или логическому принципу.

Шаг 2. Создание и каталогизация в библиотеку ПО шаблонов, сборок и процедур развертывания

Еще одна задача администратора облака — создать библиотеку ПО или репозиторий шаблонов виртуальных машин (для IaaS), процедуры развертывания БД, сборки, которые сможет выбирать конечный пользователь.



Администратор должен определить предлагаемые варианты виртуальных машин (для IaaS), баз данных (для DBaaS), серверов приложений, описать их характеристики, назначить пользователям и ролям квоты на использование дисков, памяти, процессоров, количество создаваемых виртуальных машин, баз, серверов приложений, создать тарифные планы и привязать роли и планы к зонам облака. Важно описать эти объекты библиотеки ПО на максимально понятном языке, чтобы конечный пользователь мог легко выбрать из списка именно то, что ему нужно. Oracle Enterprise Manager предоставляет удобный инструмент, который позволяет администратору быстро подготавливать эти объекты библиотеки ПО.

Если заказчик хочет использовать сервис DBaaS (Database as Service), то он может выбрать различные варианты реализации:

Первый вариант подразумевает развертывание баз данных в отдельных виртуальных машинах. Перенос БД с существующей инфраструктуры в облако при этом достаточно прост, т. к. в каждой виртуальной машине остается своя операционная система, своя БД, своя версия СУБД. Однако все эти виртуальные машины могут размещаться на одном серверном пуле, что повышает эффективность использования оборудования. Это чистая виртуализация, она выполняется просто и быстро, но при этом работа СУБД внутри разных виртуальных машин приводит к дополнительным накладным расходам и менее эффективно использует оборудование, чем вариант чистого DBaaS. Кроме того, приходится поддерживать много разных ОС, версий СУБД и т. д. Этот вариант хорош как первый шаг консолидации.

Oracle не сводит облачные вычисления только к предоставлению виртуальных машин. Другой вариант (чистая DBaaS) — когда экземпляры СУБД Oracle (или кластерные СУБД Oracle) работают на пуле физических компьютеров с единой операционной системой. Это позволяет унифицировать ОС и версию СУБД, упростить управление, повысить эффективность использования оборудования. Oracle DBaaS можно развернуть на любой платформе, где работает Oracle Database. Это не только платформы X86, но и Sparc Solaris, HP-UX, AIX, Exadata.

Более высокий уровень консолидации с целью эффективного использования оборудования — предоставление схем баз данных и логики разных приложений в единой кластерной БД Oracle. Этот вариант трудоемок и требует выполнения дополнительных работ по интеграции разных приложений в одну БД облака, обеспечение дополнительных средств разграничения доступа и безопасности хранения данных.

СУБД Oracle, начиная с версии 12c с опцией Multitenant, предоставляет возможность создания облачного сервиса DBaaS на основе контейнерной СУБД. Все базы данных в контейнерной архитектуре используют аппаратные ресурсы совместно, но при этом каждая база данных изолирована с точки зрения как безопасности, так и потребления ресурсов. В контейнерной архитектуре потреблением ресурсов каждой базы данных управляет Resource Manager, что позволяет гарантировать требуемую производительность работы и отсутствие влияния баз данных друг на друга.

Для построения шаблонов виртуальных машин OEM использует Oracle VM Template Builder, который создает шаблон на основе существующей физической или виртуальной машины. Для создания многокомпьютерных приложений в облаке создаются сборки (Assembly), описывающие все виртуальные машины такого приложения и правила их взаимодействия (имена, конфигурации сети, конфигурации дисков и т. д.). Это делается с помощью программы Oracle Virtual Assembly Builder. Она позволяет описать все компоненты такого приложения и связи между ними, после чего генерирует набор шаблонов и описаний, объединенный в сборку.

После наполнения библиотеки ПО и описания ролей и квот пользователей облако готово к работе и конечный пользователь, зайдя на портал самообслуживания, может начать создавать для себя базы данных, серверы приложений, виртуальные машины, сложные сборки, компоненты приложения.



Шаг 3. Мониторинг облака и управление им, тарификация и биллинг

Одной из важных задач работы с облаком является мониторинг его компонентов и управление ими. В облачной архитектуре мы имеем множество баз данных и серверов приложений, множество виртуальных машин, множество пользователей облака, множество запросов на выделение ресурсов (успешно и неуспешно выполнившихся), которыми нужно управлять.

Поскольку созданием БД, виртуальных машин, серверов приложений занимаются пользователи, мы получаем непрерывное и слабо контролируемое разрастание числа этих объектов облака, а то, что они автоматически (в соответствии с политиками) могут мигрировать на другие машины, пользователи могут останавливать и возобновлять их работу, еще больше усложняет управление и контроль. Необходимо также контролировать использование дискового пространства, оперативной памяти, процессоров в облаке и заранее предвидеть исчерпание этих ресурсов. Кроме того, не стоит забывать, что периодически необходимо обновлять все это огромное количество баз данных, виртуальных машин, серверов приложений, устанавливать необходимые исправления.

Из всего этого видно, что для мониторинга облака и его элементов и для управления ими нужны мощные инструменты. Причем управление и мониторинг должны обеспечиваться на групповом уровне (большим количеством объектов). Но должна быть и возможность перехода с группового уровня к мониторингу отдельного объекта группы и управлению им. Поскольку инфраструктура облака включает не только системное ПО, но и серверы, сеть, системы хранения, приложения и т. д., средства управления должны позволять работать со всем стеком: от дисков до приложения.

Oracle в качестве такого универсального средства мониторинга и управления от диска до приложения и облака предлагает продукт Oracle Enterprise Manager.

Часть работы по мониторингу созданных объектов и управлению ими отдается пользователю, создавшему объект. На портале самообслуживания пользователь, создавший БД или виртуальную машину, может перейти на экран мониторинга объекта. Здесь он видит не только информацию об использовании памяти, дисков, процессоров, но и специфическую информацию, например, для БД: наиболее ресурсоемкие операторы SQL, ожидания и т. д. На этом экране пользователь может легко включить режим автоматического полного и инкрементального резервного копирования БД. На главной странице портала самообслуживания пользователь видит, сколько баз данных, виртуальных машин или серверов приложений создано и сколько он еще может создать, какие ему выделены квоты на использование памяти, дисков или процессоров и как они используются. Владелец объекта может также остановить или запустить объект (например, виртуальную машину) или вообще удалить объект из облака. Он также может задать политику автоматического сопровождения объекта. Например, можно определить, что виртуальная машина будет автоматически останавливаться в 8 часов вечера и стартовать в 10 часов утра и т. п.

Всю остальную работу по администрированию облака выполняют администраторы облака, баз данных, серверов приложений. В OEM есть средства мониторинга зон облака и их компонентов (пулов, виртуальных машин, серверов) и управления ими. Администратор может просматривать и редактировать их характеристики, увеличивать и уменьшать ресурсы для этих объектов, удалять их и создавать. Здесь же есть средства отслеживания запросов на выделение ресурсов в облаке. Администратор видит, кто какие запросы выполнял, как они выполнялись, и может анализировать причины недостатка ресурсов, нарушение квот и политик, проактивно выявлять потенциально узкие места облака. При возникновении проблем с производительностью или работоспособностью объектов администратор может перейти непосредственно к администрированию конкретного объекта облака через OEM.

Одной из важных характеристик облачных вычислений является возможность оплаты только по факту использования ресурсов. Для этого составляются тарифные планы, учитывающие использование этих ресурсов и их цену. На сегодня Oracle имеет одну из самых мощных систем тарификации и биллинга.



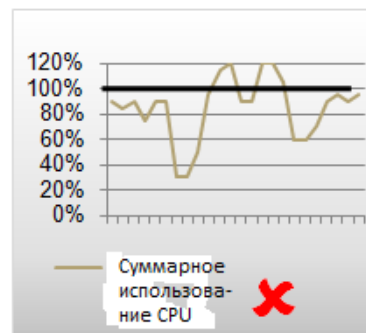
Вы можете учитывать около полусотни различных параметров, чтобы сформировать гибкий план оплаты используемых ресурсов: от диска и виртуальной машины до приложения. Базовый тарифный план учитывает использование четырех ресурсов: процессора, памяти, системы хранения и сети. Вы можете установить стоимость в единицу времени для каждого из этих ресурсов. Расширенный тарифный план может учитывать множество других метрик, например: тип операционной системы, использование опций или версий базы данных, использование резервного копирования, обеспечение высокой надежности, количество транзакций в БД и т. п. Созданные тарифные планы приписываются объектам или группам объектов (например, зоне). При использовании облачных сервисов для каждого экземпляра сервиса будут на основе тарифного плана формироваться отчеты. Их можно использовать как для реальной оплаты (печатать или загружать в биллинговые системы), так и для контроля использования ресурсов и для взаиморасчетов (например, между отделами или центрами затрат).

Oracle Consolidation Planner

Прежде чем начать консолидацию существующих приложений и баз данных на серверные пулы облака, надо понять, как их консолидировать на разделяемые ресурсы так, чтобы, с одной стороны, использовать компьютеры облака наиболее эффективно, а с другой стороны, не получить снижения качества работы этих приложений и БД. Например, если перенести на один физический компьютер два приложения, у которых пик загрузки процессора (или сети, памяти и т. д.) приходится на одно и то же время, нам потребуется очень мощный компьютер, который большую часть времени будет недозагружен. Поэтому, прежде чем принимать решение о том, на какие физические компьютеры консолидировать наши текущие приложения, надо проанализировать, как распределяется нагрузка, создаваемая этими приложениями, во времени, и консолидировать их с учетом этих данных и характеристик наших компьютеров. Для решения этой задачи используется компонент Oracle Consolidation Planner.

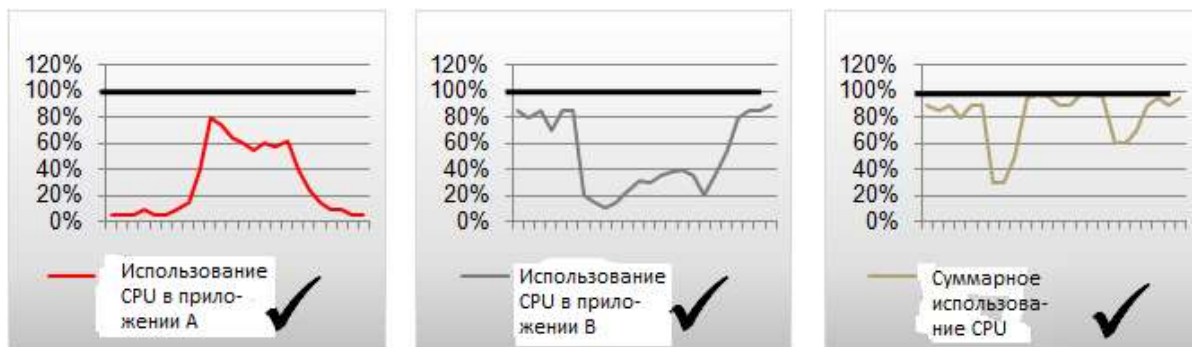
Средство Oracle для мониторинга баз данных и приложений Oracle Enterprise manager (OEM) Cloud Control^{13c} умеет сканировать сеть и находить все компьютеры, на которых работает ПО Oracle (СУБД, серверы приложений и т. д.). Получив список этих компьютеров, мы можем выбрать те из них, которые планируем консолидировать в будущем, и дать OEM команду установить на них управляющий агент OEM. Эти агенты будут собирать информацию о том, как эти компьютеры используют процессоры, память, диски и сеть. На основе этой информации Consolidation Planner выдаст рекомендации о том, какие приложения, базы, компьютеры имеет смысл объединять, при этом будет использоваться не только профиль нагрузки, но и заданные ограничения. Например, не стоит объединять на одном компьютере узлы одного кластера БД или тестовые и эксплуатационные базы. Бессмысленно объединять компьютеры с разными операционными системами.

Предположим, что мы анализируем коэффициент использования процессора приложениями А и В и видим, что их сумма в какой-то момент времени превышает 100 %. Это значит, что консолидировать А и В на одном компьютере нельзя. Если же профили нагрузки компенсируют друг друга, то эти приложения — хорошие кандидаты для консолидации на один компьютер.



Плохие кандидаты для консолидации на 1 компьютер





Хорошие кандидаты для консолидации на 1 компьютер

Мы можем указать Consolidation Planner, какие из собранных метрик (использование процессора, памяти, диска, ввод/вывод дисков, загрузка сети) учитывать при принятии решения о консолидации. Мы можем установить бизнес-ограничения для консолидации (из одного отдела, единство географического положения, назначение — эксплуатационная/тестовая и т. д.) и технические ограничения (операционная система, тип компьютера, узлы кластера). Мы также можем ограничить загрузку потенциальных серверов консолидации.

Существует три сценария консолидации:

- P2P — физические серверы на другие физические серверы.
- P2V — физические серверы превращаются в виртуальные.
- P2E — физические серверы БД консолидируются на машины баз данных Exadata.

Мы также должны указать, на какие серверы (существующие или новые) планируем консолидировать приложения и БД. На основе всей этой информации Consolidation Planner предложит нам варианты консолидации и укажет степень консолидации и планируемую загрузку узлов после консолидации.

Инженерные системы

Готовые программно-аппаратные комплексы Oracle выводят оптимизацию инфраструктуры на следующий уровень. Они разработаны, интегрированы, протестированы и оптимизированы для слаженной совместной работы всех программных и аппаратных компонент. Более того, они спроектированы вместе с программным обеспечением Oracle, благодаря чему обеспечивается высокий уровень облачной интеграции и производительности Oracle Database, который недоступен обычным системам.

При разработке таких систем (программно-аппаратных комплексов) Oracle привносит инновации на каждом уровне стека решений. Эти инновации существенно уменьшают затраты и сложность ИТ-инфраструктуры, одновременно повышая производительность и качество работы

Exadata Database Machine

Exadata Database Machine — это самая производительная и гибкая архитектура для Oracle Database. Она обеспечивает исключительную производительность, меньшее время окупаемости, быстрое развертывание и готовность системы к облачной среде.

Oracle Exadata предлагает лучшую платформу для базы данных Oracle. Это решение помогает снизить расходы до 40 % и ускорить цифровую трансформацию вашей организации, благодаря разным моделям потребления, будь то локальное развертывание, модель Cloud at Customer (облако Oracle в ЦОД заказчика)



или публичное облако Oracle Cloud, обеспечивая высокую степень идентичности приложений и повышая скорость работы в 3 раза по сравнению с любым другим решением.

Простая в развертывании машина баз данных Exadata Database Machine включает в себя все необходимое оборудование для работы с СУБД Oracle. Серверы баз данных, серверы хранения данных и сетевое оборудование предварительно сконфигурированы, настроены и протестированы экспертами Oracle, так что вам не нужно будет тратить недели рабочего времени, которое обычно необходимо для развертывания высокопроизводительных систем. Всестороннее комплексное тестирование гарантирует эффективную совместимость всех компонентов и отсутствие узких мест производительности или единых точек отказа, способных повлиять на работу системы в целом.

КЛЮЧЕВЫЕ ВОЗМОЖНОСТИ

- Настроенная и протестированная система, оптимизированная для любых приложений базы данных.
- Возможность переноса части массированных вычислений на сервер СХД.
- Скорость обработки запросов несжатых данных до 560 ГБ/сек на стойку.
- Время чтения блоков базы данных размером 8 КБ с внешней общей СХД сокращено до 19 мкс.
- Производительность системы ввода-вывода 16 млн операций блоками базы данных размером 8 КБ на полную стойку.
- Возможность гибкого масштабирования — добавления вычислительных серверов и серверов хранения по мере необходимости.
- Масштабирование путем подключения нескольких стоек машины баз данных Exadata.
- Единое управление и автоматизация процедур диагностики и обновления программного обеспечения.

Zero Data Loss Recovery Appliance X8 и X8M

Традиционные решения для хранения не всегда способны удовлетворять потребности в защите критически важных корпоративных баз данных. Использование этих систем может привести к неприемлемой потере данных и создать значительную нагрузку на продуктивные серверы во время резервного копирования. Данный подход не обеспечивает возможности гибкого восстановления и не поддерживает масштабирования при расширении баз данных. Эти проблемы возникают вследствие традиционного восприятия баз данных как набора разрозненных файлов для копирования, а не связанных данных с определенными требованиями к целостности.

Комплекс Oracle Zero Data Loss Recovery Appliance (Recovery Appliance) — это инновационное решение для защиты данных, которое интегрируется с СУБД Oracle Database и предназначено для устранения вышеперечисленных проблем. Предлагаемое решение устраняет вероятность потери данных и кардинально сокращает влияние средств защиты данных на продуктивные серверы. Помимо этого, Recovery Appliance обеспечивает проверку целостности и возможности восстановления данных, масштабирование для защиты тысяч баз данных, защиту резервных копий на протяжении всего жизненного цикла, включая резервное копирование дисков и лент, а также удаленных репликаций.

КЛЮЧЕВЫЕ ВОЗМОЖНОСТИ

- Копирование в реальном времени векторов изменений.
- Эффективная репликация между дистанционно удаленными друг от друга комплексами Recovery Appliance.
- Автономная архивация на ленту.
- Сквозная проверка данных.



- Стратегия постоянного инкрементального резервного копирования.
- Эффективная экономия места за счет возможности создания виртуальных полных резервных копий.
- Резервное копирование на ленты без участия защищаемых серверов СУБД.
- Политика защиты на уровне базы данных.
- Управление пространством хранения с учетом параметров БД.
- Архитектура с облачным масштабированием.
- Унифицированное управление и контроль.

Oracle Database Appliance

Oracle Database Appliance (ODA) — это программно-аппаратный комплекс, позволяющий сократить время и стоимость за счет упрощенного развертывания, управления и поддержки решений на основе баз данных, в организациях любого размера. Это решение оптимизировано для использования СУБД Oracle Database и объединяет в себе аппаратные и программные ресурсы для сервисов баз данных, в том числе для обработки транзакций (OLTP), In-Memory Database и аналитических приложений. Все программные и аппаратные компоненты разработаны и поддерживаются Oracle, что дает пользователям надежное и безопасное решение со встроенными средствами оптимизации и с учетом передового опыта. Oracle Database Appliance не только позволяет снизить время введения в эксплуатацию, но и предлагает гибкие возможности лицензирования Oracle Database, что помогает снизить расходы на обслуживание и поддержку.

КЛЮЧЕВЫЕ ВОЗМОЖНОСТИ

- Полностью интегрированная комплексная система для базы данных.
- Упрощенное управление через Oracle Appliance Manager.
- Пользовательский интерфейс на основе веб-браузера.
- Интеграция с Oracle Cloud.
- Можно использовать Oracle Database Enterprise Edition и Standard Edition.
- Базы данных Oracle с одним экземпляром или с Real Application Clusters.
- Oracle Automatic Storage Management.
- Oracle ASM Cluster File System.
- Oracle Linux.
- Масштабируемый процессор Intel® Xeon®.
- Сетевые порты 10GBase-T или 10/25 GbE SFP28.
- Твердотельные накопители NVM Express (NVMe).

Big Data Appliance

Единая платформа больших данных для сбора, систематизации и анализа больших объемов данных из разных источников в больших масштабах и на больших скоростях. Эффективный программно-аппаратный комплекс Oracle, предварительно сконфигурированный и готовый к работе с большими данными.

- Поддержка решений с открытым исходным кодом: система разработана специально для использования решений с открытым исходным кодом, таких как Hadoop, Spark и NoSQL, а также запуска рабочих нагрузок больших данных в Qlik, Tableau, SAS и других аналитических средах.



- Возможность использования популярных инструментов: Oracle Big Data SQL для SQL-запросов и сторонних сред от WanDisco, Cloudera и других поставщиков.
- Исключительная производительность: система интегрирована, оптимизирована и настроена таким образом, чтобы сократить время развертывания и ускорить реализацию бизнес-целей.
- Встроенные средства безопасности обеспечивают автоматическое выявление несоответствий, отслеживание подозрительной активности и защиту данных, а также упрощают аудит.
- Удобство работы с одним поставщиком: поддержка всей системы от Oracle сводит к минимуму затраты на развертывание, обслуживание и обновление вашей среды больших данных.
- Экономическая эффективность: высокая производительность и снижение затрат на обработку больших данных на Hadoop и Spark способствуют ускорению реализации бизнес-целей.

Private Cloud Appliance

Oracle Private Cloud Appliance — это гибкая интеллектуальная инфраструктура, предназначенная для быстрого автоматического развертывания частного облака и облачных приложений. Для систем на ОС Linux, Microsoft Windows или Oracle Solaris система Private Cloud Appliance поддерживает консолидацию широкого спектра рабочих нагрузок смешанного типа в центрах обработки данных среднего и крупного размера. Это система для конвергентной инфраструктуры, предоставляющая частное облако по лучшей в отрасли цене.

Высокопроизводительная, конфигурируемая программно сеть с малым временем отклика обеспечивает автоматическое управление IP-адресов вычислительных узлов и сетей хранения данных. Встроенное ПО контроллеров обеспечивает автоматическую установку, настройку всех компонент инфраструктуры и управление ими простым нажатием кнопки.

Заказчикам достаточно ввести основные параметры конфигурации и создать виртуальные машины, в том числе по предлагаемым шаблонам. Это облегчает настройку и запуск приложения в работу. С помощью Oracle Enterprise Manager комплекс Oracle Private Cloud Appliance превращается в мощную инфраструктуру частного облака, управлять которой можно с единой панели, как и публичным облаком Oracle Cloud Infrastructure.

КЛЮЧЕВЫЕ ВОЗМОЖНОСТИ

- Встроенный программный контроллер упрощает установку, настройку и управление.
- Поддержка Oracle Linux Cloud Native Environment (OL CNE) для автоматизации развертывания, масштабирования и управления приложениями в контейнерах.
- Поддержка рабочих нагрузок Oracle и сторонних производителей в системах Linux, Solaris и Windows.
- Интегрированное хранилище для приложений Oracle ZS7-2, включая 100 ТБ пространства для использования.
- Наличие возможностей виртуализации без дополнительной платы.
- Встроенное аварийное восстановление на уровне виртуальных машин с помощью Oracle Siteguard.

Продукты Oracle для управления качеством приложений

Бизнес вкладывает огромные средства в новые приложения для обеспечения лучшего и менее затратного сервиса для своих клиентов. Однако низкое качество разработки программного обеспечения создает серьезный риск для этих инвестиций. Исследования показали, что более 40 % программных систем содержат критические дефекты к моменту вывода в эксплуатацию. Стоимость выявления и исправления этих дефектов



на этапе эксплуатации обходится в 100 раз дороже, чем на стадии разработки. Например, опрос степени удовлетворенности клиентов приложениями Siebel показал четкую связь между объемом работ по тестированию и удовлетворенностью качеством. Для любых аспектов продукта, таких как общая эффективность приложения, надежность, производительность, масштабируемость, чем больше времени было потрачено на тестирование, тем лучше результаты. Вот почему управление качеством приложений так важно для поддержки современного, высококонкурентного бизнеса и обеспечения качества при снижении затрат.

Решения Oracle Enterprise Manager — Application Quality Management (AQM) предоставляют возможность высококачественного тестирования на всех уровнях стека приложения, что помогает убедиться в качестве и производительности до выхода в продуктивную эксплуатацию.

Тестирование является одним из наиболее сложных и долгих этапов цикла разработки приложения, оно же является критичным для успешной реализации всего проекта.

Oracle Enterprise Manager—AQM представляет собой уникальный набор продуктов и технологий для тестирования, которые позволяют решать следующие задачи.

- Тестирование инфраструктуры базы данных. **Real Application Testing** разработана и оптимизирована для тестирования изменений на уровне технологического стека базы данных и основан на технологии записи и воспроизведения реальной нагрузки БД, что позволяет выявить возможные ошибки и проблемы с производительностью на ранней стадии.
- Тестирование бизнес-приложений. **Application Testing Suite** поможет обеспечить качественное тестирование приложения с помощью полного набора решений: автоматизированное функциональное и нагрузочное тестирование, управление и документирование собственно процесса тестирования. Application Testing Suite — это интегрированное решение для синтетического тестирования, может быть с успехом использовано для проверки вновь разрабатываемых приложений и доработки уже существующих — внесения изменений в логику. Application Testing Suite может быть использована в комбинации с Real Application Testing.
- Управление вашими данными в тестовых средах. **Data Masking and Subsetting Pack** обеспечивает возможность сокрытия, «маскирования» конфиденциальных данных при переносе в тестовую среду, а также позволяет создавать согласованные, уменьшенные копии Production баз данных для целей функционального тестирования, разработки, обучения. Создание тестовых копий БД производится с сохранением всех правил ссылочной целостности.

Собранные вместе эти средства предоставляют исчерпывающий набор решений для обеспечения качества и надежности бизнес-приложений.

Хранилища данных и аналитические системы на основе СУБД Oracle

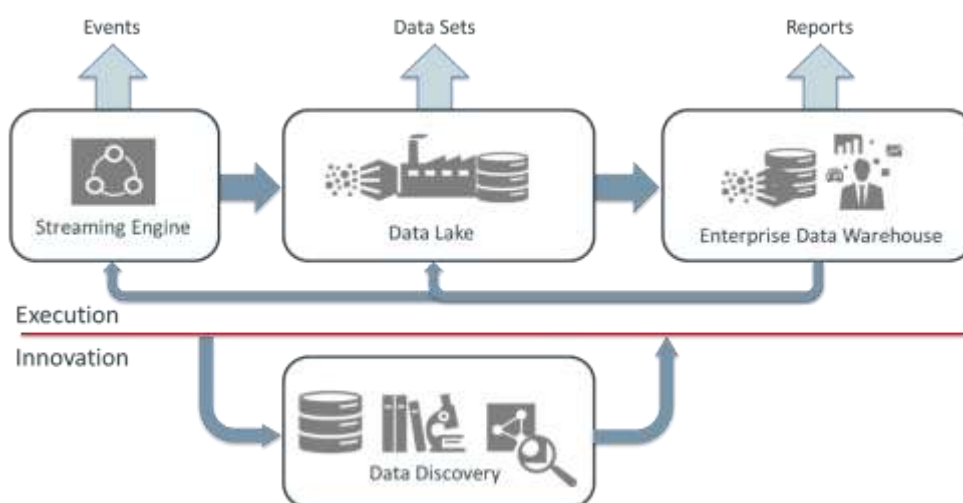
К настоящему времени во многих организациях накоплены колоссальные объемы данных, на основе которых можно решать самые разнообразные аналитические и управленческие задачи в любой сфере деятельности. Проблемы хранения, обработки и использования аналитической информации становятся все более актуальными и привлекают внимание специалистов и фирм, работающих в области информационных технологий. Современные стандарты создания корпоративных информационно-аналитических систем построены на концепции хранилищ и озер данных, а также на современных технологиях бизнес-анализа. В основе концепции хранилищ данных лежит важная идея интеграции ранее разобщенных детализированных данных, содержащихся в исторических архивах, накапливаемых в традиционных системах транзакционной обработки, поступающих из внешних источников, в единой базе данных, их предварительное согласование и, возможно, агрегация. Озера данных в рамках этой концепции могут являться как одним из подобных внешних



источников, содержащих сырые данные в неструктурированном или полуструктурированном виде, так и неотъемлемой частью кластера обработки данных, в случае если используются технологии и инструменты, позволяющие интегрировать их в общий процесс аналитической обработки данных.

В общем виде архитектура современной корпоративной информационно-аналитической системы включает в себя четыре уровня (см. рис. ниже):

- Уровень потоковых данных и их обработка в режиме реального времени.
- Уровень Data Lake и долговременного хранения данных.
- Уровень корпоративного хранилища и отчетности.
- Уровень исследования данных.



Технология функционирования системы состоит в следующем. Данные поступают из различных внутренних и внешних систем: от подчиненных структур, внешних организаций в соответствии с установленным регламентом, формами и макетами отчетности. Основная проблема при этом состоит в несогласованности и противоречивости этих информационных источников, отсутствии единого логического взгляда на корпоративные данные, многочисленные неточности и ошибки в исходных источниках. Для решения этой проблемы используются технологические инструменты ETL (Extract Transform Load), которые осуществляют проверку, согласование, преобразования и загрузку информации в хранилище и витрины данных. После этого пользователи с помощью специализированных инструментальных средств получают необходимую им информацию для построения различных табличных и графических представлений, прогнозирования, моделирования и выполнения других аналитических задач.

Направление хранилищ данных и систем бизнес-анализа является сегодня для Oracle одним из самых приоритетных. Ориентируясь на общепринятые стандарты в этой области, Oracle предлагает новые решения и технологии для создания хранилищ данных и эффективного использования накопленной в них информации.

Важной отличительной особенностью подхода Oracle является полнота и интегрированность аналитической платформы, позволяющая решать многие задачи создания и эксплуатации хранилищ данных в рамках единой технологии и продукта. Основой решения является система управления базами данных Oracle Database, которая обеспечивает надежное хранение информации как в реляционном, так и в полуструктурированном формате. Кроме того, Oracle Database включает в себя средства аналитического и интеллектуального анализа данных, встроенный OLAP-сервер, позволяющий в разы увеличить скорость выполнения аналитических запросов, средства диагностики и мониторинга базы хранилища данных.

Отдельно необходимо отметить, что опции Advanced Analytics и Spatial and Graph, начиная с релиза Oracle Database 20c, включены в основной функционал СУБД и не требуют отдельного лицензирования. Кроме того,



к свободно распространяемым программным модулям относятся также библиотеки машинного обучения OML4SQL, OML4R и т. д., которые позволяют обучать модели непосредственно в СУБД, устраняя необходимость перемещения данных. Эти библиотеки и программно-функциональные модули являются основой для организации процессов исследования данных и поиска аномалий, закономерностей, трендов, в том числе и в режиме self-service, когда специалисты Data Science получают возможность работы с актуальными данными или максимально к ним приближенными.

В качестве средств загрузки данных используются такие программные продукты, как Oracle Data Integrator, Oracle Golden Gate и Oracle Enterprise Data Quality, отвечающие соответственно за сложную трансформацию и пакетную загрузку данных, за загрузку данных в реальном режиме времени, за обеспечения качества и профилирования данных.

Для бесшовной интеграции Oracle Database и систем Big Data используется продукт Oracle Big Data SQL, позволяющий обращаться к данным, расположенным на кластерах Hadoop, средствами SQL-запросов реляционной СУБД Oracle. Данный продукт позволяет не только унифицировать механизм обращения к данным, но и построить единую модель защиты и контроля доступа к ним, независимо от их фактического расположения.

В качестве средств анализа данных используются продукты Business Intelligence, включающие в себя средства интерактивного анализа данных, построения регламентированной отчетности, мобильной аналитики, OLAP-анализа и многое другое.

Oracle Database для реализации хранилища данных

Основой хранения аналитической информации является хранилище данных, которое представляет собой базу данных, содержащую достоверную согласованную информацию, предназначенную для решения разнообразных аналитических задач. С точки зрения СУБД, под управлением которой работает хранилище данных, наиболее существенным является тот факт, что режимы функционирования базы данных для аналитических задач коренным образом отличаются от ситуации в обычных системах транзакционной обработки. Они требуют специальных настроек параметров, методов индексирования и обработки запросов.

СУБД Oracle предоставляет широкий спектр средств, направленных на работу базы в режиме хранилищ и витрин данных. К их числу относятся: параллельная обработка запросов, позволяющая наиболее полно использовать возможности многопроцессорных аппаратных платформ, эффективные битовые (bitmap) индексы и специализированные алгоритмы выполнения запросов, которые многократно повышают производительность обработки аналитических запросов, секционирование данных (partitioning), облегчающее управление и значительно ускоряющее обработку очень больших таблиц, а также сложных с точки зрения обработки запросов, но распространенных схем данных «снежинка» и «звезда».

Битовые индексы. Использование битовых индексов (Bitmap Index) позволяет резко ускорить выполнение определенных типов запросов, характерных для аналитических приложений. В случае если разброс значения какого-либо параметра или группы параметров небольшой (например, пол — мужской/женский; семейное положение — женат(замужем)/разведен(а); тип клиента — физ. лицо/юр. лицо и т. д.), возможно создать битовые индексы, которые позволяют очень быстро производить выборку по такого рода параметрам, так как значения хранятся в виде битовой карты и сравнение в запросе происходит на уровне команд процессора сервера.

Параллельное выполнение. Параллельное выполнение (Parallel Execution) позволяет значительно ускорить выполнение длительных операций, которые часто встречаются при построении и использовании хранилищ данных и аналитических систем. СУБД Oracle позволяет распараллелить такие операции, как выполнение запросов к очень большим таблицам, объединение таблиц, создание больших индексов, материализованных представлений, вставка и изменение записей в больших таблицах при загрузке данных в хранилище. Параллельное выполнение позволяет полнее раскрыть преимущества многопроцессорных систем и кластеров.



Секционирование. Вместе с параллельным выполнением еще одним механизмом увеличения производительности больших хранилищ данных и аналитических систем является секционирование (Partitioning). Этот механизм позволяет физически разбивать таблицы на части или секции по логическому условию, например по дате или по значениям каких-либо других полей, при этом логически таблица остается единой, но на физическом уровне с каждой такой секцией система работает независимо. Это позволяет автоматически заменять операции с большой исходной таблицей набором операций с отдельными ее секциями и таким образом уменьшить время обработки. Технология секционирования применяется как для физических таблиц, так и для индексов, тем самым обеспечивая оптимальную производительность и максимальную управляемость большими объемами данных.

Компрессия данных. База данных Oracle позволяет осуществлять компрессию хранимых данных. Существует нескольких вариантов компрессии, включая компрессию пакетно вставляемых данных, компрессию изменяющихся данных, а также колоночное сжатие, которое используется в аппаратно-программных комплексах Oracle и обеспечивает сжатие данных до десяти и более раз. В случае если в хранилище данных содержится такая информация, как аудио- и видеофайлы, электронные документы и так далее, помимо компрессии данных, СУБД Oracle обеспечивает дедупликацию этих данных.

Технология секционирования и компрессии позволяет реализовывать системы, где секции одной и той же таблицы могут храниться на разных физических дисках и к этим секциям будет применен разный уровень компрессии. Это позволяет сократить расходы на хранение информации путем размещения исторических данных на более медленных дисках с применением компрессии. А активно используемые данные можно разместить на быстрых дисках с применением низкого уровня компрессии или без нее.

Материализованные представления. Запросы аналитиков обычно бывают достаточно сложными, содержат большое количество связей между таблицами, агрегирующие выражения и т. п., что при больших объемах таблиц хранилища приводит к неприемлемым временным затратам на получение результатов. Материализованные представления позволяют хранить в явном виде результаты часто встречающихся запросов и не выполнять их заново каждый раз, а использовать готовый результат. Материализованные представления напоминают обычные представления (view), но, в отличие от них, в базе данных хранится не только текст запроса, но и результат его выполнения в виде некоторой промежуточной таблицы. Данные в исходных таблицах, участвующих в запросе, могут изменяться, и в этом случае необходимо обновлять и промежуточную таблицу. Существенно, что эта синхронизация материализованного представления с исходными таблицами выполняется системой автоматически в соответствии с определенным регламентом, задаваемым разработчиком. Может оказаться, что запрос пользователя не хранится непосредственно в виде материализованного представления, но выполнить его гораздо быстрее не на исходных таблицах, а используя некоторое существующее материализованное представление. Например, если исходная таблица содержит обороты за каждый день и создано материализованное представление, содержащее обороты по месяцам, было бы разумно использовать его, а не исходную таблицу при вычислении оборотов за каждый квартал. Такое переписывание запросов производится также автоматически, что освобождает разработчика от необходимости знать обо всех промежуточных результатах, которые могут оказаться полезными при оптимизации запросов, и работать непосредственно с исходными таблицами.

Одним из важнейших усовершенствований в области оптимизации выполнения аналитических запросов является технология управления агрегированными данными на основе материализованных представлений. Анализируя статистику работы системы, СУБД рекомендует администратору необходимые агрегаты, автоматически их создает и периодически обновляет. Затем при выполнении запросов с агрегированием система автоматически переписывает их таким образом, чтобы они обращались к данным, хранящимся в материализованных представлениях. Такой подход резко, иногда на несколько порядков повышает производительность хранилища данных для конечных пользователей.

Аналитические функции. Для облегчения программирования сложных аналитических запросов в SQL Oracle встроено большое количество аналитических функций. Это, например, функции ранжирования, вычисления лагов, линейной регрессии, вычисления скользящих и многие другие. Аналитические функции призваны облегчить программирование и уменьшить нагрузку на клиентские приложения. Следует отметить,



что существует возможность создавать свои сложные агрегирующие функции, в том числе не только на языке PL/SQL.

Многомерный анализ данных на основе опции OLAP. Для решения аналитических задач, связанных со сложными расчетами, прогнозированием, моделированием сценариев «что, если» и т. д., применяется особая технология многомерного анализа. Эта технология реализуется специальной компонентой Oracle OLAP, входящей в состав СУБД Oracle Database. Опция Oracle OLAP позволяет хранить и обрабатывать многомерную информацию на том же сервере баз данных, где находится реляционное хранилище.

Data Mining (ODM)

Технология Oracle Data Mining предназначена для анализа данных методами, относящимися к классу извлечения знаний или data mining.

Основная задача этой технологии состоит в выявлении в больших наборах данных скрытых закономерностей, зависимостей и взаимосвязей, полезных при принятии решений на различных уровнях управления. Такие закономерности представляются в виде моделей различного типа, позволяющих проводить классификацию ситуаций или объектов, прогнозировать их поведение, выявлять группы сходных объектов и т. п. Существенно, что модели строятся автоматически на основе анализа имеющихся данных об объектах, наблюдениях и ситуациях с помощью специальных алгоритмов.

Методы извлечения знаний применяются в различных областях для решения таких практических задач, как привлечение новых клиентов, выявление наиболее перспективных групп и удержание клиентов, повышение эффективности маркетинговой деятельности по продвижению продуктов и услуг, выявлению причин сбоев оборудования и так далее.

Основу Data Mining составляют процедуры, реализующие различные алгоритмы построения моделей классификации, регрессии, кластеризации. Версия Data Mining 20c поддерживает широкий спектр таких алгоритмов (перечислены в таблице).

Алгоритмы, реализованные в Oracle Data Mining	
Классификация	Naïve Bayes, Decision Trees, Generalized Linear Models (GLM), Support Vector Machine
Регрессия	Support Vector Machine, Generalized Linear Models (GLM)
Поиск существенных атрибутов	Minimal Descriptor Length
Кластеризация	Enhanced K-means, O-cluster
Поиск ассоциаций	Apriori
Выделение признаков	Non-Negative Matrix Factorization

Важная особенность алгоритмов состоит в том, что все они работают непосредственно в реляционной базе данных Oracle и не требуют выгрузки и сохранения данных в специальных форматах. Кроме собственно алгоритмов, Oracle предлагает средства подготовки данных, оценки результатов, применения моделей к новым наборам данных. Использовать все эти возможности можно как на программном уровне с помощью Java API или PL/SQL API, так и с помощью графической среды Oracle Data Miner, которая представляет собой расширение для среды SQL Developer и ориентирована на работу аналитиков, решающих задачи прогнозирования, выявления тенденций, сегментации и др.

Oracle R Enterprise (ORE)

Среда Oracle R Enterprise, используя возможности Oracle Database 20c, существенно повышает производительность приложений, разработанных с помощью некоммерческого языка статистических исследований и графической визуализации R, а также на несколько порядков увеличивает объемы данных, с которыми могут работать эти приложения.



Oracle R Enterprise позволяет применять статистические и аналитические модели, разработанные на R, к данным, которые хранятся в базе данных Oracle, и использовать возможности параллельного выполнения и масштабируемость базы данных.

ORE делает таблицы и представления базы данных доступными для R-среды, как если бы они были объектами R. R Enterprise умеет транслировать команды манипуляции данными R в SQL-операторы, исполнять их в базе данных и возвращать результат в привычную для аналитика среду R.

Можно также использовать встроенный в Oracle Database движок R (R Embedded) и выполнять готовые R-скрипты параллельно с несколькими экземплярами R Embedded на разных наборах данных, что позволяет значительно увеличить скорость статистической обработки данных, моделирования и прогнозирования.

Консолидация баз данных

Консолидация — это концентрация нескольких БД на небольшом количестве серверов. Серверы становятся все более мощными, в каждом новом поколении все больше ядер, больше памяти, каждое новое поколение серверов способно обрабатывать все более крупную нагрузку. Поэтому предприятия с большим количеством небольших БД могут получить значительный выигрыш от консолидации своих БД на небольшом количестве серверов.

Зачем нужна консолидация?

Как правило, оборудование всегда покупается с запасом под пиковые нагрузки приложения. Если профиль нагрузки приложения характеризуется сильными всплесками активности на короткие периоды времени (например, высокая нагрузка в течение 3–4 часов в сутки, а в остальное время активность в несколько раз ниже), то такое приложение является хорошим кандидатом на консолидацию. Если такое приложение работает в выделенном сервере, то такой сервер оказывается загружен в течение нескольких часов в сутки, а в остальное время в этом сервере имеется значительный объем свободных ресурсов. Почему бы их не использовать?

Поэтому первая цель консолидации — уменьшение стоимости. Меньше оборудования — меньше электроэнергии, меньше стойко-мест в ЦОД, меньше затрат на охлаждение, меньше системных администраторов.

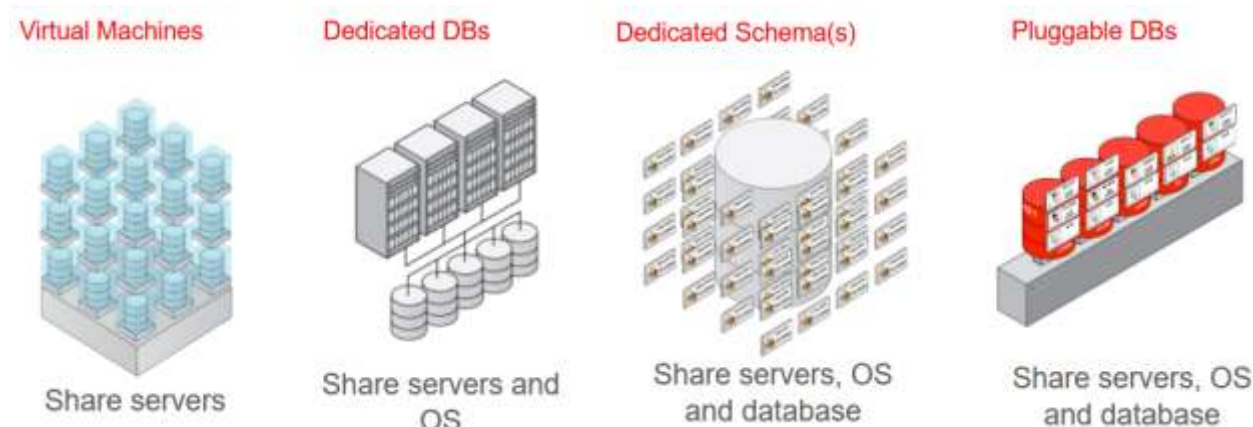
Консолидация позволяет уменьшить количество лицензий на приобретаемое ПО. Например, если до консолидации 10 БД работали на 5 серверах с общим количеством, допустим, 80 ядер, то после консолидации может оказаться, что достаточно одного сервера с 32 ядрами. В общем, консолидация значительно повышает эффективность использования оборудования.

Консолидация в ряде случаев упрощает обслуживание. Чем меньше систем, тем меньше ресурсов, способных выйти из строя, меньше работ по установке обновлений на ОС и СУБД. Меньше точек доступа для злоумышленника, которые легче контролировать. Иными словами, консолидация — это больше приложений (в том числе и баз данных) на меньшем объеме ресурсов.



Способы консолидации

Существует несколько способов консолидации.



Самый популярный способ — просто разместить несколько БД в одном сервере (Share server and OS). Этот способ хорошо подходит для совместной эксплуатации нескольких не очень важных БД. Все базы данных используют все ЦПУ, у каждой БД своя SGA, ввод-вывод делится между всеми БД. Плюсами данного способа можно назвать его простоту и легкость реализации. Однако в этом случае все БД в сервере равноправны, а небольшая и не очень важная БД может занять необычно много ресурсов, то есть негативное влияние одной БД будет ощущаться на всех остальных базах данных.

Для лучшей изоляции ресурсов можно воспользоваться возможностями виртуализации. В этом случае в одном большом сервере запускается несколько виртуальных машин, а в каждой из них своя БД или свое приложение, при этом каждой виртуальной машине гарантированно выделяются свои ЦПУ и своя память. Этот способ хорошо подходит, когда каждой БД и каждой ВМ необходимо гарантировать индивидуальный объем ресурсов, которым не смогут воспользоваться соседние ВМ. Недостаток данного метода — низкая плотность консолидации. Также работа СУБД внутри виртуальной машины приводит к дополнительным накладным расходам и менее эффективно использует оборудование, чем вариант СУБД на оборудовании без виртуализации (bare metal).

Еще один популярный и давно используемый способ, применяемый для консолидации БД — это поместить отдельное приложение в свою схему внутри одной базы данных. В этом случае всем схемам и приложениям доступны все ресурсы сервера: ЦПУ, память, В/В. Для того чтобы ограничить объем используемых ресурсов, для отдельных схем применяется Oracle Resource Manager, который позволяет гибко настраивать распределение ресурсов сервера между приложениями. Однако не все приложения могут поместиться в одну схему. Если приложение создает объекты уровня БД, то два таких приложения не смогут уместиться в одной БД. Это высокий уровень консолидации, дающий максимальную эффективность использования оборудования, но он трудоемок и требует выполнения дополнительных работ по интеграции разных приложений в одну БД, обеспечение дополнительных средств разграничения доступа и безопасности хранения данных.

Четвертый и относительно новый способ консолидации — это использование опции Multitenant. В этом случае в сервере создается одна контейнерная БД, к которой подключаются полноценные базы данных. Каждое приложение в этом случае располагается в отдельной БД. Multitenant — это новая опция СУБД Oracle 12c, в которой одна контейнерная база данных может содержать множество подключаемых баз данных. Любую существующую базу данных можно легко и без каких-либо изменений использовать как подключаемую, при этом на уровне приложения не требуется никаких изменений. Подключаемые базы данных в одной базе-контейнере совместно используют ее память и фоновые процессы. Благодаря этому на одном сервере можно разместить больше баз данных, чем в других случаях.



Функциональность **Resource Manager** была расширена, чтобы динамически управлять ресурсами ЦПУ, памяти, и распределять В/В в нужной пропорции внутри одной контейнерной БД.

Подключаемую БД можно отсоединить от базы-контейнера и подключить к другой контейнерной БД. Ее также можно клонировать в той же самой или другой контейнерной БД. Эти операции, как и создание подключаемой базы данных, производятся с помощью новых команд SQL и занимают считанные секунды. Многие терабайты могут быть скопированы практически мгновенно, используя ключевое слово **snapshot** в команде SQL.

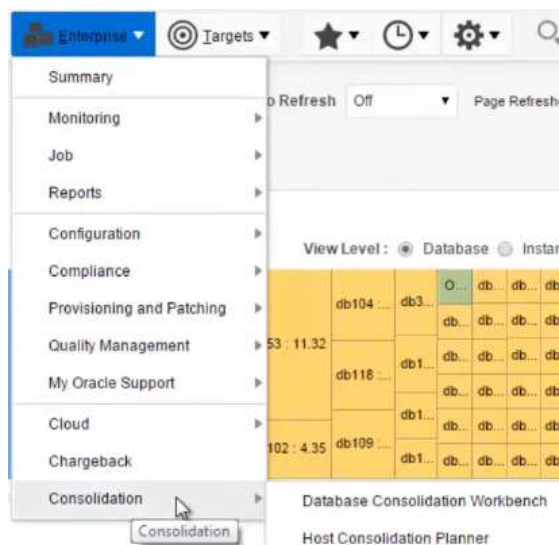
Это самый высокий уровень консолидации, дающий максимальную эффективность использования оборудования.

OEM и консолидация

В Oracle Enterprise Manager встроены два инструмента для оценки эффективности консолидации баз данных:

- OEM **Host** Consolidation Planner,
- OEM **Database** consolidation Workbench.

Host Consolidation Planner анализирует профиль нагрузки с точки зрения операционной системы, собирает статистики производительности ОС и позволяет сделать прогноз для консолидации серверов: Physical → Physical, Physical → Virtual. Host Consolidation Planner знает и умеет сравнивать производительность экздат, OVM, Oracle Cloud Shapes, то есть позволяет оценить потребление ресурсов на новом оборудовании в новой конфигурации. Результаты его работы оформляются в отчет со сценарием консолидации. Возможна ручная корректировка сценария консолидации, например, чтобы указать, какие конкретные хосты нельзя консолидировать на одном новом сервере. Host Consolidation Planner не требует лицензирования.

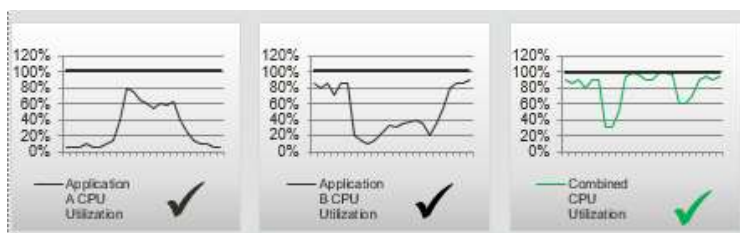


Database Consolidation Workbench — инструмент для анализа нагрузки на основе исторических значений метрик базы данных и операционной системы. Автоматизирует все этапы консолидации: от анализа до развертывания и проверки корректности. Предоставляет рекомендации по консолидации: определяет, какие БД целесообразно совмещать в одном сервере, а какие нецелесообразно, и покажет причину: одновременная высокая потребность в ЦПУ, большой сетевой трафик, который в случае совмещения БД превосходит возможности оборудования и т. д. Рассчитывает консолидацию конкретных БД, работающих на конкретных серверах перед их переносом на новое оборудование. Рекомендует оптимальную стратегию консолидации, основываясь на исторических значениях метрик нагрузки. Позволяет предсказать конфликтные моменты на основе исторических данных о нагрузке.

На рисунке ниже показано, что при совмещении данных БД суммарная нагрузка превосходит ресурсы сервера. Следовательно, консолидация этих БД нежелательна.



Тогда как в следующем случае профиль нагрузки соответствующих баз данных позволяет им работать в одном сервере, не оказывая негативного влияния друг на друга.



Database Consolidation Workbench знает сравнительную мощность разных ЦПУ, анализирует потребление SGA и PGA для каждого инстанса, анализирует объемы баз данных (объемы табличных пространств и свободное место в них, объемы сегментов и свободное место внутри сегментов (space allocated, space used)), оценивает, сколько пространства можно высвободить. Database Consolidation Workbench анализирует нагрузку V/B для каждой БД (IOPS, MBPS), умеет учитывать компрессию и ее влияние на объемы данных и на V/B каждой БД, делает рекомендации по реструктуризации данных и вносит эти рекомендации в отчет, умеет рассчитывать консолидацию баз данных с переносом на Exadata и в облако Oracle (Oracle Public Cloud).

В качестве источника для Database Consolidation Workbench подходят базы версии от 10.2 и более новые. Для использования Database Consolidation Workbench требуется лицензия Diagnostic Pack.

Консолидация с помощью OEM Database Consolidation Workbench выполняется в три шага: **планирование, миграция, проверка**.

Планирование. На стадии планирования пользователь запускает Consolidation Advisor, указывает базы, которые необходимо обследовать. Consolidation Advisor подключается к каждой БД и анализирует ее профиль нагрузки, анализирует потребление ЦПУ, памяти SGA и PGA, анализирует сетевой трафик, анализирует действительный объем данных (Compression Advisor).

Пользователь может указать желаемый вариант консолидации: Database-to-Database (консолидация в базу Multitenant), Database-to-Server (консолидация БД на отдельный сервер, в том числе на Exadata и Oracle Public Cloud). Пользователю также предоставляется возможность указать, какие БД нежелательно совмещать в одном сервере.

Consolidation Advisor выявляет конфликтующие нагрузки и формирует отчет с возможными вариантами размещения баз данных по хостам (сценарии консолидации).

Следующий шаг — **миграция**, это фаза реализации выбранного пользователем сценария консолидации, то есть непосредственный перенос БД на новое «место жительства». Database Consolidation Workbench умеет создавать стендбай и переключаться на него (если перемещение БД в рамках одной аппаратной платформы), умеет использовать RMAN для клонирования БД, умеет использовать DataPump для кросс-платформенной миграции, умеет подключать в CDB любую базу версии 11.2.0.3, 11.2.0.4, 12c.

После перемещения наступает третья фаза — автоматическая **проверка** успешности консолидации. На этой стадии происходит сравнение планов выполнения на новой и старой БД. Database Consolidation Workbench запускает SQL Performance Analyzer, который выполняет сравнение планов и формирует отчет с выявленными регрессиями планов выполнения. Пользователь может проанализировать результаты консолидации и при необходимости поправить планы.

При необходимости более глубоко протестировать результаты консолидации пользователь может выполнить на консолидированной платформе Database Replay (опция СУБД Oracle — Real Application Testing). Для этого необходимо записать нагрузку на исходных БД, объединить ее в консолидированный replay и проиграть записанную нагрузку на консолидированной системе.



Обновление версии и миграция СУБД

Переход на более свежую версию СУБД Oracle Database может осуществляться в рамках перехода на новую версию СУБД (в которой есть новые полезные функции или если техподдержка на старые версии истекает), обновления парка серверного оборудования, перехода на иную технологию хранения, например Oracle Automatic Storage Management, или миграции на другую операционную систему. Во многих случаях переход на новую версию Oracle Database предусматривает перевод базы данных в облако Oracle Cloud. Сценарии обновления и миграции могут значительно различаться, поэтому Oracle предлагает несколько методов для обновления версии СУБД.

Выбор метода миграции или обновления

Термины «обновление» (upgrade) и «миграция» (migration) часто используются как синонимы, но в контексте баз данных Oracle им придается разный смысл. Обновление означает переход на новую версию базы данных на том же самом оборудовании и платформе, а миграция — переход на другое серверное оборудование и/или систему хранения с возможной сменой платформы.



Рис. 1. Различие между обновлением и миграцией

Между обновлением и миграцией имеются два существенных различия. Во-первых, обновление предполагает только модификацию словаря данных (Oracle Dictionary) и не затрагивает данные пользователей и приложений, тогда как миграция обычно предусматривает перемещение или изменение этих данных. Соответственно, размер базы данных оказывает существенное влияние на проект миграции. Во-вторых, миграция может осуществляться и без смены версии.

Выбор наилучшего метода обновления или миграции зависит от характеристик конкретного проекта:

- исходная версия базы данных, включая установленные патчи;
- исходная и целевая операционная системы и их версии;
- исходная и целевая аппаратные платформы и порядок записи байтов (Endian);
- изменение формата данных (набор символов, партиционирования, шифрования, сжатия и т. д.);
- допустимое время простоя;
- размер мигрируемой базы данных;
- архитектура исходной и целевой баз данных (non-CDB или PDB).

Универсального метода, пригодного для всех сценариев миграции и обновления, не существует, но для каждого можно подобрать наилучший, с учетом вышеперечисленных факторов.



Обновление посредством DBUA или командной строки

Наиболее простой метод перейти на новую версию СУБД Oracle Database — обновить существующую базу данных с помощью графической утилиты Database Upgrade Assistant (DBUA) или сценариев командной строки (утилита dbupgrade). Эти два варианта отличаются главным образом пользовательским интерфейсом. Они не предполагают создания копии или нового экземпляра БД, поэтому их иногда называют обновлением по месту.

Во время обновления DBUA вызывает те же самые сценарии, что и используемые в командной строке. Кроме того, он выполняет необходимые проверки перед обновлением и позволяет автоматизировать процессы, выполняемые после обновления.

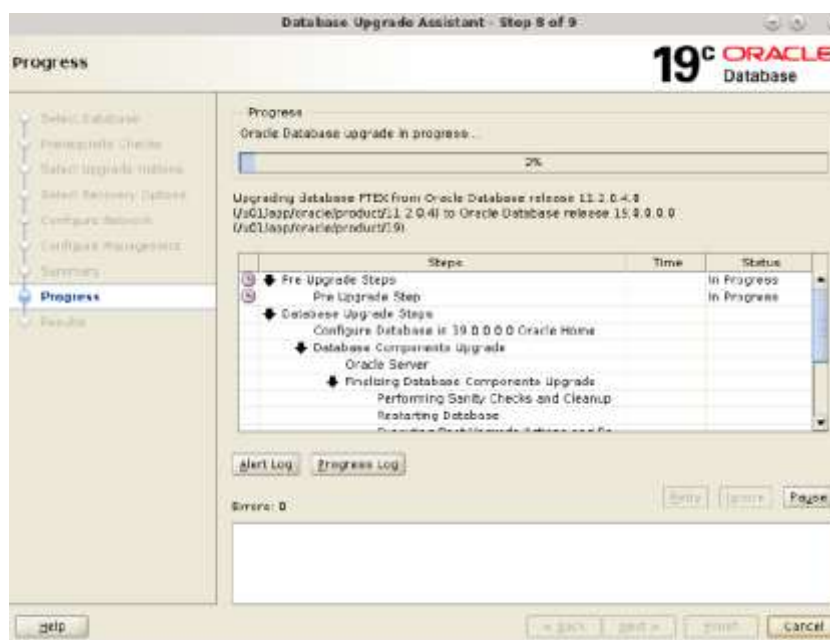


Рис. 2. Oracle Database Upgrade Assistant

Графический пользовательский интерфейс DBUA значительно сокращает объем ручных операций, но он менее гибок, чем командная строка.

DBUA предназначен для обновления базы данных Oracle на существующем оборудовании.

При использовании DBUA пользователь может задать степень параллелизма. Хотя DBUA автоматизирует многие задачи, которые необходимо выполнить перед обновлением, тем не менее некоторые действия может потребоваться осуществить вручную.

Кроме того, для обновления до новой версии СУБД Oracle Database можно использовать утилиту командной строки (dbupgrade). Многие администраторы баз данных предпочитают использовать ее, поскольку это позволяет им более полно контролировать процесс обновления. Утилита dbupgrade также позволяет распараллелить процессы при обновлении базы данных, что сокращает время простоя. Обновление с помощью командной строки состоит из тех же шагов (этапов) и занимает столько же времени, что и при использовании DBUA.

Инструмент Pre-Upgrade Information Tool (preupgrd.jar) автоматически генерирует скрипты для подготовки исходной БД к обновлению и завершению обновления уже на БД новой версии.

Помимо личных предпочтений DBA, типичный сценарий использования утилиты dbupgrade — миграция на новое оборудование с той же самой системной архитектурой. Возможность задавать количество параллельных процессов позволяет сбалансировать скорость обновления с загруженностью системных



ресурсов. В отличие от DBUA, утилита dbupgrade может возобновить обновление с этого места, когда ее работа была прервана.

Детальное описание обновления при помощи DBUA и утилит командной строки представлено в документе [Oracle Database Upgrade Guide](#).

Обновление и миграция с помощью транспортируемых табличных пространств

Технология транспортируемых табличных пространств позволяет копировать набор табличных пространств из одной базы данных в другую. Эта процедура намного быстрее, чем экспорт и импорт данных из этих табличных пространств, потому что табличные пространства копируются в виде физических файлов, без необходимости создания логических объектов, таких как таблицы или индексы, находящиеся в файлах. Однако, помимо копирования табличных пространств, необходимо перенести в новую базу данных метаданные, описывающие объекты в исходной базе данных, с помощью экспорта/импорта Data Pump.

Транспортируемые табличные пространства можно скопировать в базу данных, находящуюся на другой платформе, и/или на другую версию ПО Oracle Database. Это предоставляет относительно быстрый способ миграции или обновления базы данных. Раньше платой за быстроту была относительная сложность процедуры, поскольку пользователь должен был сам перенести метаданные, такие как процедуры, пакеты, ограничения целостности и т. п. Начиная с Oracle Database 12c полностью транспортируемый экспорт/импорт (Full Transportable Export/Import) значительно упрощает процесс переноса метаданных и данных.

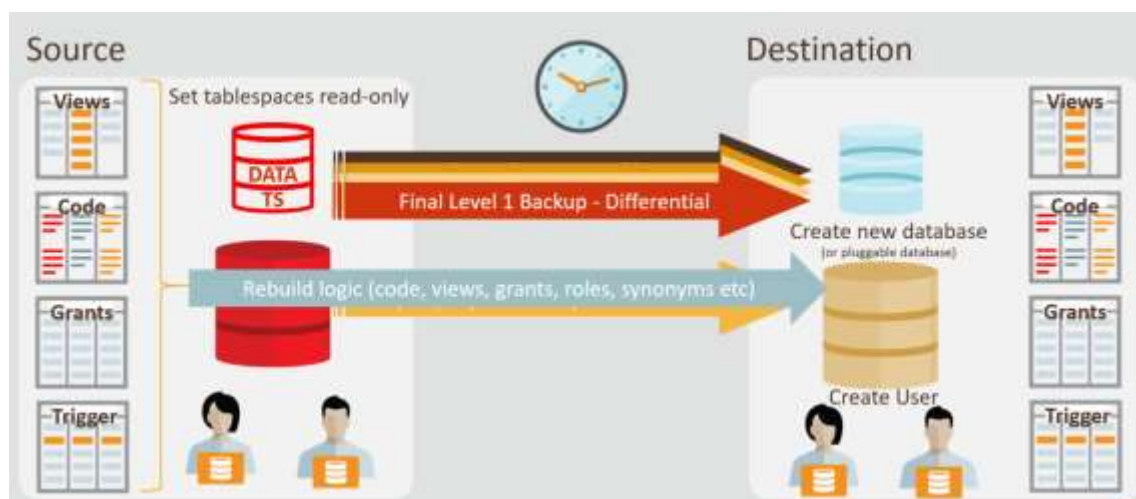


Рис. 3. Обновление и миграция с помощью транспортируемых табличных пространств

Полностью транспортируемый экспорт/импорт позволяет переместить всю базу данных с помощью транспортируемых табличных пространств. Он автоматизирует процесс перемещения метаданных и может перенести пользовательские метаданные, находящиеся в нетранспортируемых табличных пространствах, таких как SYSAUX и SYSTEM. Кроме того, он позволяет переносить зашифрованные табличные пространства.

Перенос метаданных осуществляется посредством файлов дампа или связи базы данных (database link). Комбинация скорости и простоты превращает транспортируемый экспорт/импорт в хороший выбор для многих сценариев.

Данный метод можно использовать для миграции баз данных начиная с Oracle Database 11g Release 2 (11.2.0.3). Задание на экспорт/импорт нельзя остановить: если операция будет прервана, то все задание придется запустить сначала. Если оборудование и операционная система используемых платформ для исходной и целевой баз данных имеют различный порядок записи байтов в машинном слове (Endian), то для конвертации табличных пространств необходимо использовать команду RMAN CONVERT.

Подробное описание процедуры можно найти в техническом документе [Oracle Database 12c: Full Transportable Export/Import](#). (Для более новых версий СУБД Oracle Database процесс транспортируемого экспорта/импорта остался тем же самым.)



Экспорт/импорт с помощью Oracle Data Pump

Oracle Data Pump обеспечивает быстрый перенос данных и метаданных внутри баз данных и между ними. Благодаря высокой гибкости и простоте использования утилиты Oracle Data Pump для экспорта (expdp) и импорта (impdp) технология Oracle Data Pump широко используется для миграции таблиц, схем и баз данных на новые серверы, платформы с иной операционной системой и новые релизы программного обеспечения Oracle Database.

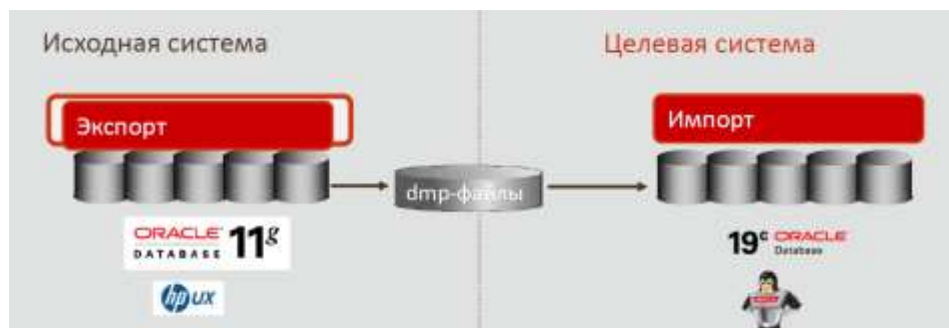


Рис. 4. Обновление и миграция с помощью Oracle Data Pump Export/Import

При использовании этого метода данные могут записываться в файлы дампа на диск или перемещаться из исходной базы данных в целевую непосредственно по сети. При импорте данных они преобразуются в соответствии с характеристиками целевой базы данных, включая миграцию на новый набор символов, реализацию шифрования или сжатия, изменение формата LOB с BasicFiles на SecureFiles для LOB и изменение партиционирования таблиц в базе данных.

На протяжении многих лет экспорт данных в файл дампа остается наиболее распространенным способом перемещения данных между базами данных. Экспорт Data Pump выбирает наилучший метод для извлечения данных из исходной базы данных в плоский файл, импорт Data Pump принимает аналогичное решение относительно чтения файлов дампа и вставки данных в целевую базу данных.

Главные преимущества экспорта файлов дампа — создание постоянной копии данных на диске и высокий уровень распараллеливания при операциях экспорта и импорта. Файлы дампа могут таким образом использоваться повторно для многократного импорта, например, когда необходимо предварительно протестировать или настроить импорт. Экспорт в файл дампа может быть также полезен, когда сеть имеет невысокую пропускную способность. В этом случае может оказаться быстрее отключить диск от исходной системы и подключить его к целевой, чем копировать данные по сети.

Базу данных можно мигрировать и без экспорта в файлы дампа. Импорт Data Pump по сети исключает необходимость в хранении, управлении и переносе файлов дампа. Данные извлекаются из исходной базы данных и добавляются напрямую в целевую базу данных по сетевому соединению. Помимо сокращения потребности в ресурсах для хранения файлов, сетевой режим упрощает миграцию, позволяя перейти на другую платформу и новый релиз Oracle Database за один шаг. Сетевой режим позволяет возобновлять прерванные задания. За простоту сетевого режима приходится платить тем, что экспорт и импорт метаданных осуществляется последовательно, а не параллельно, как объекты данных.

Полную документацию по Data Pump можно найти в [Oracle Database Utilities Guide](#).

Обновление или миграция на новую версию СУБД Oracle Database предоставляет множество преимуществ. Новые функции позволяют увеличить производительность, усилить безопасность и расширить функциональность. В конечном итоге это помогает улучшить операционную эффективность, сократить затраты и повысить продуктивность.



Продукты Oracle



Опции Oracle DB Enterprise Edition



Active Data Guard

Для защиты от катастрофических сбоев (Disaster Recovery — DR) обычно в другом ЦОД создается резервная БД, являющаяся копией основной БД. Она непрерывно синхронизируется с основной БД, является ее копией, и на нее можно переключиться в случае сбоя или недоступности основной БД или во время проведения регламентных работ. Однако обычно эта резервная (standby) БД большую часть времени простаивает и используется только при сбоях. Опция Active Data Guard позволяет расширить использование резервной БД и открыть ее на чтение, в то время как она продолжает синхронизироваться с основной БД. Это повышает эффективность использования ПО и оборудования.

Самое простое решение, способное повысить производительность, — это перенос рабочих процессов, которым необходим доступ только на чтение, на синхронизированную копию основной базы данных. Oracle Active Data Guard позволяет переносить ресурсоемкие работы с основной на одну или несколько резервных баз данных, поддерживаемых технологией Data Guard, которая включена в Oracle Database Enterprise Edition. Oracle Active Data Guard предоставляет доступ для чтения к физической резервной базе данных для создания отчетов, выполнения резервного копирования и других ресурсоемких операций, не требующих значительных изменений в базе данных. Active Data Guard также расширяет возможности Oracle Data Guard по обеспечению отказоустойчивости базы данных, что позволяет добиться высоких уровней доступности систем.

Традиционный Data Guard

Технология Oracle Data Guard обеспечивает синхронизацию физической резервной базы данных (standby) с основной БД для защиты данных и их высокой готовности.

Передача изменений от основной к резервной базе данных производится автоматически с помощью пересылки журналов транзакций, обязательной структуры любой базы данных Oracle Database. Data Guard обеспечивает передачу журналов транзакций как в асинхронном режиме, так и в синхронном режиме, при котором завершение транзакции на основной базе данных происходит только при успешной передаче журнальной информации на резервную базу данных. Синхронный режим передачи журналов транзакций позволяет построить решение для аварийного восстановления «с нулевой потерей данных».

Data Guard в режиме Real Time Apply позволяет применять изменения к резервной базе данных сразу же после их получения с основной БД, не дожидаясь архивирования текущего журнала транзакций на резервной БД. Это обеспечивает оперативную синхронизацию резервной и основной баз данных и дает возможность получения актуальных отчетов в реальном времени. Кроме этого, Real Time Apply сокращает время штатного (switchover) и аварийного (failover) переключения, минимизируя время плановых и внеплановых простоев системы.

Data Guard в режиме Fast-Start Failover позволяет организовать мониторинг работы и аварийное автоматическое переключение на резервную базу данных в случае проблем с основной. При этом прежняя основная база данных, после того как становится доступной вновь, автоматически восстанавливается и преобразуется в резервную базу данных — это предотвращает необходимость пересоздания основной базы данных и очень быстро восстанавливает требуемый уровень защиты системы.

Использование Active Data Guard только для чтения (режим read only with apply)

До выхода Oracle Database 11g физические резервные БД только применяли журналы транзакций основной базы данных. Чтобы предоставить доступ на чтение к резервной базе данных, применение журналов транзакций должно было быть остановлено.

Начиная с версии Oracle Database 11g опция Oracle Active Data Guard позволяет открывать физические резервные базы данных на чтение для создания отчетов, простых или сложных запросов, сортировки и т. д., в то время как изменения из производственной базы данных продолжают применяться на резервной БД. Все запросы к резервной базе данных выполняются в реальном времени и возвращают текущие результаты.



Это означает, что любая операция, которая требует быстрый доступ на чтение, может быть выполнена в резервной базе данных, что увеличивает производительность и разгружает основную базу.

Active Data Guard Reader Farm

Одна база данных способна отсылать изменения на множество (до тридцати) резервных БД, что позволяет создать так называемый Reader Farm. Дополнительные резервные базы данных могут быть использованы для увеличения производительности запросов, отчетов. На рисунке ниже приведен пример обычного Reader Farm.

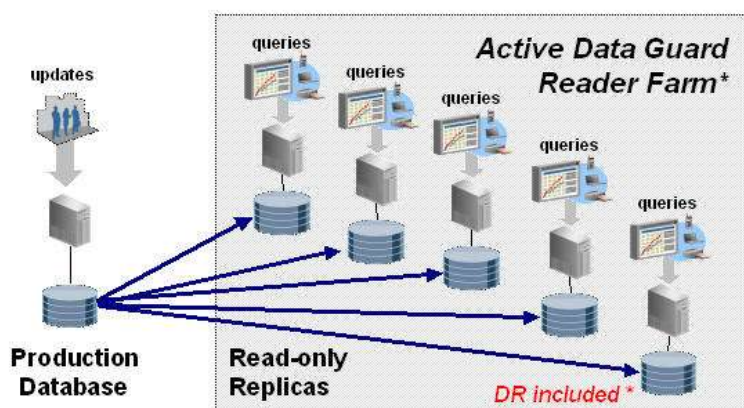
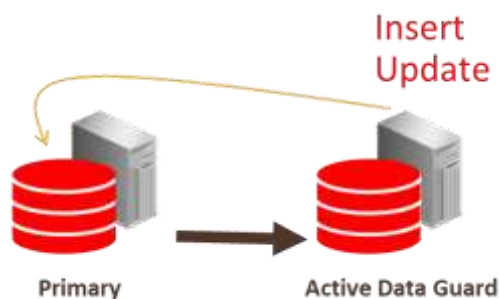


Рис. 1. Reader Farm

Reader Farm также включает в себя функции защиты данных и высокой готовности. Если в основной базе данных произойдет сбой, то любая резервная база данных может легко перехватить ее роль. Это позволяет автоматически синхронизировать оставшиеся базы данных с самыми последними транзакциями.

Поддержка DML-операций на Active Data Guard

Начиная с версии 19c приложениям, которые часто производят операции чтения, но иногда модифицируют данные, возможно выполнять операции Insert и Update в резервной базе данных, несмотря на то что она находится в режиме read only with apply. Для этого необходимо выполнить ALTER SYSTEM ENABLE ADG_REDIRECT_DML (эту установку параметра можно выполнить также и на уровне сессии). В таком режиме все DML операции на Standby с Active Data Guard перенаправляются на primary, а затем через журналы транзакций передаются обратно на резервную базу данных.



Block Change Tracking на Active Data Guard

Active Data Guard также поддерживает отслеживание изменений данных с помощью RMAN, что позволяет создавать быстрые инкрементальные резервные копии основной БД на резервной базе данных. Скорость



создания инкрементальных резервных копий может в несколько раз превышать скорость бэкапов, выполняемых на резервной базе данных в ранних версиях продукта.

Active Data Guard Automatic Block Repair

Также Oracle Active Data Guard обеспечивает взаимное автоматическое восстановление разрушенных блоков в случае их обнаружения на основной или резервной базах данных. Испорченный блок данных на основной БД может автоматически быть заменен корректной копией блока резервной БД, которая работает в режиме real-time query. Испорченный блок данных на резервной БД также может автоматически быть заменен корректной копией блока основной БД.

Active Data Guard Far Sync

До версии 12c администраторам баз данных приходилось применять сложные решения для репликации данных без потери на большие расстояния. Задержка каналов связи между территориально далеко расположенными друг от друга ЦОД ограничивала использование синхронного режима передачи журналов транзакций Data Guard, т. к. транзакция ждала подтверждения от резервной БД. Асинхронный режим мог приводить к потере данных при их передаче на резервную БД при сбое. С появлением технологии Far Sync появилась возможность обеспечивать нулевую потерю данных при сбое на основной площадке, экземпляр Far Sync получает журналы транзакций в синхронном режиме и передает их в резервную базу данных уже независимо от задержек в сети, поэтому восстановление после сбоя возможно вплоть до последней выполненной транзакции на основной базе данных.



Application Continuity и Active Data Guard

Технологии Active Data Guard предоставляют наилучшие средства обеспечения отказоустойчивости базы данных, но со стороны приложения также должны быть предусмотрены меры по автоматическому переключению на резервную базу данных при сбое. Особую сложность представляет обработка ошибок в приложении при сбое во время фиксации транзакций. Application Continuity маскирует любые сбои и запланированные простои, эта технология после переключения на резерв проверяет, все ли завершённые транзакции приложения зафиксированы в резервной базе данных, и если нет, то автоматически повторяет эти транзакции; таким образом, после переключения на резервную базу данных приложение даже не получает каких-либо ошибок в обработке транзакций.

Преимущества Active Data Guard

- **Повышает скорость работы производственной базы данных:** перенос части нагрузки в актуальную физическую копию основной базы данных.



- **Упрощает операции:** устраняет сложности настройки и управления, свойственные традиционным решениям для репликации.
- **Упрощает контроль целостности:** копия базы данных обновляется через журналы транзакций и всегда находится в режиме онлайн, что невозможно с использованием традиционной технологии зеркалирования на уровне дисковых массивов.
- **Сокращает затраты:** резервная база данных Active Data Guard также обеспечивает наилучшие средства для защиты от катастроф и высокую готовность, и кроме того, может служить в качестве тестовой системы.
- **Сокращает время резервного копирования:** совершает инкрементальное резервирование значительно быстрее посредством функции RMAN Block Change Tracking на физической резервной базе данных.

Advanced Analytics

Опция Oracle Advanced Analytics бесплатная и входит в состав Oracle Database. Она объединяет среду статистических исследований Oracle R Enterprise и Oracle Data Mining (ODM).

Технология Oracle Data Mining предназначена для анализа данных методами, относящимися к классу извлечения знаний или «data mining». Основная задача этой технологии состоит в выявлении в больших наборах данных скрытых закономерностей, зависимостей и взаимосвязей, полезных при принятии решений на различных уровнях управления. Такие закономерности представляются в виде моделей различного типа, позволяющих проводить классификацию ситуаций или объектов, прогнозировать их поведение, выявлять группы сходных объектов и т. п. Существенно, что модели строятся автоматически на основе анализа имеющихся данных об объектах, наблюдениях и ситуациях с помощью специальных алгоритмов.

Методы извлечения знаний применяются в различных областях для решения таких практических задач, как привлечение новых клиентов, выявление наиболее перспективных групп и удержание клиентов, повышение эффективности маркетинговой деятельности по продвижению продуктов и услуг, выявлению причин сбоев оборудования и так далее.

Основу Data Mining составляют процедуры, реализующие различные алгоритмы построения моделей классификации, регрессии, кластеризации. Версия Data Mining 11g поддерживает широкий спектр таких алгоритмов (перечислены в таблице).

Алгоритмы, реализованные в Oracle Data Mining	
Классификация	Naïve Bayes, Decision Trees, Generalized Linear Models (GLM), Support Vector Machine
Регрессия	Support Vector Machine, Generalized Linear Models (GLM)
Поиск существенных атрибутов	Minimal Descriptor Length
Кластеризация	Enhanced K-means, O-cluster
Поиск ассоциаций	Apriori
Выделение признаков	Non-Negative Matrix Factorization



Важная особенность алгоритмов состоит в том, что все они работают непосредственно в реляционной базе данных Oracle и не требуют выгрузки и сохранения данных в специальных форматах. Кроме собственно алгоритмов, Oracle предлагает средства подготовки данных, оценки результатов, применения моделей к новым наборам данных. Использовать все эти возможности можно как на программном уровне с помощью Java API или PL/SQL API, так и с помощью графической среды Oracle Data Miner, которая представляет собой расширение для среды SQL Developer и ориентирована на работу аналитиков, решающих задачи прогнозирования, выявления тенденций, сегментации и др.

Среда Oracle R Enterprise (ORE), используя возможности Oracle Database 11g, существенно повышает производительность приложений, разработанных с помощью некоммерческого языка статистических исследований и графической визуализации R, а также на несколько порядков увеличивает объемы данных, с которыми могут работать эти приложения.

Oracle R Enterprise позволяет применять статистические и аналитические модели, разработанные на R, к данным, которые хранятся в базе данных Oracle, и использовать возможности параллельного выполнения и масштабируемость базы данных.

ORE делает таблицы и представления базы данных доступными для R среды, как если бы они были объектами R. R Enterprise умеет транслировать команды манипуляции данными R в SQL-операторы, исполнять их в базе данных и возвращать результат в привычную для аналитика среду R.

Можно также использовать встроенный в Oracle Database движок R (R Embedded) и выполнять готовые R-скрипты параллельно несколькими экземплярами R Embedded на разных наборах данных, что позволяет значительно увеличить скорость статистической обработки данных, моделирования и прогнозирования.

Advanced Compression

ВВЕДЕНИЕ

Экспоненциальный рост объема данных, который переживают в данный момент предприятия, ставит перед ними сложные задачи. Компаниям нужно быстро адаптироваться к изменчивому миру бизнеса без потерь. ИТ-руководителям необходимо эффективное управление существующей ИТ-инфраструктурой для контроля затрат, и возможность обеспечивать высокую скорость работы приложений.

Опция Advanced Compression Option предоставляет широкий набор разнообразных возможностей сжатия данных, который позволяет ИТ-руководителям добиваться успеха в этой сложной среде. Используя опцию Advanced Compression Option, предприятия могут успешно бороться с экспоненциальным ростом данных, что уменьшает затраты на ИТ и позволяет достичь максимального уровня производительности.

На сегодняшний день опция Advanced Compression включает следующие возможности:

- **Сжатие табличных данных:** Advanced Row Compression, Advanced LOB Compression, Advanced LOB Deduplication
- **Сжатие индексов:** Advanced Index Compression
- **Температурные карты и автоматическая оптимизация данных:** Heat Map и Automatic Data Optimization
- **История изменений каждой строки:** Flashback Data Archive
- **Сжатие файлов экспорта:** Data Pump Export Data Compression
- **Сжатие бэкапов:** RMAN Backup Compression



- **Сжатие трафика Data Guard:** Redo Transport Compression
- **Сжатие трафика SQL*Net:** Advanced Network Compression
- **Снапшоты средствами СХД:** Storage Snapshot Optimization

Advanced Compression Option (ACO) предлагает комплексный набор функций для сжатия всех типов данных: будь то реляционные (структурированные) данные, неструктурированные данные (документы, электронные таблицы и т. д.) или данные резервных копий. Самым ощутимым преимуществом сжатия данных является экономия места на жестком диске, увеличение плотности (объемов) данных, находящихся на диске и в оперативной памяти, а также увеличение объемов данных, передаваемых по каналам связи в единицу времени.

СЖАТИЕ ТАБЛИЧНЫХ ДАННЫХ

BASIC TABLE COMPRESSION

Oracle является одним из первопроходцев в технологиях сжатия, применяемых в базах данных. В Oracle Database 9iR2 появилась возможность Basic Table Compression, которая позволила сжимать данные в процессе загрузки данных (direct load). Этот способ сжатия идеально подходит для хранилищ данных, где данные загружаются автоматически большими порциями.

Сжатие таблиц Basic Compression, появившееся в Oracle Database 9iR2, является базовой функцией редакции Enterprise Edition и не требует приобретения Advanced Compression Option. Этот способ работает только для операции Insert и не работает для операций Update, Delete.

Oracle использует алгоритм сжатия, который специально спроектирован для работы с реляционными данными. При проектировании этого решения разработчики исходили из необходимости обеспечить приемлемые уровни сжатия и одновременно обеспечить высокую скорость работы пользовательских процессов, выполняющих сжатие. Алгоритм анализирует повторяющиеся последовательности байтов в одном блоке БД и заменяет их короткими последовательностями, устраняя таким образом дублирование повторяющихся значений. В заголовке такого блока БД создается таблица замены. Этот алгоритм означает самодостаточность блока БД, потому что сжатые данные и метаданные, необходимые для их распаковки, хранятся в одном и том же блоке базы данных. Этот метод обладает значительными преимуществами, избавляя от дополнительных операций ввода-вывода при доступе к сжатым данным.

Степень сжатия зависит от природы сжимаемых данных, а также от способа их представления. Применяя Basic Table Compression, пользователи СУБД Oracle могут рассчитывать на сокращение занимаемого на диске места в 2–4 раза.

Преимущества сжатия данных не ограничиваются одной лишь экономией дискового пространства. При доступе к сжатым данным происходит улучшение производительности SQL-запросов за счет уменьшения количества операций ввода-вывода, поскольку данные размещаются в блоках более плотно и СУБД Oracle придется обращаться к меньшему числу блоков, а одна операция ввода-вывода приносит в SGA больше данных. Данные в буферном кэше хранятся в сжатом виде, поэтому буферный кэш также содержит больше данных.

Если над сжатой таблицей или партицией будет выполнено достаточно большое количество DML-операций, то сжатая таблица со временем потеряет свою сжатость и станет достаточно «рыхлой», т. е. в каждом блоке образуется достаточно большой объем свободного пространства, данные вырастут в объеме, и производительность SQL-запросов к такой таблице может снизиться в несколько раз.



Поэтому начиная с версии 11gR1 в СУБД Oracle появилась опция Advanced Row Compression, которая позволяет пользователям СУБД Oracle держать блоки таблиц в сжатом состоянии при выполнении над ними DML-операций.

ADVANCED ROW COMPRESSION

Алгоритм Advanced Row Compression выполняет сжатие данных при выполнении над ними операций DML. Обработка сжатых данных существенно ускоряет работу системы, сокращая число необходимых операций ввода-вывода, что увеличивает производительность SQL-запросов. Производительность современных процессоров означает, что «дешевле» выполнить, например, 100 тыс. процессорных инструкций для распаковки сжатых данных, чем выполнять дополнительные операции В/В, чтобы прочитать эти данные с диска.

Возможность OLTP-сжатия таблиц является частью опции Advanced Compression, ее нужно приобрести в дополнение к лицензиям Enterprise Edition.

Алгоритм Advanced Row Compression основан на алгоритме Basic Table Compression с небольшими изменениями. В целях экономии алгоритм OLTP Table Compression не сжимает весь блок после каждой операции DML над этим блоком. А именно, пока в блоке есть свободное место, алгоритм Advanced Row Compression не применяется, т.е. операции INSERT и UPDATE выполняются в этом блоке как обычно, записывая строку в блок в несжатом виде. Однако, когда очередная операция DML приводит к переполнению блока, все содержимое блока сжимается. В этом случае в блоке опять появляется значительное свободное пространство и операции DML выполняются штатным порядком, до тех пор, пока блок не будет заполнен. Этот процесс повторяется до тех пор, пока система Oracle не определит, что сжимать этот блок дальше не имеет смысла.

При использовании Advanced Row Compression подавляющее большинство DML-операций будут происходить так же быстро, как если бы они выполнялись над несжатыми блоками. И только транзакции, для которых выполняется анализ и сжатие данных, могут незначительно замедлить систему. Данный алгоритм увеличивает потребление ЦПУ на 2–5 % только для сессий выполняющих сжатие данных. На остальные сессии и на операции SELECT данный алгоритм не оказывает никакого влияния.

Степень сжатия для Advanced Row Compression около 2–5 раз для разных типов данных.

Улучшение сжатия

Улучшить степень сжатия можно следующими способами:

- отсортировав данные перед загрузкой (чтобы похожие строки находились рядом);
- изменив порядок столбцов в таблице;
- используя более крупный размер блока.

Сжатие табличных неструктурированных данных

Начиная с версии 11g в СУБД Oracle был значительно переработан функционал работы с неструктурированными данными (LOB: документы, изображения, электронные таблицы и файлы XML), появился формат хранения SecureFiles. Опция ACO применительно к SecureFiles предоставляет лучшие в своем классе способы сжатия неструктурированного контента. Опция Advanced Compression Option включает технологии, которые в несколько раз сокращают объемы хранимых в SecureFiles данных.



SecureFiles специально спроектированы для максимального увеличения производительности при работе с данными типа LOB, сохраняя при этом преимущества базы данных Oracle. Используя Oracle SecureFiles, компании могут управлять всеми реляционными данными и связанными с ними файлами Oracle, используя единую модель безопасности/аудита, единый процесс резервного копирования и восстановления, а также свободный поиск любой информации.

SecureFiles Deduplication

Часто бывает так, что приложению приходится хранить одни и те же данные несколько раз. Например, приложение для работы с электронной почтой, где несколько пользователей могут получить одно и то же вложение. SecureFiles Deduplication — «умная» технология, которая хранит единственный экземпляр данных SecureFiles и заменяет дубликаты ссылками на этот образ. Возьмем, к примеру, почтовый сервер, где 10 пользователей получают одно и то же письмо с вложением размером в 1 МБ. Без использования SecureFiles Deduplication система хранила для каждого пользователя свою копию вложения, что требует 10 МБ дискового пространства. Если в этом примере использовалась бы опция SecureFiles с дедупликацией, то пришлось бы хранить только 1 МБ. Это экономит дисковое пространство на 90 %. Помимо экономии дискового пространства, SecureFiles Deduplication также увеличивает скорость работы приложений. Операции записи и копирования в особенности эффективны, поскольку в образ SecureFiles записываются только ссылки. Кроме того, операции чтения также в большинстве случаев происходят быстрее, поскольку данные на жестком диске занимают меньше места.

SecureFiles Compression

Опция Advanced Compression применительно к SecureFiles использует отраслевые стандарты сжатия данных с целью свести к минимуму пространство, занимаемое LOB-данными. Сжатие файлов, таких как документы или XML-файлы, позволяет сократить их размеры в 2–3 раза. Технология SecureFiles Compression не применяет сжатие данных, если оно окажется неэффективным с точки зрения освобожденного места, на стадии анализа выполняется тестирование поступивших данных и оценка сжимаемости. Например, если документ или изображение ранее было упаковано сторонним приложением перед добавлением в базу данных.

В версии 19c поддерживаются три уровня сжатия данных: High, Medium, Low. Более высокий уровень требует больше ресурсов процессора, но достигает большей степени сжатия. Обычно SecureFiles Compression отнимает от 3–5 % ресурсов процессора. Приложения по-прежнему могут обращаться к сжатым файлам на чтение и запись, поскольку сжатие данных полностью прозрачно для приложений.

Сжатие индексов

Индексы широко используются в OLTP-базах данных. Обычно на одну таблицу приходится достаточно много индексов, иногда по нескольку десятков. Поэтому индексы могут занимать значительные объемы данных. Типичная ситуация, когда объем индексов составляет около 50% объемов БД или даже больше. Версия 12c СУБД Oracle предлагает возможность применить к индексам Advanced Index Compression и тем самым уменьшить объем дискового пространства, занимаемого индексами в несколько раз + увеличить плотность данных на диске и в оперативной памяти + увеличить скорость выполнения SQL-выражений.

Index Prefix Compression

Начиная с версии 8.1 в СУБД Oracle имеется возможность сжатия ключа индекса Index Prefix Compression.

Обычный несжатый индекс состоит из значений «ключ-ROWID». И если в таблице в индексируемых полях несколько раз повторяется одно и то же значение (ключ), то и в обычном несжатом индексе (в индексном блоке) сочетание «ключ-ROWID» повторится столько же раз, сколько и в индексируемой таблице.

Index Prefix Compression сохраняет только одну копию уникального индексного ключа, за которым следует список rowid. Данный метод сжатия хорошо работает, когда в индексе много повторяющихся значений (низко



избирательный индекс), но уникальные и РК-индексы слабо сжимаются этим методом и в ряде случаев могут стать даже больше оригинала.

Для успешного применения этого метода сжатия требуется глубокое понимание структур данных и тщательный отбор столбцов в префикс. Однако набор префиксных столбцов может быть не оптимальным, чтобы обеспечить хороший уровень сжатия во всех листовых блоках, и в отдельных случаях можно получить «отрицательное сжатие», когда сжатый индекс получается больше исходного индекса.

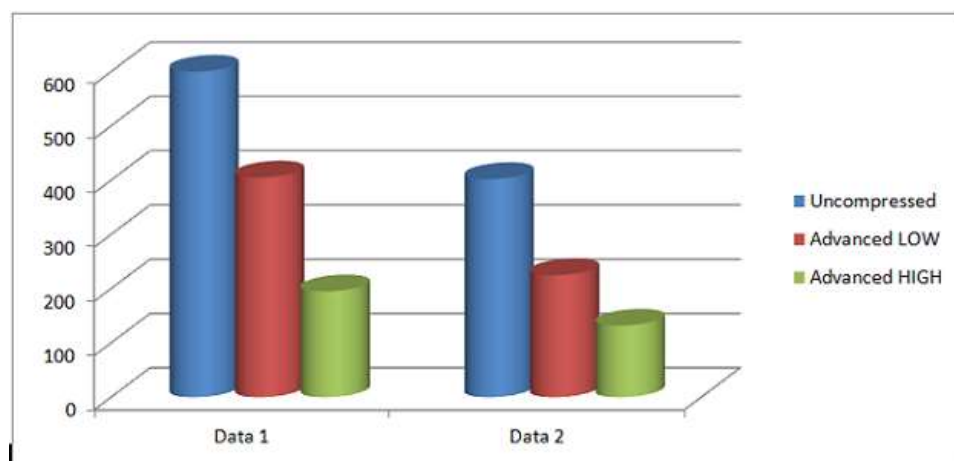
Advanced Index Compression

Advanced Index Compression — это еще один способ сжатия индексов, основанный на алгоритме Advanced Row Compression и предназначенный устранить проблемы, возникающие при применении Prefix Index Compression. Advanced Index Compression является частью Advanced Compression Option, т. е. требует приобретения опции Advanced Compression.

Этот алгоритм аналогичен алгоритму, описанному в **ADVANCED ROW COMPRESSION**, но применяется не к блокам таблицы, а к блокам индекса, которые содержат повторяющиеся «ключ-ROWID». Advanced Index Compression эффективно сжимает любые индексы, даже уникальные и даже те, которые плохо поддаются сжатию методом Index Prefix Compression.

Advanced Index Compression работает на уровне блоков, поэтому не требуется изменение кода приложения, данная функциональность невидима для приложений. Но при этом Advanced Index Compression обеспечивает гораздо лучшую степень сжатия.

Примерные степени сжатия индексов при использовании Advanced Index Compression:



Температурные карты и автоматическая оптимизация данных (Heat Map & Automatic Data Optimization)

В процессе работы базы данных некоторые данные требуются чаще, а некоторые — реже. Обеспечить наилучшее качество доступа ко всем данным слишком затратно, а часто и невозможно. Решением данной проблемы является многоуровневое хранение данных, при котором данные, к которым требуется быстрый и частый доступ («горячие» данные) находились на самых быстрых носителях, а данные, с меньшей интенсивностью доступа («теплые», «холодные») перемещались на более дешевые носители, обеспечивающие доступ к данным с меньшей скоростью.

Многоуровневое хранение данных удачно сочетается с различными типами компрессии. Например, «холодные» данные могут быть сжаты для экономии дискового пространства.



Но даже имея под рукой многоуровневые системы хранения, не всегда понятно, где и какие данные должны располагаться и когда их переносить с одного уровня на другой. Oracle Database 12c решает такую задачу с помощью функциональности, которая автоматически выявляет «температуру» данных (Heat Map) и использует эту информацию, чтобы автоматически переносить данные между табличными пространствами (Automatic Data Optimization).

Heat Map (температурная карта) — автоматическое отслеживание частоты обращения к данным на уровне строк и сегментов с предоставлением детальной информации о том, насколько часто происходил доступ к данным и какого типа: чтение или запись.

Automatic Data Optimization (ADO, автоматическая оптимизация данных) — выполняет перемещение и, возможно, сжатие данных в соответствии с заданными политиками на основе информации, собранной Heat Map.

Технология Heat Map & Automatic Data Optimization подразумевает, что в ИТ-инфраструктуре реализовано многоуровневое хранение данных, когда «горячие» данные располагаются на более дорогом и быстром хранилище, а «холодные» данные перемещаются на более дешевые и медленные хранилища. В процессе перемещения эта функциональность позволяет использовать различные типы/уровни сжатия таблиц, позволяет создавать политики для сжатия и перемещения данных. Данные операции выполняются в фоновом режиме без участия пользователя, обычно ночью в течение maintenance window. СУБД Oracle использует информацию Heat Map, чтобы автоматически определять какие сегменты созрели для перемещения.

Heat Map

СУБД Oracle отслеживает операции чтения, записи на уровне строк и сегментов (таблиц и партиций). СУБД Оракл различает 3 температуры.

Hot — означает, что объект активно подвергается операциям чтения и записи.

Warm — означает, что объект в основном читается и редко изменяется.

Cold — означает, что объект не является активным участником.

Интерфейс Enterprise Manager Cloud Control оснащен командами управления Heat Map и наглядно показывает температуру таблиц и партиций.

Automatic Data Optimization

ADO позволяет компаниям создавать политики данных для автоматического перемещения «теплых» и «холодных» данных на более дешевые носители. Политики ADO задают условия и последующие действия для соответствующих объектов. Политики ADO могут быть заданы на уровне строки, таблицы, отдельной партиции или табличного пространства.

ADO различает политики двух типов:

- политики, описывающие сжатие данных (политики типа «Compression»);
- политики, описывающие перемещение данных из одного табличного пространства в другое (политики типа «Storage Tiering»).

Политики ADO, не основанные на Heat Map: разработчики политик ADO предоставили пользователям СУБД Oracle возможность программировать свои собственные условия для запуска политик ADO в виде PL/SQL-функций.

История изменения строки (Flashback Data Archive)

В промышленной СУБД данные изменяются каждую секунду, и бывают случаи, когда требуется запросить состояние данных на какой-то момент в далеком прошлом. Основная причина такой архитектуры — требования регулирующих органов и такие нормативные документы, как акты Sarbanes-Oxley, HIPAA и Basel-II,



которые требуют строгого контроля изменений и ведения истории данных клиентов. Большинство этих нормативных документов выдвигают длительные периоды хранения истории, обычно 5 и более лет, в течение которых архивы должны быть доступны.

Flashback Data Archive — это возможность хранить в базе данных историю изменения каждой строки для конкретных таблиц. Являясь частью Oracle Advanced Compression Option, Flashback Data Archive дает пользователям СУБД Oracle централизованное и эффективное решение, которое обеспечивает надежное хранение истории изменений данных.

Архивы можно хранить в течение любого количества времени, как этого требуют нормативные законодательные акты или деловые принципы.

Создание архива событий осуществляется простым способом, который не влияет на работу пользовательских сессий. Если для какой-нибудь таблицы настроить хранение истории Flashback Data Archive, то при выполнении DML-операций СУБД Oracle записывает предыдущее состояние строки в табличное пространство, предназначенное для хранения FDA. Для каждого архива задается параметр «RETENTION», который определяет длительность хранения изменений. Flashback Data Archive гарантирует, что данные истории будут храниться столько, сколько указано в параметре «RETENTION».

Сжатие файлов экспорта (DataPump Compression)

Утилита DataPump позволяет пользователям СУБД Oracle сжимать файлы дампов. В версии 19c поддерживаются четыре уровня сжатия данных: High, Medium, Low, Basic.

Файлы меньшего размера занимают на диске меньше пространства. В отличие от используемых в операционной или файловой системе утилит для сжатия файлов, сжатие данных Data Pump также полностью встроено в процесс импорта, поэтому нет необходимости разархивировать сжатый файл перед тем, как его импортировать. Сжатые дампы-файлы импортируются без каких-либо дополнительных вмешательств со стороны администратора базы данных.

Коэффициент сжатия зависит от избыточности данных, но в среднем файлы дампов уменьшаются в 3–5 раз по сравнению с несжатыми.

Сжатие бэкапов (RMAN Backupset Compression)

Пространство для хранения резервных копий напрямую зависит от размера базы данных. Утилита для резервного копирования и восстановления Recovery Manager (RMAN) впервые представила функции сжатия в Oracle Database 10g. Сжатие данных в RMAN значительно сокращает место, необходимое для хранения резервных копий. Резервные копии сжимаются перед тем, как попасть на ленту или диск, их не нужно разархивировать до восстановления — это существенно сокращает затраты на хранение.

Oracle Advanced Compression вводит новые функции сжатия RMAN, которые увеличивают производительность, при этом заметно сокращая требования к дисковому пространству. RMAN создан на базе отраслевого алгоритма сжатия ZLIB. Быстрое сжатие RMAN Compression — идеальное решение для инкрементального резервирования, которое будет безупречно работать в условиях нормальной рабочей загрузки. В версии 19c поддерживаются четыре уровня сжатия данных: High, Medium, Low, Basic. Basic не требует приобретения ACO.

Сжатие трафика Data Guard (DataGuard Redo Transport Compression)

Oracle Data Guard обеспечивает мониторинг и автоматическое обслуживание резервных баз данных, чтобы защитить предприятия от сбоев и повреждения данных. Data Guard обеспечивает синхронизацию боевой и резервной баз данных, используя файлы журналов. DataGuard Redo Transport передает транзакции из боевой БД в резервные БД. С помощью Advanced Compression журнальные данные могут передаваться в сжатом формате, чтобы уменьшить объем сетевого трафика.



Сжатие трафика SQL*Net

В Oracle Advanced Compression появилась возможность сжимать данные SQL*Net перед отправкой по сети. В процессе такого сжатия повышается пропускная способность канала связи, что обеспечивает высокую производительность обмена данными с пользователями.

Storage Snapshot Optimization

Некоторые пользователи СУБД Oracle используют технологии создания снимков на СХД. В этом случае на СХД создается снимок соответствующих томов (как правило, это все дата-файлы, контрольные файлы и онлайн-файлы журнала), затем эти тома монтируются/подключаются к другому хосту/серверу. Эти технологии известны под названиями «split-mirror», BCV, EMC² TimeFinder и т. д.

Такие технологии выполняют создание снимка без перевода БД в режим Begin Backup, в результате чего возникает несогласованная копия дата-файлов. Но при этом пользователи имеют желание восстановить БД в полноценном состоянии.

Снимки, выполненные этим способом, называются «crash-consistent». СУБД Oracle не может отличить такие снимки от упавшей базы данных. Файлы базы данных, выполненные таким способом, требуют применения архивных журналов для приведения БД к консистентному состоянию. Для выполнения восстановления дата-файлов с таких снимков в версии 12c введена специальная команда **SNAPSHOT TIME**, использование которой требует приобретения опции Advanced Compression. Подробнее об этой технологии — в заметке [604683.1 Supported Backup, Restore and Recovery Operations using Third Party Snapshot Technologies](#).

Advanced Security

Oracle Advanced Security предоставляет широкий спектр возможностей по шифрованию (Transparent Data Encryption) и редактированию отображаемых данных (Data Redaction), жизненно важных для защиты критичных данных приложения. Transparent Data Encryption и Data Redaction помогают предотвратить несанкционированный доступ к критичной информации на прикладном уровне, в операционной системе, на резервных носителях и в экспортных файлах базы данных. Oracle Advanced Security полностью поддерживает Oracle Multitenant и интегрируется с инженерными системами Oracle, обеспечивающими беспрецедентную производительность.

Шифрование и редактирование данных для обеспечения конфиденциальности и соответствия требованиям регуляторов

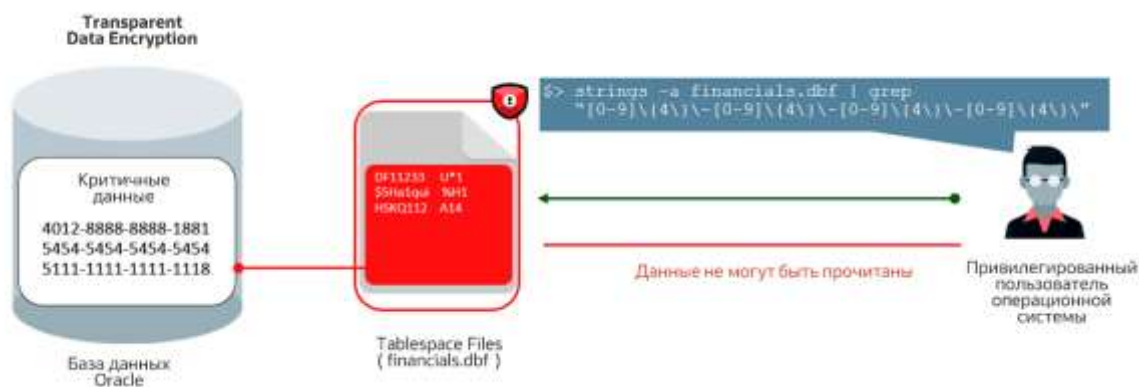
Защита данных должна быть эшелонированной и включать превентивный, детективный и административный контроль. Oracle Advanced Security обеспечивает превентивный контроль, который помогает выполнить многочисленные нормативные требования, предотвратить утечку данных и защитить конфиденциальную информацию. Например, данные кредитной карты могут быть автоматически зашифрованы при записи на физические носители, а при чтении — расшифрованы и отредактированы согласно заданной политике, прежде чем результаты запроса будут отображены в интерфейсах пользователей (см. рис. ниже). Эти две возможности имеют решающее значение для соблюдения требований к конфиденциальности и стандартов, таких как стандарт безопасности данных индустрии платежных карт (PCI-DSS).





Прозрачное шифрование данных

Transparent Data Encryption обеспечивает защиту критичных данных от несанкционированного доступа на физическом уровне путем шифрования файлов данных. Это предотвращает возможность прямого доступа привилегированных пользователей операционной системы к критичной информации путем обхода средств защиты базы данных на логическом уровне и непосредственно просмотра содержания файлов базы данных.



Transparent Data Encryption (TDE) также защищает от воровства, потери или ненадлежащей утилизации (списания) физических носителей с файлами базы данных и резервными копиями.

Это решение является прозрачным для приложений, поскольку данные автоматически зашифровываются при записи на физические носители и расшифровываются при чтении. Элементы управления доступом, применяемые на уровне базы данных и приложений, остаются без изменений. Т. е. никогда не изменяются SQL-запросы, не требуется проводить изменения в коде приложения или его конфигурации.

Операции зашифровывания и расшифровывания выполняются чрезвычайно быстро, поскольку прозрачное шифрование данных использует оптимизацию кэширования. Кроме того, TDE использует аппаратное ускорение на базе процессора, доступное на платформах Intel® AES-NI и Oracle SPARC, включая Oracle Exadata и SuperCluster. Прозрачное шифрование данных еще больше выигрывает от интеллектуального сканирования Exadata, быстрого параллельного расшифровывания данных на нескольких ячейках хранения, а также от гибридного колоночного сжатия Exadata, сокращающего общее количество криптографических операций, которые необходимо выполнить.



Стандартные алгоритмы шифрования и хеширования, используемые TDE:

ENCRYPTION ALGORITHMS	KEY LENGTH
Advanced Encryption Standard (AES)	128, 192, 256 bits
Triple Data Encryption Standard (TDES)	168 bits
ARIA (Korea)	128, 192, 256 bits
SEED (Korea)	128 bits
GOST (Russia)	256 bits

Управление ключами шифрования имеет решающее значение для обеспечения безопасности. TDE предоставляет готовую двухуровневую архитектуру управления ключами, состоящую из ключей шифрования данных и главного ключа шифрования. Ключи шифрования данных управляются автоматически базой данных и, в свою очередь, шифруются главным ключом шифрования. Главный ключ шифрования хранится и управляется вне базы данных в Oracle Wallet, стандартном файле PKCS12, который защищает ключи, или в Oracle Key Vault, централизованной платформе управления ключами, которая соответствует отраслевому стандарту Oasis Key Management Interoperability Protocol (KMIP). Сохранение мастер-ключа отдельно от зашифрованных данных смягчает атаки, поскольку и ключи, и зашифрованные данные должны быть отдельно скомпрометированы, чтобы получить доступ к охищенным данным. Двухуровневая архитектура ключей также позволяет выполнять ротацию главных ключей без необходимости повторного шифрования всех конфиденциальных данных.

Прозрачное шифрование данных легко разворачивается и устанавливается по умолчанию в ходе установки базы данных. Существующие табличные пространства могут быть зашифрованы в режиме онлайн с нулевым временем простоя в производственных системах или зашифрованы в автономном режиме. Кроме того, прозрачное шифрование данных работает без дополнительных настроек с Oracle Automatic Storage Management для защиты данных в хранилищах файлов ASM.

TDE является уникальным по сравнению с альтернативными подходами, которые шифруют целые тома хранилища или требуют дополнительные наборы инструментов и программных интерфейсов. Эти подходы не защищают от многих атак злоумышленников, могут потребовать значительных изменений в приложениях, имеют сложное (или вообще не имеют) управление ключами и не интегрированы с дополнительными технологиями, такими как Oracle Advanced Compression, Oracle Real Application Cluster (RAC), Oracle Recovery Manager (RMAN), Oracle Multitenant, Oracle GoldenGate и Oracle Data Guard.

Шифровать можно не только данные в файлах БД и бэкапов, но также трафик, передаваемый по сети. ASO позволяет выявить и предотвратить перехват и изменение передаваемых по сети (например, от БД к клиенту, серверу приложений, другой БД) данных.

Ограничение доступа к конфиденциальной информации с помощью редактирования данных

Конфиденциальность и соответствие требованиям требуют экономически эффективного подхода к управлению доступом к данным в приложениях. Использование смартфонов и планшетных устройств для работы с приложениями делают проблему безопасности конфиденциальных данных еще более актуальной, поскольку доступ к данным за пределами традиционной офисной среды становится обычным



делом. Даже традиционные приложения требуют более комплексного решения для уменьшения воздействия конфиденциальных данных, например приложение call-центра с экраном, который предоставляет информацию о кредитной карте клиента и личную информацию операторам call-центра. Предоставление этой информации даже действительным пользователям приложения может нарушить правила конфиденциальности и подвергнуть данные неоправданному риску.

Data Redaction обеспечивает выборочное, непрерывное редактирование критичных данных при отображении результатов запроса пользовательскими приложениями так, чтобы неавторизованные пользователи не могли просмотреть критичные данные. Data Redaction минимизирует необходимость вносить изменения в приложения, потому что не изменяет фактические данные во внутренних буферах базы данных, кэшах или устройстве хранения данных, сохраняет исходный тип данных и их форматирование, когда отредактированные данные передаются приложению. Data Redaction не оказывает влияния на операции с базой данных, такие как резервное копирование и восстановление, обновление и патчирование.

Пример вариантов редактирования данных:

	Хранимые данные		Отредактированные значения данных
Full	10/09/1079	→	01/01/2001
Partial	987-65-4328	→	XXX-XX-4328
Regex	fname@example.com	→	[hidden]@example.com
Random	5105105105105100	→	5500000000000004

В отличие от подходов, основанных на внесении изменений в программный код приложений или использовании дополнительных программных компонентов, политики редактирования данных применяются непосредственно в ядре базы данных. Эти политики декларативные и могут задавать различные способы преобразования данных, такие как частичное, случайное и полное редактирование. Редактирование может быть условным, основанным на различных факторах, которые отслеживаются базой данных или передаются в базу данных приложениями, такими как идентификаторы пользователей, идентификаторы приложений или IP-адреса клиентов. Библиотека формата редактирования предоставляет предварительно настроенные шаблоны столбцов, которые могут быть применены к общим типам конфиденциальных данных, таким как номера кредитных карт и национальные идентификационные номера. Как только режим редактирования будет включен, политика немедленно вводится в действие, даже для активных сеансов.

Редактирование данных может быть быстро подготовлено для существующих приложений с помощью API командной строки или Oracle Enterprise Manager. API командной строки — это процедура PL/SQL, в которой выбираются защищаемые колонки, задаются типы преобразований и условия. Oracle Enterprise Manager предоставляет удобный конструктор политик, который позволяет администраторам определять и применять политики редактирования для существующих приложений.



Применение шифрования и редактирования в контейнерной архитектуре Oracle

Oracle Advanced Security полностью поддерживает контейнерную архитектуру базы данных Oracle. Как TDE, так и атрибуты редактирования данных автоматически следуют за подключаемыми базами данных (PDB) при перемещении между контейнерными базами данных (CDB). При перемещении PDB, содержащей политики редактирования, политики передаются непосредственно в новый контейнер как часть PDB. При перемещении зашифрованного PDB мастер-ключи TDE для этого PDB передаются отдельно от зашифрованных данных для поддержания надлежащего разделения безопасности во время транзита. Шифрование и редактирование немедленно возобновляют свою нормальную работу после подключения и настройки PDB.

Шифрование данных в облаке Oracle

В Oracle Database Cloud Service безопасность данных обеспечивается при передаче и хранении данных. Безопасность передаваемых данных достигается с помощью шифрования сетевого трафика. Безопасность хранимых данных достигается за счет шифрования данных, хранящихся в файлах данных базы данных и резервных копиях, с использованием Oracle Advanced Security Transparent Data Encryption. По умолчанию все новые табличные пространства, созданные пользователем в базе данных Oracle Database Cloud Service, шифруются. Для контейнерных баз данных в облаке Oracle TDE поддерживает хранилище ключей для каждой подключаемой базы данных. Эта конструкция обеспечивает большую изоляцию между арендаторами и позволяет проводить независимые операции по управлению ключами. Кроме того, чтобы предоставить клиентам возможность управлять своими собственными основными ключами шифрования, Oracle Key Vault поддерживает гибридное облачное развертывание. В этом сценарии Oracle Key Vault может быть развернут локально для одновременной поддержки развертывания TDE как в облачных, так и в локальных (on-premises) базах данных.

Опция Oracle Database In-Memory

Oracle Database In-Memory — это технология, которая появилась в СУБД Oracle Database 12c, и смысл ее в том, что рядом с привычным буферным кэшем, который хранит строки таблиц и блоки индексов, находится новый кэш, точнее, новая разделяемая область для данных в оперативной памяти, в которой данные из таблиц хранятся в колоночном формате.

Oracle Database In-Memory прозрачным образом на порядки ускоряет аналитику данных и выполнение OLTP-операций для приложений со смешанными типами нагрузки. С помощью Oracle Database In-Memory пользователи получают результаты бизнес-запросов сразу, тогда как раньше это занимало несколько часов.

Двухформатная архитектура

Эта принципиально новая возможность обеспечивается с помощью двухформатной архитектуры Oracle Database In-Memory, представляя таблицы как в традиционном строчном формате, так и в новом колоночном для хранения в оперативной памяти. Оптимизатор SQL-запросов Oracle автоматически направляет аналитические запросы к данным в колоночном формате, а запросы OLTP — к данным в строчном формате, обеспечивая прозрачным образом наилучшую производительность в обоих случаях. Oracle Database автоматически обеспечивает полную транзакционную согласованность между строчным и колоночным форматами (такую же, как между таблицами и индексами).





Рис. 1. Уникальная двухформатная архитектура Oracle Database In-Memory

Новый колоночный формат — это формат хранения исключительно в оперативной памяти. На диске таблицы хранятся в прежнем строчном формате Oracle или гибридном колоночном формате (если таблицы сжаты с помощью технологии Hybrid Columnar Compression). Поскольку для постоянного хранения данных колоночный формат не используется, нет никаких дополнительных затрат или проблем синхронизации на уровне дисковой подсистемы. Изменения данных в новом колоночном формате в оперативной памяти выполняются очень быстро, так как для них не требуется дорогостоящее журналирование с сохранением на диске.

Представление данных одновременно в колоночном и строчном форматах не удваивает требования к объему памяти. Oracle использует алгоритмы управления буферным кэшем с высоким уровнем оптимизации, чтобы сохранять в памяти только активно используемые строчные данные. Многолетний опыт показал, что кэширование небольшого процента блоков данных в оперативной памяти позволяет исключить подавляющее большинство операций ввода-вывода, а практически все остальные операции ввода-вывода можно исключить путем кэширования во флэш-памяти. Следовательно, большую часть ресурсов памяти на сервере базы данных можно выделить под колоночный формат.

Колоночный формат в памяти СУБД Oracle Database использует продвинутые алгоритмы сжатия для увеличения объема памяти и ускорения обработки запросов. Коэффициенты сжатия варьируются от 2 до 20 в зависимости от выбранного алгоритма и избыточности данных. Для разных столбцов, секций или таблиц можно использовать разные методы сжатия. Например, одну группу секций таблицы можно оптимизировать для увеличения скорости сканирования данных, другую — для экономии памяти, а третью — для эффективного выполнения частых операций DML.

Можно кешировать в In-Memory-кэше не всю таблицу, а отдельные секции или отдельные столбцы. Специальный алгоритм позволяет автоматически подгружать новые данные на место давно не используемых данных. Утилита In-Memory Advisor подскажет, какие таблицы и столбцы стоит кэшировать, сколько они займут места и какую выгоду по производительности это даст. Затем Advisor поможет пометить для Oracle таблицы и секции таблиц, которые он будет кэшировать. Даже если часть таблиц одного запроса присутствует в In-Memory-кэше, а часть нет, оптимизатор построит план, использующий оба представления.

Всесторонние In-Memory-оптимизации

Oracle Database In-Memory использует современные алгоритмы для операций сканирования, соединения и агрегирования в памяти. Например, приведенные ниже.

- Аналитические запросы, как правило, ссылаются только на небольшое число столбцов в таблице. Oracle Database In-Memory минимизирует объем работы и обеспечивает максимальную производительность, обращаясь только к тем столбцам, которые нужны для запроса, и обрабатывая их напрямую, без предварительной распаковки.
- Таблицы разбиваются на логические секции, и для каждой секции определяются минимальное и максимальное значение каждого столбца. Это позволяет запросам быстро пропускать те секции таблиц, которые содержат данные за пределами нужного диапазона.
- Некоторые столбцы могут содержать множество повторяющихся значений. Например, столбец, в котором указан географический регион для каждой записи продаж, будет содержать много



повторений одних и тех же регионов. Oracle Database In-Memory сжимает повторяющиеся значения в целях экономии памяти и оптимизирует обработку, выполняя предикаты запросов только один раз для каждого уникального значения в столбце (например, один раз для каждого уникального региона).

- Современные микропроцессоры поддерживают векторные SIMD-команды (Single Instruction Multiple Data - один поток команд, много потоков данных) для ускорения обработки графики и научных расчетов. Oracle Database In-Memory может использовать векторные SIMD-команды для обработки нескольких значений столбца за одну команду процессора.

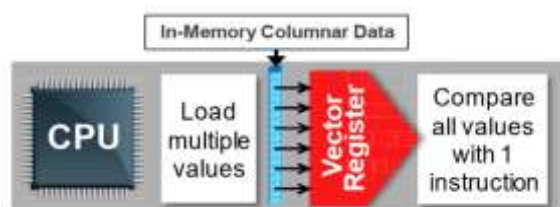


Рис. 2. Векторная обработка ОКМД сканирует миллиарды значений в секунду

- Преимущества нового колоночного формата используются при соединении таблиц в оперативной памяти: условия соединения преобразуются в фильтры, которые применяются во время очень быстрого сканирования данных.
- С помощью колоночного формата алгоритмы агрегирования в памяти ускоряют выполнение аналитических запросов и формирование отчетов, делающих агрегирование больших объемов данных.

Благодаря всем этим приемам оптимизации Oracle Database In-Memory обеспечивает огромную скорость обработки запросов — несколько миллиардов строк в секунду для каждого процессорного ядра. Аналитические операции, которые раньше занимали несколько часов или дней, теперь выполняются за считанные секунды, что позволяет принимать бизнес-решения в реальном времени.

OLTP-системы со смешанными типами нагрузки

Помимо OLTP-транзакций, базы данных со смешанной нагрузкой создают отчеты и выполняют нерегламентированные запросы. Как правило, для ускоренного формирования отчетов создается несколько десятков индексов для наиболее важных таблиц. Благодаря колоночному формату Oracle In-Memory большинство этих аналитических индексов не нужны, поскольку обеспечивается такая же производительность, как при создании индекса для каждого столбца, но с гораздо меньшими транзакционными накладными затратами. Операции OLTP ускоряются, так как каждой транзакции больше не требуется обновлять аналитические индексы. Не нужны и длительные процессы настройки и администрирования, связанные с определением и созданием аналитических индексов.

Еще важнее то, что теперь пользователи могут в кратчайшие сроки получать ответ на любые свои запросы. Пользователи могут не ограничиваться только теми запросами или отчетами, которые были предварительно оптимизированы с помощью аналитических индексов. Используя Oracle Real Application Clusters (RAC) в сочетании с Oracle Database In-Memory, можно изолировать аналитические рабочие нагрузки от OLTP, выполняя их на разных узлах кластера.

В машинах баз данных Exadata можно зеркалировать данные, загруженные в область колоночного хранения, посредством указания ключа DUPLICATE атрибута IN-MEMORY. Это означает, что у каждого объекта в области In-Memory будет зеркальная копия на другом узле кластера. При сбое узла кластера In-Memory-кэш быстро восстановится на основе своей копии.





Рис. 3. Объекты в In-Memory Area в Exadata могут зеркалироваться для повышения отказоустойчивости

Предприятие реального времени

Современные бизнес-транзакции выполняются в реальном времени, но обработка бизнес-запросов, требующих анализа данных, может занимать несколько часов. Технология Oracle Database In-Memory использует широкие возможности масштабирования оборудования вместе с новыми структурами и алгоритмами обработки данных в оперативной памяти, чтобы обеспечить мгновенный ответ на любой вопрос. Oracle Database In-Memory обрабатывает миллиарды, а не миллионы строк в секунду.

Кроме того, аналитические запросы можно выполнять непосредственно в базах данных OLTP, что еще больше сокращает задержки, повышает точность результатов. Сочетая анализ данных с обработкой транзакций в реальном времени для всех существующих приложений, организация может реализовать концепцию предприятия реального времени, обладающего следующими особенностями.

- Использование оперативных и точных ответов на запросы для принятия решений в режиме реального времени.
- Мгновенное удовлетворение потребностей заказчиков в информации, вариантах для выбора, персонализации и взаимодействии.
- Непрерывная оптимизация всех ключевых процессов, включая продажи, маркетинг, производство, кадровое обеспечение, оценку затрат и т. д., на основе актуальных данных.

Используя Oracle Database In-Memory, организация может вести свою деятельность в реальном времени, чтобы опередить конкурентов, добиться лояльности заказчиков и повысить прибыль.

Database Vault

Oracle Database Vault (DBV) обеспечивает мощные средства контроля безопасности для защиты конфиденциальных данных от несанкционированного доступа и обеспечивает разделение обязанностей между администраторами баз данных и владельцами данных в соответствии с требованиями конфиденциальности и нормативными требованиями. Контроль может производиться для блокирования доступа привилегированных учетных записей к данным приложения и управления конфиденциальными операциями внутри базы данных. DBV обеспечивает прозрачную защиту баз данных приложений, устраняя необходимость в дорогостоящих и трудоемких изменениях приложений.

Контроль над привилегированными учетными записями

Скомпрометированные привилегированные учетные записи баз данных являются одним из наиболее часто используемых путей получения доступа к конфиденциальным данным. В то время как их широкий и неограниченный доступ облегчает обслуживание базы данных, та же самая привилегированная учетная запись также создает точку атаки для получения ненадлежащего доступа к большим объемам данных.



Защитные области (Realms), определенные вокруг схем приложений, таблиц и хранимых процедур, обеспечивают средства управления, предотвращающие использование привилегированных учетных записей злоумышленниками для доступа к конфиденциальным данным. Различные нестандартные факторы, такие как IP-адрес, метод аутентификации и имя программы, помогают реализовать авторизацию «доверенного пути» для предотвращения атак, использующих украденные пароли.

Защитная область DBV блокирует доступ привилегированного пользователя к данным:

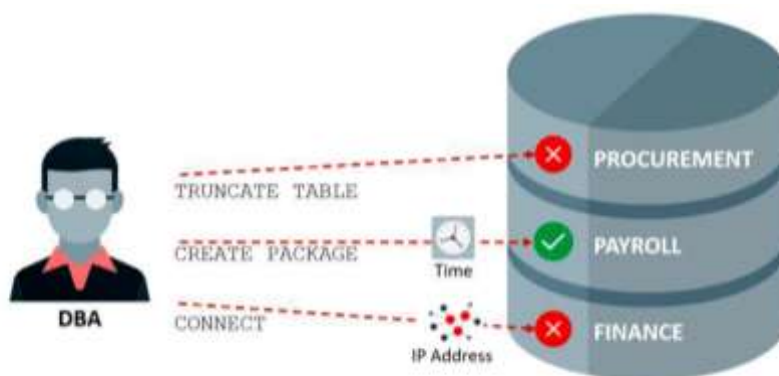


Правила для контроля выполнения команд

К числу наиболее распространенных выводов по результатам аудита относятся замечания в части несанкционированных изменений прав доступа к базе данных, включая предоставление роли DBA, а также бесконтрольное создание новых учетных записей и объектов базы данных. Предотвращение несанкционированных изменений в производственных средах важно не только для обеспечения функциональной безопасности, но и для соблюдения регулятивных требований, поскольку эти изменения могут ослабить безопасность и открыть двери для хакеров, нарушая правила конфиденциальности и требования по обеспечению безопасности. Правила для контроля выполнения команд DBV позволяют управлять операциями внутри базы данных, включая такие команды, как ALTER DATABASE, ALTER SYSTEM, TRUNCATE TABLE и CREATE USER. Эти механизмы защиты DBV предотвращают несанкционированные изменения конфигурации, а также предотвращают вмешательство хакеров и злонамеренных инсайдеров в работу приложений и внесение в них изменений.

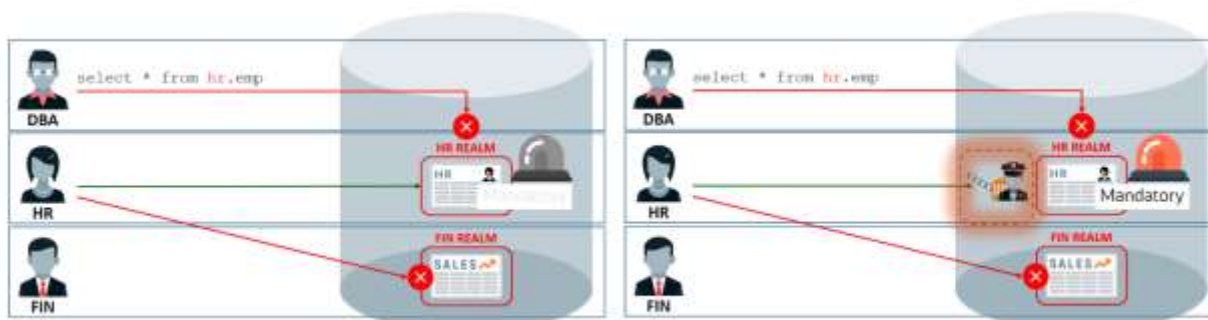
Правила для контроля выполнения команд можно использовать для ограничения доступа к базам данных для конкретной подсети, сервера приложений и программы. Создавая доверенный путь от приложения к базе данных. Встроенные факторы, такие как IP-адрес, имя хоста и имя пользователя сеанса, могут использоваться для принудительного применения элементов управления SQL-командами в базе данных. Метки безопасности (Oracle Label Security) могут также использоваться, чтобы контролировать доступ пользователя. Кроме того, собственные функции и факторы приложений Oracle APEX могут использоваться с правилами контроля выполнения команд DBV для определения того, следует ли разрешить доступ к определенным операторам DML или DDL.





Защитные области с особым правилом контроля доступа (Mandatory)

DBV в базах данных Oracle, начиная с версии 12c, обеспечивает новый вид защитных областей (Mandatory Realm), которые позволяют прекращать доступ к объектам приложения, даже тем пользователям, которые имели «непосредственные» гранты (direct object grants) к этим объектам, включая владельца объекта. Это полезно, когда необходимо обеспечить доступ к схеме приложения непосредственно как владельца приложения (при выполнении операций по технической поддержке, например). Mandatory Realm могут включаться во время выполнения и использоваться в ответ на киберугрозу, прекращая доступ к данным.



Контроль доступа для Oracle Multitenant

DBV обеспечивает защиту подключаемых баз данных (PDB), позволяя клиентам создавать защитные области вокруг конфиденциальных данных приложений внутри PDB, который предотвращает доступ к конфиденциальным данным Администратора контейнерной базы данных (CDB) администраторов других PDB этой же CDB. Правила контроля выполнения команд DBV могут применяться внутри PDB, исходя из того, где и как осуществляется доступ к защищаемой PDB, а также из того, какие операции могут выполняться в этой PDB.

Изменения в системе защиты DBV, начиная с версии Oracle Database 19c, позволяет предотвратить доступ обычных пользователей к локальным данным PDB без необходимости настраивать и активировать DBV в каждой PDB. Этот элемент управления подходит для облачных операций и консолидированных клиентских контейнерных баз данных, где команда DBA инфраструктуры управляет множеством PDB из CDB, используя общие учетные записи пользователей. Таким образом, конфиденциальные данные в PDB защищаются без необходимости включения DBV и настройки защитных областей для защиты локальных данных.



Простота развертывания и эксплуатации

Oracle Database Vault устанавливается по умолчанию начиная с версии Oracle Database 12c (в более ранних версиях, начиная с версии Oracle 9.2 и выше, требовалось дополнительно устанавливать и настраивать), и может быть включен в командной строке. После инициализации DBV базу данных Oracle необходимо перезапустить. Установка дополнительного программного обеспечения или повторная компоновка исполняемых файлов базы данных Oracle не требуется.

DBV защищает данные приложений, обеспечивая при этом возможность администраторам базы данных (DBA) выполнять свои обычные обязанности, такие как настройка, диагностика, резервное копирование и восстановление, как обычно. Однако DBV действительно обеспечивает соблюдение дисциплины, когда речь заходит об администрировании или попытке несанкционированного доступа к защищаемым конфиденциальным данным. Прежде чем администраторы смогут экспортировать, импортировать или перемещать защищаемые конфиденциальные данные, им потребуется соблюсти ряд ограничений, предписываемых политикой информационной безопасности. Например, проводить операции с определенного рабочего места или с помощью специализированной учетной записи.

Политики безопасности DBV применяются в ядре базы данных Oracle, обеспечивая беспрецедентную безопасность и очень малое влияние на производительность защищенной базы данных.

Начиная с версии Oracle Database 12cR2 в DBV реализована возможность использовать режим имитации, который позволяет снизить риски влияния DBV на рабочую среду. Вместо включения механизмов безопасности они переводятся в режим имитации, чтобы фиксировать нарушения установленных правил контроля выполнения команд и защитных областей в специальном журнале, а не блокировать инструкцию SQL. Это может быть использовано для более быстрой сертификации приложения с новыми правилами политики безопасности DBV, так как приложение сможет завершить свой регрессионный тест без блокировки.

In-Memory Database Cache

Oracle TimesTen In-Memory Database (TimesTen) — это реляционная СУБД, оптимизированная для работы в оперативной памяти. Она обеспечивает приложениям возможность мгновенного реагирования и очень высокую скорость обработки данных, необходимые современным предприятиям и отраслям, работающим в реальном времени (телекоммуникации, рынки ценных бумаг, системы обороны и т. п.). СУБД Oracle TimesTen работает с данными, которые полностью размещаются в физической памяти.

TimesTen может быть развернута тремя следующими способами.

- TimesTen Classic
 - База данных, установленная на одном узле, предназначенная для приложений, которым требуется минимальное и очень стабильное время отклика. Высокая доступность в TimesTen Classic обеспечивается с помощью репликации на другой узел в режиме «active-standby». Также может поддерживаться несколько дополнительных узлов-подписчиков, работающих в режиме только для чтения, чтобы обеспечить масштабирование тяжелых рабочих нагрузок на чтение. Приложения работают с базами данных TimesTen, используя стандартные языки SQL и PL/SQL через программные интерфейсы JDBC, ODBC, ODP.NET, Oracle Call Interface (OCI) и Pro*C/C.
- TimesTen Cache
 - Опция Oracle Database Enterprise Edition, позволяющая приложениям выборочно кэшировать критичные к времени доступа данные из таблиц Oracle Database в TimesTen для повышения



производительности приложения. TimesTen Cache основан на той же технологии, что и TimesTen Classic с добавлением функций кэширования.

- TimesTen Scaleout
 - Горизонтально масштабируемая реляционная СУБД в оперативной памяти без разделения ресурсов, которая обеспечивает высочайшую пропускную способность, как того требуют многие приложения OLTP в широком спектре отраслей. В основе TimesTen Scaleout лежит ведущая в своем классе технология баз данных TimesTen In-Memory. Архитектура горизонтального масштабирования без разделения ресурсов позволяет масштабировать базу данных на десятки узлов, увеличить размер базы данных до сотен терабайт и поддерживать сотни миллионов транзакций в секунду без ручного разбиения базы данных на шарды или разделения рабочей нагрузки. TimesTen Scaleout поддерживает полный SQL, ACID-транзакции, стандартные API-интерфейсы баз данных и имеет встроенный механизм отказоустойчивости.

TimesTen Classic

Во многих отношениях СУБД TimesTen Classic (TimesTen) похожа на другие системы RDBMS, поэтому значительная часть его интерфейса и администрирование могут показаться знакомыми.

Большая часть работы, которая делается обычной, оптимизированной для работы с диском RDBMS, производится в предположении, что данные изначально размещаются на диске. Алгоритмы оптимизации, управление буферным пулом и методы поиска по индексам разработаны на этом фундаментальном предположении. Основное отличие TimesTen от обычной RDBMS заключается в том, что TimesTen спроектирована для работы с данными, целиком расположенными в оперативной памяти, и поэтому она может выбирать более прямые маршруты к данным, сокращая время поиска данных и упрощая как его алгоритм, так и структуру данных. Более того, кроме использования соединения клиент/сервер к БД, база данных TimesTen может быть развернута на уровне приложения, что существенно сокращает время доступа к БД TimesTen за счет исключения межпроцессного взаимодействия (IPC) и накладных расходов сети. Все это позволяет достигать времени отклика операции чтения меньше 2 микросекунд и около 7 микросекунд операции записи, что также выводит пропускную способность БД TimesTen на уровень от десятков до сотен тысяч транзакций в секунду (измерялось на оборудовании с ОС Oracle Linux на Intel Xeon E5-2680@2.7GHz).

TimesTen Classic обслуживает многопользовательские и многопоточные приложения, поддерживает блокировку на уровне строки и уровень изолированности транзакций read committed. Приложения получают доступ к базе данных TimesTen, используя стандартные языки SQL и PL/SQL посредством доступа через интерфейсы JDBC, ODBC (родной интерфейс), Oracle Call Interface (OCI), ODP.NET и Pro*C/C++.

База данных TimesTen является надежной и восстанавливаемой. Надежность достигается благодаря комбинации журналов транзакций и периодических обновлений снимка базы данных на диске. Записи системного журнала сохраняются на диске в синхронном или асинхронном режиме по отношению к окончанию выполнения транзакции и управляются на уровне приложения. TimesTen позволяет сохранить всю БД из памяти на диск. Это осуществляется с помощью механизма контрольных точек. Он работает в фоновом режиме и оказывает очень небольшое воздействие на производительность базы данных.

Для обеспечения высокой доступности TimesTen использует схему репликации Active/Standby (A/S). Схема репликации A/S состоит из двух автоматически синхронизирующихся баз данных, развернутых на разных машинах и связанных LAN-сетью. В любой заданный момент времени только одна база данных в паре A/S может быть активной, т. е. выполнять DML-операции. Вторая база данных, выступающая в роли standby, может выполнять только запросы от приложений. При необходимости база данных standby может быть очень быстро (за миллисекунды) переведена в активный режим. Oracle Grid Infrastructure позволяет автоматизировать процесс обнаружения сбоев основной базы данных и переключение на резервную базу данных.



Дополнительно схема репликации A/S может содержать до 127 баз подписчиков read-only, которые могут быть связаны с парой A/S баз данных через LAN- или WAN-сеть. Подписчики могут быть использованы для масштабирования нагрузок read-only и защиты от катастроф.

В схеме A/S по умолчанию реплицируется вся база данных, при желании можно исключить заданные объекты из репликации.

В TimesTen обеспечена поддержка как асинхронной репликации, которая обеспечивает максимальное быстродействие, так и синхронной репликации, которая гарантирует максимальную доступность и согласованность данных между серверами TimesTen. Репликация может работать в параллельном многопоточном режиме.

Механизм репликации TimesTen основан на захвате изменений, записываемых в журнал транзакций базы данных. При правильной настройке процесс захвата изменений целиком осуществляется в оперативной памяти без чтения данных с диска.

Установка, настройка и администрирование СУБД Oracle TimesTen не требуют много времени и специально выделенных администраторов. Простота дисковых структур и относительно небольшой и постоянный размер базы данных в оперативной памяти делают ненужным большинство традиционных задач администрирования баз данных. Задачи администрирования выполняются в основном через утилиты командной строки и часто встраиваются в приложения для их выполнения без участия оператора. Утилиты обеспечивают резервирование, восстановление, копирование и миграцию баз данных, установку политик и выполнение интерактивных запросов в базе данных, а также контроль состояния транзакций и системы в целом.

Задачи администрирования также могут выполняться через Oracle SQL Developer, а отслеживать состояние БД TimesTen можно через специальный плагин для Oracle Enterprise Manager.

TimesTen Cache

TimesTen Cache — опция СУБД Oracle, позволяющая для повышения производительности приложения выборочно кэшировать критичные к времени доступа данные из таблиц Oracle Database в TimesTen In-Memory Database для повышения производительности приложения. TimesTen Cache обеспечивает персистентность, транзакционную целостность и автоматическую синхронизацию данных с Oracle Database для операций read/write, выполняемых приложениями на кэшируемых таблицах с использованием SQL и PL/SQL.

TimesTen Cache основан на той же технологии, что и TimesTen Classic, с добавлением функций кэширования, разворачивается на уровне приложения и поддерживает многопользовательские и многопоточные приложения. Приложения работают с кэшируемыми таблицами, используя стандартный SQL через программные интерфейсы JDBC, ODBC, ADO.NET, Oracle Call Interface (OCI), Pro*C/C++ и Oracle PL/SQL. Кэшированные таблицы работают как обычные реляционные таблицы в базе данных TimesTen и являются персистентными и восстанавливаемыми.

Приложения могут выбрать следующие комбинации опций кеширования TimesTen Cache.

- Read-only-кеши — транзакции сначала выполняются в Oracle Database, и изменения затем передаются в базу данных TimesTen-кэш.
- Read-write (или сквозные) кэши — транзакции сначала выполняются в базе данных TimesTen-кэш, а потом передаются в Oracle Database.
- Кэширование по требованию или предварительная загрузка данных в кэш — данные могут быть загружены в кэш по требованию или заранее.

TimesTen Cache использует понятие cache group (кэш-группа), которое описывает совокупность таблиц базы данных Oracle, которые будут кэшированы в памяти. Кэш-группа может состоять из всех или из подмножества столбцов или строк этих таблиц.



Синхронизация данных с Oracle Database осуществляется автоматически.

- Асинхронный сквозной кэш сначала фиксирует транзакции локально в базе данных TimesTen кэш, а затем асинхронно отправляет изменения в Oracle Database. Асинхронные сквозные кэш-группы обеспечивают самое быстрое время отклика приложения и самую высокую пропускную способность транзакций.
- Синхронный сквозной кэш гарантирует, что будет сделан откат транзакции в базе данных TimesTen кэш, если Oracle Database не может принимать изменения. Приложение должно ожидать завершения фиксации транзакции как в Oracle Database, так и в TimesTen Cache.
- Для кэшей read-only используется инкрементальное обновление через заданные интервалы времени.

TimesTen Cache может продолжить работу даже после того, как будет прервано соединение с сервером базы данных Oracle Database. Обновления кэш-групп будут поставлены в очередь и отправлены в Oracle Database, как только восстановится соединение. Точно так же обновленные в Oracle Database версии кэшированных данных будут поставлены в очередь и позже переданы в кэш-группы TimesTen, если те заняты или недоступны.

TimesTen Cache предлагает встроенные механизмы репликации кэш-баз данных такие же, как в TimesTen Classic. Большинство корпоративных приложений должны работать непрерывно, поэтому они часто используют репликацию баз данных TimesTen кэш для обеспечения высокой доступности и балансировки нагрузки.

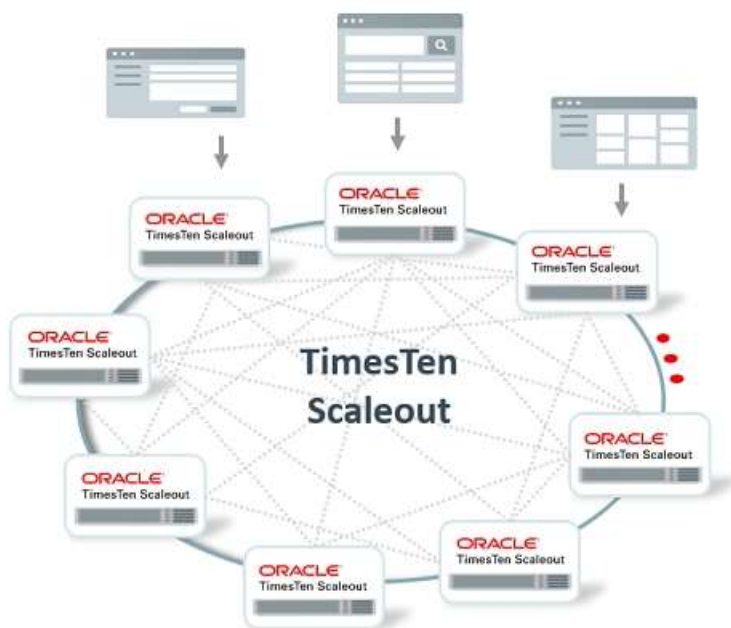
TimesTen Scaleout

TimesTen Scaleout — это горизонтально масштабируемая реляционная СУБД в оперативной памяти без совместного использования ресурсов (Shared Nothing), предназначенная для экстремально высоких OLTP-нагрузок следующего поколения. СУБД TimesTen Scaleout отличается гибкой масштабируемостью, полным SQL, стандартными API-интерфейсами баз данных, транзакциями ACID и встроенной отказоустойчивостью.

Решение TimesTen Scaleout разработано на основе ведущей в классе баз данных в оперативной памяти Oracle TimesTen In-Memory Database.

Новая архитектура горизонтального масштабирования без совместного использования ресурсов позволяет масштабировать базу данных на десятки хостов, увеличить размер базы данных до сотен терабайт и поддерживать сотни миллионов транзакций в секунду без ручного разбиения базы данных на шарды или разделения рабочей нагрузки. Приложения также получают выигрыш от параллельного выполнения SQL на узлах базы данных и прозрачного автоматического механизма высокой доступности, благодаря которому все узлы активны. Эластичная масштабируемость TimesTen Scaleout предоставляет приложениям возможность динамически наращивать (или сокращать) объем базы данных и процессорную мощность в зависимости от потребностей бизнеса без ущерба для производительности.





TimesTen Scaleout база данных выглядит для приложений как единое целое, несмотря на то, что данные распределены по многим узлам. Приложения имеют полный доступ ко всем данным независимо от того, к какому узлу базы данных они подсоединены. TimesTen Scaleout поддерживает ACID-транзакции, стандартные SQL и PL/SQL и стандартные API-интерфейсы баз данных, такие как JDBC, ODBC, ODP.NET, OCI (Oracle Call Interface), Pro*C/C++. Соединения клиент/сервер автоматически распределяются между узлами для балансировки нагрузки.

База данных TimesTen Scaleout имеет свойства персистентности, восстановимости и высокой доступности. Персистентность обеспечивается комбинацией журналов транзакций и периодических обновлений версии базы данных на локальном диске на каждом узле. Высокая доступность достигается за счет механизма многократного копирования, база данных автоматически поддерживает несколько копий данных, располагая каждую копию на отдельном узле. Это позволяет базе данных обеспечивать полную доступность даже в случае одновременного сбоя нескольких узлов одновременно при условии, что оставшиеся узлы совместно содержат полную копию базы данных.

Централизованное управление и администрирование — ключевая особенность TimesTen Scaleout. Установка, настройка, управление и мониторинг выполняются из одной точки (через экземпляр управления). Это упрощает повседневные операции и снижает нагрузку на администратора базы данных. Основные функции управления включают в себя следующее:

- Установка и обновление программного обеспечения TimesTen.
- Конфигурирование и развертывание грида узлов с TimesTen.
- Создание, конфигурирование, администрирование базы данных.
- Создание резервной копии и восстановление базы данных.
- Расширение и уменьшение конфигурации TimesTen Scaleout (добавление и удаление узлов).

Label Security

Опция Oracle Label Security (OLS) представляет собой технологическое решение для организаций, которым необходим низкоуровневый, построчный контроль доступа для защиты конфиденциальной информации. Основанная на многоуровневой технологии безопасности, OLS позволяет сохранять в одной базе данных



информацию с разной степенью конфиденциальности, при этом доступ к данным ограничивается категориями допуска. Построчный контроль доступа дополняет безопасность, основанную на правах доступа к объектам, позволяя реализовать низкоуровневую модель привилегий.

OLS предоставляет доступ к строке в таблице базы данных, основываясь на метке, содержащейся в строке, метке, ассоциированной с каждой сессией, и привилегией, присвоенной сессии. Метки служат для классификации данных по уровням конфиденциальности. Так как данные классифицированы по уровням, каждый конкретный пользователь получает ограниченный доступ к данным. Он может оперировать только с данными, находящимися на том уровне конфиденциальности, который соответствует его статусу, и на уровнях ниже. Любой пользователь может в своем SQL-запросе потребовать выдать все записи из таблицы. СУБД проверит уровень конфиденциальности пользователя и в ответ на запрос возвратит только те строки, которые удовлетворяют условиям, сформулированным в запросе, и соответствуют уровню доступа этого пользователя. Создание новых таблиц путем копирования данных из таблиц, защищенных политиками безопасности OLS, невозможно.



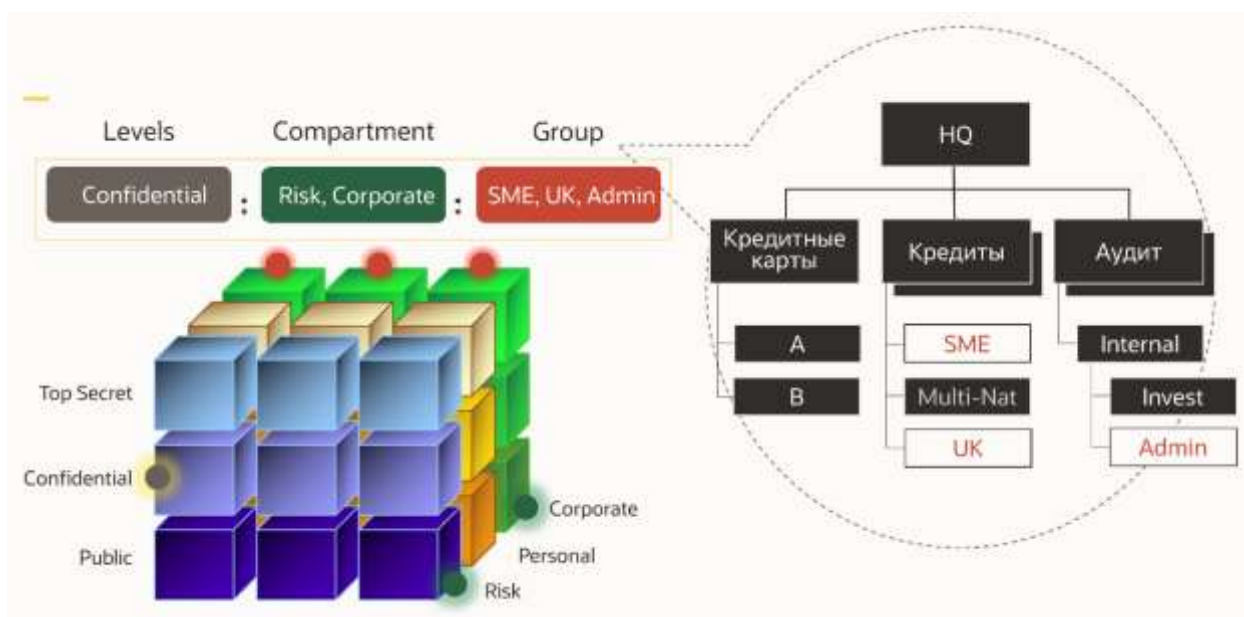
Важной является возможность формировать метку доступа более сложным путем (составная метка), чем на основе только уровня конфиденциальности информации. В состав метки могут быть включены еще компоненты: разделы данных (compartment), позволяющие сгруппировать данные по категориям с общим режимом доступа, например «административные данные», «финансовые данные», «операционные данные»; группы безопасности пользователей (security groups), позволяющие сгруппировать пользователей данных по принципу общих правил доступа, например «главное управление», «южное отделение», «северное отделение», «предприятие X», «предприятие Y». В отличие от разделов данных, группы пользователей могут формировать древовидную подчиненность. Начиная с версии 10.1 информацию об уровнях, разделах данных и группах пользователей стало возможным помещать в сервер имен OID/LDAP, что существенно для ценности самого подхода. В отличие от простых меток, соответствующих уровням конфиденциальности информации, составные метки имеют более сложные правила упорядочения и обеспечивают большую функциональность средств защиты информации. Составная метка может содержать до 10 000 значений для каждой из ее составляющих.

Компоненты меток данных

Oracle Label Security предоставляет возможность определять метки классификации данных в соответствии с конкретными бизнес- и организационными требованиями. Например, государственная организация может иметь такие уровни, как «для служебного пользования» и «конфиденциальный», в то время как коммерческая организация может иметь такие уровни, как «конфиденциальный» и «публичный».



Компоненты	Описание
Compartment	Является необязательным и иногда называется как «категория», не является иерархическим. Используется, как правило, для разделения данных по типу, области знаний, географии или проекта
Group	Является необязательным и похож на Compartment с несколькими исключениями. Каждая группа может иметь отношения «родитель-ребенок». Группы обычно используются для разделения данных по организациям
Level	Уровень - это иерархический компонент, который обозначает чувствительность (степень конфиденциальности) данных. Каждая метка данных должна иметь свой уровень. Организация может определить такие уровни, как конфиденциальный, конфиденциальный и высокочувствительный. Если не требуется несколько уровней, то необходимо определить один уровень по умолчанию.



Использование меток данных

Первым и наиболее важным шагом в планировании развертывания Oracle Label Security является определение требований к меткам данных вашей организации. Это означает определение того, какие метки данных или чувствительность вам необходимы для защиты вашей информации. Определение требований к меткам



данных обычно означает анализ вашего приложения и определение таблиц, которые вы планируете защитить с помощью Oracle Label Security. Это лучше всего сделать с помощью администратора приложения или разработчика, который знает схему приложения. В большинстве случаев только небольшой процент таблиц приложений требует применения политики безопасности OLS. После того как таблицы-кандидаты были определены, данные, содержащиеся в этих таблицах, должны быть оценены. Может потребоваться помощь аналитика данных или кого-то, кто разбирается в этих данных. Рекомендуется также учитывать потребности в будущих прикладных данных.

Multitenant опция

Новая архитектура СУБД Oracle позволяет снизить сложность сопровождения множества БД Oracle и повысить эффективность использования оборудования. В традиционной архитектуре СУБД Oracle, если мы создаем новую БД, то создается набор файлов этой БД на дисках и для работы с этой БД запускается экземпляр Oracle (Instance), который занимает часть оперативной памяти компьютера (SGA и PGA) и запускает набор фоновых процессов. Если на одном компьютере мы хотим создать 10 БД, у нас создается 10 наборов файлов на дисках, запускается 10 наборов фоновых процессов и в оперативной памяти выделяется 10 областей.

В случае архитектуры Multitenant создается одна контейнерная БД (CDB — container database), которую обслуживает один экземпляр Oracle. А все вновь создаваемые БД (они называются PDB — pluggable database, или подключаемые базы данных) помещаются в эту контейнерную БД. При этом один набор процессов и одна область оперативной памяти используются для обслуживания множества независимых БД. Причем эти PDB независимы и не видят друг друга (рис. 1). В такой архитектуре старый словарь БД разделяется на 2 части. Одна — общая для всех PDB — хранится в CDB, а в каждой PDB хранится информация словаря, специфичная для данной PDB. БД Multitenant имеет один набор журнальных файлов (redo logs) и один набор управляющих файлов, общих для всех PDB в контейнерной БД.



Рис. 1. Мультиарендная архитектура

Такой подход позволяет на одном компьютере разместить гораздо больше БД, то есть выполнить консолидацию БД, не сталкиваясь с проблемами дублирования и несовместимости объектов, которую мы имеем при консолидации схем БД. Было проведено тестирование: на одном и том же компьютере создали сначала базы данных старой архитектуры, а потом — новой архитектуры. Поместилось в 50 раз больше БД за счет более эффективного использования памяти, процессоров и дисков. Подключаемые БД (PDB) не зря называют pluggable и изображают в виде флэшки. Их можно легко отключить от одной CDB и подключить к другой. Например, при переносе PDB из CDB версии 12c в CDB версии 20c PDB автоматически обновляется до версии 20c.



Кроме консолидации БД, новая архитектура сильно упрощает администрирование множества БД. Если раньше ДБА (администратору БД) приходилось администрировать 10 БД и экземпляров, то, превратив эти БД в PDB, он будет администрировать только один экземпляр СУБД Oracle. Если раньше нужно было регулярно делать бэкап для каждой БД, то теперь можно делать один бэкап для всей CDB и из него восстанавливать нужную PDB. Если раньше для повышения надежности работы приложения мы создавали, конфигурировали и сопровождали резервную (standby) БД для каждой БД, то теперь можно создать одну резервную БД для CDB, и все ее текущие и вновь создаваемые PDB будут иметь standby БД. Кроме того, для CDB можно сконфигурировать кластер (RAC), и все ее PDB резко повысят свою надежность. Можно, кстати, закрепить отдельные PDB за конкретными узлами RAC, чтобы обеспечить изолированность и масштабируемость. В CDB очень просто делать клоны PDB, существующих в этой или другой CDB, причем клоны могут обновляться по мере обновления исходных PDB.

Ну и конечно, упрощаются процедуры обновления БД (как смена версий, так и установка отдельных обновлений). Вместо 10 процедур для 10 БД, мы можем обновить CDB, и все ее PDB автоматически обновятся до версии CDB. Если же обновление требуется не для всех PDB, а лишь для части, то их можно перенести в CDB новой версии. Перенос PDB из одной CDB в другую очень прост. Одной командой (или мышкой в Enterprise Manager) выгружается в xml-файл метаданные о PDB, и затем эта метаданные загружаются в другую CDB. Если эти CDB размещаются на одном компьютере, то даже копировать файлы БД PDB не требуется. Можно не только переместить PDB в другую CDB, но и сделать в новой CDB клон PDB из другой CDB. При этом можно сделать так называемый hot refreshable клон, поскольку во время клонирования исходная PDB открыта для изменений, а после окончания клонирования эти изменения применяются к клону, т. е. клон и исходная PDB синхронизируются. И далее эта синхронизация периодически повторяется в автоматическом или ручном режиме. Таким образом, мы всегда имеем свежую копию мастер-клона в новой CDB, открытую для чтения, и можем делать открытые для изменения новые клоны PDB в этой CDB. Это очень удобно для разработчиков приложений.

Итак, у нас есть исходная PDB и ее синхронизирующийся клон в другой CDB. Это некоторый вариант резервной БД, похожий на standby. И так же, как при БД standby, мы можем переключить PDB-клон в режим основной открытой для изменений PDB, а ее исходная PDB превратится в доступный для чтения обновляемый мастер-клон. То есть происходит переключение (**PDB switchover**) — смена ролей.

Разработчикам и тестировщикам приложений часто бывает нужно восстанавливать свою БД на момент времени в прошлом, это можно делать с помощью механизма flashback (можно делать flashback отдельной PDB), но можно применить новый механизм — **карусель снапшотов** (snapshot carousel). Если для PDB включен режим карусели, Oracle автоматически ежедневно будет делать копию этой PDB и хранить ее в архивном файле (такая копия называется снапшотом). По умолчанию хранятся последние 8 снапшотов. Если, например, в среду необходимо восстановить PDB на состояние 5 часов вечера понедельника, то мы просто восстанавливаем PDB из снапшота за понедельник и далее накатываем архивные журналы, чтобы применить изменения, сделанные до 5 вечера.

Еще один интересный механизм мультиарендной БД — **Application Container (AC)**. Если несколько PDB имеют одинаковые объекты (таблицы, процедуры, функции и т. д.), то их можно поместить в отдельную PDB, называемую application container. Все PDB, наследующие объекты этого контейнера, будут видеть эти объекты. Таким образом, мы убираем дублирование и облегчаем сопровождение этих объектов (они изменяются в одном месте — application container). Причем, если таким разделяемым объектом является таблица, то у вас есть 3 варианта:

- Хранить таблицу со всеми данными в AC (тогда PDB будут видеть ее как таблицу, открытую на чтение).
- Хранить таблицу и часть ее данных в AC (тогда каждая PDB будет видеть эти данные в режиме чтения, но может иметь свою открытую на изменение часть этой таблицы).
- Хранить только описание структуры таблицы в AC (тогда каждая PDB будет иметь свой, скрытый от других, открытый для изменений вариант этой таблицы).



PDB-базы изолированы и не видят друг друга, администратор БД может использовать менеджер ресурсов, чтобы управлять разделением ресурсов компьютера (память, процессоры, ввод/вывод, параллелизм) между БД. С помощью механизма lockdown profiles можно ограничить для PDB выполнение отдельных команд SQL и их частей, запретить выполнение некоторых потенциально опасных команд (например: alter system), запретить выполнение команд операционной системы (ОС) и даже запретить прямой доступ к этой PDB.

Однако серверные процессы ОС имеют доступ к файлам БД и могут читать или модифицировать файлы чужих PDB. Чтобы избежать этого, в версии 20c вводится механизм **DB Nest**. Все процессы ОС экземпляра Oracle можно разделить на две группы: фоновые процессы и серверные процессы (они обслуживают сессии и SQL отдельных пользователей). DB Nest позволяет запретить серверным процессам конкретной PDB доступ к файлам БД, файлам трассировки, файлам настройки других PDB, им также можно запретить доступ к областям памяти (pga) других PDB и выполнение команд ОС. Механизм очень похож на механизм контейнеризации в ОС, каждая PDB со своими серверными процессами и файлами как бы работает в отдельном контейнере и изолирована от других PDB.

Поскольку теперь с PDB можно работать как с обычной традиционной БД, у нее есть и традиционные средства настройки БД. Она имеет свой AWR, можно запускать ADDM и строить AWR-отчеты уровня PDB, получать рекомендации по настройке БД. Можно запускать RAT на отдельной PDB, чтобы захватить, а потом и воспроизвести нагрузку и оценить влияние изменений на эту PDB.

Мультиарендная архитектура доказала свои преимущества, на ней построены все автономные БД Oracle, и с версии 20c Oracle будет поддерживать только эту новую архитектуру. Те, кто хочет по-прежнему иметь один экземпляр Oracle для каждой БД, могут создавать CDB с одной PDB. Бесплатно можно создавать до 3 PDB в одной CDB, если нужно больше — следует лицензировать опцию Multitenant.

OLAP

Опция Oracle OLAP предназначена для построения аналитических систем, основанных на принципах многомерного анализа и технологии OLAP. Этот компонент позволяет хранить и обрабатывать в одной и той же базе данных не только реляционную, но и многомерную информацию.

Основой опции является полномасштабная многомерная модель данных, с помощью которой можно определять и работать с такими понятиями, как измерения, иерархии, многомерные показатели и т. п. При этом не требуется установка специального многомерного сервера, так как вся метainформация и сами многомерные данные хранятся и обрабатываются в СУБД Oracle. Создание и управление многомерными данными осуществляется с помощью визуального конструктора OLAP кубов — Analytic Workspace Manager или с помощью специального API, реализованного в PL/SQL пакете.

Важной отличительной особенностью опции OLAP является то, что доступ к многомерным данным осуществляется посредством языка SQL. Это делает процесс разработки приложений более гибким и простым, благодаря возможности использования универсального доступа к реляционным и многомерным данным одновременно. Кроме того, существует возможность использования многомерных кубов в качестве материализованных представлений, а функция query rewrite делает доступ приложений к многомерным данным абсолютно прозрачным. Это позволяет начать использовать возможности OLAP без какого-либо изменения уже существующих клиентских приложений.

Встроенные в Oracle Database средства OLAP обеспечивают высокий уровень надежности, масштабируемости и производительности при одновременной работе большого числа пользователей. Важным преимуществом полной интеграции технологии OLAP с реляционным сервером баз данных Oracle является единый подход к администрированию как реляционных, так и многомерных баз данных, включая управление пользователями, ограничение прав доступа, оптимизацию выполнения запросов, использование кластерных технологий, средств распараллеливания, секционирования и управления агрегированными данными.



Partitioning

Опция Partitioning необходима для работы с большими базами данных, например, хранилищами данных. Использование этой опции позволяет повысить производительность, надежность и управляемость приложений, работающих с большими таблицами. Идея механизма секционирования (партиционирования) состоит в том, что таблица физически разбивается на отдельные сегменты-секции (или подсекции для комбинированных схем секционирования) в соответствии с некоторым условием — ключом секционирования. Ключом секционирования может быть, например, значение в поле даты или комбинация значений в каких-либо столбцах таблицы. При этом на логическом уровне таблица остается единой, монолитной, а на физическом уровне с каждой такой секцией система работает независимо.

Каждая индивидуальная секция имеет собственное имя и может иметь собственные характеристики хранения, такие как режимы сжатия или указание на табличное пространство, в котором будет храниться данная секция. Поэтому, с точки зрения администратора базы данных, секционированный объект состоит из нескольких частей, которыми можно управлять как вместе, так и каждым в отдельности. Это значительно упрощает обслуживание крупных объектов хранения в базе данных и предоставляет администратору значительную свободу в выборе оптимальных схем хранения таких объектов.

Однако с точки зрения приложения секционированная таблица выглядит как обычная несекционированная, и для доступа к ней не требуются какие-либо специальные модификации текста SQL-предложений. Кроме того, возрастает надежность системы, так как уменьшается влияние сбоев в отдельных секциях на работу всего приложения.

Применение секционирования упрощает задачи администрирования крупных объектов хранения. Каждая из секций может быть индивидуально переведена в онлайн или офлайн. Ее можно копировать и восстанавливать при сбое независимо от других секций. Можно импортировать, экспортировать или загружать данные только в требуемые секции, уменьшая тем самым время, требуемое для выполнения таких операций. Операции с секциями могут выполняться параллельно. Для каждой секции таблицы может быть построен индивидуальный индекс, что также сокращает время, необходимое для выполнения операций с данными и для сопровождения самих индексов. Возможны разнообразные локальные и глобальные стратегии индексирования, причем некоторые секции (например, секции с архивными данными) могут не иметь локальных индексов или не быть включенными в глобальный индекс.

При секционировании сокращается время, требуемое для выполнения большинства операций с таблицами. Это уменьшение достигается путем применения этих операций к отдельным секциям крупных таблиц и увеличения производительности вследствие их параллельного выполнения. Так, например, запрос к большой исходной таблице автоматически заменяется запросами к отдельным ее секциям, и эти запросы выполняются одновременно и независимо друг от друга, уменьшая тем самым общее время обработки.

Если таблица велика, то административные операции над всей таблицей могут не уложиться в отведенное для них время, операции же над секциями выполняются гораздо быстрее. Если в операции соединения таблиц (Join) используются секционированные таблицы, то оптимизатор запросов применяет умный алгоритм (wise join) при котором соответствующие секции двух таблиц соединяются в параллельном режиме. При выполнении запроса над секционированной таблицей оптимизатор пропускает секции, не соответствующие критерию where запроса. Все это ускоряет выполнение запросов.

Допускается несколько методов секционирования:

- секционирование по диапазону (range partitioning);
- секционирование по интервалу (interval partitioning);
- секционирование по списку значений (list partitioning);
- секционирование хешированием (hash partitioning);



- секционирование по ссылке (REF partitioning);
- комбинированное секционирование (composite partitioning);
- секционирование по виртуальному столбцу (virtual column based partitioning).

Секционирование по диапазону или интервалу позволяет задать разбиение на секции по попаданию в диапазон значений некоторого поля. При секционировании по списку значений секция определяется перечислением значений соответствующего поля, причем есть возможность использовать функцию Auto-List Partitioning, с помощью которой при появлении в ключевом поле нового значения автоматически создается дополнительная секция. Секционирование хешированием производится с помощью некоторой хеш-функции, что позволяет равномерно распределить данные по секциям. Ссылочное секционирование позволяет сходным образом секционировать связанные между собой таблицы, при этом поддерживается интервальное секционирование родительской таблицы.

При комбинированном секционировании разрешается использовать сочетание различных методов, например, можно разбить исходную таблицу на секции по диапазону значений одного поля, а каждую такую секцию дополнительно разбить на подсекции по списку значений в другом поле. Секционирование по виртуальному столбцу дает возможность секционировать данные на основе результата выполнения определяемой разработчиком функции. Данные XML- Type- и LOB-столбцов, которые хранят бинарные и большие текстовые данные в отдельных физических сегментах, также разбиваются на секции согласно ключу секционирования таблицы. Partitioning on External Tables позволяет секционировать также и внешние таблицы, данные которых располагаются не в табличных пространствах БД, а в файлах на файловой системе. Технология Hybrid partitioning позволяет комбинировать секционирование таблицы и секции на основе внешних таблиц.

Возможность секционирования таблиц и индексов базы данных при использовании опции Partitioning позволяет администраторам делить большие таблицы на более управляемые части меньшего размера и выполнять операции обслуживания по отдельным секциям. Кроме того, большое количество операций над секциями, такие как слияние, разделение, удаление и очистка (truncate), смена метода секционирования и даже создание секционированной таблицы на основе обычной таблицы, можно производить в режиме online, без остановки. После удаления или очистки партиции глобальные индексы на таблице не переводятся в недействительное состояние, а производится их перестройка в фоновом режиме. DML-операции при выполнении этих операций не блокируются и продолжают свою работу.

Real Application Clusters

Технология Real Application Clusters (RAC) (и одноименная опция) позволяет объединить несколько независимых серверов (узлов кластера) в единую отказоустойчивую, хорошо масштабируемую вычислительную систему. При этом база данных одновременно обслуживается сразу несколькими экземплярами СУБД Oracle, каждый из которых работает на отдельном узле кластера. По существу, с точки зрения приложения — это единая СУБД. Таким образом, на все приложения, работающие на подобной системе, распространяются следующие преимущества.

- Гибкость и экономическая эффективность при планировании мощностей, так как мощность системы может быть изменена до требуемой величины при изменении требований бизнеса (сессии «размазаны» по узлам кластера, и новые узлы можно добавлять «на лету»).
- Отказоустойчивость приложений против системных сбоев (при выходе из строя узла кластера БД доступна через другие узлы, прерванная транзакция может повториться через оставшийся узел, а сессии переезжают на оставшиеся узлы).
- Постепенный рост системы (не нужно сразу покупать большой, мощный и дорогой компьютер), узлы кластера добавляются постепенно и увеличивают мощность системы.



- Создание GRID-архитектуры, возможность собрать из нескольких небольших узлов кластер, превышающий возможности мощного коммерческого сервера.

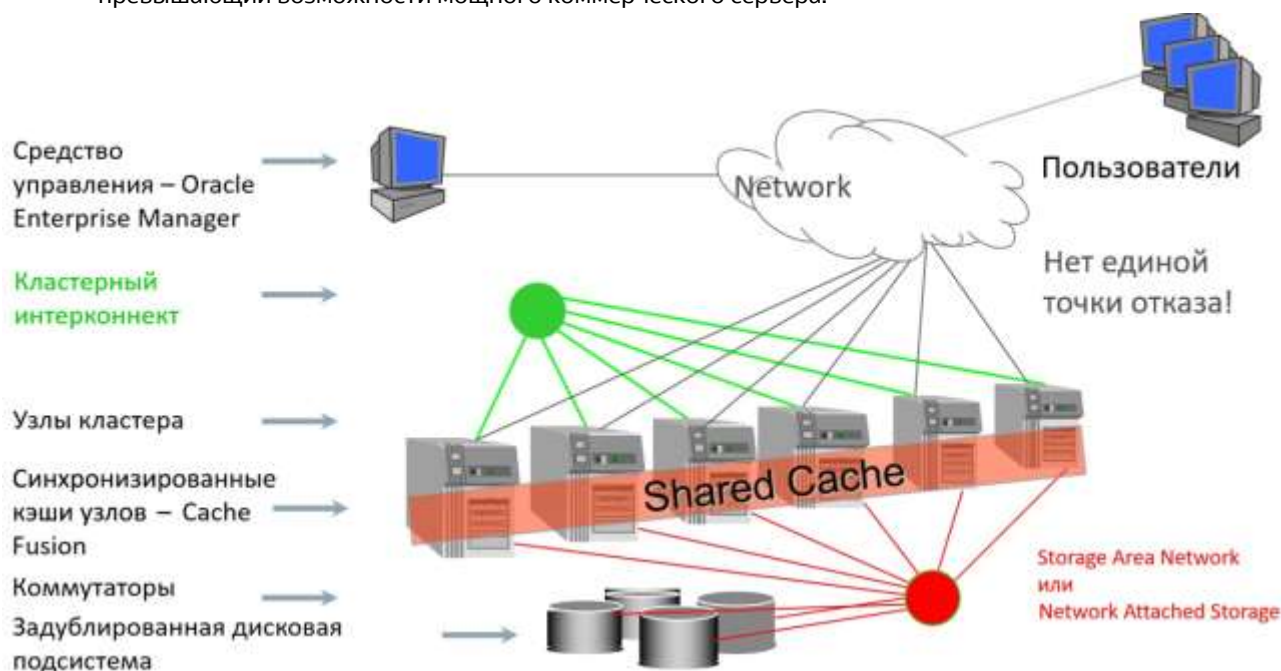


Рис. 1. Архитектура Oracle Real Application Cluster

Решение Real Application Clusters используется для построения корпоративных Grid-систем. Корпоративные Grid-системы строятся из большого количества стандартизованных недорогих компонентов: процессоров, серверов, сетевых устройств и устройств хранения данных. RAC — единственная технология, позволяющая объединить все эти компоненты в эффективную вычислительную систему для всего предприятия. Real Application Clusters и Grid-технологии дают возможность радикально снизить эксплуатационные затраты и обеспечить новый уровень гибкости, делая корпоративные системы более адаптивными, проактивными и динамичными. Возможности динамического изменения аппаратной конфигурации кластера, например добавление или удаление узлов, позволяют быстро и эффективно гарантировать необходимые уровни сервиса при одновременном снижении затрат за счет лучшего использования ресурсов. Кроме того, среда RAC полностью прозрачна для приложения, работающего с кластерной базой данных.

Real Application Clusters дает пользователям возможность добавлять в кластер новые узлы при возрастании требований к ресурсам, производить постепенное увеличение мощности системы при оптимизации затрат и исключить необходимость замены небольших систем более мощными.

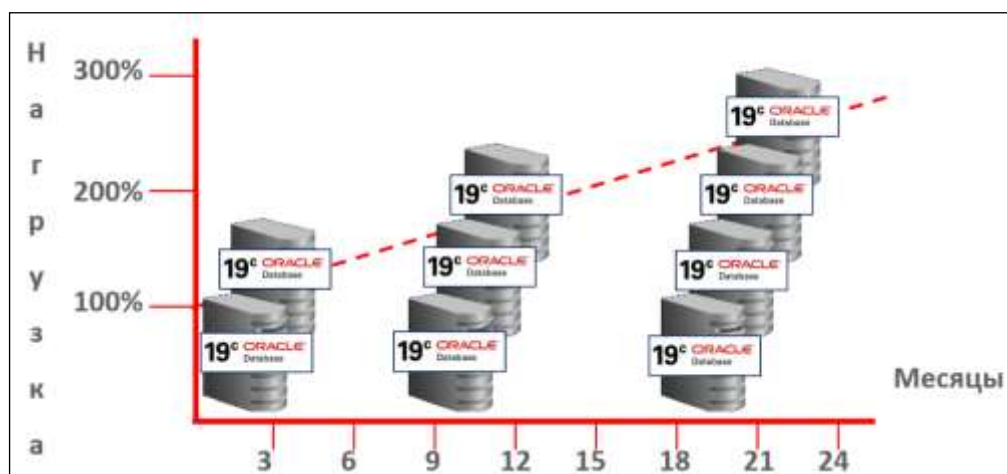


Рис. 2. Постепенная оплата и расширение в RAC



Каждый новый узел кластера обеспечивает дополнительные вычислительные ресурсы для обработки данных, в том числе поддерживается параллельное выполнение запросов между узлами кластера, конвейерный параллелизм, и тем самым обеспечивается масштабируемость сервера базы данных. В случае сбоя одного из узлов кластера программное обеспечение RAC автоматически переносит все сессии на другой узел.



Рис. 3. Автоматическое переключение пользователей после сбоя узла

До версии Oracle Database 12c, в случае если приложение в момент сбоя имело активную транзакцию (модифицировало данные), то это приводило к откату данной транзакции. Начиная с версии Oracle Database 12c появилась технология Application Continuity. Application Continuity — это технология, обеспечивающая автоматический повтор транзакции в случае сбоя соединения с кластерной БД Real Application Cluster.

Технология Application Continuity позволяет решить проблему восстановления активных транзакций при сбое.

Application Continuity задействует технологию Transaction Guard. Transaction Guard — это технология для определения факта фиксации транзакции (или НЕфиксации) после сбоя.



Рис 4. Application Continuity автоматически повторяет транзакцию после сбоя



Application Continuity реализована посредством специального драйвера повтора (Replay Driver) на стороне клиента. После сбоя, с помощью Transaction Guard, этот драйвер проверяет факт фиксации транзакции (успели ли выполниться оператор commit), и в случае, если транзакция не была зафиксирована, производится ее повтор. Если транзакция успела зафиксироваться, то никаких действий не производится и приложение продолжает работу.

В случае если повтор транзакции требует восстановления контекста сессии (например, в транзакции есть вызовы PL/SQL, которые используют глобальные переменные пакета), разработчик может определить свою функцию обратного вызова (callback), которая будет вызвана перед повтором транзакции. В случае невозможности повтора транзакции (например, между DML-операторами происходила запись в файл с помощью вызова пакета UTL_FILE) разработчик может явно запретить повтор такой транзакции в коде приложения.

Для поддержки сохранения значений последовательностей (если перед сбоем был вызов функции получения следующего значения — nextval, повтор транзакции ошибочно увеличит счетчик последовательности) введена возможность сохранения их значений для повтора. Аналогичная возможность есть для сохранения значений результатов вызовов функций SYSGUID (получение уникального глобального идентификатора) и SYSDATE (получение текущего системного времени).

Oracle RAC дает возможность радикально снизить эксплуатационные затраты и обеспечить новый уровень гибкости, делая корпоративные системы более динамичными и адаптивными. Динамическое обеспечение приложений процессорным ресурсом, оперативной памятью и устройствами хранения позволяет гарантировать необходимые уровни сервиса при одновременном снижении затрат за счет лучшего использования ресурсов.

Real Application Clusters One Node

Введение

RAC One Node иногда называют «временный RAC» или «RAC на короткое время для плавного переезда на другой узел». Oracle RAC One Node представляет собой своеобразный одноузловой кластер Real Application Cluster. Эта технология обеспечивает защиту как от внеплановых простоев, вызванных сбоем ПО или оборудования, так и от плановых простоев, вызванных необходимостью проведения регламентных работ по обслуживанию ИТ-инфраструктуры (обновление системного и прикладного ПО, изменение аппаратной конфигурации сервера, обновление сетевых настроек и т. д.). Основной особенностью RAC One Node является плавное перемещение экземпляра БД на другой узел, без остановки приложения (с возможностью плавного перемещения сессий на другой узел). После того как сессии с первого узла были переподключены на второй узел, экземпляр на первом узле может быть остановлен и на нем могут быть проведены регламентные работы (например, установлены патчи или обновлено оборудование, в том числе также могут быть обновлены бинарные файлы ПО Oracle). После завершения регламентных работ экземпляр аналогичным образом может быть возвращен на первый узел.

Технология работы

Рассмотрим пример конфигурации, в которой имеется пять одиночных экземпляров (Single Instance), запущенных на кластере из 3-х серверов. На каждом сервере запущена одна операционная система. На первом и третьем серверах запущено по две базы (вернее: запущено два экземпляра БД), которые консолидированы в одной ОС.



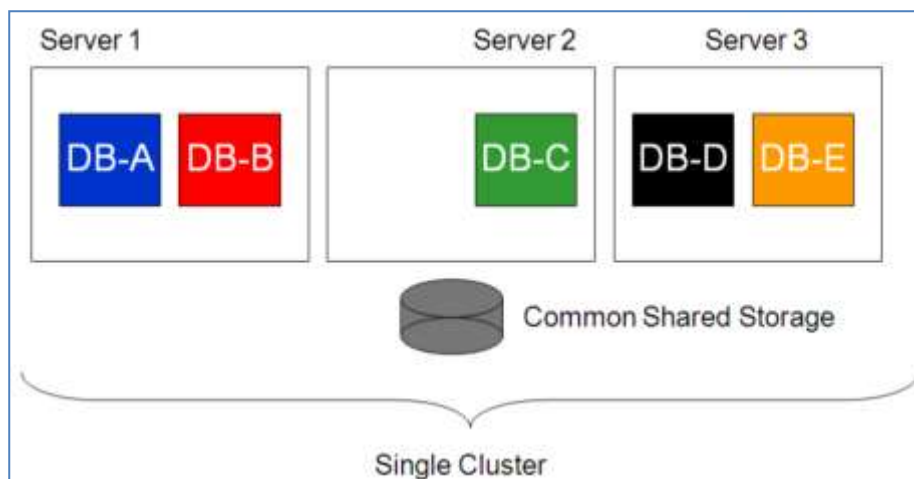


Рис. 1. Пример схемы защиты пяти БД с помощью RAC One Node

Перемещение экземпляра БД (например, БД DB-B) на другой узел начинается с того, что на втором узле, выбранном для работы БД RAC One Node, запускается второй экземпляр БД DB-B. На короткий период времени, который задается в параметрах перемещения, БД переходит в кластерный режим «Active-Active» — оба экземпляра активны на двух узлах.

В течение этого периода времени происходит следующее:

- после завершения текущих активных транзакций сессии на первом узле закрываются, и новые сессии открываются только на втором узле;
- при переключении сессии приложение может обработать события переключения сессии с помощью технологий Transparent Application Failover или Fast Application Notification, либо это вообще может произойти в автоматическом режиме с помощью технологии Application Continuity;
- все новые сессии направляются на второй узел.

По окончании периода переключения происходит останов экземпляра БД на первом узле. При этом сессии, которые не успели завершить свои активные транзакции, принудительно закрываются и переводятся на второй узел.

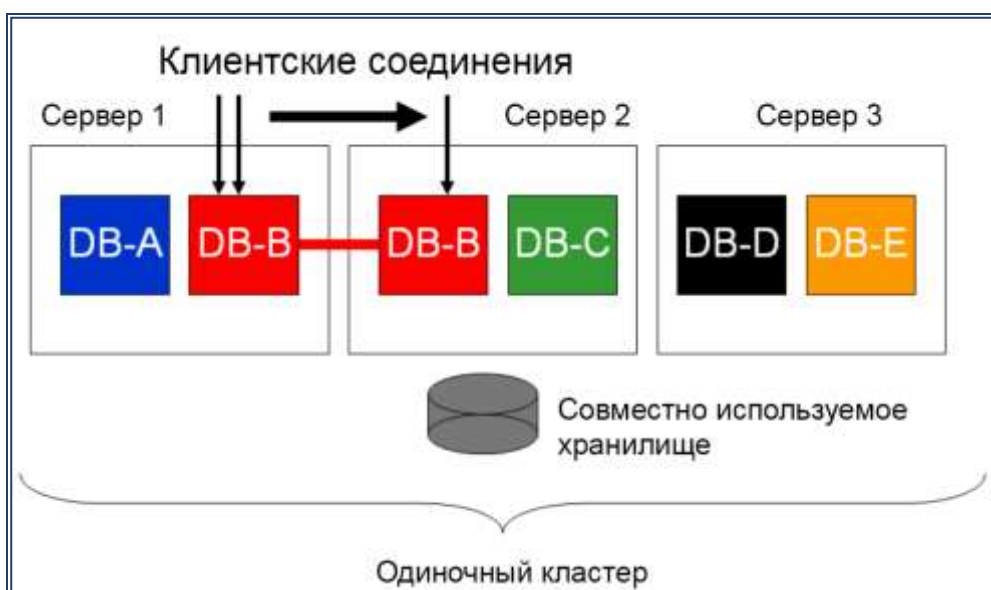


Рис. 2. БД временно переходит в режим работы «Active-Active» в кластере



После остановки экземпляра на первом сервере можно проводить регламентные работы, например: обновить бинарные файлы ПО Oracle Database. Поскольку поддерживается перемещение экземпляра БД на узлах с бинарными файлами, которые имеют разный уровень обновления (patch level), технология RAC One Node обеспечивает полностью прозрачную для пользователей установку патчей.

В случае, если необходим полный останов аппаратного сервера первого узла, точно так же следует переместить экземпляр DB-A на другой узел кластера.

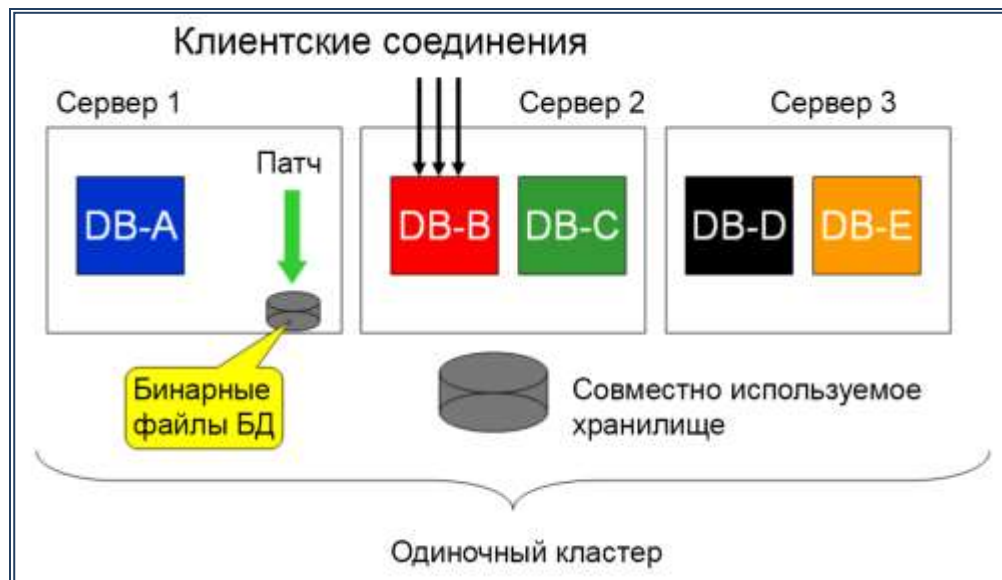


Рис. 3. Обновление бинарных файлов БД после перемещения экземпляра на другой узел

После окончания обновления бинарных файлов ПО Oracle Database на первом узле производится возвращение экземпляра DB-B на первый узел: по команде администратора происходит «мягкое» перемещение с первого узла на второй по аналогичной схеме.

Защита от сбоев

Oracle RAC One Node также обеспечивает защиту от сбоя БД и автоматический перезапуск БД на работоспособном узле кластера.

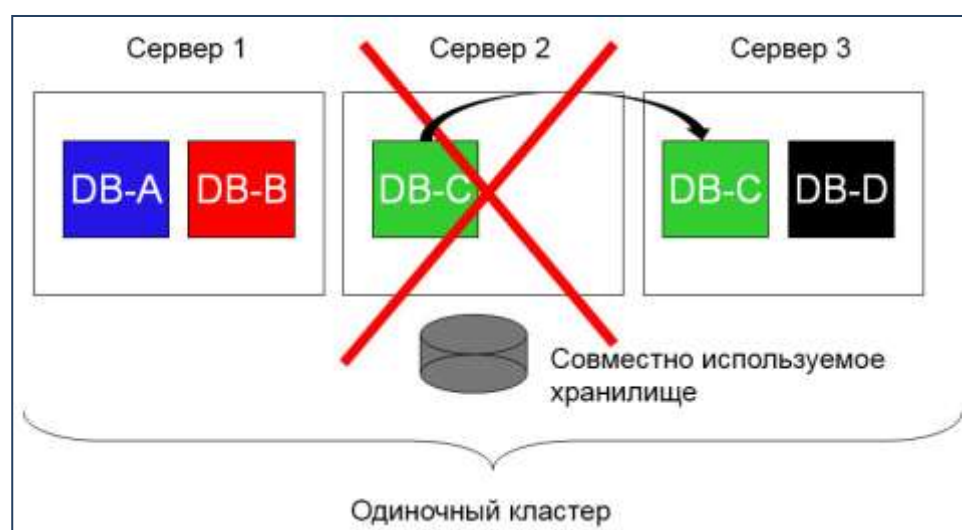


Рис. 4. Защита от сбоя узла с помощью RAC One Node



На *Рис. 4* изображена ситуация отказа второго узла, и экземпляра БД, который работал на втором узле, автоматически был перезапущен на третьем узле.

Oracle RAC One Node интегрирован с Oracle Grid Infrastructure, которое производит непрерывный мониторинг работоспособности экземпляра БД и обеспечивает доступность сервиса. В случае возникновения сбоя в работе экземпляра (например, из отказа оборудования) Oracle Grid Infrastructure определяет факт сбоя и рестартует экземпляр БД на отказавшем сервере. Если это не удастся, то происходит перезапуск экземпляра на другом узле.

По сравнению с другими поставщиками решений для «холодного» переключения («Cold Failover Cluster») Oracle RAC One Node обеспечивает полностью автоматическое восстановление работоспособности БД с полным временем восстановления от 5 до 30 минут. Это идеальное решение для приложений, которые допускают короткое время простоя, но для которых неприемлем длительный простой, связанный с длительным ожиданием времени, в течение которого администраторы определяют факт наличия проблемы в работе БД и примут меры для их устранения. Для приложений, которые вообще не приемлют простоя в работе, рекомендуется использовать Oracle RAC.

Заключение

Oracle RAC One Node обеспечивает постоянную работоспособность одиночного экземпляра БД (Single Instance), защищая как от незапланированных, так и от плановых простоев.

Oracle RAC One Node полностью поддерживается в среде виртуализации Oracle VM и KVM и может быть использован в виртуальных средах со всеми преимуществами виртуализации.

Заказчики, которым необходимо улучшить доступность, уменьшить время простоя и упростить сопровождение своих БД, найдут Oracle RAC One Node идеальным решением.

RAC One Node намного дешевле, чем RAC, и его достаточно купить на 1 узел кластера (где расположена переезжающая БД), а не на 2 и более, как в случае RAC.

Real Application Testing

Real Application Testing (RAT) является уникальным решением для высококачественного тестирования Баз Данных Oracle. RAT использует уникальную оптимизированную технологию записи/воспроизведения реальной нагрузки на БД, что позволяет охватить практически все сценарии тестирования. Типичными примерами таких сценариев являются: обновления системного программного обеспечения — ОС, переход на новую версию СУБД Oracle, изменения на аппаратном уровне, — новый сервер или система хранения данных, установка рекомендуемых исправлений patches системного ПО, изменения параметров СУБД или статистики оптимизатора, миграция на платформу машины БД Exadata, добавление/удаление индексов, внедрение table partitioning, переход к архитектуре Real Application Cluster, консолидация баз данных на сервере, в частном или публичном Облаке или в контейнерной БД.

В состав Real Application Testing входят три модуля:

- **SQL Performance Analyzer (SPA)** для оценки влияния изменений на производительность и эффективность отдельных SQL-запросов. Детально анализируются и сравниваются планы и статистика выполнения запросов «до» и «после»;
- **Database Replay** для полного интеграционного тестирования при помощи захвата реальной нагрузки на продуктивной базе данных с последующим воспроизведением в тестовой БД с целью оценки влияния изменений на общую производительность;



- **Database Consolidation Workbench** для планирования и оценки различных сценариев консолидации баз данных на основе исторического анализа нагрузок. Позволяет минимизировать ошибки, автоматизируя все этапы консолидации от планирования до развертывания.

Database Replay, SQL Performance Analyzer (SPA) и Database Consolidation Workbench вместе образуют гибкое решение для тестирования баз данных, они позволяют полностью и всесторонне оценить влияние изменений на производительность БД и предотвратить возможные негативные последствия до выхода в рабочий режим эксплуатации. Управление сценариями тестирования Real Application Testing доступно как через графический интерфейс Oracle Enterprise Manager, так и с помощью командной строки с использованием PL/SQL API.

SQL Performance Analyzer (SPA)

Внесение изменений в БД, таких как, например, обновление версии ПО, установка патчей, изменение режима или параметров сбора статистики оптимизатора, создание/удаление индексов, безусловно, может влиять на планы выполнения SQL-запросов, что часто приводит к серьезным проблемам с производительностью и стабильностью систем. Следовательно, возможность детальной оценки статистики выполнения запросов после внесения изменений очень важна. Именно для этих целей создавался SQL Performance Analyzer (SPA). SPA позволяет проводить тестовые сценарии, где при помощи интуитивного интерфейса автоматически выполняется и тестируется каждый SQL-запрос — до и после внесения изменений. По итогам выполнения представляется подробный отчет, показывающий детальную статистику по SQL-запросам. Из отчета можно выяснить, производительность каких запросов осталась на прежнем уровне, какие стали выполняться эффективнее, а какие показали ухудшение. Функциональность SPA интегрирована с другими решениями Oracle для управления производительностью базы данных: SQL Tuning Advisor и SQL Plan Management; таким образом, SPA значительно упрощает и автоматизирует сложный, рутинный и продолжительный процесс поиска и устранения проблем с производительностью, даже в системах с большой нагрузкой и сотнями тысяч SQL-запросов. Также SPA имеет специальный режим Quick Check, который полностью автоматизирует тестовый сценарий и позволяет периодически проводить тестирование эталонного набора SQL-запросов. **SPA Quick Check** автоматически выполняет четыре тестовых прогона, выявляя на первой стадии запросы, у которых из-за изменений меняются планы выполнения, и вторым шагом сравнивая производительность этих запросов. **SPA Quick Check** может использоваться для быстрой проверки влияния рекомендаций SQL Tuning Advisor, применения SQL Profiles, изменений статистики оптимизатора, изменений параметров БД.

Для облегчения работы SPA имеет готовые преднастроенные тестовые сценарии, которые с помощью экранов wizard полностью проведут администратора через все необходимые шаги процесса тестирования; примеры тестовых сценариев: DB Upgrade, Parameter change, Exadata simulation, Optimizer Statistics и т. д.

В новых версиях SPA позволяет сравнивать не только производительность, но и корректность результатов запросов (result set validation), что позволяет выявлять ошибки, при которых один и тот же запрос может выдавать разные результаты.

Отчеты SPA оценивают влияние изменений как на общую нагрузку в целом, так и на отдельные SQL-запросы. Также имеется возможность автоматической настройки тех запросов, в которых SPA выявил снижение производительности. Автоматическая настройка не требует изменений на уровне кода приложения.



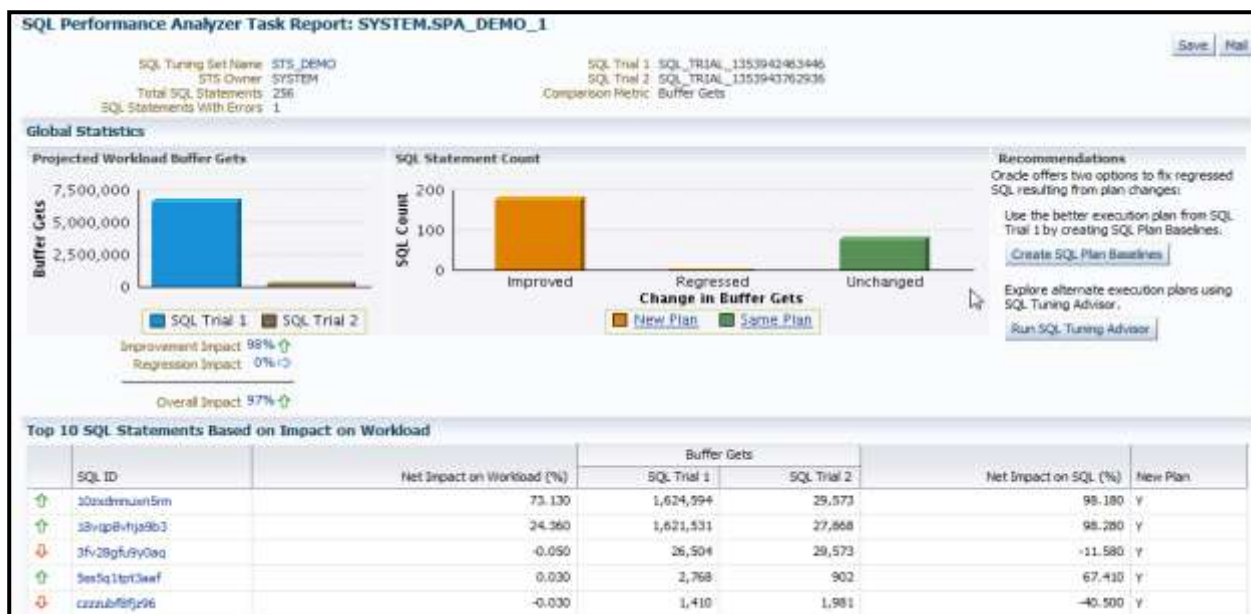


Рис. 1. Пример отчета SQL Performance Analyzer.

Database Replay

Следующий компонент продукта Real Application Testing носит название Database Replay, он предоставляет возможность захвата реальной нагрузки на базу данных продуктивной системы и последующего ее воспроизведения в тестовой среде. Нагрузка на БД записывается с сохранением всех характеристик: запросы, DML, вызов процедур, user calls, порядок транзакций, временные интервалы, конкуренция, количество открытых сессий, connect/disconnect. Возможность полного воспроизведения нагрузки на тестовой базе данных обеспечивает достоверную и всестороннюю оценку влияния изменений на производительность БД, включая возможную деградацию производительности или конкуренцию за ресурсы. По итогам воспроизведения нагрузки создаются подробные отчеты, показывающие расхождение по различным категориям показателей производительности между продуктивной и тестовой базами данных и об ошибках, возникших при воспроизведении. Таким образом, применение Database Replay сокращает сроки тестирования системных изменений БД с месяцев до дней.

Database Replay «умеет» корректно работать с любыми типами нагрузки и обеспечивать точное синхронное воспроизведение вызовов (user calls), как при использовании SQL, так и PL/SQL-кода, например, специальный расширенный (Extended) режим захвата нагрузки работает с отдельными вызовами внутри PL/SQL-процедур, позволяя соблюдать точные временные интервалы при воспроизведении.

Захват нагрузки Database Replay производится на уровне базы данных, следовательно, может быть использован для тестирования любых типов системных изменений БД, таких как:

- Database Upgrade, установка патчей, изменение параметров, optimizer statistics, индексов и т. д.
- изменение конфигурации, например миграция с single instance на RAC и/или переход на ASM;
- изменения на уровне системы хранения данных;
- миграция на платформу Exadata;
- изменения на уровне ОС, аппаратного обеспечения (миграции, установка патчей, изменение параметров);
- миграция в Контейнерную БД (Multitenant CDB).

Database Replay имеет различные режимы масштабирования нагрузки (Workload Scale-Up), такие как:



- Consolidated Replay, при котором происходит одновременное воспроизведение нагрузок на разные БД, что особенно полезно для оценки производительности при сценариях миграции отдельных баз данных в общую контейнерную БД;
- Time Shifting и Workload Folding — возможность совмещать и «вырезать» наиболее интенсивные фрагменты нагрузки и «накладывать» друг на друга при одновременном воспроизведении, что позволяет организовать эффективный стресс-тест.

Тестирование с помощью Database Replay состоит из следующих основных этапов.

- Захват нагрузки (**Capture**) — перехватывает и записывает все обращения к БД со стороны пользовательских сессий, сохраняется вся необходимая информация, такая как: текст SQL-предложения, используемые переменные, транзакционная информация, временные характеристики и т. д. Запись нагрузки производится в виде платформонезависимых бинарных файлов.
- Обработка нагрузки (**Preprocessing**) — пакетное задание для анализа файлов нагрузки и формирования метаданных, необходимых для воспроизведения. Preprocessing выполняется один раз и используется для многократного воспроизведения нагрузки.
- Воспроизведение нагрузки (**Replay**) — непосредственное применение нагрузки в виде запуска последовательности вызовов к тестовой БД с соблюдением всех временных характеристик, т. е. создание идентичной с промышленной нагрузочной среды. Для обеспечения максимальной корректности теста Replay необходимо иметь физическую копию продуктивной БД, восстановленную на момент начала захвата нагрузки.
- Анализ результатов (**Analysis & Reporting**) — по итогам воспроизведения нагрузки предоставляется детальная информация для оценки результатов теста БД: сравнительные диагностические отчеты, отчеты об ошибках. Отчеты включают в себя как стандартные диагностические за период «Compare Period AWR» и «Compare Period ADDM», так и специальные отчеты RAT Database Replay, в которых приводится ключевая информация об основных показателях тестирования: расхождение при воспроизведении user calls, запросы, по которым выявлена наибольшая разница в производительности (DB Time), запросы, у которых выявлены разные количества строк результата при захвате и воспроизведении, сравнительное количество ошибок и т. д.:

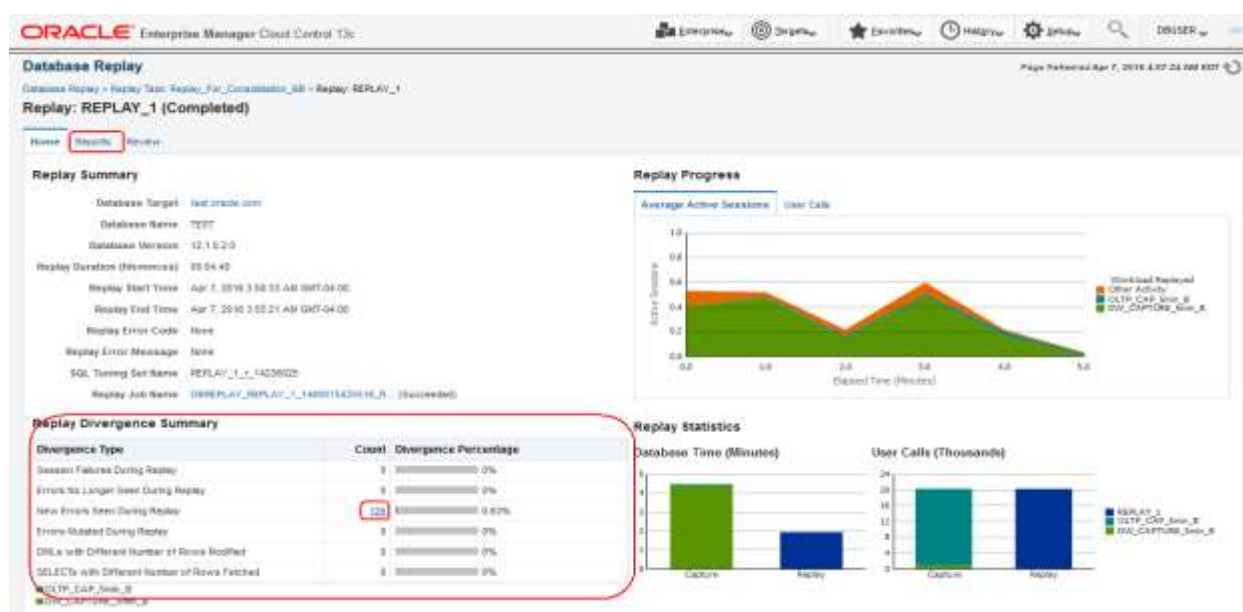


Рис. 2. Пример экрана Enterprise Manager — Database Replay Summary



Database Consolidation Workbench — модуль в составе Real Application Testing, позволяющий планировать различные сценарии консолидации БД. По мере того как серверы становятся все более мощными, у многих заказчиков с большим количеством баз данных, работающих на разных серверах, часто возникают ситуации, когда их серверы сильно недогружены, следовательно, ресурсы используются неоптимально. Цель Database Consolidation Workbench состоит в том, чтобы помочь консолидировать эти небольшие базы данных на меньшем количестве серверов, тем самым снижая затраты на закупку оборудования и текущие расходы на техническое обслуживание.

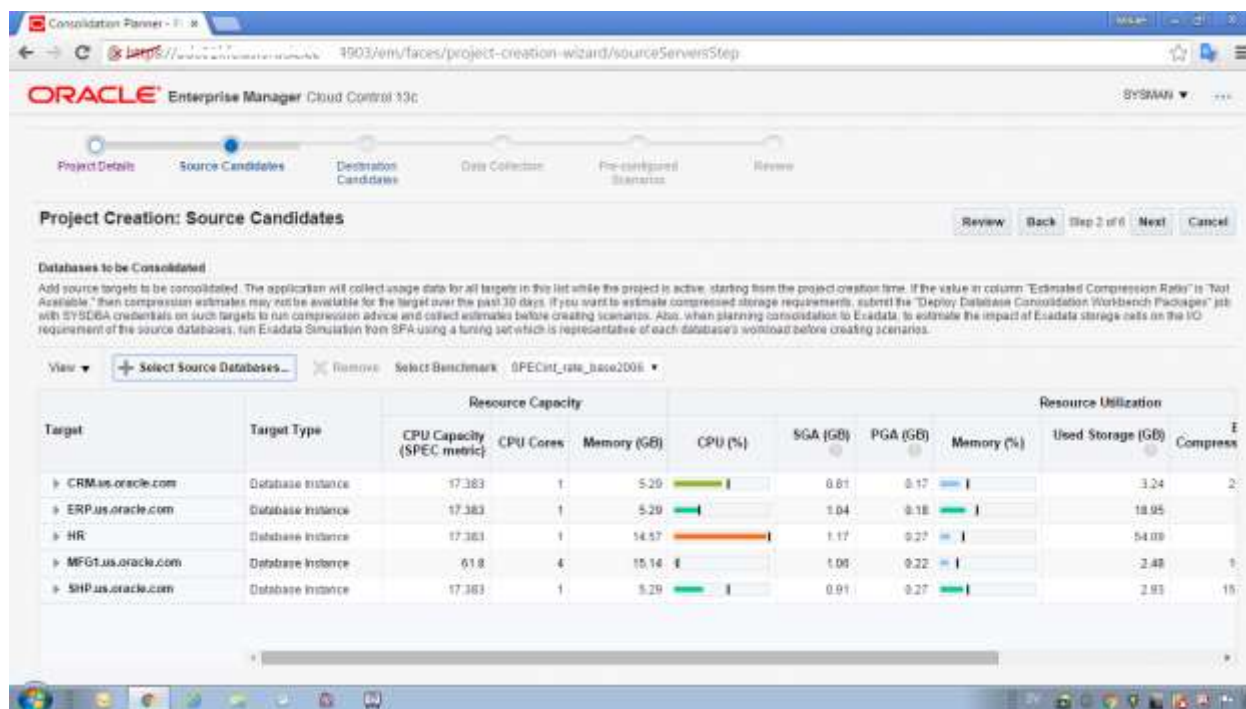
Рекомендации Database Consolidation Workbench по оптимизации консолидации позволяют оценить распределение ресурсов в различных сценариях консолидации, от ультраконсервативных (peak maximum) до агрессивных (average). Сценарии консолидации могут быть такими:

- Database-to-Database (Multitenant) — консолидация отдельных БД в контейнерную CDB;
- Database-to-Server (напр., Exadata) — консолидация отдельных БД на одном сервере;
- Oracle Public Cloud Consolidation — консолидация БД в публичном облаке Oracle.

Основой для оценки сценариев консолидации является анализ исторической нагрузки баз данных-кандидатов. В зависимости от выбранного сценария специальный компонент Database Consolidation Workbench, который называется Advisor, в течение определенного времени снимает данные о производительности исходных БД, используя в качестве источника диагностический репозиторий AWR, и на основе этих данных Advisor генерирует рекомендации.

Управление сценариями и анализ рекомендаций Database Consolidation Workbench осуществляется через графический интерфейс Oracle Enterprise Manager 13c.

Примеры экранов:



Spatial and Graph

Бесплатная опция Oracle Spatial and Graph входит в состав Oracle Database EE. Она обеспечивает поддержку со стороны базы данных геоинформационных систем (ГИС), а также корпоративных приложений и сервисов, использующих информацию о местоположении.



Функционал по хранению и анализу геопространственных данных, реализованный в Oracle Spatial and Graph, доступен во всех редакциях СУБД Oracle. Функционал обеспечивает мощную инфраструктуру для создания приложений, требующих продвинутого пространственного анализа и обработки пространственных данных в СУБД Oracle. Oracle Spatial and Graph поддерживает все основные пространственные типы и модели данных, тем самым обеспечивая решение задач, возникающих в различных отраслях, таких как транспорт, коммунальные службы, энергетика, государственный сектор, оборона и коммерция. Oracle Spatial and Graph содержит следующий функционал.

- более 400 пространственных функций, включая поиск центроидов и пространственную агрегацию;
- тип данных GeoRaster, который обеспечивает встроенную поддержку геопривязанных растровых изображений, таких как спутниковые снимки и GRID-данные;
- поддержку 3D-объектов, облаков точек, а также управление и анализ данных лазерного сканирования (LIDAR);
- геокодирование и маршрутизацию;
- топологическую модель данных для управления земельными ресурсами (например, кадастр).

Возможности сетевой модели данных

Сетевая модель данных в Oracle обеспечивает хранение и управление сетевыми (графовыми) структурами, хранящими информацию о вершинах сети и ребрах, соединяющих эти вершины. Помимо хранения в сетевой модели, Oracle реализовал алгоритмы для анализа сети, такие как:

- поиск кратчайшего пути;
- поиск ближайших соседей;
- поиск вершин, достижимых в рамках ограниченной стоимости.

Алгоритмы анализа в сетевой модели Oracle реализованы таким образом, что они автоматически секционируют сеть на более мелкие разделы. Загрузка разделов в оперативную память осуществляется по требованию во время анализа, что позволяет решить проблему анализа больших сетей, не помещающихся целиком в оперативную память. Возможности сетевой модели данных Oracle включают:

- моделирование и анализ графовых структур, описывающих физические и логические сети, используемые в транспорте, коммунальных службах, энергетике и телекоммуникациях;
- Java APIs для доступа к функциям анализа;
- хранение атрибутивной информации, привязанной к элементам графа, таким как вершина и ребро;
- поддержку направленных и ненаправленных графов с возможностью хранения стоимости.

Возможности управления семантическими базами данных

Oracle Spatial and Graph включает широкие возможности по управлению семантическими данными со встроенной поддержкой стандартов World Wide Web Consortium. Семантические RDF-графы разработаны для использования в социальных сетях, а также для хранения информации о социальных взаимодействиях, требуемых в исследовательской деятельности, медицинских науках, средствах массовой информации и связях с общественностью. Возможности RDF Semantic Graph включают:

- индексирование, управление и выполнение запросов к онтологиям;
- хранение связей в графах в виде триад в компрессированных и партицированных таблицах;
- RDFS, OWL и построение выводов (параллельный, пакетный и инкрементальный режимы).



Property Graph

Используя функционал Property Graph, разработчики могут создавать системы для сложного анализа различных графовых данных. Например, исследование и выявление связей в социальных сетях, IoT, выявление мошеннических транзакций, анализ поведения клиентов и др.

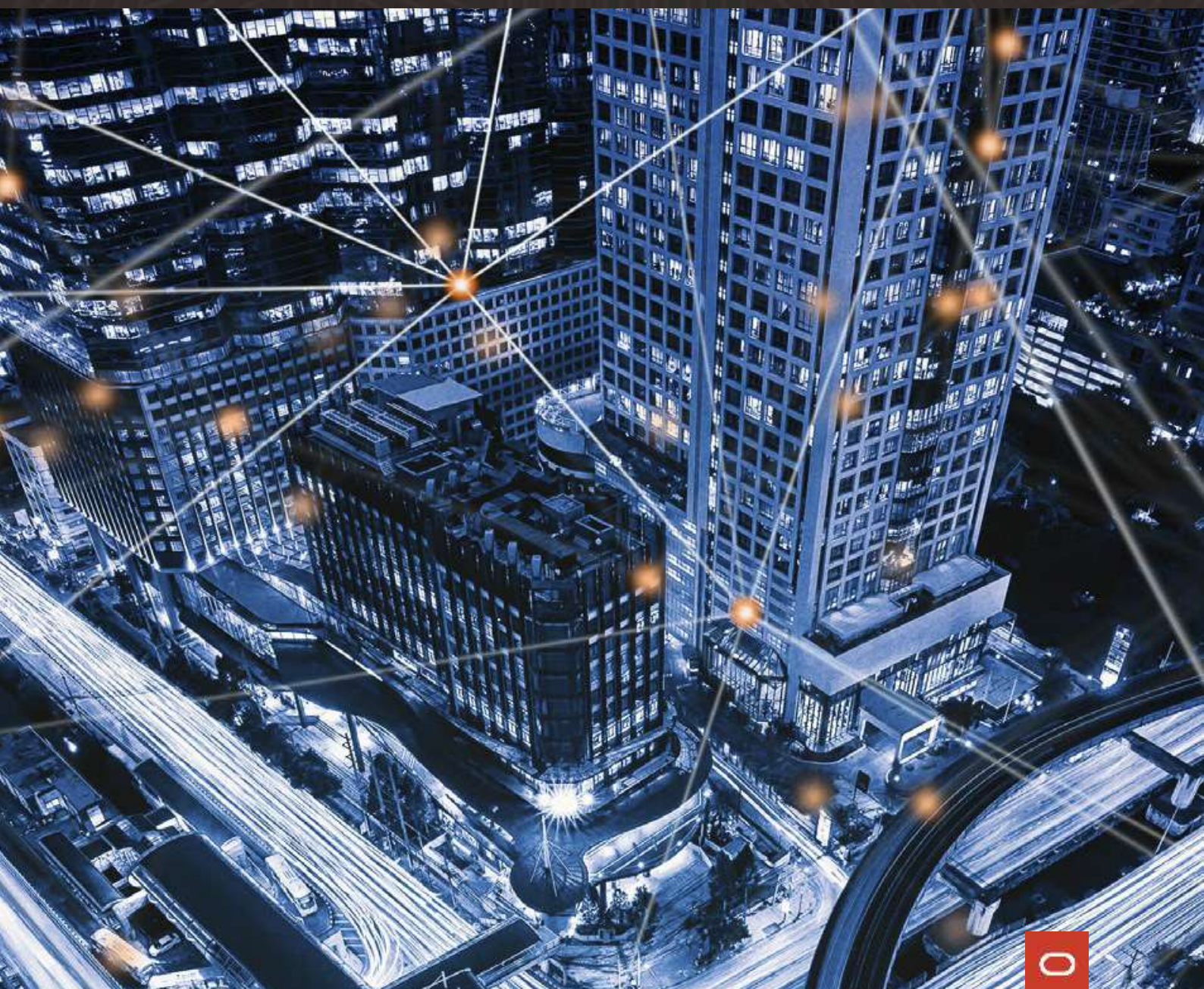
Возможности Property Graph

- Функционал Property Graph построен на In-Memory-движке, позволяющем масштабировать обработку графов до триллионов ребер.
- Более 50 встроенных аналитических функций, позволяющих делать ранжирование, центрирование, выдачу рекомендаций, поиск путей в графе, выявление сообществ и др.
- Property Graph содержит специальный SQL-подобный язык для работы с графами и API для разработчиков (REST, SQL, Java), что позволяет упростить разработку приложений.



Продукты Oracle

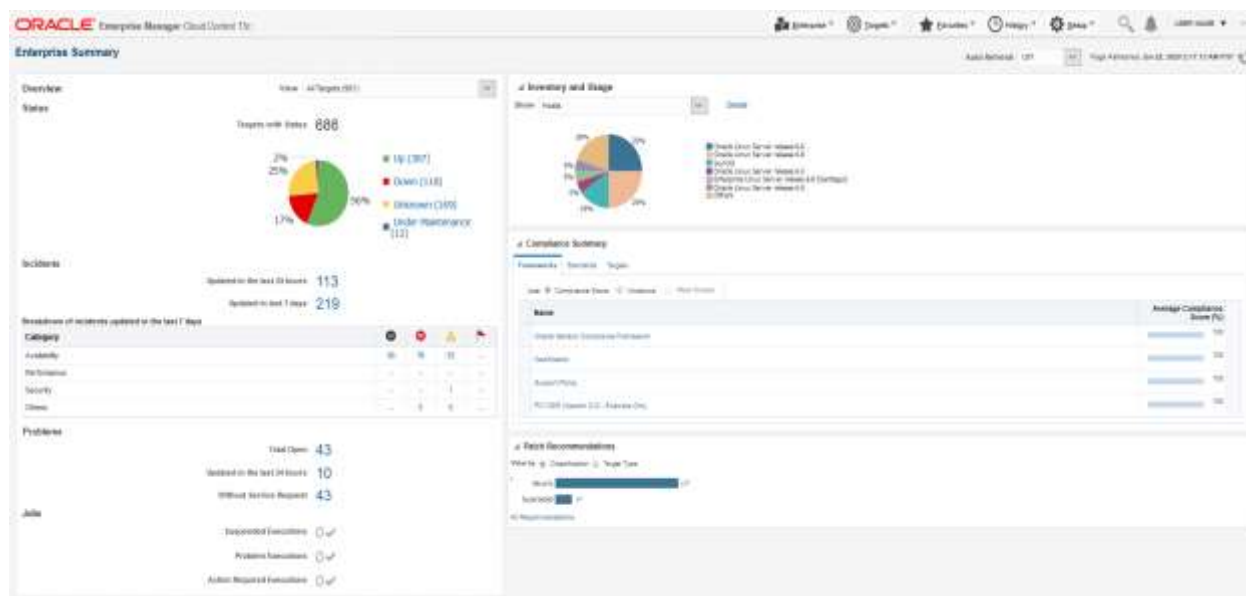
Средства управления
инфраструктурой и СУБД
(ОМС, OEM)



Oracle Enterprise Manager

Oracle Enterprise Manager (OEM) — это гибкая расширяемая платформа для мониторинга и управления ИТ-инфраструктурой и корпоративными облаками, также **OEM** — общее название линейки продуктов Oracle, относящихся к категории Enterprise Manageability.

Oracle Enterprise Manager представляет собой единственное в отрасли полное, интегрированное решение для управления всем технологическим стеком Oracle в традиционных архитектурах (On-Premise), в публичных облаках (Public Cloud), в гибридных средах (Hybrid Cloud), а также в частном корпоративном облаке (Private Cloud). Какой бы вариант вы ни выбрали, Oracle Enterprise Manager обеспечивает возможность всестороннего мониторинга и управления вашими системами «от приложения до диска» из единого окна, позволяя достичь беспрецедентного качества и эффективности сопровождения и максимальной отдачи от инвестиций в ИТ-менеджмент за счет лучших решений для интеллектуального управления стеком Oracle и инженерными системами. Начиная с версии 12 полное название продукта: **Oracle Enterprise Manager Cloud Control**.



Ключевые возможности Oracle Enterprise Manager:

- высокопроизводительная, масштабируемая платформа мониторинга, основанная на Oracle Database и WebLogic, поддержка расширений (Plug-Ins) и REST API;
- автоматизация и отчеты (Jobs, Incident Management, Reporting);
- управление жизненным циклом компонентов ИТ (Patching, Provisioning, Upgrade);
- всесторонний мониторинг Oracle Database (Diagnostic, Tuning, полная поддержка опций БД);
- управление конфигурациями (Comparison & Drift Management);
- управление Compliance (соблюдение промышленных и корпоративных стандартов);
- управление качеством приложений (функциональное и нагрузочное тестирование, управление тестовыми средами БД);
- управление частным облаком (Self-Service Portal, Metering & Chargeback, Virtualization);
- автоматизация массовых операций (Fleet Maintenance);
- бизнес-транзакции и диагностика «узких мест» приложений (JVMD).

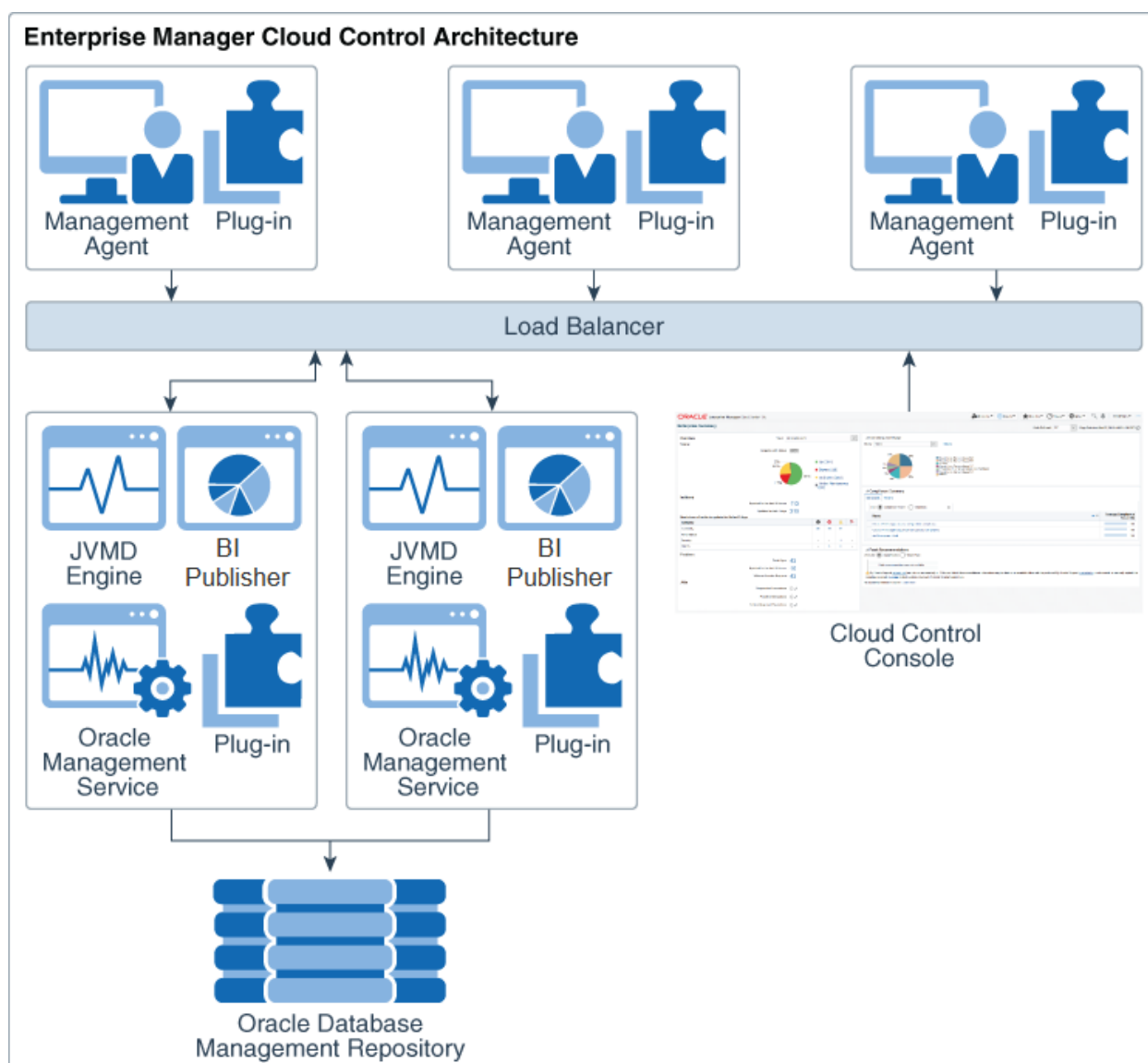


Архитектура Oracle Enterprise Manager

Oracle Enterprise Manager (OEM) — система мониторинга и управления корпоративного уровня, поэтому в его архитектуру были заложены принципы максимальной производительности, масштабируемости и надежности. В основе платформы OEM лежат проверенные временем компоненты технологического стека: Oracle Database для репозитория (Management Repository) и Oracle WebLogic для сервера управления (Oracle Management Service). Схематично архитектура OEM выглядит следующим образом.

1. На целевые компоненты ИТ, которыми нужно управлять (БД, сервер, сервер приложений и т. д.), устанавливаются специальные агенты (**Management Agent**), которые занимаются сбором и передачей метрик производительности на центральный сервер управления (Oracle Management Service), также через эти агенты на целевые системы передаются команды управления, например: запуск/остановка БД, установка патча, создание или клонирование Pluggable Database, изменения параметров.
2. Центральный сервер управления (Oracle Management Service) — «сердце» Oracle Enterprise Manager, его задачи: сбор, анализ, визуализация метрик производительности, взаимодействие с репозиторием хранения (Oracle Management Repository), управление интерфейсом пользователя UI, планирование и запуск заданий (jobs), формирование отчетов, рассылка уведомлений и т. д. Oracle Management Service представляет собой web-приложение, написанное на Java, пользователи взаимодействуют с ним через графическую консоль в браузере.
3. Oracle Management Repository — центральное хранилище диагностической информации и система поддержки работы сервера управления. В репозитории хранится история метрик и необходимая информация для их отображения, благодаря чему графические экраны мониторинга доступны разным пользователям Enterprise Manager. Технологически Oracle Management Repository выполнен в виде набора объектов в Oracle Database: tables, views, procedures, jobs, packages и т. д.
4. В архитектуру OEM были заложены возможности расширения, поэтому необходимо отметить еще один ключевой компонент «Plug-In». Благодаря Plug-Ins можно, с одной стороны, расширять вашу систему Oracle Enterprise Manager дополнительными возможностями, с другой — не перегружать установку OEM лишним функционалом, которым вы пользоваться не будете. Например, если ваш ИТ ландшафт состоит только из Oracle Database, MySQL, Exadata Database Machine, то вам для мониторинга и управления нужны соответствующие Plug-Ins, которые можно установить прямо из консоли, после этого в OEM появятся возможности для детального мониторинга ваших объектов. Plug-Ins — специальные программные модули, которые устанавливаются на Management Service и в некоторых случаях на Agent.





Oracle Enterprise Manager Cloud Control — одно из высокоприоритетных направлений корпорации Oracle в разработке продуктов семейства Enterprise & Cloud Manageability. График выпуска новых версий похож на модель обновления СУБД Oracle — выпуск основной версии (Major Release) и регулярные обновления/исправления (Release Update — RU). Для Oracle Enterprise Manager обновления, исправления и добавления новых возможностей в рамках пакета Release Update выходят каждый месяц! Актуальная на момент написания этой статьи версия: Oracle Enterprise Manager Cloud Control 13.4 Release Update 5.

Новый релиз Oracle Enterprise Manager включает много интересных возможностей: улучшенный и доработанный интерфейс пользователя, доступ к OEM через новое мобильное приложение, возможность интеграции с системами OpenSource (Grafana), поддержку новых типов и версий объектов, соответствие стандартам Center for Internet Security, обновление технологического стека и многое, многое другое (подробнее: <https://blogs.oracle.com/oem/>).

Необходимо отметить один очень важный момент относительно лицензионной политики продукта Oracle Enterprise Manager. **Базовый функционал Oracle Enterprise Manager, такой как: установка и конфигурирование платформы OEM, управление агентами, обнаружение объектов, создание пользователей, автоматизация рутинных операций через OEM jobs, управление инцидентами через консоль, отчеты Information Publisher — бесплатен для всех заказчиков!**

Также в OEM являются бесплатными такие частые операции с Oracle Database, как: просмотр и поиск ошибок в alert.log, управление загрузкой/выгрузкой данных, управление Grid Infrastructure, управление



резервным копированием, параметрами БД, управление storage, tablespaces, datafiles, undo, управление статистикой оптимизатора, управление пользователями, ролями, привилегиями БД.

Таким образом, лицензировать нужно лишь строго определенные функциональные наборы (Management Pack); например, если вы хотите работать с интерфейсом Active Session History Analytics для поиска первопричин плохой производительности или анализируете работу SQL через экраны Real-Time SQL Monitoring, то вам потребуется лицензия на опцию Diagnostics Pack для соответствующей БД.

В Oracle Enterprise Manager всегда можно гибко управлять доступом к соответствующему функционалу, в настройках имеется пункт Management Pack Access, с помощью которого администратор может разрешать или запрещать пользователям доступ к лицензируемым опциям. Также с помощью пункта меню «Packs for this Page» всегда можно посмотреть, требуется ли лицензия для того функционала OEM, который вы сейчас используете.

Какую бы архитектуру развертывания ИТ-ландшафта вы ни выбрали (On-Premise, Cloud, Hybrid, Private Cloud), с помощью Oracle Enterprise Manager Cloud Control у ваших ИТ-специалистов всегда будет единый источник информации о состоянии инфраструктуры, возможность оперативно реагировать на инциденты, возможность в разы повысить продуктивность работы администраторов за счет автоматизации рутины, что в конечном счете позволит обеспечить стабильную и качественную работу ваших ИТ-сервисов!

Data Masking and Subsetting Pack

Тестовые среды и среды разработки являются потенциальными целями для нападения злоумышленников, поскольку они часто содержат копии производственных данных. Обычно эти среды менее защищены, чем производственные системы. Необходимо замаскировать данные и, возможно, использовать меньшее количество данных, чем в производственной среде. Пакет OEM Oracle Data Masking and Subsetting извлекает из базы данных все данные приложения или заданное их подмножество, проводит маскирование конфиденциальной информации так, чтобы получаемые в итоге тестовые данные могли быть безопасно использованы при тестировании и разработке. Затраты на обеспечение информационной безопасности тестовых сред при этом снижаются, т.к. тестовые данные более не содержат конфиденциальной информации. Замаскированные данные можно без опасения передавать другим компаниям для тестирования или разработки приложений.



Рис. 1. Маскирование подмножества данных

Sensitive Data Discovery and Modeling

Поиск конфиденциальных данных, формирование модели данных приложения.

Нахождение конфиденциальных данных приложения в базе данных является нетривиальной задачей. Процедура Application Data Modeling автоматизирует поиск столбцов, содержащих конфиденциальные данные, и соответствующих им столбцов в других таблицах. При поиске используются встроенные расширяемые шаблоны, такие как номера кредитных карт и национальные идентификаторы, чтобы найти



данные отвечающие шаблону и так идентифицировать столбцы с конфиденциальной информацией. Получающаяся в итоге модель данных приложения (Application Data Model, ADM) содержит полный набор столбцов конфиденциальных данных вместе с их отношениями (связями), гарантирующими, что целостность данных приложений будет сохранена в процессе маскирования и формирования подмножества данных.

Masking Sensitive Data

Маскирование конфиденциальных данных.

Пакет Oracle Data Masking and Subsetting обеспечивает всестороннюю и расширяемую библиотеку маскирования форматов, функций/трансформаций и шаблонов приложений. Конфиденциальные данные, такие как номера кредитных карт, национальные идентификаторы, персональные данные и другие, могут быть легко замаскированы с использованием библиотеки форматов маскирования.

Для простых случаев библиотека обеспечивает трансформации маскирования, такие как фиксированные символы или числа, замена столбца и SQL-выражение или регулярные выражения. Библиотека также обеспечивает несколько усовершенствованных трансформаций для создания пользовательских форматов маскирования.

- **Shuffle Masking** — производится перестановка значений столбца таблицы в случайном порядке, нарушающая взаимно однозначное соответствие между элементами конфиденциальных данных в строках. Например, значения в столбце «зарплата» могут быть переставлены (перемешаны) для искажения отображаемого значения зарплаты сотрудника.
- **Blurring or perturbation** — «размытие или возмущение» — генерируются случайные числа, символы и даты в определяемом пользователем диапазоне.
- **Encryption** — зашифровываются конфиденциальные данные с помощью назначаемого ключа шифрования. При этом формат данных сохраняется. Эта трансформация полезна, когда замаскированные данные, отправленные третьему лицу, должны быть возвращены обратно (восстановлены) для дальнейшего использования.
- **Format Preserving Randomization (or auto mask format)** придает случайный характер данным (рандомизирует данные), сохраняя длину строки, позицию, случайные символы в верхнем или нижнем регистрах и специальные символы.
- **Conditional Masking** — маскирование по заданному условию. Например, идентификаторы Social Security Number (USA), National Insurance Number (UK) могут быть замаскированы с использованием соответствующих шаблонов.
- **Compound Masking** — групповое маскирование для выбранных столбцов. Например, если вы хотите одновременно переместить группу полей адреса (город, область, страна, улица и т. п.), то эти поля будут перемещаться вместе во время процесса перестановки.
- **Deterministic Masking** — производит согласованное маскирование полей для связанных приложений и баз данных. Это позволяет одинаково замаскировать связанные поля для группы приложений.
- **User Defined PL/SQL Masking** — использование функций, чтобы сформировать пользовательскую маскирующую логику или мигрировать (адаптировать) существующие маскирующие сценарии.

Data Subsetting

Подмножество данных.

Сокращает угрозы безопасности путем удаления ненужных строк из базы данных прежде, чем данные будут перенесены в тестовое окружение. Это также помогает минимизировать затраты на хранение



в непродуманных средах. Способность должным образом формировать подмножество данных, не нарушая работоспособность приложения, является сложной задачей.

Oracle Data Masking and Subsetting обеспечивают формирование подмножества данных согласно цели тестирования. Целью может быть относительный размер таблицы, такой как извлечение 10%-го подмножества данных таблицы, содержащей 10 миллиардов строк, для тестирования. Условие формирования подмножества может минимизировать риск утечки путем отбрасывания всех пользовательских записей, создаваемых до указанной даты, или использование данных для определенного региона, например Азии (на рисунке). Подмножества строк, связанных в модели данных таблиц, формируются с учетом указанных связей, т. е. мы получаем логически связанные наборы данных.



Рис. 2. Формирование подмножества данных по заданному условию

Centralized Administration and Flexible Deployment

Централизованное управление и гибкое использование.

Oracle Enterprise Manager Cloud Control (EM) является универсальным средством управления. Oracle Data Masking and Subsetting Pack установлен в EM по умолчанию.

Маскирование и формирование подмножества данных может быть выполнено на клонированной копии данных (In-Database), снижая накладные расходы в производственной системе. Также маскирование и формирование подмножества может быть проведено во время экспорта базы данных (In-Export), что избавляет от необходимости подготовки «промежуточных» серверов.



Высокая эффективность маскирования достигается через интеграцию с механизмами БД Oracle и утилитой Oracle Data Pump. После построения модели данных приложения процесс маскирования может быть повторен, минимизировав накладные расходы.

Формирование подмножества данных и маскирование могут быть выполнены на данных в других реляционных базах данных (MySQL, SQL Server, Sybase, DB2, Informix и Teradata) путем подготовки данных с использованием Oracle Database Gateway.

Software Lifecycle Integration

Интеграция жизненного цикла программного обеспечения.

Маскирование и формирование подмножества интегрированы с управлением данными Oracle и инструментами тестирования. Например, интеграция с пакетом управления жизненным циклом БД (Oracle Database Life Cycle Management Pack) упрощает процедуру маскирования и формирования подмножества в едином потоке операций. Интеграция с Oracle Real Application Testing Pack позволяет маскировать конфиденциальные данные в составе полученной рабочей нагрузки и воспроизвести его в системах тестирования без риска утечки конфиденциальной информации.

Cloud Management Pack for Oracle Database

Продукт Oracle Enterprise Manager (OEM) является интегрированным решением уровня предприятия для эффективного управления ИТ-системами, а также для поддержки полного жизненного цикла облачных инфраструктур. Бизнес-ориентированная функциональность позволяет быстро настраивать, управлять и поддерживать корпоративные облака и традиционные ИТ-окружения на всех уровнях их архитектуры от приложений до дисков. В состав бесплатного продукта OEM входит ряд платных пакетов для выполнения отдельных задач. В частности, для создания и управления частным облаком предприятия служит пакет Oracle Cloud Management Pack (OCMP). Он предоставляет уникальные в отрасли облачные возможности, включая портал самообслуживания, централизованное развертывание сервисов, управление политиками доступа, интегрированную функциональность для расчета и планирования использования ресурсов.

Платформа как сервис (DBaaS и MWaaS)

Oracle Cloud Management Pack for Oracle Database для баз данных Oracle и Cloud Management Pack for Oracle Fusion Middleware для серверов приложений WebLogic позволяют использовать в Oracle Enterprise Manager функциональность работы с сервисами DBaaS и MWaaS.

Пользователи могут заказывать сервисы баз данных и серверов приложений в удобном интерфейсе портала самообслуживания и начинать полноценное использование подготовленной для них платформы в кратчайшие сроки. Администраторы имеют возможность гибкой настройки шаблонов сервисов, правил их развертывания, пользовательских ролей и квот на использование ресурсов.

Мощный функционал мониторинга облачной системы предоставляет администраторам детальную информацию по всем ее компонентам, позволяя оперативно реагировать на происшествия и гибко изменять ее параметры. Наконец, он позволяет обеим сторонам понять расходы на предоставленные сервисы, а также установить ответственность за потребление ресурсов.

Инфраструктура как сервис (IaaS)

OCMP позволяет создавать и развертывать как сервисы из шаблона виртуальные машины Oracle VM. Машины могут содержать различные операционные системы для платформы x86 и предустановленное ПО. Ведется сбор статистики по использованию ресурсов каждой виртуальной машины. Размеры машины — количество процессоров, дисков, памяти — можно менять.



Установка, создание и конфигурирование

Администраторы облака выделяют пулы ресурсов для использования в облачной инфраструктуре. Одновременно администраторы могут установить политики доступа для определения прав пользователей на доступ к сервисам в облаке. Чтобы ограничить чрезмерное потребление ресурсов компании, Enterprise Manager позволяет определять необходимые квоты. В результате, пользователь не сможет задействовать больше ресурсов, чем ему позволено соответствующим правилом. Администратор также может удобно управлять расписанием пользовательских запросов для оптимизации загрузки облачной системы.

PaaS-каталог и жизненный цикл облака

Администраторы облака могут создавать как простые одноузловые сервисы, так и более сложные многоузловые, многокомпонентные сервисы (запакованные в специальные сборки — Assemblies) и загружать их во встроенную в библиотеку приложений Oracle Enterprise Manager. После этого в зависимости от настроенных политик доступа пользователи имеют возможность заказать ту или иную платформу. Также поддерживается контроль версий для платформ, и при добавлении новой версии она автоматически становится доступна в каталоге сервисов.

Портал самообслуживания

OSMP поставляется с готовым к использованию порталом самообслуживания, в котором пользователи могут как заказывать новые сервисы, так и управлять жизненным циклом предоставленных сервисов. Портал не только отображает доступные сервисы из каталога, но и позволяет создавать собственные сервисы, если соответствующие права предоставлены администратором облака. Пользователям доступна информация по их предыдущим запросам, квотам, текущей утилизации ресурсов и статистике использования сервисов. Портал также позволяет пользователям запросить дополнительные ресурсы для их окружений (например, дополнительное пространство на диске) или изменить размер их платформы в соответствии с текущими потребностями. Кроме портала самообслуживания, пользователям облака также доступен интерфейс командной строки (CLI) и программный интерфейс (API), которые могут использоваться разработчиками для доступа к облачным сервисам по аналогии с порталом самообслуживания.

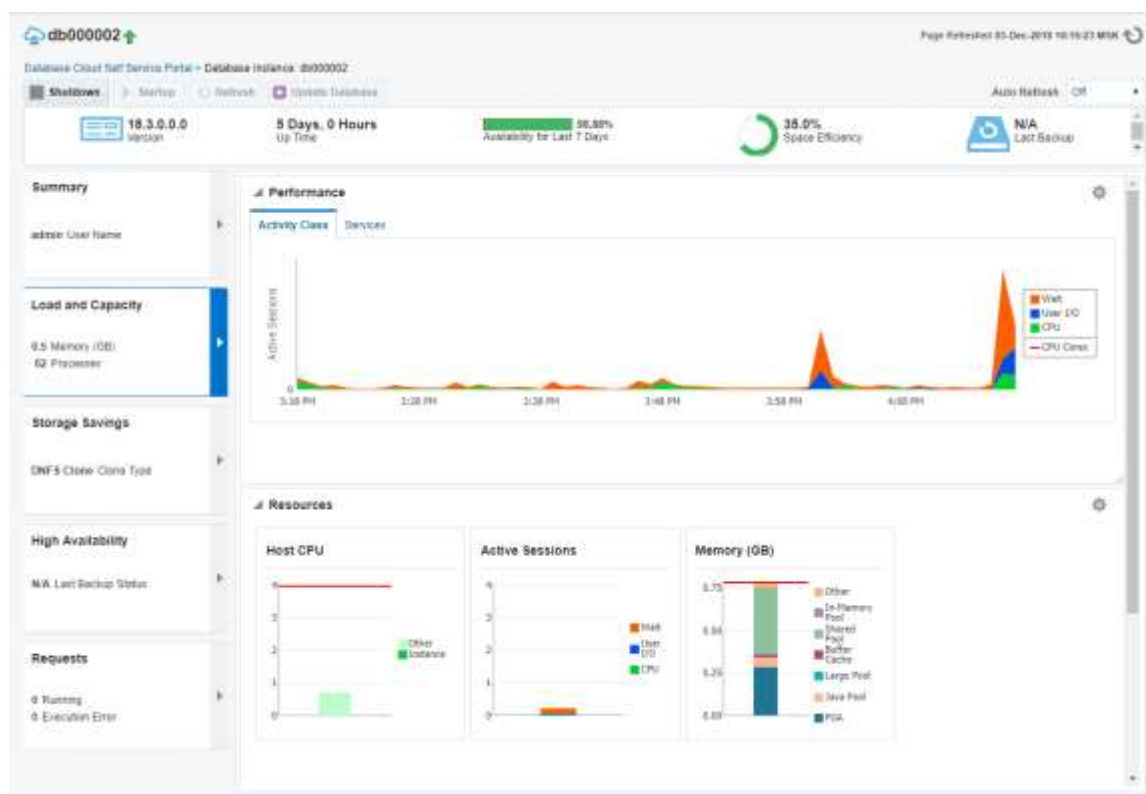


Рис. 1. Портал самообслуживания для корпоративного облака



Аудит и монетизация ресурсов

Важный аспект использования облаков заключается в возможности прозрачного распределения затрат между пользователями облачных сервисов, основанном на объеме предоставленных ресурсов. Oracle Enterprise Manager предоставляет функциональность для настройки детальных тарифных планов, охватывающих различные метрики, доступные для физических и виртуальных ресурсов.

Тарифные планы могут быть основаны не только на использовании ресурсов, но и на использовании различных конфигураций или версий, а также могут использовать фиксированную ставку оплаты.

Oracle Enterprise Manager поддерживает измерение потребления всех необходимых ресурсов для Oracle PaaS, включая хост-серверы, виртуальные машины, инстансы БД и сервера приложений. Иерархия кост-центров для распределения затрат и агрегации отчетности может быть загружена из LDAP. А при использовании аналитического инструмента BI Publisher отчетность может быть представлена в таких общепринятых форматах, как PDF, HTML, Word, Excel или PowerPoint.

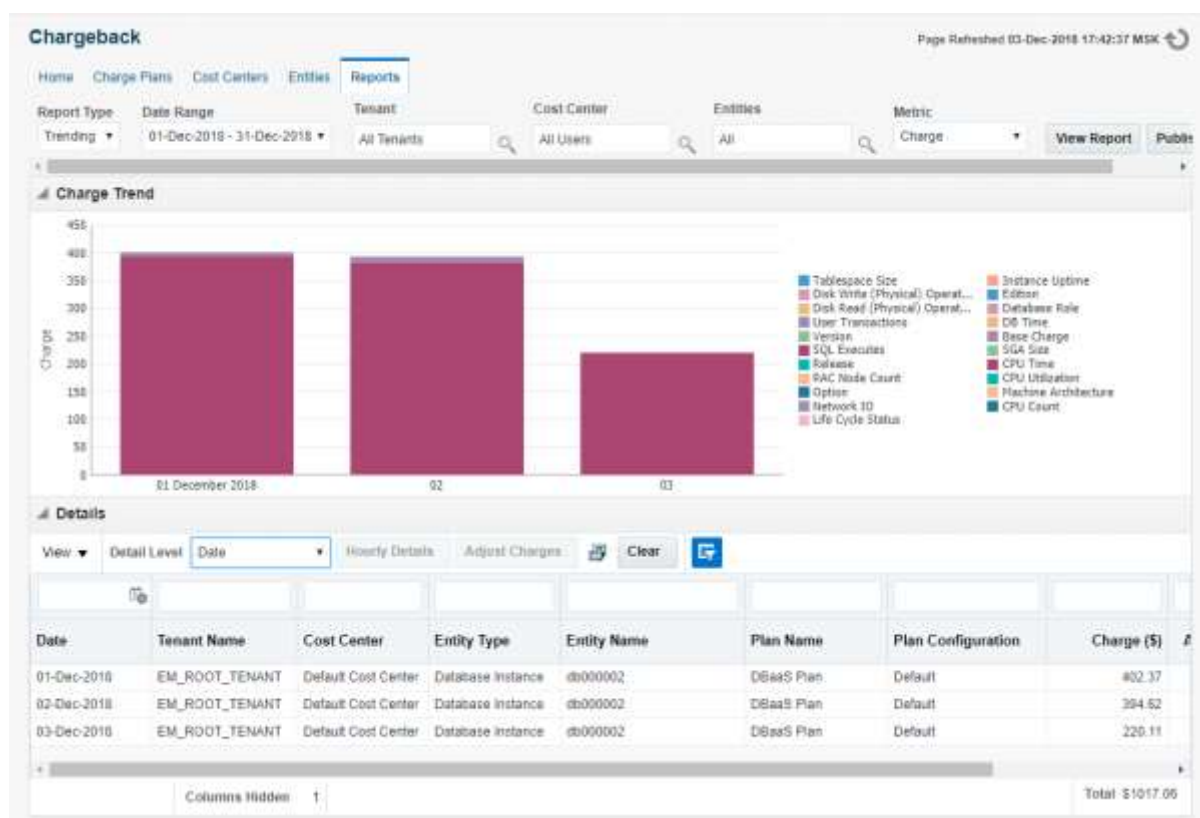


Рис. 2. Функциональность аудита облачных ресурсов

Создание новых сервисов

Одним из главных достоинств ОСМР является возможность быстрого и простого (без программирования) создания новых сервисов. Достаточно заполнить экранные шаблоны, выбрать, например, конфигурацию БД (SI, RAC, DG, Multitenant и т. д.), указать источник создания БД (если создается не пустая БД), и сервис будет создан автоматически. Во время работы сервиса возможны апгрейд ПО и накатка патчей для множества сервисов (fleet management), при этом у пользователя сервиса появится кнопка, нажав на которую он может перейти на новую версию сервиса (т. е. время он выбирает сам).

Создавать сервисы, пулы, зоны, пользователей можно не только через GUI стандартного портала самообслуживания, но и через CLI и REST API. Это позволяет создать собственный портал самообслуживания. При развертывании сервиса могут срабатывать заранее заготовленные pre- и post-скрипты.



Если у заказчика есть БД, созданные без участия OCMР, то их можно добавлять в портал самообслуживания и работать с ними как с другими сервисами (запустить, остановить, удалить, собрать статистику и т. д.).

Создавать сервисы DBaaS можно не только для x86, но и для любой платформы, где можно вручную установить СУБД и создать БД (Sparc, HP, AIX).

Diagnostics Pack

Oracle Diagnostics Pack предоставляет всестороннюю функциональность по мониторингу, автоматической диагностике производительности и диагностике производительности в реальном времени, встроенную в ядро базы данных и Oracle Enterprise Manager Cloud Control 13c. Управляете ли вы одной или многими базами данных, Oracle Diagnostics Pack предоставляет полное, экономически эффективное и простое в использовании решение для управления производительностью Oracle Database. При использовании с Enterprise Manager Oracle Diagnostics Pack дополнительно предоставляет отчеты о производительности и доступности на уровне предприятия, централизованный репозиторий с данными производительности, кросс-системную агрегацию данных производительности, что значительно упрощает процесс управления большим количеством баз данных.

Диагностика медленно работающей системы является отнимающей много времени задачей, с которой чаще всего сталкиваются администраторы баз данных. Oracle Diagnostics Pack включает в себя модуль самодиагностики, встроенный непосредственно в ядро базы данных Oracle, который называется Automatic Database Diagnostic Monitor (ADDM) и который полностью освобождает администраторов от сложной и тяжелой задачи по диагностике проблем производительности.

ADDM начинает свой анализ, фокусируясь на активности, которая занимает больше всего времени в базе данных, и затем идет по сложному дереву классификации проблем, чтобы найти первопричину проблемы. Возможность ADDM обнаруживать настоящую причину проблемы производительности, а не только сообщать об ее симптомах, — лишь один из нескольких факторов, которые ставят ADDM намного выше любых других средств для управления производительностью Oracle. Каждая рекомендация ADDM имеет соответствующий вес, что позволяет расставлять приоритеты при решении наиболее критичных проблем. Чтобы лучше понять влияние выявленных ADDM проблем на работу базы данных в течение долгого времени, каждая рекомендация имеет описательное имя, что позволяет применять фильтры, делать поиск по этому имени и связывать эту рекомендацию с предыдущими аналогичными рекомендациями, которые были даны за последние 24 часа. ADDM полностью поддерживает контейнерную архитектуру Multitenant и может работать на уровне отдельных PDB.

Для Oracle Real Application Cluster (RAC) ADDM имеет специальный режим для анализа производительности на уровне кластера. Он делает анализ глобальных ресурсов базы данных, таких как высоконагруженный SQL, эффективность межкластерного соединения, сетевые задержки, отклонения во времени отклика базы данных, пропускная способность I/O и т. д.

Другая возможность Oracle Diagnostics Pack — Real-Time ADDM. Эта возможность обеспечивает инновационный способ анализа проблем в зависших базах данных. Real-Time ADDM запускает набор предопределенных процедур, используя нормальный и диагностический режим связи с базой данных, для анализа текущей производительности и помогает администратору решать проблемы взаимоблокировок, зависаний, конфликтов в разделяемом пуле и многих других исключительных ситуаций, из-за которых приходится перестартовать базу данных и которые часто приводят к значительным потерям доходов.

Oracle Diagnostics Pack включает в себя встроенный в каждую базу данных Oracle репозиторий, который называется Automatic Workload Repository (AWR). В нем содержится статистика о работе базы данных и другая релевантная информация. Через регулярные интервалы времени (раз в час по умолчанию) база данных делает снимки всех основных метрик и сохраняет их в AWR. AWR спроектирован так, чтобы оказывать малое



влияние на работу базы данных и автоматически управлять своим пространством, чтобы не быть дополнительным бременем для администраторов.

AWR является основой для механизмов самоуправления базы данных Oracle Database. Он является источником информации о работе базы данных в прошлом, о том, как она использовалась, и дает ей возможность самой принимать решения, как наиболее точно настроиться под окружение, в котором она работает. AWR также поддерживает создание базисных снимков производительности. Готовое скользящее окно протяженностью 80 дней доступно для того, чтобы сравнивать производительность базы данных с производительностью за предыдущую неделю. Окно может настраиваться при необходимости. Эти базисные снимки производительности AWR могут использоваться для последующего сравнения с текущей производительностью системы для выявления отклонений в производительности и их первопричин.

Ключевым компонентом AWR является Active Session History, или ASH. ASH делает каждую секунду снимки текущего состояния всех активных сессий и сохраняет их в оперативной памяти. Собранные в оперативной памяти данные доступны через V\$-представление. Эти данные также записываются каждый час в AWR для последующей диагностики производительности. ASH позволяет проводить анализ неустойчивых проблем производительности, которые длятся очень короткий промежуток времени, и избавляет от необходимости использовать SQL-трассировку. ASH подобно AWR тоже может работать с RAC, и полученная информация о распределении активности сессий по кластерным классам ожидания позволяет выявить специфичные для RAC потенциальные проблемы. Возможности ASH также были расширены для работы на резервных базах данных, чтобы помогать анализировать производительность Oracle Data Guard.

Oracle Enterprise Manager Cloud Control 13c включает в себя ASH Analytics, инструмент для анализа ASH данных, позволяющий администраторам агрегировать, детализировать и разбивать данные производительности по разным измерениям. Благодаря возможности создавать фильтры по различным измерениям выявление проблем производительности стало как никогда простой задачей. Встроенное представление активности сессий в виде плоского дерева (treemap) позволяет администраторам исследовать данные производительности, используя predetermined иерархии измерений производительности.



Рис.1. Представление производительности ASH Analytics

Oracle Diagnostics Pack включает в себя полный набор возможностей для мониторинга и уведомлений, позволяющий администраторам заблаговременно выявлять IT-проблемы для всего стека приложения и реагировать на них. Администраторы могут воспользоваться как готовыми настройками Enterprise Manager для мониторинга объектов управления, так и изменить их под нужды своего центра обработки данных. Для баз данных существуют адаптивные пороговые значения метрик, автоматически вычисляемые самой базой данных на основе анализа своей предыдущей производительности; администратор автоматически получает уведомление, если метрика приняла статистически необычное значение. Для других типов объектов предусмотрен легкий доступ к истории метрик, позволяя администраторам определить подходящие пороговые значения, основанные на типичных значениях метрик. Если требуется осуществлять мониторинг каких-то специфичных для центра обработки данных условий, то администраторы могут определить новые



метрики для любого объекта управления, используя расширения метрик. Администраторы могут настроить скрипты с корректирующими действиями, если уведомление о проблеме не имеет готового решения. Когда возникнет уведомление, эти скрипты будут автоматически выполняться и решать проблему, тем самым сводя к минимуму необходимость ручного вмешательства. Кроме того, история уведомлений также легкодоступна и позволяет администраторам видеть, какие действия были осуществлены в ответ на предыдущие уведомления.

Как только мониторинг настроен и события обнаруживаются на контролируемых объектах, уведомления об этих событиях могут быть отправлены соответствующим администраторам. Уведомления включают в себя уведомления по электронной почте, выполнение пользовательских скриптов и PL/SQL-процедур и отсылку SNMP-прерываний. Кроме того, могут быть использованы управляющие коннекторы (management connectors) для открытия сервисных запросов в автоматизированных системах технической поддержки для инцидентов (основанных на важных событиях) и/или для отправки информации в системы управления третьих фирм. Наконец, для поддержки периодов планового обслуживания объектов предоставляется возможность блэкаута, позволяя администраторам временно приостановить мониторинг объектов и предотвратить генерирование ложных уведомлений во время планового простоя.

Tuning Pack

Oracle Tuning Pack — дополнительная опция для управления Oracle Database, наиболее эффективное и легкое в использовании решение, которое полностью автоматизирует процесс настройки приложений. Улучшение производительности SQL достигается с помощью мониторинга выполнения SQL в реальном времени и SQL-советников, интегрированных с Oracle Enterprise Manager Cloud Control 13c, и все это вместе предоставляет всестороннее решение для сложной и требующей много времени задачи по настройке приложений.

Для администраторов баз данных и разработчиков приложений настройка приложений является критически важной задачей, и они тратят значительное время на ее выполнение. Плохо настроенное бизнес-приложение может потенциально повлиять не только на нескольких пользователей, но и на всю операционную деятельность организации, поэтому компании вкладывают значительное количество ресурсов, чтобы обеспечить хорошую работу приложений, критически важных для их бизнеса.

Первый шаг в настройке SQL — это выявление «тяжелых» SQL-операций, потребляющих значительное количество системных ресурсов. Традиционно долго выполняющиеся SQL-операции доставляли администраторам баз данных массу проблем на живых рабочих системах, так как у них никогда не было средств, чтобы выяснить, сколько времени осталось до завершения долгого запроса: несколько мгновений или несколько часов. Real-Time SQL Monitoring, появившийся еще в Oracle Database 11g, предоставляет самый быстрый и самый легкий способ для выявления и исправления проблем с производительностью долго выполняющихся SQL-операторов. Теперь администраторы могут видеть выполнение SQL в реальном времени через специальные экраны Enterprise Manager Performance Hub с детальной, динамически обновляющейся статистикой по каждому шагу плана выполнения SQL. Сбор статистики выполнения SQL практически не оказывает влияния на производительность рабочей системы.



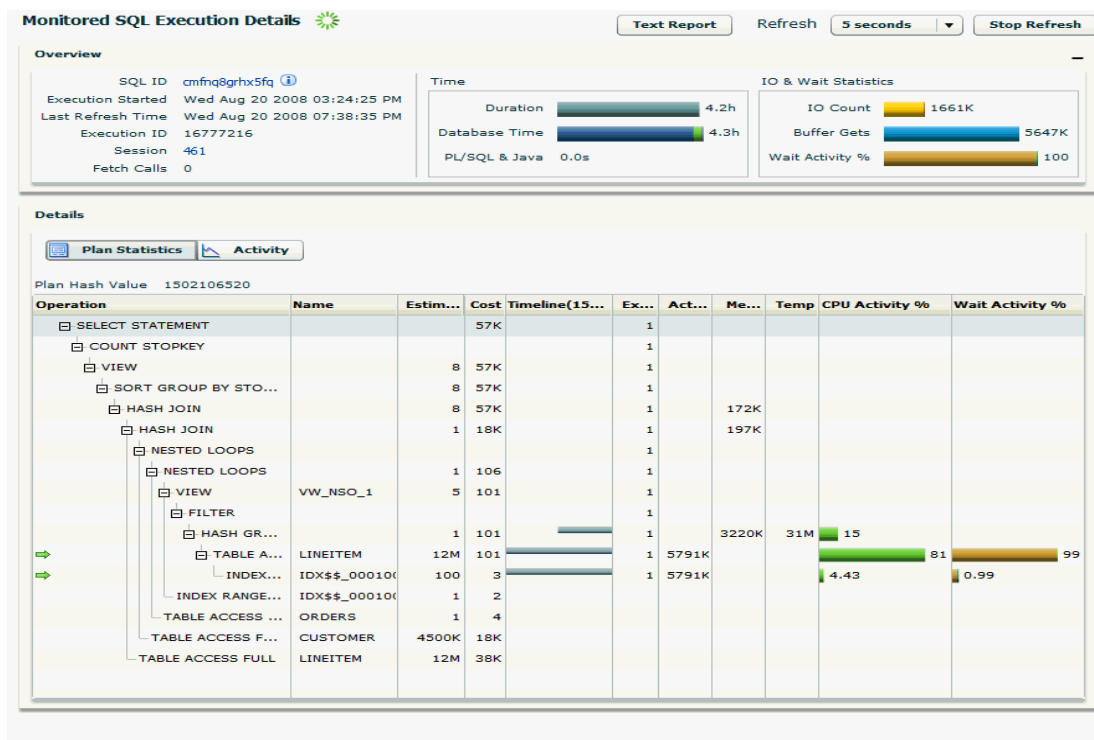


Рис. 2. Real-Time SQL Monitoring

Администраторы теперь имеют мощное средство для отслеживания сложных планов выполнения запросов, выявления плохих механизмов индексации и перекосов в параллельном выполнении запросов — все в реальном времени.

Ручная настройка SQL представляет собой очень сложный и проблемный процесс. Он требует глубокого знания в различных областях, отнимает много времени, требует детального знания структуры данных и модели использования данных приложения. Все эти факторы делают процесс ручной настройки SQL сложной и ресурсоемкой задачей, которая в конечном счете обходится очень дорого для бизнеса.

SQL Tuning Advisor является ответом Oracle на все недостатки и проблемы ручной настройки SQL. Он автоматизирует процесс настройки SQL путем всестороннего исследования всех возможных вариантов настройки SQL-предложения. Анализ и настройка осуществляются с помощью существенно улучшенного оптимизатора запросов, встроенного в ядро базы данных. SQL Tuning Advisor проводит шесть типов анализа.

- Анализ статистики: выявление объектов с отсутствующей или устаревшей статистикой, выдача соответствующих рекомендаций по устранению проблемы.
- SQL-профилирование: эта возможность, появившаяся в Oracle Database 10g, революционизировала подход к настройке SQL. SQL-профилирование позволяет настраивать SQL-предложения без каких-либо изменений кода приложения.
- Анализ путей доступа: во время этого анализа определяются новые индексы, которые могут значительно улучшить производительность запросов.
- Анализ структуры SQL: здесь проверяется неявное преобразование типов и даются рекомендации по изменению кода SQL.
- Степень параллелизма: SQL Tuning Advisor определяет, можно ли улучшить время выполнения с помощью параллельных потоков на определенных этапах выполнения SQL.
- Альтернативные планы: во время этого анализа SQL Tuning Advisor находит другие планы выполнения запроса, используя текущие и исторические данные производительности.



Настройка SQL-предложений больше не является прерогативой только специалистов. Oracle встроила эксперта по настройке SQL в ядро базы данных, позволив администраторам баз данных выполнять эту очень важную задачу за доли времени и затрат, необходимых для выполнения той же задачи вручную.

SQL Tuning Advisor может также работать в автоматическом режиме. В этом режиме советник по настройке запускается автоматически во время окна для плановых работ. Во время каждого выполнения советник выбирает тяжелые SQL-запросы в системе и генерирует рекомендации по их настройке.



Рис. 3. Результаты автоматической настройки SQL

Дизайн схемы базы данных может оказывать огромное влияние на производительность приложения. SQL Access Advisor предоставляет исчерпывающие рекомендации, как оптимизировать дизайн схемы, чтобы достичь максимальной производительности приложения. SQL Access и SQL Tuning Advisors вместе предоставляют полное решение для настройки приложений баз данных. SQL Access Advisor принимает данные из всех представляющих интерес источников, таких как кэш курсоров, Automatic Workload Repository (AWR), пользовательская нагрузка (набор SQL-предложений приложения), и даже может сам создавать гипотетическую нагрузку, если схема содержит измерения или отношения первичного/вторичного ключа.

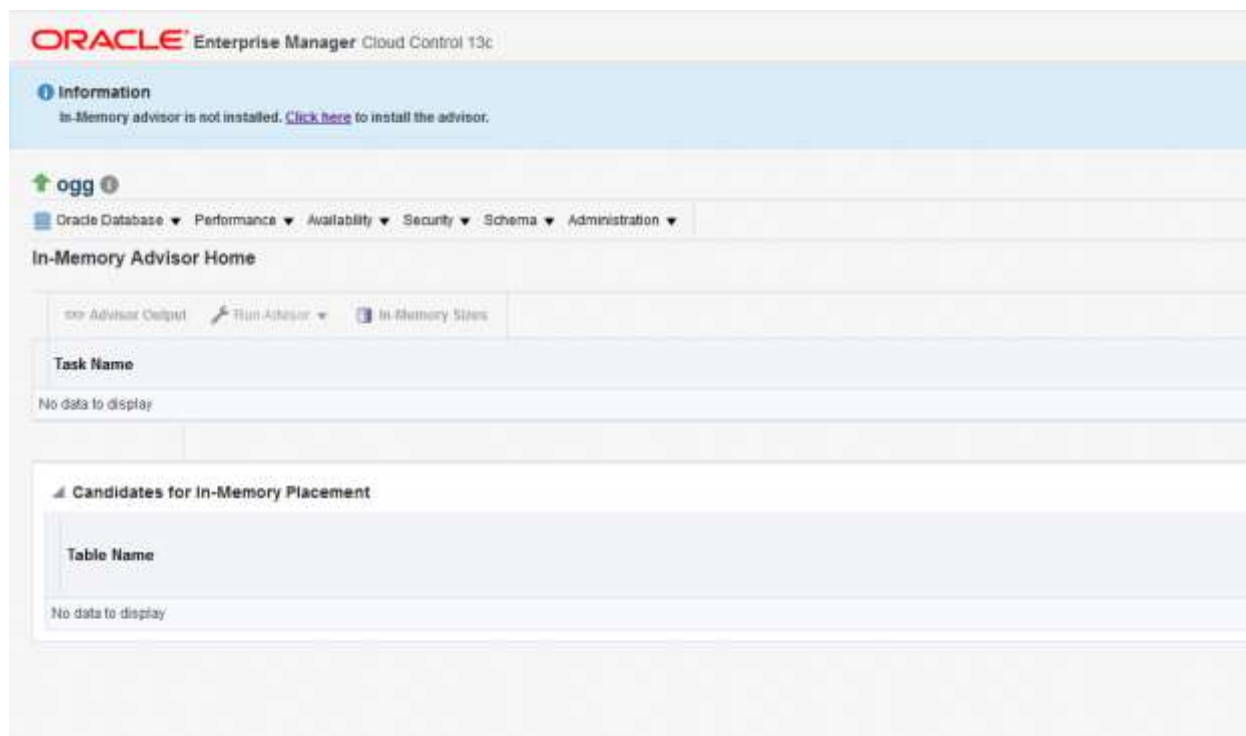
Он всесторонне анализирует всю нагрузку и дает рекомендации по созданию новых секций таблицы или индексов, удалению неиспользуемых индексов, созданию новых материализованных представлений и журналов. Определение оптимальной стратегии секционирования или индексирования для конкретной нагрузки является сложным процессом, требующим опыта и времени. SQL Access Advisor учитывает стоимость операций ввода/обновления/удаления в дополнение к запросам и дает соответствующие рекомендации, сопровождаемые количественной мерой ожидаемого выигрыша в производительности, а также скрипты, необходимые для реализации этих рекомендаций. SQL Access Advisor снимает тайну с процесса проектирования структур доступа. За счет автоматизации этой очень важной функции SQL Access Advisor устраняет необходимость в длительном, дорогостоящем, подверженном ошибкам процессе ручной настройки.

Oracle Tuning Pack также предоставляет возможность реорганизации объектов. Эффективное управление табличными пространствами путем удаления неиспользуемого пространства является не только хорошей практикой управления пространством, но также повышает производительность за счет сокращения ненужных операций ввода/вывода. Реорганизация используется для: 1) перестройки фрагментированных индексов и таблиц; 2) перемещения объектов в другое табличное пространство; и 3) пересоздания объектов с оптимальными атрибутами хранения.



Oracle Database In-Memory Advisor, входящий в Oracle Tuning Pack, позволяет оценить эффективность применения революционной технологии Oracle In-Memory — построчного хранения таблиц БД в памяти для ускорения аналитических запросов. In-Memory Advisor помогает подобрать подходящие таблицы-кандидаты для загрузки в память и установить оптимальные параметры для построчного хранения. In-Memory Advisor использует данные диагностического репозитория AWR для анализа нагрузки и выдачи рекомендаций. Также In-Memory Advisor полностью интегрирован в Oracle Enterprise Manager Cloud Control начиная с версии 13c.

Пример экрана In-Memory Advisor:



Database Lifecycle Management Pack

Database Lifecycle Management Pack является комплексным решением, которое помогает администраторам баз данных, систем и приложений автоматизировать процессы, необходимые для управления жизненным циклом базы данных Oracle, как в локальном ЦОД, так и в облаке Oracle Cloud. Он устраняет необходимость в ручных и трудоемких задачах, связанных с обнаружением, начальным провизионированием, патчиванием, управлением конфигурациями, непрерывным управлением изменениями и автоматизацией защиты от катастроф. Кроме того, Database Lifecycle Management Pack предоставляет средства управления соответствием промышленным и нормативным стандартам.

Database Lifecycle Management покрывает весь жизненный цикл баз данных, включая:

- обнаружение и отслеживание IT активов;
- начальное провизионирование, возможность развертывания базы данных за считанные минуты, в том числе и клонирование баз данных;
- непрерывное управление изменениями, комплексное управление патчами, обновлениями, изменениями схем и данных;
- управление конфигурациями, отслеживание IT активов, изменений конфигураций и детальный поиск систем, соответствующих стандартным конфигурациям;



- управление соответствием регламентным требованиям, управление промышленными и нормативными стандартами и отчеты соответствия им;
- автоматизация защиты от катастроф на уровне сайта;
- широкие возможности для управления, миграции, тестирования и развертывания в гибридном облаке, обеспечивая 100% повторное использование существующих навыков и практик управления ИТ-операциями.

Database Lifecycle Management Pack устраняет необходимость вручную отслеживать ИТ-активы, включая базы данных. Он предоставляет безагентные возможности для обнаружения физических серверов. Как только серверы обнаружены, их статус легко может быть поднят до управляемого с автоматическим обнаружением всех баз данных и серверов приложений, работающих на этих серверах. Автоматическое обнаружение ИТ-активов оказывает существенную помощь при консолидации и оптимизации ИТ инфраструктуры.

Database Lifecycle Management Pack поставляется с готовыми Deployment Procedures (процедурами развертывания) для провизионирования и патчирования базы данных Oracle (как для одиночного экземпляра БД, так и для RAC), включая Grid Infrastructure. Можно также создавать базу данных из эталонной системы или из золотого образа. Золотой образ вместе с конфигурационными деталями может быть упакован в профиль провизионирования (Provisioning Profile). Профили провизионирования можно сформировать из эталонной системы или загрузить с сайта Oracle.

Database Lifecycle Management Pack поддерживает весь жизненный цикл управления патчами, в том числе советники по патчированию, предустановочный анализ, установку и отчетность. Он интегрирован с My Oracle Support, чтобы предоставлять синхронизированное представление доступных и рекомендованных патчей. Эти патчи могут быть затем проанализированы перед установкой на наличие конфликтов. После этого можно установить одновременно несколько патчей для нескольких баз данных. Процедуры развертывания патчей призваны обеспечить максимальное удобство и минимальное время простоя. Enterprise Manager также предлагает другие методы минимизации простоя, такие как поочередное патчирование узлов для RAC и патчирование в другом Oracle Home.

Database Lifecycle Management Pack обеспечивает полную автоматизацию процесса развертывания схемы, позволяя захватывать метаданные объектов схемы приложения в виде базовых снимков словаря базы данных. Когда все изменения схемы завершены, DBA может сохранить их в этих снимках и перенести изменения на любую базу данных. Эти изменения проверяются на целевой базе данных, чтобы выявить любые возможные несоответствия и конфликты, такие как несоответствие типов данных или дублирование объектов. Это позволяет администраторам проактивно исправлять изменения до их применения. Когда нужно развернуть новый набор изменений приложения, они могут быть легко установлены, используя более новые версии этих базовых снимков словаря.

Также может быть выполнен анализ воздействия обновлений приложения на кастомизацию за счет автоматического выявления изменений схемы, специфичных для каждой кастомизации. И наоборот, если нет никаких изменений, влияющих на эти модули, менеджеры приложений могут пропустить тестирование больших частей приложения, тем самым ускоряя процесс обновления.

Database Lifecycle Pack предоставляет ведущие в отрасли решения по сравнению и поиску конфигураций, выявлению изменений конфигураций, контролю соблюдения регламентных требований. Администраторы могут определять золотые стандарты и базисные снимки для своих конфигураций, чтобы стандартизовать свое окружение согласно этим стандартам. Сравнивать конфигурации можно в режиме «один к одному» или «один ко многим», автоматически по расписанию или вручную. Enterprise Manager позволяет автоматически собирать информацию о конфигурациях и сохраняет ее в базе данных управления конфигурациями (CMDB). Отношения между конфигурационными элементами могут отображаться в виде топологии. Топологию приложения можно просматривать совместно с анализом влияния изменений, которые предполагается сделать, или с анализом первопричин в случае возникновения проблемы.



Для удовлетворения растущих потребностей клиентов по контролю и отчетности соблюдения стандартов, таких как PCI, SOX, CIS и COBIT, Enterprise Manager предлагает наборы стандартов (Compliance Frameworks). Эти наборы стандартов могут быть использованы как есть или расширены для поддержки пользовательских требований безопасности. Изменения в конфигурации базы данных и ее окружении могут обнаруживаться в реальном времени. Интеграция с модулем управления изменениями позволяет выявлять и получать отчетность о санкционированных и несанкционированных изменениях.

Oracle Site Guard — решение, позволяющее администраторам автоматизировать полное восстановление работоспособности сайтов после аварии. Site Guard исключает необходимость в специализированных навыках, освобождая ИТ-персонал от бремени вручную выполнять сложные операции по переходу на другой ресурс при сбое, уменьшая вероятность человеческой ошибки, которая может привести к длительному простое и потере данных. Операции по восстановлению работоспособности после отказа выполняются быстро и надежно, уменьшая риск и повышая уверенность, что план восстановления работоспособности будет работать, когда потребуется. Site Guard также может быть использован для координации частичного переключения при отказе или может быть использован для перемещения рабочей нагрузки между сайтами, чтобы облегчить плановое техническое обслуживание. Site Guard оркеструет координированное переключение на резервный сайт Oracle Fusion Middleware, Oracle Database и может быть расширен, чтобы обеспечить отказоустойчивость других компонентов центра обработки данных. Site Guard интегрируется с репликационными механизмами, с помощью которых осуществляется синхронизация основной и резервной сред и защита от потерь критически важных данных, Oracle Data Guard для Oracle данных и дисковой репликацией для данных файловых систем, внешних по отношению к Oracle Database.

Enterprise Manager теперь предоставляет единую консоль мониторинга и управления локальным развертыванием и развертыванием в Oracle Cloud. Развертывая агенты управления на виртуальных хостах Oracle Cloud, вы можете управлять базами данных в Oracle Cloud точно так же, как и локальными. Связь между агентами управления и локальными экземплярами службы управления Oracle защищена от внешних помех. Предоставляется поддержка для управления целевыми объектами PaaS облачных сервисов баз данных (DBCS) и облачных сервисов Java (JCS), а также поддержка JVMD для мониторинга JVM на ваших виртуальных хостах Oracle Cloud.

Oracle Management Cloud (OMC)

В 2015 году компания Oracle представила семейство новых облачных сервисов под общим названием Oracle Management Cloud (OMC) — интегрированное решение для проактивного мониторинга и ИТ-аналитики, в составе которого четыре основных модуля:

- Application Performance Monitoring,
- Log Analytics,
- Infrastructure Monitoring,
- IT Analytics.

Эти новые инструменты Oracle помогают решать ряд задач:

1. обеспечивать эффективный сквозной мониторинг производительности бизнес-приложений в реальном времени, следить за статистикой работы конечных пользователей, определять, насколько быстро открываются веб-страницы, какие возникают ошибки, какова их первопричина;
2. анализировать, коррелировать и визуализировать события из лог-файлов, находить первопричины различных проблем с помощью механизмов Machine Learning;



3. осуществлять планирование ресурсов ИТ-ландшафта: измерять загрузку баз данных и серверов приложений, оценивать их при помощи алгоритмов ML, прогнозировать утилизацию основных ресурсов: CPU, I/O, памяти на будущие периоды;
4. обеспечивать мониторинг компонентов ИТ-ландшафта в проактивном режиме, настроить систему «раннего предупреждения», прогнозирования инцидентов, выявлять аномалии и негативные тренды нагрузки.

Если Oracle Enterprise Manager наиболее полезен для оперативного администрирования и управления ИТ-инфраструктурой, то ОМС добавляет множество возможностей для проактивного мониторинга on-premise и облачных баз данных, приложений, инфраструктур. Он удобен для мониторинга автономных БД. ОМС не требует сложной установки и позволяет мониторить не только продукты Oracle, но и продукты третьих фирм. Использование метода мониторинга отклонений поведения позволяет ОМС на базе огромного потока собираемых данных выявлять (и показывать DBA) только ситуации, требующие пристального внимания, влияющие на текущую и будущую работу приложений.

Решение ОМС может быть очень полезно для службы эксплуатации ИТ-инфраструктуры, администраторов приложений, ИТ-архитекторов, руководителей ИТ-подразделений, специалистов DevOps. Когда речь идет о бизнес-ориентированном мониторинге, о том, чтобы локализовать проблему, найти ее в сложном технологическом стеке, состоящем из множества систем и компонентов, таких как базы данных, серверы приложений и т. д., которые установлены на разных физических и виртуальных серверах, различных дата-центрах, — для этого нужны специализированные инструменты. Именно таковыми и являются сервисы Oracle Management Cloud. Несмотря на то что сервисы ОМС являются частью публичного облака Oracle Cloud, они успешно работают с обычными on-premise ИТ-ландшафтами и не требуют от заказчика миграции его инфраструктуры в облако, также ОМС может интегрироваться с локальным инструментом мониторинга и управления Oracle Enterprise Manager Cloud Control (OEM), таким образом, оба инструмента ОМС + OEM составляют законченную «экосистему», охватывающую все задачи мониторинга.

Сервисы Oracle Management Cloud интегрированы с облачной платформой Oracle Cloud Infrastructure (OCI) и построены на современных технологиях обработки и анализа данных, которые включают Big Data, обработку событий в реальном времени, библиотеку Machine Learning. В основе платформы ОМС используются программно-аппаратные комплексы обработки данных от Oracle. Возможности платформы позволяют обрабатывать огромные объемы телеметрии практически в реальном времени.

ОМС построен с учетом обработки и анализа только диагностических данных, что полностью исключает риски переноса персональных данных в облако. Тем не менее весь поток передаваемых технологических данных в обязательном порядке шифруется, также есть средства маскирования чувствительной, по мнению заказчика, информации перед отправкой в облако Oracle Management Cloud. Исходными данными для инструментов мониторинга ОМС являются: лог-файлы, машинные данные, трассировочные файлы, метрики производительности, диагностические данные из репозитория EM, журналы аудита и т. д. (рисунок 1).





рис. 1

Рис. 1. Oracle Management Cloud собирает, анализирует и визуализирует различные виды телеметрии

Также важным источником данных для Oracle Management Cloud являются данные репозитория Enterprise Manager, если он используется для мониторинга инфраструктуры заказчика (рисунок 2).

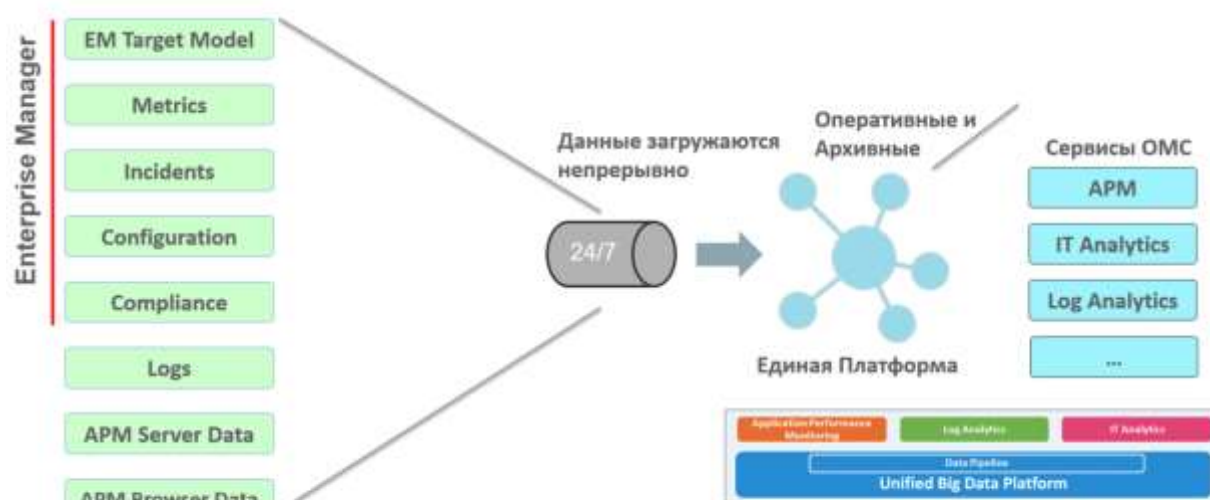


рис. 2

Рис. 2. Основа Oracle Management Cloud — аналитическая платформа, позволяющая обрабатывать огромные объемы данных

Oracle Application Performance Monitoring Cloud Service (APM)

Сервис Application Performance Monitoring (рисунок 3) предназначен для мониторинга веб-приложений, построенных на платформе Oracle WebLogic, Jboss, Apache Tomcat, node.js, WebSphere и т. д. Задачи APM — «профилирование» веб-приложения на различных уровнях: от конечного пользователя до инфраструктуры, помощь в поиске первопричин инцидентов, предоставление единого интерфейса для службы эксплуатации и разработчиков и, конечно, проактивный мониторинг конечных пользователей; сервис Application Performance Monitoring позволяет определить опыт взаимодействия конечного пользователя с Вашим веб-приложением. При этом APM позволяет производить контекстный нисходящий анализ производительности приложения от уровня пользователя, визуализировать взаимодействие модулей приложения и, если нужно,



переключаться на «нижний» уровень, чтобы анализировать возможные связанные проблемы инфраструктуры.

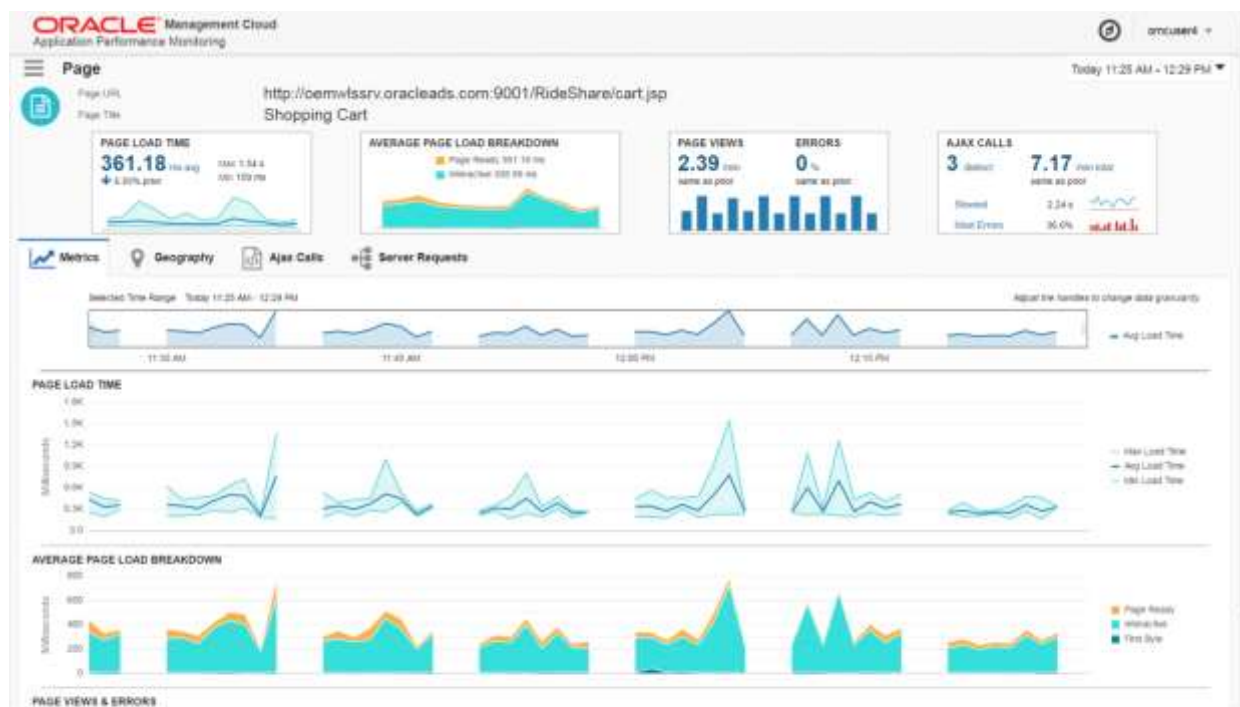


рис. 3

Рис. 3. Пример экрана сервиса APM: метрики производительности со стороны пользователя веб-приложения

Служба эксплуатации и разработчики могут использовать Application Performance Monitoring в качестве единого источника информации при диагностике производительности вплоть до уровня кода и SQL-запросов. Разработчики получают доступ к ценным диагностическим данным и работают совместно со службой эксплуатации — они дополняют друг друга, это обеспечивает быстрое устранение проблем, повышается качество сопровождения (DevOps). Сервис Application Performance Monitoring основан на технологии «инструментирования» кода. Перекомпиляция не требуется, специальные агенты встраиваются в готовый код приложения и веб-страниц, измеряя производительность и передавая метрики в публичное облако через прокси-сервер или Gateway в безопасном режиме.

Log Analytics Cloud Service

Инструмент Log Analytics (рисунок 4) может работать в связке с Application Performance Monitoring. Именно этот сервис непосредственно работает с журнальными файлами — он занимается поиском, сбором, индексированием, хранением данных из лог-файлов, а также анализом событий в журналах и визуализацией.



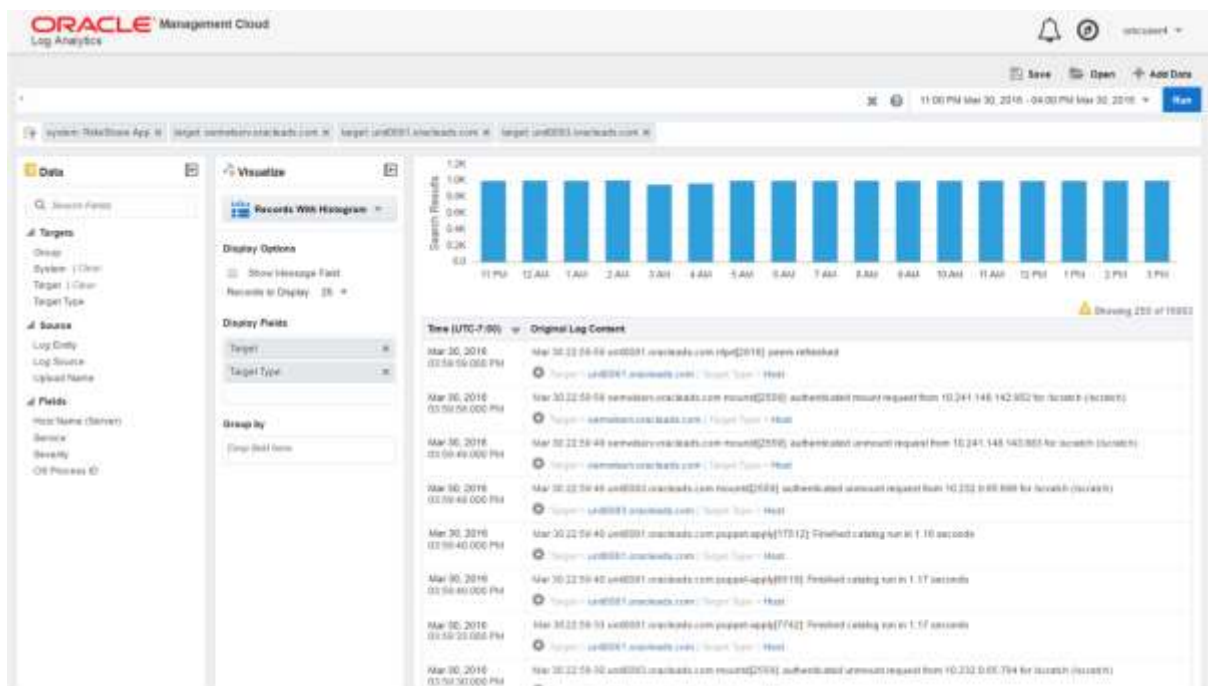


Рис. 4. Сервис Log Analytics — консоль для визуализации и анализа событий из логов инфраструктуры и приложений

Особенностью инструмента Log Analytics является то, что он не зависит ни от компонентов инфраструктуры, ни от форматов журнальных файлов. Сервис «из коробки» поддерживает около 250 форматов лог-файлов — журналы баз данных, Enterprise Manager, Fusion Middleware, веб-серверов, кластерного программного обеспечения, операционных систем Linux, планировщиков и т.д. Для поддержки нестандартного журнального файла есть простые визуальные средства для создания т. н. парсера — модуля анализа записей журнала. Используя алгоритмы Machine Learning, сервис Log Analytics может из тысяч рядовых сообщений, накопившихся в Ваших журнальных файлах, например, за последние сутки, выявить несколько ценных сообщений об ошибке — те самые ошибки, которые, возможно, и являются первопричиной инцидента. После обнаружения проблемы появляется меню Show 1 Minute context — т. е. отображение контекста системы за минуту до сообщения об ошибке и спустя минуту после него. Log Analytics является готовым решением и может автоматически применять алгоритмы кластеризации, корреляции и классификации событий из логов, не требуя от пользователя специальных навыков Data Science.

Вся инфраструктура самого сервиса Log Analytics находится в публичном облаке Oracle. Для пользователя предоставляется удобный веб-интерфейс, стандартные информационные панели, визуализирующие ключевые события, зафиксированные в журнальных файлах через диаграммы различного типа, а также можно создавать виджеты и комбинировать их в собственные информационные панели.

Log Analytics собирает данные из лог-файлов в режиме реального времени, индексирует и хранит в публичном облаке Oracle на высокопроизводительной платформе Big Data, что позволяет эффективно работать с огромными объемами журнальных данных, выполнять сортировку, поиск, группирование, машинный анализ. Наконец, Log Analytics прозрачно интегрирован с другими сервисами ОМС: Application Performance Monitoring и Infrastructure Monitoring. Например, кнопка Drilldown to Log позволяет при анализе транзакции веб-приложения переходить с уровня сервера приложений на уровень конкретного, связанного элемента инфраструктуры (например, базы данных Oracle).

IT Analytics Cloud Service

Бизнесу необходимы инструменты для понимания функционирования текущего ИТ-ландшафта, планирования мощностей, выявления проблемных участков и т.д. Существующие системы этих задач



не решают, т. к. они фокусируются на отдельных компонентах ИТ-ландшафта, таких как база данных или промежуточное ПО, и не позволяют определить, как системы функционируют в масштабе ЦОД — за определенный период времени, в разрезе приложений, систем центров затрат и т. д.

Поэтому еще одной важной частью набора сервисов Oracle Management Cloud является сервис IT Analytics (рисунок 5). Область его применения — анализ текущего ИТ-ландшафта. Задачи, которые он выполняет: анализ использования ресурсов, распознавание систематических проблем производительности и планирование ресурсов. Цели анализа ИТ-ресурсов — выявление проблемных участков, сравнение нагрузок по различным периодам, выявление максимального потребления ресурсов по различным измерениям, оптимизация ресурсов и планирование роста через предсказание нагрузок.

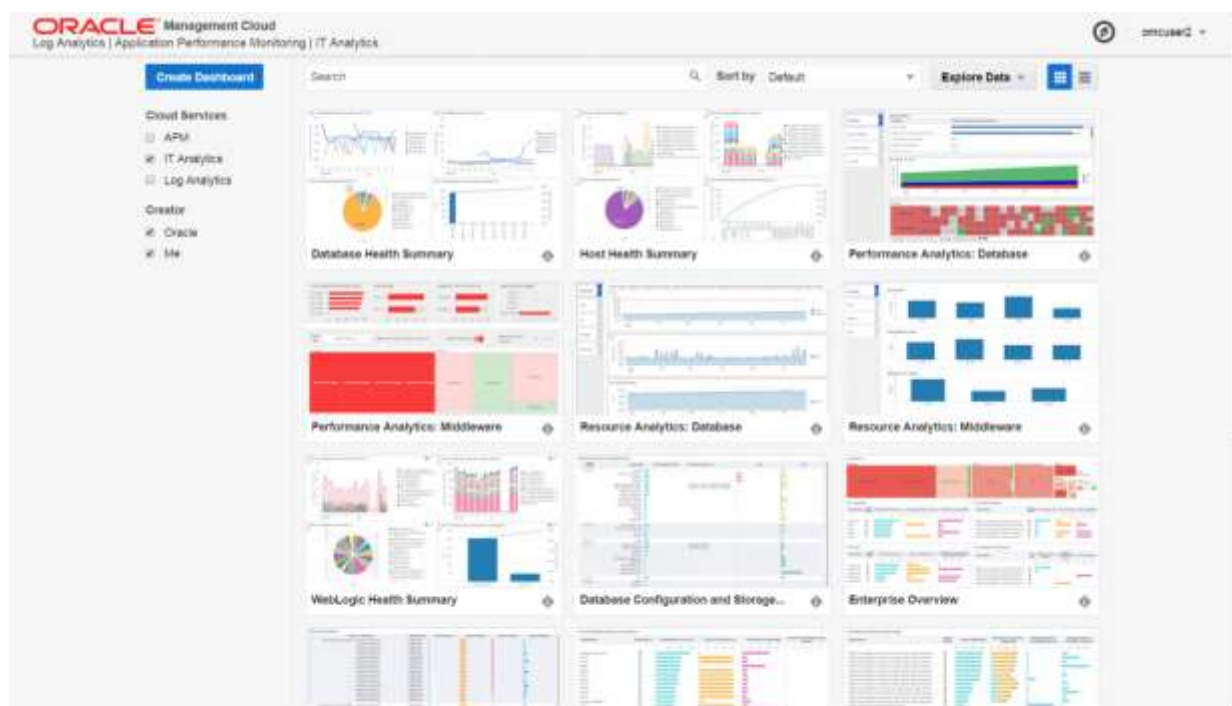


Рис. 5. Различные типы готовых аналитических дашбордов в Oracle Management Cloud

Сервис позволяет выполнять также анализ производительности по различным критериям, таким как процессные ресурсы, ввод-вывод, память, объемы хранения данных, производительность промежуточного ПО, производительность Java. Классификация и прогнозы производительности баз данных и даже отдельных SQL-запросов. Возможности визуализации данных позволяют упростить поиск по различным критериям, визуализировать KPI по ключевым компонентам для анализа текущего использования и трендов, а также использовать информационные панели. Например, показать график суммарного использования CPU по бизнес-критичным БД версии 12.1.0.2 за последние 3 месяца. Результат использования упомянутых возможностей для бизнеса — проактивный подход к мониторингу и планированию (улучшение использования ресурсов через выявление недостаточно или избыточно нагруженных узлов), снижение операционных расходов (затрат на анализ и выявление проблем) и помощь в трансформации ИТ (переход к проактивному подходу и стратегическому планированию).

Infrastructure Monitoring Cloud Service

Современные требования к системам мониторинга включают в себя не только возможности визуализации различных метрик и отсылки уведомлений при достижении критических значений, но и способы прогнозирования или обнаружения проблем на ранней стадии, а также способности систем «обучаться» на конкретном профиле нагрузки и автоматически распознавать аномалии.



Сервис Infrastructure Monitoring в составе пакета Oracle Management Cloud позволяет осуществлять интеллектуальный мониторинг по широкому списку систем гетерогенного ИТ-ландшафта. В составе решения имеются возможности настройки системы раннего предупреждения по ряду метрик (Early Warning Alert), таким образом, Infrastructure Monitoring непрерывно следит за распределением значений метрики по времени и применяет статистический алгоритм для прогнозирования достижения установленного порога. Также Infrastructure Monitoring обучается на профиле нагрузки и автоматически фиксирует аномальные «выбросы», которые не укладываются в область нормальных значений. Очень интересная возможность Infrastructure Monitoring — корреляция метрик, которая особенно полезна при поиске первопричин аномального поведения системы.

Пример экрана Infrastructure Monitoring

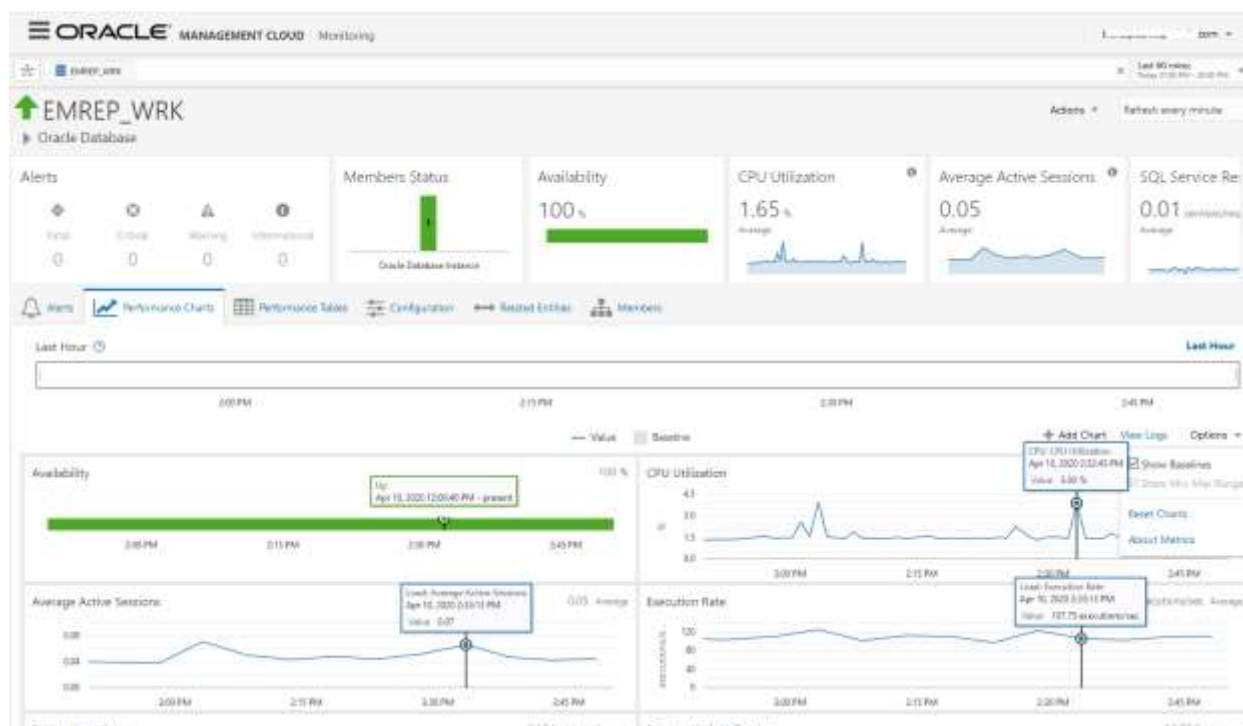


Рис. 6. Infrastructure Monitoring: корреляция метрик и определение аномалий за выбранный период

Еще раз подчеркнем, что сервисы Oracle Management Cloud и Ваш локально установленный Oracle Enterprise Manager (при его наличии) работают совместно. Консолидация и визуализация данных из репозитория Enterprise Manager и различных метрик, собираемых агентами Oracle Management Cloud, дает очень интересные возможности: проактивный подход к решению проблем, своевременный бизнес-ориентированный мониторинг, улучшение взаимодействия разработчиков и службы эксплуатации (DevOps).

Oracle Management Cloud — набор сервисов в публичном облаке Oracle, предлагаемых по подписке; это значит, что заказчику нет необходимости дополнительно инвестировать и поддерживать дорогостоящую, сложную инфраструктуру для инструментов мониторинга. Все, что требуется — установить агентов и настроить передачу данных в Облако — система немедленно готова к эксплуатации. За стабильность, производительность и своевременное обновление сервисов ОМС отвечает Oracle. Сервисы Oracle Management Cloud не собирают и не передают в облако бизнес-данные или персональные данные! Собирается и анализируется только диагностическая информация, метрики, журнальные данные, которые при необходимости можно «замаскировать» перед отправкой в облако.

С помощью Oracle Management Cloud различные команды ИТ-службы, такие как разработчики и служба эксплуатации, смогут найти общий язык и работать эффективно.



Продукты Oracle



Прочие серверные продукты



Golden Gate

Oracle GoldenGate обеспечивает сбор, трансформацию, маршрутизацию и доставку данных между БД в гетерогенной среде в реальном времени. Он перемещает данные с очень большой скоростью, низкой нагрузкой на сервер СУБД и задержкой менее секунды, но при этом сохраняя транзакционную целостность. Компонентная архитектура Oracle GoldenGate помогает компаниям решать задачи, связанные с обеспечением непрерывной доступности и интеграции данных в реальном времени, возникающие при построении корпоративных систем. Oracle GoldenGate состоит из трех компонентов: Capture, Pump и Delivery. Capture отвечает за захват изменений в базе данных — источнике, Pump передает изменения на другой сервер, Delivery вносит изменения в базу данных — приемник.

Такой модульный подход позволяет каждому компоненту выполнять свои задачи независимо, что гарантирует ускорение репликации и целостность данных.



Oracle GoldenGate может быть развернут в двух вариантах: GoldenGate Classic и GoldenGate Microservices Architecture (OGG MA).

GoldenGate Classic — это традиционная архитектура GoldenGate, которая содержит стандартные функции захвата, доставки и применения изменений и управляется с помощью утилиты командной строки ggsci.

OGG MA — это архитектура на основе REST API микросервисов, которая позволяет устанавливать, настраивать, отслеживать службы Oracle GoldenGate и управлять ими через веб-интерфейс.

В будущем OGG MA должна полностью заменить GoldenGate Classic, поэтому рекомендуется использовать OGG MA, если есть такая возможность.

Использование GoldenGate позволяет решать следующие задачи.

- **Строить хранилища данных реального времени.** GoldenGate обеспечивает постоянный захват измененных данных и их перемещение из OLTP-источников в единое хранилище данных. При необходимости Oracle GoldenGate интегрируется с ETL-инструментами (например, Oracle Data Integrator).
- **Получать оперативную отчетность.** GoldenGate позволяет разгрузить промышленную базу данных от решения задач отчетности за счет переноса необходимых данных на более дешевые серверы в реальном времени.
- **Выносить «нагрузку на чтение».** Гарантировать высокую производительность промышленных систем за счет выноса нагрузки «на чтение» на отдельную систему (возможно, на другой платформе или СУБД).
- **Снижать время простоя.** Не останавливать бизнес во время обновления систем, миграции и плановых работ.
- **Интегрировать оперативные данные.** GoldenGate позволяет интегрировать оперативные данные между транзакционными системами в реальном времени. Наличие в организации SOA-инфраструктуры (построенной, например, на базе Oracle SOA Suite) позволяет обмениваться измененными данными по сервисной шине с использованием адаптеров GoldenGate.
- **Интеграция с Big Data.** GoldenGate обеспечивает потоковую интеграцию в реальном времени со всеми основными Big Data приемниками, такими как Apache Hadoop (HDFS, HBase, Hive), инфраструктурами



обмена сообщениями (Apache Kafka, JMS), NoSQL базами данных ((MongoDB, Cassandra и Oracle NoSQL), Elasticsearch, JDBC, Amazon WebServices (AWS S3, Kinesis, Redshift), Microsoft Azure и облачными хранилищами данных (Google BigQuery, Snowflake).

Потоковая аналитика в реальном времени. GoldenGate прозрачным образом интегрируется с Oracle Stream Analytics, позволяя пользователям идентифицировать интересующие их события, выполняя запросы к потокам событий в реальном времени. Это позволяет создавать настраиваемые операционные информационные панели, которые обеспечивают мониторинг в реальном времени, преобразовывают потоковые данные или генерируют оповещения на основе анализа потоков.

Ключевые особенности:

- Работа в режиме, близком к реальному времени: задержка менее секунды.
- Поддержка гетерогенных систем: может получать изменения из СУБД Oracle, MSSQL, DB2, MySQL, Sybase и доставлять их в вышеперечисленные СУБД, а также TimesTen, PostgreSQL, плоские файлы, JMS-очереди, системы Big Data.
- Надежность: гарантируется доставка и сохраняется транзакционная целостность.
- Высокая производительность.
- Минимальная нагрузка на источник.

Oracle Big Data SQL

Как говорилось ранее, альтернативой наполнения данными централизованного реляционного хранилища является его объединение с внешними системами специальными интеграционными механизмами, позволяющими работать с данными в этих источниках так же, как и с данными самой СУБД. Это особенно оправданно, если объемы данных в этих источниках значительно превышают объемы самого хранилища или представлены в нереляционном формате. Примером таких систем являются хранилища Big Data, построенные на продуктах стека Hadoop. Перемещение данных из них часто нецелесообразно, особенно если в них применяется подход schema on read.

Big Data SQL — это инновация в области виртуализации данных от Oracle, представляющая собой специализированный коннектор, предназначенный для интеграции реляционной СУБД Oracle и кластера Hadoop. Однако едва ли не более ценной при этом является возможность распространить политики безопасности СУБД на данные хранимые в Big Data источниках, обеспечив таким образом единую систему контроля доступа как для реляционных данных, так и для данных, расположенных в кластере Hadoop.

Используя Oracle Big Data SQL, организации могут:

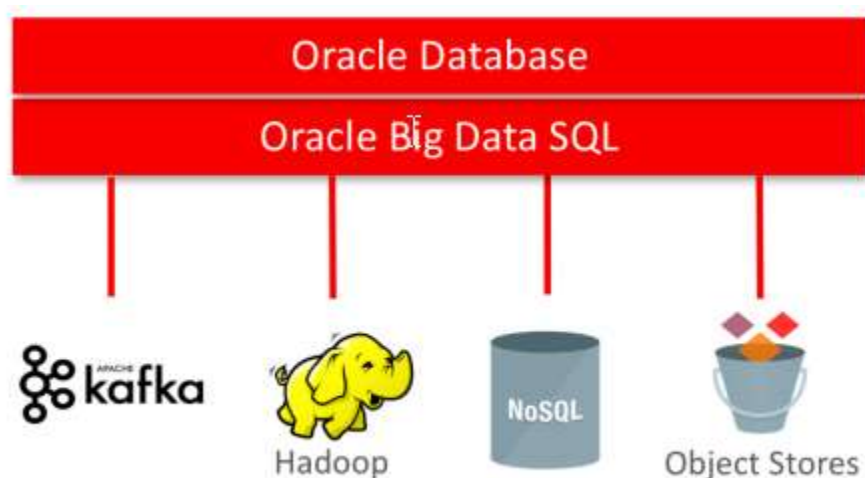
- Использовать Oracle SQL для запроса и анализа данных в Apache Hadoop, Object Store, Apache Kafka и NoSQL
- Максимизировать производительность запросов ко всем данным, используя передовые методы, такие как Smart Scan, Aggregation Offload, Partition Pruning, Storage Indexes, Bloom Filters and Predicate Push-Down в распределенной архитектуре
- Интегрировать анализ больших данных в существующие приложения и архитектуры
- Расширить политики безопасности и доступа из базы данных Oracle к данным в Apache Hadoop, Object Store, Apache Kafka и NoSQL

При работе с большими наборами данных, хранящимися в разнородных системах, иногда трудно понять, где данные находятся, не говоря уже о том, чтобы понять, как данные структурированы. Big Data SQL использует



внешние таблицы в Oracle Database, которые дают пользователям единое расположение для каталогизации и защиты данных в Hadoop, Object Store, Kafka и NoSQL системы. Big Data SQL отслеживает метаданные о внешних источниках данных — кластеры и таблицы внутри них — без перемещения или копирования данных. Внешние таблицы для Big Data SQL обеспечивают:

- Полную интеграцию метаданных и запросы, объединяющие данные из базы данных Oracle с данными из Hadoop, Object Store, Apache Kafka и NoSQL
- Автоматическое сопоставление метаданных, хранящихся в HCatalog (или Hive Metastore), с таблицами Oracle
- Поддержку нескольких кластеров, позволяющую одной базе данных Oracle работать с несколькими Hadoop кластерами.
- Расширенные параметры доступа, позволяющие администраторам баз данных управлять отображением столбцов и доступом к данным.



Oracle Secure Backup

Oracle Secure Backup обеспечивает централизованное управление резервным копированием для разнородных файловых систем и базы данных Oracle на диск, ленту и в облако. Oracle Secure Backup предлагает расширенные функциональные возможности резервного копирования и восстановления с минимальными затратами на сопоставимые решения, что снижает сложность и увеличивает возврат инвестиций для клиентов Oracle. Oracle Secure Backup обладает высокой масштабируемостью и управляется централизованно с использованием единой консоли и общего интерфейса управления для всего спектра серверов и устройств NAS. Помимо защиты файловой системы, Oracle Secure Backup имеет встроенную интеграцию с базой данных Oracle, обеспечивая оптимизированное резервное копирование/восстановление базы данных.

Oracle Secure Backup обеспечивает централизованное управление резервным копированием для всей ИТ-среды:

- Интеграция базы данных Oracle с Recovery Manager (RMAN). Оптимизированная производительность, ускоряющая резервное копирование на 25–40 % при меньшей загрузке ЦП на 10 %
- Повышение скорости передачи данных для Exadata и/или Oracle Database Appliance с использованием RDS/RDMA (Reliable Datagram Sockets over Remote Data Memory Access)
- Защита данных на уровне файловых систем для серверов UNIX/Windows/Linux



- Защита сетевых систем хранения (NAS) усиливается с использованием Network Data Management Protocol (NDMP)
- Создание резервной копии данных БД Oracle не только на дисковые и ленточные библиотеки, но и в облачные системы хранения, включая Oracle Cloud Infrastructure (OCI) Object Storage
- Дублирование резервных копий на основе правил: с диска на ленту или с диска в облако
- Сжатие и проверка контрольной суммы резервных копий

Экспоненциальный рост данных приводит к хаосу в инфраструктурах резервного копирования и восстановления, затрудняя соблюдение окон резервного копирования и соглашений об уровне обслуживания (SLA). Oracle Secure Backup решает эти проблемы путем динамического совместного использования ленточных библиотек и пулов дисков на нескольких медиасерверах для повышения эффективности использования устройств и производительности. При использовании Oracle ZFS Storage Appliance в качестве дискового пула Oracle Secure Backup данные резервного копирования и восстановления передаются через NDMP. Возможно использование дисковых пулов в качестве временных контейнеров. Резервная копия, записанная на промежуточное устройство пула дисков, копируется или перемещается на ленточное или облачное устройство позднее. Образы резервных копий, размещенные в пуле дисков, могут быть запланированы для перемещения на ленточное или облачное устройство на основе либо расписания, либо объема данных, либо для контроля минимального времени, в течение которого образ резервной копии гарантированно остается на устройстве. Поддержка облачных хранилищ позволяет заменить процедуру хранения резервных копий на основе ротации лент на автоматическое решение на основе правил «диск-облако», которое проще в управлении и экономически выгодно.

Для обеспечения безопасности данных как на магнитной ленте, дисках, так и в облачном хранилище Oracle Secure Backup выполняет шифрование резервных копий на основе политик, а ключи шифрования резервных копий надежно хранятся на административном сервере. Ключи шифрования могут быть сгенерированы прозрачно (случайным образом) или с использованием ключевой фразы и регулярно обновляются в соответствии с заданным пользователем расписанием регенерации ключей. Резервные копии могут быть зашифрованы с использованием опций шифрования LTO и T10000.

Secure Backup 18.1 обеспечивает еще больший контроль резервных копий. Контрольная сумма генерируется и сохраняется до того, как резервная копия записывается или копируется на целевое устройство, и используется для проверки целостности данных во время следующих заданий резервного копирования и восстановления.

Oracle Secure Backup предоставляет широкий набор инструментов управления: веб-приложение OSB, командная строка или Oracle Enterprise Manager (EM) Cloud Control.

ШИРОКИЙ НАБОР ПОДДЕРЖИВАЕМЫХ УСТРОЙСТВ

Oracle Secure Backup поддерживает более 200 ленточных устройств, использующих различные интерфейсы SCSI, fibre, SAS. К серверам физические или виртуальные ленточные устройства могут быть подключены как непосредственно, так и по сети (Storage Area Network, SAN).

ПОСТАВЩИКИ ЛЕНТОЧНЫХ УСТРОЙСТВ: ADIC, Breece Hill, Copan, Dell, FalconStor, HP, IBM, Overland Storage, Qualstar, Quantum, Sepaton, SONY, Spectra Logic, StorageTek

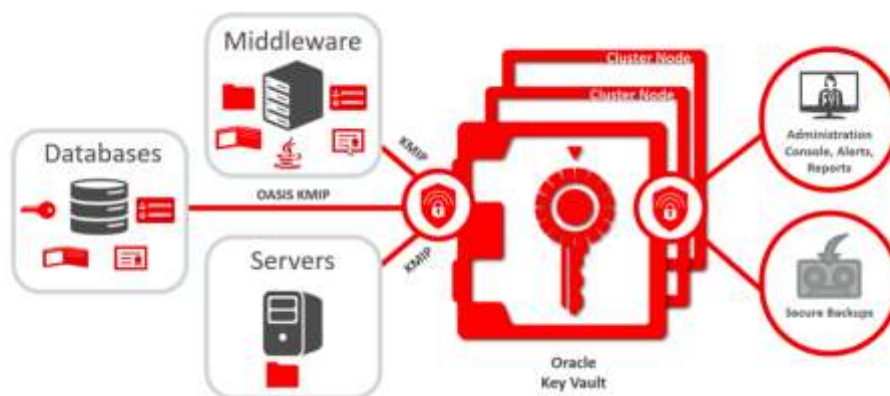
ПОДДЕРЖКА NDMP УСТРОЙСТВ: EMC, Network Appliance Pillar Data Systems, Sun ZFS Storage Appliance

ОПЕРАЦИОННЫЕ СИСТЕМЫ: Linux, Windows, SolarisSPARC, Solaris x-86, AIX, HP-UX Itanium



Key Vault

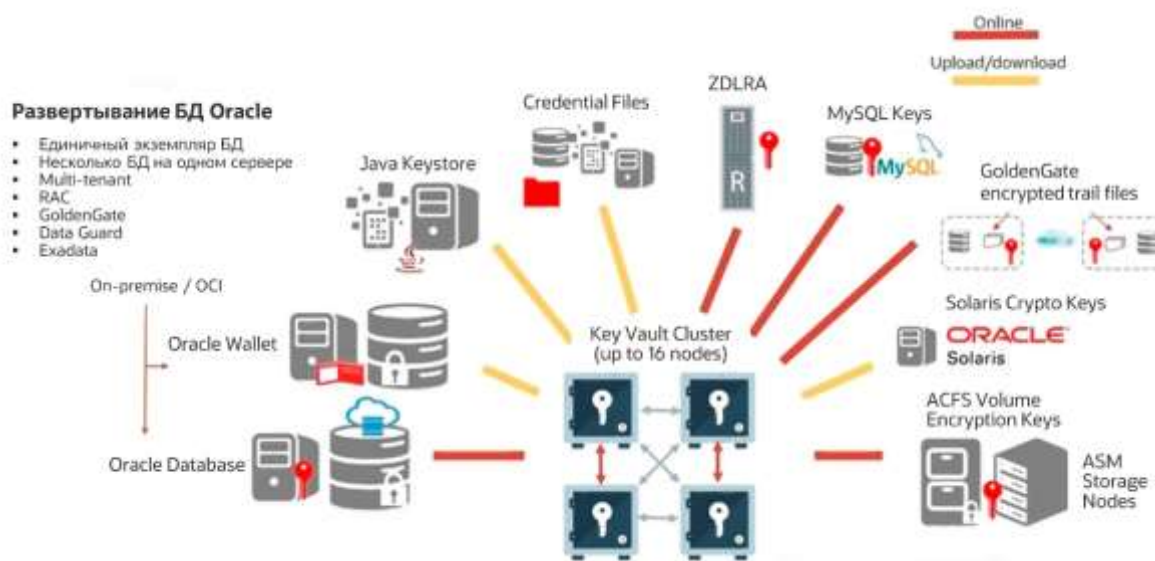
Oracle Key Vault (OKV) позволяет клиентам применять шифрование и другие решения для обеспечения безопасности, централизованно управляя ключами шифрования базы данных Transparent Data Encryption (TDE), Oracle Wallet, хранилищами ключей Java и файлами учетных данных. OKV поддерживает архитектуру высокой доступности для обеспечения непрерывного функционирования служб и географического охвата.



Управление мастер-ключами шифрования данных

OKV обеспечивает физическое разделение ключей шифрования и зашифрованных данных, которые часто требуются для соблюдения нормативных требований. Он централизованно управляет мастер-ключами TDE через прямое сетевое подключение в качестве альтернативы использованию локальных файлов Oracle Wallet. Это также устраняет операционные проблемы управления файлами кошелька, такие как периодическая ротация паролей, резервное копирование файлов кошелька и восстановление из ситуаций забытых слов. Общий доступ к мастер-ключам поддерживает Oracle Multitenant Database instances, а также Oracle Real Application Clusters (RAC), Oracle Data Guard и Oracle GoldenGate. Существующие мастер-ключи, используемые для шифрования данных в базах данных Oracle, можно легко перенести из кошельков Oracle в хранилище ключей Oracle.

Oracle Key Vault обеспечивает оперативное управление мастер-ключами без необходимости использования локальных Oracle Wallet



Управление Oracle Wallet, хранилищами ключей java и файлами учетных данных

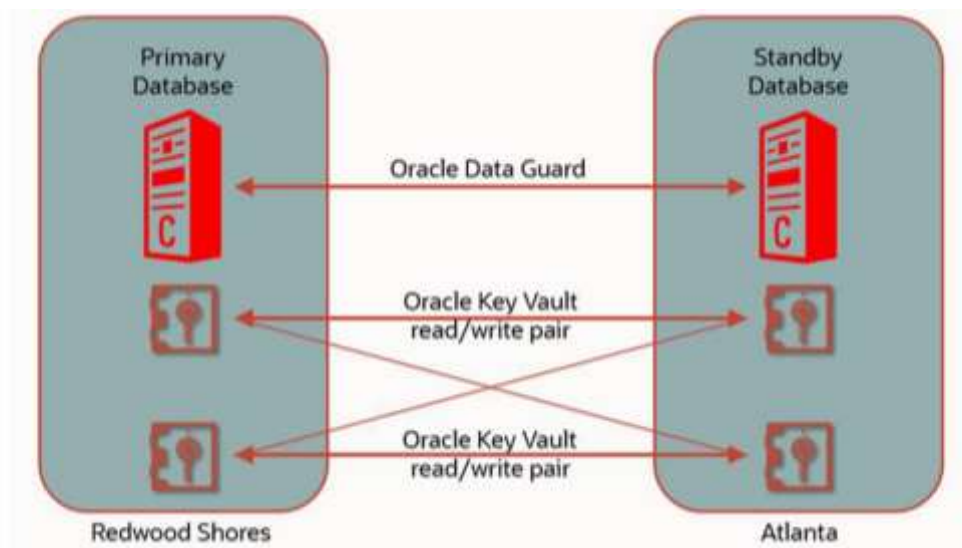
Администраторы часто вручную копируют Oracle Wallet и хранилища ключей Java между серверами и кластерами серверов. OKV упрощает совместное использование кошельков между кластерами баз данных, такими как Oracle RAC, Oracle Data Guard и Oracle GoldenGate. Безопасный обмен кошельками также облегчает перемещение зашифрованных данных с помощью Oracle Data Pump и Oracle Recovery Manager (RMAN). OKV надежно архивирует эти файлы и позволяет восстанавливать кошельки и хранилища ключей, когда они ошибочно удалены или если их пароли забыты.

Во многих организациях файлы учетных данных, содержащие SSH-ключи, файлы Kerberos keytab и аналогичные файлы учетных данных, также широко распространяются без соответствующих защитных механизмов. OKV создает резервные копии файлов учетных данных для долгосрочного хранения и восстановления. OKV легко восстанавливает эти файлы, когда это необходимо, проверяет доступ к ним и совместно использует их в доверенных конечных точках.

Постоянно доступная и масштабируемая кластерная архитектура

Узлы Oracle Key Vault разворачиваются как часть кластера, обеспечивая непрерывную доступность и географический охват. OKV поддерживает до 16 узлов в кластере, автоматически синхронизируя любые изменения, внесенные на одном узле во всем кластере.

Каждая конечная точка базы данных прозрачно поддерживает свой собственный список доступных узлов и постоянно осведомлена об изменениях в кластере. Если текущий узел становится недоступным, конечная точка прозрачно переходит на другой соседний узел. Чтобы еще больше повысить устойчивость к сетевым сбоям, Oracle Key Vault позволяет дополнительно создавать постоянный кэш на серверах баз данных, чтобы базы данных оставались полностью функциональными, если сетевое подключение ко всем узлам будет нарушено.



Уникальная архитектура разворачивания кластеров Oracle Key Vault обладает высокой масштабируемостью. Клиенты могут разворачивать пары узлов чтения и записи в разных центрах обработки данных, чтобы обеспечить конечным точкам доступ к локальному узлу для выполнения операций чтения и записи. Кроме того, кластерная архитектура поддерживает разворачивание дополнительных узлов только для чтения, чтобы обеспечить локальное присутствие службы ключей для небольших центров обработки данных. Наконец, каждый сервер Oracle Key Vault разворачивается на стандартных аппаратных платформах, размер которых может быть определен для удовлетворения самых высоких нагрузок. В результате получается служба поддержки управления ключами, которая может работать с тысячами баз данных, развернутых по всему миру, с экстремальной доступностью и высоким уровнем обслуживания.



Администрирование

Консоль управления на основе браузера позволяет легко администрировать серверы Oracle Key Vault, управлять кластерами, предоставлять конечные точки серверов, безопасно управлять группами ключей и отчитываться о доступе к ключам. Администраторы получают уведомления по электронной почте о важных обновлениях состояния и системных действиях, таких как предстоящий срок окончания действия пароля и ключа. Регистрация и подготовка конечных точек могут быть автоматизированы с помощью защищенных интерфейсов RESTful для массового развертывания в базах данных.

Cloud File System

Oracle Cloud File System (Oracle CloudFS) разработана для хранения любых файлов вне базы данных Oracle и поддерживает различные операционные системы. Oracle Cloud File System имеет единый интерфейс для управления, единые инструменты для инсталляции и конфигурации, единый стек кластерного ПО. Поддержка и решение проблем предоставляется одним вендором. Таким образом, заказчикам не нужны сторонние файловые системы и инструменты для управления дисковыми томами.

Возможности, предоставляемые Oracle Cloud File System:

- **Snapshots.** Возможность делать read-only снимки, использующие технологию Copy-On-Write
- **Tagging.** Возможность задать файлам дополнительный атрибут и обрабатывать эти файлы, находящиеся в разных каталогах, совместно.
- **Репликация.** Возможность создать Standby-копию основной файловой системы. Изменения в основной копии передаются на standby и применяются.
- **Security.** Возможность назначать права на уровне так называемых realms, а также на уровне операционной системы (owner, group, other для Linux, ACLs для Windows)
- **Шифрование.** Может быть применено как на уровне всей файловой системы, так и на уровне отдельного файла.
- **Динамическое изменение размера.** Можно изменять размер разделов файловой системы в online-режиме.
- **Отказоустойчивость.** Отказоустойчивость обеспечивается инфраструктурой Oracle ASM.
- Oracle Cloud File System — технология, работающая на базе Automatic Storage Management (ASM), позволяющая избавиться от затрат на приобретение, установку и сопровождение специализированных продуктов управления дисковыми томами.

Oracle Cloud File System — технология, работающая на базе Automatic Storage Management (ASM), позволяющая избавиться от затрат на приобретение, установку и сопровождение специализированных продуктов управления дисковыми томами.

CloudFS состоит из кластерной файловой системы, называемой ASM Cluster File System, и менеджера томов, называемого ASM Dynamic Volume Manager. Технология ASM Cluster File System (ACFS) на операционных системах Linux, Solaris, AIX и Windows позволяет хранить файлы приложений на дисковых группах ASM. ASM Cluster File System (ACFS) — это кластерная файловая система, файлы которой доступны с любого узла кластера как для чтения, так и для записи. Также поддерживаются снимки файловой системы в режиме «только для чтения» — до 64 копий данных файловой системы одновременно.

ASM Dynamic Volume Manager (ADVM) позволяет использовать дисковые группы ASM для хранения данных не только файловой системы ACFS, но и других файловых систем, таких как EXT4 для Linux. ADVM — это



загружаемый в ядро операционной системы модуль, который обеспечивает основные функции управления дисковыми томами. ASM Dynamic Volume Manager также предоставляет ряд полезных функций. В частности, пользователи могут добавлять в том дополнительные диски, не заботясь о равномерном распределении данных между всеми доступными ресурсами хранения.

Oracle CloudFS способна функционировать и как самостоятельная файловая система; кроме того, поверх нее может быть развернута другая файловая система, скажем, EXT4 или ZFS (Zetabyte File System), что дает возможность использовать функции, поддерживаемые этими системами. При таком подходе пользователь теряет возможность использовать эти разделы совместно в кластере, состоящем из нескольких серверов, но другие полезные функции Oracle Cloud File System остаются доступными.

Cloud File System и ASM — части Grid Infrastructure, которые бесплатно поставляются с любой редакцией СУБД.

Oracle VM и виртуализация

Oracle VM — бесплатный гипервизор от компании Oracle на основе Xen, позволяющий создавать и выполнять виртуальные машины. Он обеспечивает поддержку виртуальных машин как архитектуры x86, так и SPARC, делая возможным развертывание ПО в гетерогенной среде с минимальным привлечением аппаратного обеспечения. Клиенты могут задействовать Oracle VM для консолидации виртуальных машин, быстрого развертывания ПО, быстрого восстановления после отказа системы и изменения потребления ресурсов в соответствии с рабочей нагрузкой.

Oracle VM Server for x86 — гипервизор для серверов на базе процессоров x86 архитектуры, позволяющий создавать виртуальные машины с операционными системами Linux, Windows и Solaris for x86. Oracle VM Server for SPARC — высокоэффективный гипервизор для серверов Sun SPARC, позволяющий использовать различные версии Solaris for SPARC на едином сервере.

Управление виртуальным окружением осуществляется с помощью Oracle VM Manager, который поддерживает как Oracle VM Server for x86, так и Oracle VM Server for SPARC.

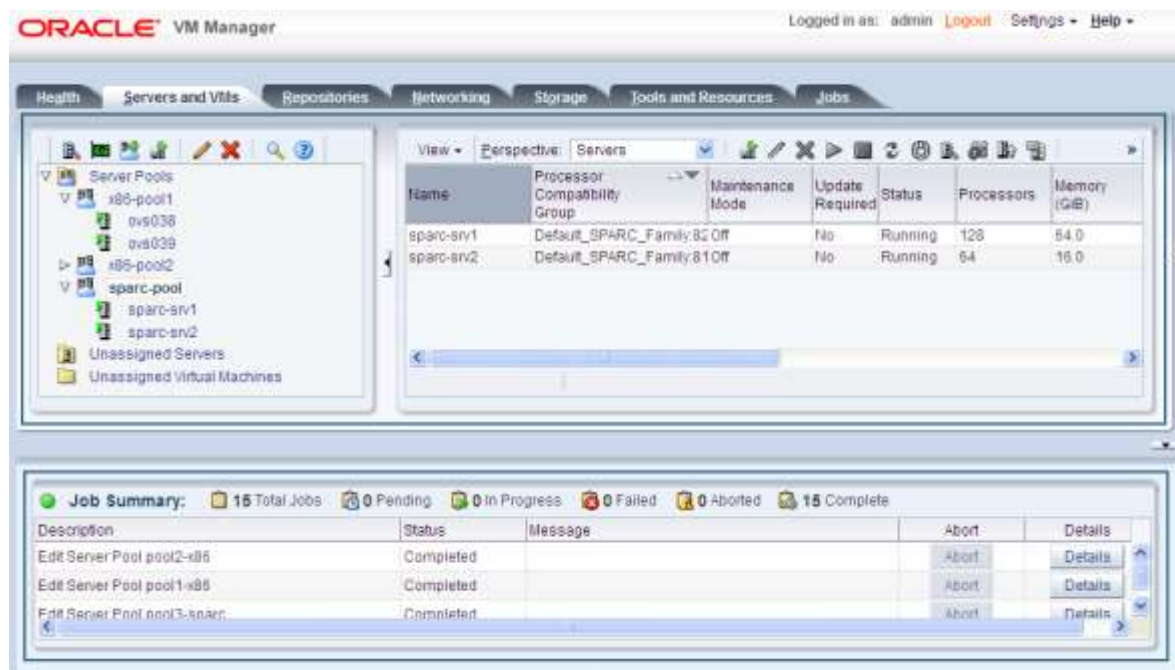


Рис. 1. Oracle VM Manager



Oracle VM Manager позволяет объединять серверы виртуализации (компьютеры) в единый пул серверов, осуществлять контроль и управление ресурсами пула, сетевыми интерфейсами, использованием дискового пространства, обеспечивает отказоустойчивость. Виртуальные машины могут мигрировать на соседние сервера пула в случае отказа оборудования или при необходимости сервисного обслуживания аппаратного сервера. VM Manager автоматизирует разворачивание виртуальных машин из шаблонов. Шаблоны виртуальных машин хранятся в едином репозитории, доступном каждому серверу виртуализации. Oracle предоставляет готовые шаблоны виртуальных машин не только с предустановленными операционными системами, но и с другими программными продуктами Oracle, например Oracle Database.

Oracle VM Manager тесно интегрирован с Oracle Enterprise Manager, что позволяет управлять виртуальным окружением совместно с другими компонентами ИТ-инфраструктуры.

Кроме того, для виртуализации Oracle также предлагает использовать Oracle Linux KVM — ПО с открытым исходным кодом, для управления которым Oracle создал Oracle Linux Virtualization Manager. Oracle Linux Virtualization Manager является средством конфигурирования, управления и мониторинга серверов виртуализации Oracle Linux KVM на корпоративном уровне, с высокопроизводительным современным графическим веб-интерфейсом.



Рис. 2. Oracle Linux Virtualization Manager

Как Oracle VM, так и Oracle Linux KVM можно использовать для уменьшения количества лицензий на продукты Oracle за счет того, что можно ограничить использование в виртуальной машине ядер процессора. Для гипервизоров других производителей Oracle требует лицензирование продуктов Oracle внутри виртуальных машин на основе всех ядер процессора аппаратного сервера.

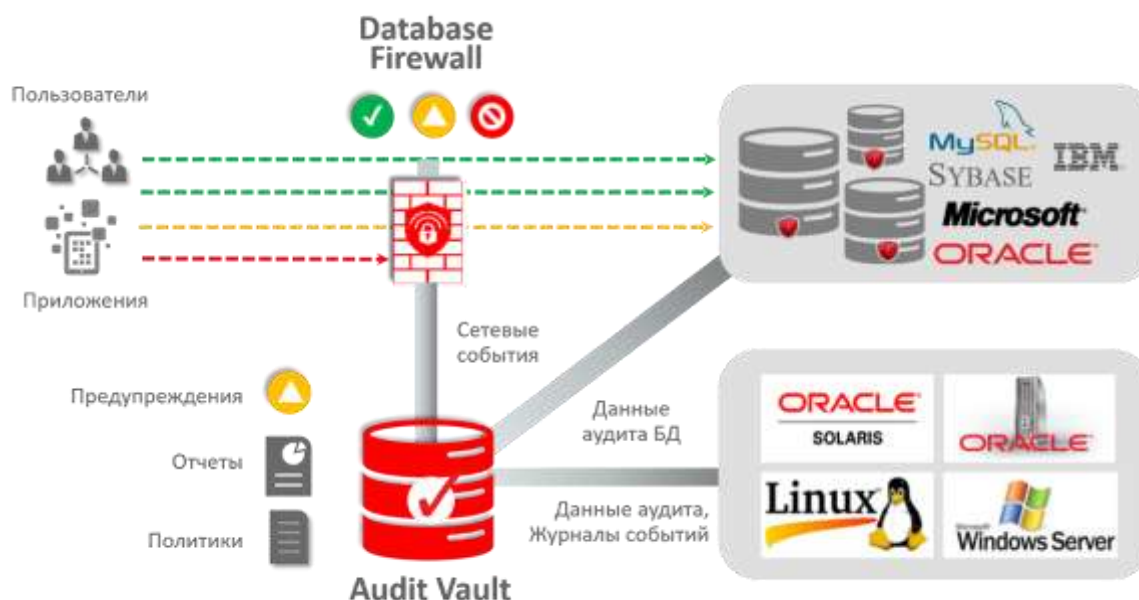
Audit Vault and Database Firewall

Oracle Audit Vault and Database Firewall обеспечивает первый рубеж обороны для баз данных и консолидирует данные аудита баз данных, операционных систем и каталогов. Очень точный, основанный на грамматике SQL механизм контролирует и блокирует несанкционированный трафик SQL, прежде чем он достигнет базы данных. Данные аудита сетевого трафика объединены с подробными данными аудита от других источников для подготовки отчетности о выполнении нормативных требований, формирования предупреждений при нарушениях. С использованием Oracle Audit Vault and Database Firewall задачи аудита и контроля могут быть легко адаптированы для обеспечения соответствия требованиям корпоративной безопасности.



Детективный и превентивный контроль

Брандмауэры периметра играют важную роль в защите центров обработки данных от несанкционированного внешнего доступа, но средства и способы нападения на базы данных становятся все более и более сложными. Злоумышленники обходят безопасность периметра, используют в своих интересах уязвимости серверов приложений и даже привлекают привилегированных инсайдеров. В результате задача обеспечения эффективной защиты баз данных стала критически важной. Эффективный контроль и аудит могут предупредить и заблокировать нарушения политик информационной безопасности, а также предоставить всесторонние отчеты для оценки уровня защищенности баз данных от несанкционированного доступа.



Oracle Audit Vault and Database Firewall консолидируют события контроля сетевых запросов к базам данных и журналы аудита. Устанавливаемые политики безопасности приводят в исполнение ожидаемое поведение приложения, помогают предотвратить SQL-инъекции, обход приложения и другие злонамеренные операции. Также проводится аудит за действиями привилегированных пользователей и другими операциями в базе данных. Oracle Audit Vault and Database Firewall могут также консолидировать данные аудита от Microsoft Active Directory, Microsoft Windows, Oracle Solaris, Oracle Linux и Oracle ASM Cluster File System. Использование плагинов позволяет обеспечивать консолидацию пользовательских данных аудита из таблиц приложения и других источников.

Брандмауэр баз данных для контроля активности и блокирования

Oracle Database Firewall обеспечивает сложный аналитический механизм грамматического анализа SQL-выражений, который проверяет SQL-операторы, поступающие в базу данных и определяет с высокой точностью, какое из предписанных политикой безопасности действий нужно произвести: разрешить выполнение; разрешить, но зарегистрировать; сформировать предупреждение; заменить или заблокировать SQL. Oracle Database Firewall поддерживает технологии «белый список», «черный список» и списки исключений в политиках безопасности. «Белый список» является простым набором утвержденных SQL-операторов, которые брандмауэр базы данных будет пропускать. Они могут изучаться в течение долгого времени или разрабатываться в тестовой среде. «Черный список» включает SQL-операторы от определенных пользователей, IP-адресов или определенные типы SQL-выражений, которые не разрешены для защищаемой базы данных. Списки исключений обеспечивают дополнительную гибкость развертывания для уточнения политик «белого» и «черного» списков. Политики могут быть приведены в исполнение с учетом атрибутов, включая категорию SQL, название программы, имя пользователя и IP-адрес. Эта гибкость, объединенная сочень точным анализом грамматики SQL, позволяет минимизировать уровень ложных срабатываний



системы защиты и собирать только те данные, которые действительно важны. События Oracle Database Firewall поступают в Audit Vault Server, где они обрабатываются совместно с данными аудита от других источников.

Консолидация данных аудита и управление жизненным циклом

Десятки встроенных отчетов предоставляют простую, адаптированную отчетность о выполнении положений нормативных документов, таких как SOX, DSS PCI и HIPAA. Отчеты формируются на основе консолидированных данных аудита баз данных, сетевых событий и данные аудита от других контролируемых систем. Для подробного анализа трендов определенные системы или события могут быть объединены или отфильтрованы, представлены в интерактивном режиме или в форматах Excel и PDF. Менеджеры безопасности могут произвести настройки системы оповещения о критичных событиях безопасности. Детальный механизм настройки прав доступа позволяют менеджеру безопасности ограничить аудиторов информацией из определенных источников, позволяя собирать данные в едином хранилище для предприятия или группы организаций.

Гибкость развертывания и масштабируемость

Средства обеспечения безопасности могут использоваться в режиме контроля и блокирования для некоторых баз данных и только в режиме контроля для остальных. Database Firewall может быть развернут различными способами (in-line, out-of-band, proxy) в зависимости от конкретной конфигурации сети. Кроме того, имеется использовать Host Monitor, который устанавливается на сервер базы данных и передает сетевой трафик к удаленному Database Firewall. Установленный как Soft Appliance (программное обеспечение устанавливается на выбранную аппаратную платформу), один Audit Vault Server может консолидировать данные аудита и события брандмауэров для тысяч баз данных. И Audit Vault Server, и Database Firewall могут быть сконфигурированы в режиме высокой доступности (High Availability, HA) для повышения отказоустойчивости.

Oracle Linux

Основное преимущество операционной системы Oracle Linux — это **бесплатное использование, распространение и обновление**. Она доступна для скачивания в виде скомпилированных дистрибутивов, в виде исходных кодов, а также в виде образов виртуальных машин. Кроме того, компания Oracle предоставляет неограниченный и бесплатный доступ к yum-репозиторию с обновлениями через сайт <http://public-yum.oracle.com>, где Вы можете получить свежие исправления, обновленные версии пакетов.

Высокая производительность — операционная система Oracle Linux оптимизирована для современных приложений, включая СУБД Oracle и Oracle Applications, она содержит исправления, внесенные специалистами Oracle, такие как оптимизация обращений к памяти, оптимизация размещения процессов и других структур, особенно при работе в NUMA-архитектуре, улучшение производительности до 3,6 раз за счет уменьшения конкуренции блокировок на примере SPECjbb, оптимизации при использовании SSD, оптимизации для работы с Infiniband.

Компания Oracle обеспечивает 10-летний срок поддержки каждого релиза. На сегодняшний день актуальной является версия 7, для которой Premier Support запланирован до июля 2024 года, а для выпущенной в 2019 году версии 8 Premier Support запланирован до июля 2029 года.

Oracle Linux версии 7 сертифицирована на соответствие стандартам компанией Edison Group и стандартам Linux Standard Base (LSB). Edison Group, независимая третья сторона, которая выполняет лабораторные испытания на совместимость и публикует результаты, а соответствие стандарту LSB (стандарт linuxfoundation.org) гарантирует пользователям надежную работу приложений на Oracle Linux. Oracle Linux 7 имеет сертификат на соответствие Common Criteria (CC), который подтверждает выполнение условий National



Information Assurance Partnership (NIAP) General Purpose Operating System Protection Profile (OSPP) v4.1, а также сертификат FIPS 140-2 на сертификацию криптографических модулей.

Таким образом, ОС Linux полностью совместима с существующими на рынке приложениями. За всю историю существования ОС Linux не зарегистрировано ни одного SR, в котором было бы отмечено нарушение совместимости ОС Linux с каким-либо приложением.

Oracle Linux содержит встроенные средства **виртуализации и контейнеризации**, которые позволяют пользователям безопасно запускать много приложений на одном и том же хосте без негативного взаимного влияния. В качестве решений возможно использовать cgroup, Oracle Container Runtime for Docker, Oracle Container Services, Kubernetes. Применение технологий виртуализации и контейнеризации повышает плотность приложений в сервере, что положительно сказывается на снижении себестоимости.

Подсистема **Performance Counters for Linux**, позволяет отслеживать аппаратные и программные события в системе, не затрагивая ее производительности. **DTrace** — это мощный инструмент, позволяющий системным администраторам, разработчикам и обслуживающему персоналу обстоятельно обследовать и анализировать проблемные операции в ОС и в приложении. Предельно полная и очень подробная **трассировка** с помощью встроенного **инструмента отладки DTrace** позволит системным администраторам разобраться в сложных проблемах.

Oracle Unbreakable Enterprise Kernel

Дистрибутив Oracle Linux предлагает клиентам на выбор два ядра: ядро, разработанное компанией Oracle **Unbreakable Enterprise Kernel**, и ядро Red Hat. Unbreakable Enterprise Kernel (UEK) — это вариант ядра Linux от компании Oracle, ориентированный на промышленное применение.

Ядро UEK полностью совместимо с ядром Red Hat и проверено на совместимость с большим количеством приложений, для запуска которых не потребуются какие-либо изменения. Приложения с ядром UEK работают быстрее и стабильнее. За всю историю работы ядра UEK не зарегистрировано ни одного SR по поводу негативного влияния ядра UEK на приложения.

В отличие от стандартного ядра RedHat, ядро Oracle UEK оптимизировано для выполнения промышленного программного обеспечения Oracle и для работ на оборудовании Sun/Oracle (в том числе и на Exadata Database Machine), позволяя достичь повышения производительности OLTP-тестов на 75 %, а Infiniband-тестов на 200 %.

- Оптимизации под архитектуру **NUMA**. Улучшение системы управления питанием и контроля эффективности энергопотребления. Более точный контроль потребления ресурсов CPU и памяти
- Оптимизация для работы на инженерных системах Oracle: **Exadata, ZDLRA, Private Cloud Appliance, BigData, ODA** и т. д.
- Оптимизации для работы с **Persistent Memory NVDIMM**
- TCP-стек теперь использует модель **Early Departure Time** вместо предыдущей модели As Fast As Possible. Реализация **ZERO-COPY TCP** при передаче пакетов по сети. Поддержка **RDMA**-протоколов, **RDMA over Converged Ethernet**
- **KSTACK** — распараллеливание ЦПУ-интенсивных приложений. Позволяет ЦПУ-интенсивным задачам выполняться в многопоточном режиме более чем на одном ядре при наличии свободных ЦПУ-ресурсов
- **Параллельный KSWAPD** — увеличение производительности при освобождении страниц в случае дефицита оперативной памяти



- Интегрирована подсистема **Performance Counter**, позволяющая задействовать для выявления узких мест, отладки и профилирования специальных PERFMON регистров, доступных в современных CPU (например, Intel Core 2); поддержка **DTrace 2.0** — решение, полностью работающее в User Space
- Реализован ряд оптимизаций, специфичных для продуктов Oracle: дополнительная поддержка оборудования Oracle/Sun и более полное тестирование ядра на аппаратных платформах Oracle/Sun;

Ядро Oracle Unbreakable Enterprise Kernel является единственным Linux-ядром, рекомендуемым компанией Oracle для использования со своими продуктами.

Сервис поддержки

Наличие технической поддержки на Oracle Linux не является обязательным при использовании данной ОС, тем не менее при использовании Oracle Linux в промышленной среде полезно иметь техническую поддержку. Услуга технической поддержки является платной. При этом стоимость контракта будет значительно ниже, чем у RedHat.

Лицензионная политика Oracle дает возможность заказчику выбрать системы, на которые он желает оформить поддержку, и уровень поддержки. Возможны три уровня технической поддержки ОС Linux:

- **Basic support** — дает возможность завести SR по технической проблеме
- **Premier Support** = Basic Support + backports + zero-downtime kernel updates with Ksplice + Oracle Container Services for use with Kubernetes and Gluster Storage
- **Extended Support** = поддержка устаревших версий Oracle Linux на системах, которые ранее имели поддержку Premier Support + program updates + fixes + security alerts + assistance with service requests

Техническая поддержка включает обязательства компании Oracle по внесению исправлений и устранению ошибок в версии ОС, которой пользуется клиент. Oracle предлагает для Linux такой же уровень корпоративной поддержки, какой предоставляется другими компаниями для других операционных систем. Пользователи других продуктов Oracle (например, владельцы лицензий СУБД Oracle или Fusion Middleware) дополнительно получают техническую поддержку ОС Linux автоматически, без дополнительной оплаты. Дополнительно все клиенты Oracle Linux получают бонус в виде бесплатного кластерного программного обеспечения Oracle Clusterware. Ранее это ПО было доступно только клиентам, которые приобрели лицензию Oracle Real Application Clusters (RAC), но теперь это кластерное решение бесплатно для всех, кто приобрел поддержку Oracle Linux.

Обновление ОС Linux без остановки приложений

Заказчики, которые приобрели сервис поддержки Linux Premier Support, дополнительно получают доступ к технологии Ksplice — обновления ядра Linux без остановки приложений («на лету»).

Технология Ksplice позволяет обновить пакеты ядра, гипервизор и пользовательские пакеты, не останавливая работу приложений и без перезапуска ОС Линукс. В основе таких обновлений лежит технология Ksplice. Компания Oracle использует технологию Ksplice для автоматической установки обновлений в своих облачных системах, облачных ЦОД, в инженерных системах.

Подробнее об «обновлениях на лету» см. на сайте <https://www.ksplice.com/>

Oracle Autonomous Linux — это строительный кирпич облака Oracle, это версия ОС Linux, работающая в Oracle Cloud. Autonomous Linux — это надстройка над Oracle Linux с Ksplice, которая периодически опрашивает репозиторий с обновлениями и выполняет автоматическую установку исправлений и настройку ОС без вмешательства человека. Этот подход увеличивает производительность труда системных администраторов, безопасность и надежность. Еще одной особенностью Oracle Autonomous Linux является возможность autoscaling: можно увеличивать и уменьшать количество ядер, выделенных виртуальной машине без остановки этой VM и без остановки баз данных в этой VM. Autonomous Linux позволяет настроить



автоматическое увеличение и уменьшение ядер и памяти (автоматическое масштабирование) как скриптами, так и с помощью REST API. Oracle Autonomous Linux минимизирует простои и уязвимости от некорректно установленных исправлений или их отсутствия, что снижает общую стоимость владения на 30–50 % по сравнению с on premise и другими облачными провайдерами.

Oracle Recovery Manager (RMAN)

Стратегия высокой доступности и аварийного восстановления требует надежных процедур резервного копирования и восстановления данных. Oracle Recovery Manager (RMAN) обеспечивает комплексную основу для эффективного резервного копирования и восстановления базы данных Oracle.

Возможности RMAN включают следующее:

- выполнение полного и инкрементального резервного копирования
- выполнение холодного/горячего резервного копирования
- обнаружение поврежденных блоков
- параллельное выполнения операций ввода-вывода и передачи данных по сети
- автоматическое протоколирование операций резервного копирования и восстановления
- эмуляция процедур резервного копирования и восстановления для тестирования скорости и готовности окружения к экстренному восстановлению
- отслеживание устаревших резервных копий на основе политик хранения

При выполнении резервного копирования утилита RMAN не выполняет резервное копирование файла БД, если уже существуют резервные копии идентичных версий этого файла, RMAN никогда не выполняет резервную копию дубликатов блоков, блоков Undo для завершенных транзакций или неиспользованных блоков (параметр BACKUP OPTIMIZATION). Существует два режима выполнения инкрементального резервного копирования. RMAN позволяет выполнять как разностно-накопительное (кумулятивное) инкрементальное резервное копирование (копирование изменений, произошедших после выполнения полного резервного копирования), так и просто разностное (дифференциальное) копирование (копирование изменений, произошедших после последнего инкрементального резервного копирования).

Во время процедур инкрементального резервного копирования требуется сканировать файлы базы данных в поиске изменившихся блоков БД, что делает процедуры копирования излишне долгими. Для решения этой проблемы применяется файл отслеживания изменений (Block Change Tracking), в который записывается информация о физическом местонахождении изменившихся блоков базы данных. Во время резервного копирования утилита RMAN считывает этот файл для выяснения того, какие именно блоки данных ей надлежит прочитать и скопировать, и тем самым избегает необходимости считывать с диска файлы данных целиком, что позволяет значительно сократить время, затрачиваемое на выполнение инкрементального резервного копирования. Инкрементальное резервное копирование особо эффективно при работе RMAN с Zero Data Loss Recovery Appliance. Полная резервная копия делается только один раз, а затем благодаря инкрементальным копиям Zero Data Loss Recovery Appliance может на лету воссоздавать «виртуальную полную копию БД» на любой момент времени.

При каждом создании резервной копии файлов данных RMAN автоматически выполняет резервное копирование ключевых файлов БД: управляющего файла и файла параметров экземпляра БД (CONTROLFILE AUTOBACKUP).

При резервном копировании можно использовать также параметр SECTION SIZE для создания многосекционной резервной копии, который ограничивает размер каждой секции резервной копии БД, этот параметр удобно использовать для выполнения параллельного резервного копирования очень большого



файла данных. Также для работы с большими файлами БД в случае прерывания процедуры резервного копирования в RMAN предусмотрено возобновление резервного копирования с того же места, где оно было прервано, вместо того чтобы начинать выполнять процедуру заново.

Если резервное копирование сказывается на производительности какого-то критически важного задания базы данных, то при указании параметра `DURATION` утилита RMAN будет вычислять наиболее подходящую для задания скорость выполнения резервного копирования. Кроме того, можно указывать базе данных либо сводить к минимуму время выполнения резервного копирования (`MINIMIZE TIME`), либо сводить к минимуму загрузку системы (`MINIMIZE LOAD`).

Восстановление данных может осуществляться как на уровне файлов, так и на уровне отдельных блоков — в обоих случаях это не прерывает доступ пользователей к остальной части базы данных. Если экземпляр БД функционирует в режиме архивации журналов (`ARCHIVELOG`), то это гарантирует возможность выполнения в RMAN процедуры восстановления базы данных до состояния, в котором она находилась на определенный момент времени в прошлом. Возможен также вариант восстановления не всей базы данных, а только определенной таблицы (`RECOVER TABLE`), в этом случае RMAN восстановит в новый экземпляр БД все необходимые для работы системные данные и только те пользовательские табличные пространства, в которых содержатся данные нужной таблицы (при этом основной экземпляр БД будет продолжать работать), а затем RMAN восстановит данные таблицы в основную БД в отдельную схему. Эта процедура значительно увеличивает скорость восстановления при потере данных из-за человеческих ошибок.

RMAN предлагает уникальные функции, например распараллеливание потоков данных резервного копирования/восстановления, политику хранения файлов резервных копий (`RETENTION POLICY` и `ARCHIVELOG DELETION POLICY`) и подробную историю обо всех созданных резервных копиях (`RECOVERY CATALOG`). В случае запуска сеанса резервного копирования в параллельном режиме с несколькими каналами передачи данных выход из строя одного из каналов, например, из-за неполадки с ленточным устройством не будет приводить к прекращению процесса резервного копирования. Вместо этого RMAN завершит процесс резервного копирования с оставшимися каналами (функция `Automatic Channel Failover`).

RMAN разработан для тесной работы с СУБД Oracle, обеспечивая обнаружение повреждений на уровне блоков во время резервного копирования / восстановления или при выполнении команд `CROSSCHECK` и `VALIDATE`. Есть возможность создавать зашифрованные резервные копии RMAN, для этого используется опция Oracle `Advanced Security`. RMAN оптимизирует производительность и потребление пространства во время резервного копирования за счет мультиплексирования файлов и сжатия набора резервных копий, а также RMAN интегрируется с Oracle `Secure Backup`, `Zero Data Loss Recovery Appliance` и с продуктами сторонних производителей для управления носителями при резервном копировании на ленту.

RMAN предоставляет интерфейс через командную строку или графический интерфейс `Enterprise Manager` и упрощает процедуры резервного копирования, позволяя использовать мощные и в то же время легкие в написании сценарии резервного копирования и восстановления. Он автоматически обеспечивает создание резервных копий всех необходимых файлов базы данных, что исключает вероятность допущения человеческой ошибки, которая присутствует в методах резервного копирования, основанных на применении команд операционной системы.

Oracle Database также имеет интеллектуальное средство идентификации проблем базы данных — `Data Recovery Advisor`, тесно интегрированное с RMAN, которое сокращает время для определения необходимых шагов по восстановлению работоспособности базы данных в кратчайшие сроки, выбирая для восстановления БД резервные копии базы данных как на ленте, так и на диске, а также данные резервной базы данных, созданной с помощью `Oracle Data Guard`.

Также с помощью RMAN можно легко клонировать базы данных (команда `DUPLICATE`), причем это можно делать не останавливая БД (`DUPLICATE ... FROM ACTIVE DATABASE`). Эта возможность также очень полезна для создания и поддержки резервных баз данных (`DUPLICATE ... FOR STANDBY`). RMAN может проверять standby базу данных на наличие блоков, которые не соответствуют состоянию блоков на основной БД из-за `no-logging` операций, и восстановить эти блоки (`VALIDATE/RECOVER .. NONLOGGED BLOCK`). RMAN также поддерживает создание клонов БД на sparse дисковых группах на Exadata (`DUPLICATE ... FROM SPARSE`).



Продукты Oracle



Средства разработки



Oracle Application Express (APEX)

Oracle Application Express (APEX) — это платформа разработки с минимальным объемом программирования для быстрого создания масштабируемых, безопасных веб-приложений для бизнеса с самыми передовыми возможностями, с широким выбором варианта развертывания. С помощью APEX разработчики могут быстро создавать и разворачивать профессиональные приложения, которые решают реальные проблемы и позволяют быстро достигать нужного результата. Нет необходимости быть экспертом в массе технологий для достижения прекрасного результата. Нужно только сфокусироваться на решении проблемы, а остальное сделает APEX.

Oracle APEX поддерживает командная разработка приложений. Созданные веб-приложения немедленно становятся доступны пользователям и специалистам по тестированию через браузер, и они могут опробовать работу с ними. Это очень удобно для прототипирования, быстрой доработки и развития приложений.

Oracle APEX обеспечивает всеми необходимыми инструментами для создания приложений в единой расширяемой платформе, реализованной как часть базы данных Oracle. APEX не требует развертывания и установки, он есть в любой БД Oracle.

Архитектура

Oracle APEX использует встроенную в СУБД простую архитектуру, работа которой определяется метаданными. Эта архитектура обеспечивает быстрый доступ к данным, высокую производительность и масштабируемость «сразу из коробки». В реальности и сам APEX, и создаваемые с его помощью приложения — это таблицы и PL/SQL-объекты в базе данных.



Веб-запрос от веб-браузера посылается на Oracle REST Data Services (ORDS), откуда он передается в СУБД Oracle для выполнения. В СУБД запрос обрабатывается Oracle APEX. Как только запрос выполняется, ответ посылается через ORDS к веб-браузеру.

Стек Oracle **RAD** — это самодостаточный стек технологий, основанный на трех основных компонентах: **REST**, **APEX** и **Oracle Database**. Этот стек обеспечивает все необходимые компоненты для создания и развертывания передовых, мощных, масштабируемых приложений с красивым дизайном. Кроме того, и ORDS, и APEX являются уже встроенным функционалом СУБД Oracle: если у Вас в распоряжении есть СУБД Oracle, то значит, Вы уже имеете весь стек Oracle RAD.

REST Data Services — это Java приложение, которое позволяет разработчикам со знаниями SQL и СУБД разрабатывать REST API для базы данных Oracle, хранилища JSON-документов Oracle и Oracle NoSQL Database.

Когда разработчик создает или изменяет приложение, Oracle APEX создает или модифицирует метаданные, сохраняемые в таблицах базы данных. Когда приложение выполняется, Oracle APEX на основании этих данных создает и отображает страницы или выполняет действия, связанные с ответом пользователя.

Вся обработка производится в PL/SQL прямо в схемах данных базы данных. Именно потому, что манипуляция данными осуществляется непосредственно в базе данных и результат напрямую отправляется в веб-браузер,



Oracle APEX очень эффективен. Единственный вызов API порождает целую последовательность действий, основанных на определении метаданных, единственный запрос вместо множества обращений к базе данных.

Oracle APEX очень хорошо масштабируем и может поддерживать одновременную работу десятков тысяч пользователей именно благодаря тому, как обрабатываются запросы к базе данных. Вызовы API к «движку» Oracle APEX используют стандартный пул соединений СУБД Oracle. Это значит, что, как только запрос обработан и результат отправлен в браузер, использованное им соединение возвращается в пул соединений базы данных и может быть использовано для других запросов.

Сессия базы данных активная только тогда, когда выполняется обработка запроса, в остальное время сессия не использует никаких ресурсов.

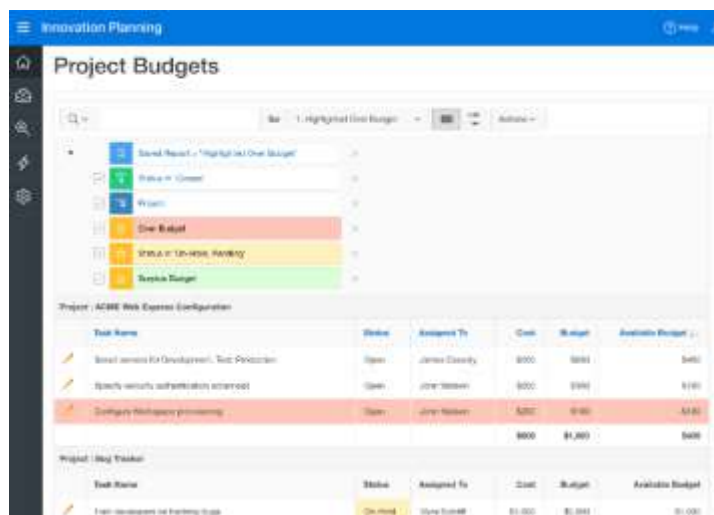
Одним из преимуществ архитектуры, работа которой основана на метаданных в базе данных, является то, что Oracle APEX может обеспечивать автоматизированную платформу для создания приложений с полным самообслуживанием — любой пользователь Вашей организации может подключаться к собственному рабочему пространству. В этом случае все подразделения могут продолжать создавать собственные приложения, оставаясь в сфере компетенции службы ИТ. У службы ИТ появляется возможность управлять политиками и процедурами для доступа и использования корпоративных данных.

Это стимулирует к улучшению рабочих взаимоотношений между ИТ и департаментами, которым не приходится создавать свой «теневой ИТ».

Основные возможности

Oracle APEX обеспечивает весь необходимый инструментарий для продуктивной работы по созданию приложений в срок и с впечатляющими результатами для бизнеса.

APEX обеспечивает простую обработку и визуализацию данных, создание из них диаграмм. Не имеет значения, что является источником данных, локальная база данных, другая база данных или веб-сервис, встроенная функциональность Oracle Apex позволяет превратить данные в информацию.



Каждый изучает данные по-разному. Интерактивные отчеты дают возможность пользователю представить просматриваемые данные в том уникальном виде, который удовлетворяет его потребности. Начиная от простого определения, какие столбцы показывать, до гораздо более сложных подстроек, таких как сводные (pivot) таблицы, а также поиск, фильтрация, подсветка по условиям, группировка, расчет и агрегация, построение диаграмм. Произведенные изменения могут быть сохранены как отдельный вариант отчета для каждого посетителя, чтобы можно было им воспользоваться в следующий визит.



PeopleSoft	Name	Country	State	Start	App	Category	Action
10-000001	Lorenza de Silva	Italy	1000	1000	01	Category A	View
10-000002	John Carter	Italy	1000	1000	02	Category B	View
10-000003	John Carter	Italy	1000	1000	03	Category B	View
10-000004	John Miller	Italy	1000	1000	04	Category B	View
10-000005	John Miller	Italy	1000	1000	05	Category B	View
10-000006	John Miller	Italy	1000	1000	06	Category B	View
10-000007	John Miller	Italy	1000	1000	07	Category B	View
10-000008	John Miller	Italy	1000	1000	08	Category B	View
10-000009	John Miller	Italy	1000	1000	09	Category B	View
10-000010	John Miller	Italy	1000	1000	10	Category B	View
10-000011	John Miller	Italy	1000	1000	11	Category B	View
10-000012	John Miller	Italy	1000	1000	12	Category B	View
10-000013	John Miller	Italy	1000	1000	13	Category B	View
10-000014	John Miller	Italy	1000	1000	14	Category B	View

Редактирование данных должно быть простым. С интерактивными таблицами редактировать множество строк данных просто — «кликнуть» мышкой на ячейку и изменить значение. Если пользователи хорошо знакомы с электронными таблицами, то они будут себя комфортно чувствовать и здесь с такими возможностями, как выбор столбца и строки, копирование вниз, заполнение и многое другое.

Интерактивными таблицами можно строить на одной странице множество таблиц с разными уровнями

взаимоотношений, такими как перекрестные ссылки, многоуровневая детализация.

Мощные возможности диаграмм и визуализации данных обеспечиваются Oracle JET. Возможно простое создание красивых диаграмм и графиков быстро и с большим количеством подстроек.

Диаграммы с полной поддержкой HTML5 работают в любом современном браузере независимо от платформы, размера экрана и возможностей. Огромные возможности по настройке с большим диапазоном декларативных атрибутов. Широкий спектр вариантов визуализации — включая bar, line, area, range и их комбинации, scatter, bubble, polar, pyramid, radar, pie, donut, funnel, dial gauge, stock и Gantt диаграммы.



APEX позволяет интегрировать данные из различных источников: из локальной базы данных, из внешней БД или из любого веб-сервиса. Точно так же Oracle APEX позволяет публиковать веб-сервисы, используя встроенный REST-воркшоп.



Возможно создание и использование веб-сервисов с доступом к данным из различных REST-источников, включая простейшие REST data feeds, REST-сервисы из Oracle REST Data Services и Oracle Cloud Applications REST Services. Дополнительно имеются возможности по интеллектуальному кэшированию REST-данных и уникальные возможности по прямому манипулированию результатами REST-источников с помощью стандартного SQL.

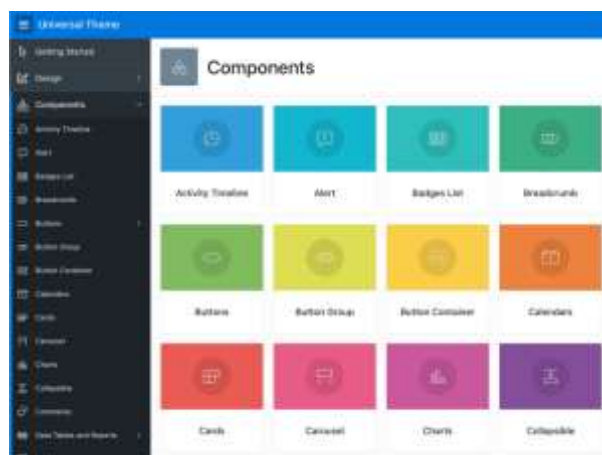
С Oracle APEX больше нет необходимости использовать database link. Сейчас манипуляция данными на внешних базах данных может осуществляться с помощью SQL через HTTP и REST.

Страницы форм позволяют конечному пользователю увидеть и отредактировать одиночные строки таблицы базы данных или представления. Выбирая из множества различных типов элементов, разработчик может создавать простые в использовании формы, которые взаимодействуют с базой данных через автоматические процедуры DML, динамические действия и AJAX.



Встроенная тема оформления Universal Theme — это пользовательский интерфейс к APEX, который позволяет разработчикам строить современные веб-приложения без требования глубоких знаний HTML, CSS или JavaScript. С базовыми рабочими умениями APEX можно использовать Universal Theme для простого создания отзывчивого приложения, легко настраиваемого, с простой поддержкой.

Встроенная тема оформления Universal Theme прекрасно работает как на самых маленьких устройствах, так и на самых больших. Это значит, что все пользовательские компоненты, независимо от устройства, обладают такой же или похожей функциональностью.



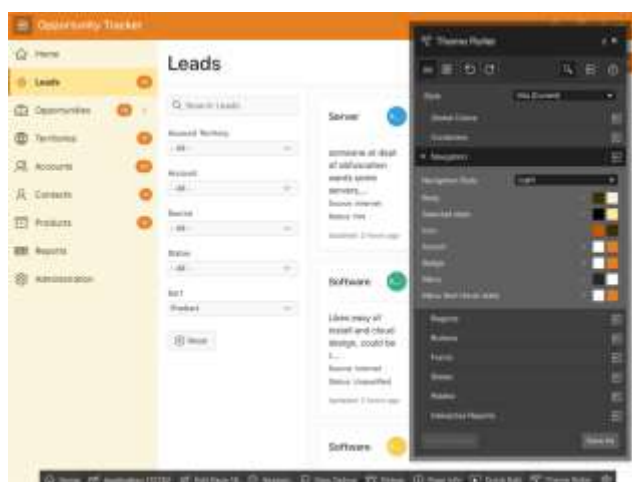
Встроенная тема оформления предоставляет компоненты и целые блоки, необходимые для построения практически любого типа пользовательского интерфейса для бизнес-приложений.

Формы, карты, меню, вкладки, отчеты и многое другое разработаны с использованием лучших практик на основании общепринятых и популярных шаблонов пользовательского интерфейса.

В дополнение к встроенным базовым требованиям, широкие возможности настроек и кастомизация являются ключом и необходимым ингредиентом для успеха приложения.

Встроенная тема имеет все возможности для широкого спектра различных настроек без необходимости быть экспертом в пользовательском интерфейсе, HTML, CSS или JavaScript.

Однако при необходимости можно создавать собственные шаблоны и даже элементы/блоки с помощью встраиваемых плагинов.



Oracle APEX создана для построения веб-приложений с высокой степенью безопасности «прямо из коробки». Существуют различные встроенные схемы аутентификации и авторизации доступа к разным частям приложения и отдельных компонентов. Кроме этого, разработчики могут встроить собственные схемы аутентификации и авторизации на SQL и PL/SQL.

Мониторинг и журналирование обеспечивают полный обзор того, как пользователи работают с приложением. Эта информация может быть использована для выявления хакерских атак, ошибок приложений, проблем с производительностью. С Oracle APEX приложение уже защищено, так как мониторинг и журналирование уже встроены в «движок» системы.

Oracle APEX может генерировать массу диагностических данных с высокой детализацией, которые могут помочь в поиске ошибок или проблем с производительностью в приложении. Администратор приложения может включать диагностическое журналирование для определенных пользовательских сессий или для всего экземпляра. Разработчики получают простой доступ к диагностической информации с помощью страниц отчетности, доступных в окружении разработки Oracle APEX.



Развертывание

Приложение — это бизнес. И именно поэтому Oracle APEX позволяет выбрать стратегию развертывания, максимально удобную для потребителя. Приложение является органичным для облака любого типа — от частного облака предприятия до публичного облака любого поставщика. Кроме того, оно легко разворачивается на любом сервере в ЦОД или офисе заказчика или даже на виртуальной машине и на портативном переносном компьютере. Может быть развернуто везде, где может быть запущена СУБД Oracle, включая бесплатную Oracle Express Edition.

Разработано Oracle для СУБД Oracle

Oracle APEX — полностью поддерживаемая встроенная возможность СУБД Oracle без дополнительных расходов на приобретение. Это значит, что если Вы уже имеете СУБД Oracle, то Вы уже владеете и Oracle APEX! Это значит, что из Oracle APEX есть прямой доступ ко всем возможностям СУБД Oracle.

Основные возможности Oracle Application Express

- Мобильные приложения: поддержка мобильных платформ, включая возможность визуализации форм без HTML-таблиц.
- Интерактивные отчеты: позволяет создать страницы с отчетами, которые имеют инструменты для конечных пользователей для модификации отчетов, сохранения данных отчета в различных форматах, включая документы PDF.
- Формы: автоматическая генерация форм ввода данных на основе описания таблицы, связанных между собой нескольких таблиц, хранимых процедур или веб-сервисов.
- Поддержка веб-сервисов RESTful: обеспечивает интеграцию с веб-сервисами RESTful через декларативный интерфейс.
- Dynamic Actions: позволяет разработчикам в описательной форме встраивать динамические процессы на стороне клиента, что уменьшает необходимость серьезного изучения языков программирования JavaScript и AJAX для создания пользовательских функций на базе технологий Web 2.0.
- Plug-Ins: дополнительные подключаемые программные модули расширяют среду разработки, позволяя разработчикам встраивать собственные компоненты.
- Загрузка данных: разработчики с легкостью могут добавлять в свои приложения интерфейс для конечных пользователей, позволяющий загружать данные в существующие внутри приложения таблицы.
- Обработка ошибок: определяемая разработчиком обработка исключений позволяет с легкостью перехватывать сообщения, генерируемые базами данных Oracle Database, и отображать доступное и понятное конечным пользователям сообщение.
- Календарь: позволяет разработчикам включать в свои приложения возможности управления посредством операций drag-and-drop и генерировать календари с редактируемыми страницами.
- Websheets: обеспечивает быстрый и простой способ интеграции и совместного использования данных в сети Интернет.

Oracle SQL Developer

Oracle SQL Developer — интегрированная среда разработки на языках SQL и PL/SQL, с возможностью администрирования баз данных, ориентирована на применение в среде Oracle Database. Корпорация Oracle



предоставляет продукт бесплатно. Сама среда написана на языке программирования Java, работает на всех платформах, где доступна среда выполнения Java SE.

Расширенная версия SQL Developer, включающая возможность визуального моделирования ER-моделей, генерацию схем данных и реверс-инжиниринг баз данных, поставляется как отдельный продукт (также бесплатно) под наименованием SQL Developer Data Modeler.

Oracle SQL Developer изначально поддерживает работу с Oracle Database, существуют плагины, обеспечивающие подключение из среды к другим системам управления базами данных, в частности, реализован доступ к IBM DB2, Microsoft Access, Microsoft SQL Server, MySQL, Sybase ASE, Teradata Database.

В среде поддержана возможность создания расширений, обеспечивающих ее дополнительными возможностями. Некоторые расширения разрабатываются в корпорации Oracle, есть большая серия расширений, написанная сторонними разработчиками. Например, есть расширения, обеспечивающие картографический просмотр хранимой в базе данных геоинформации или обеспечивающие визуальное создание ER-диаграмм.

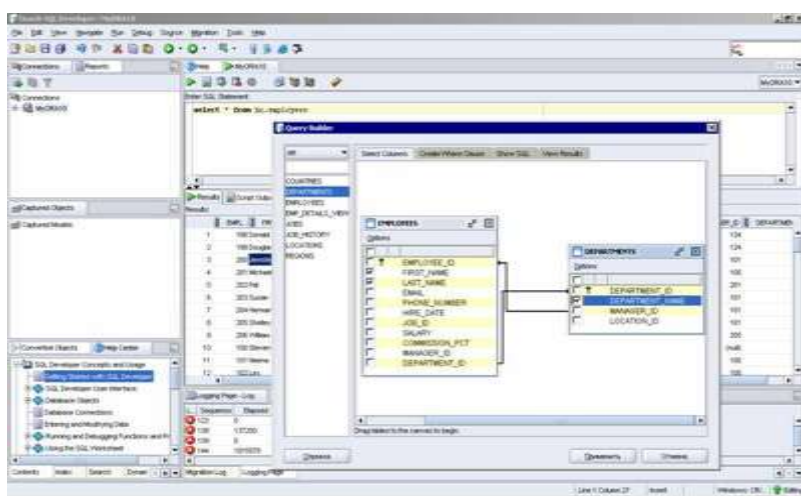


Рис. 1. Создание нового соединения в Oracle SQL Developer

Основные возможности Oracle SQL Developer:

- I. Просмотр, поиск и изменение объектов БД, декларативная модификация параметров объектов БД, автоматическая генерация DDL;
- II. SQL Worksheet — инструмент для редактирования и исполнения SQL-запросов и хранимых процедур, графический редактор SQL-запросов с поддержкой drag and drop, удобный просмотр плана выполнения запроса, отладка хранимых процедур, интеграция с Subversion, CVS (Concurrent Versions System), Serena Dimensions и Perforce для контроля версий кода, поддержка скриптов SQL*Plus;
- III. Удобный импорт и экспорт данных БД в файл для резервной копии и загрузки в другую БД, возможность миграции объектов БД и данных из MySQL, Microsoft SQL Server, Sybase Adaptive Server, Microsoft Access или IBM DB2 (UDB) в Oracle Database
- IV. Инструменты для администрирования БД, настройка резервного копирования, безопасности, параметров хранения, определяемые пользователем отчеты о параметрах БД
- V. Интеграция с Oracle Application Express
- VI. Data Modeler позволяет создать логическую, реляционную и многомерную модели данных БД



В Oracle SQL Developer реализованы возможности прямого редактирования записей в таблицах баз данных. Для этого нужно в контекстном меню для таблицы в окне Connections выбрать команду Edit, а затем перейти на вкладку Data.

В Oracle SQL Developer предусмотрены средства для отладки кода программных модулей PL/SQL (включая точки останова, пошаговое выполнение, просмотр значений переменных и т. п.). Эти средства в основном доступны из меню Debug.

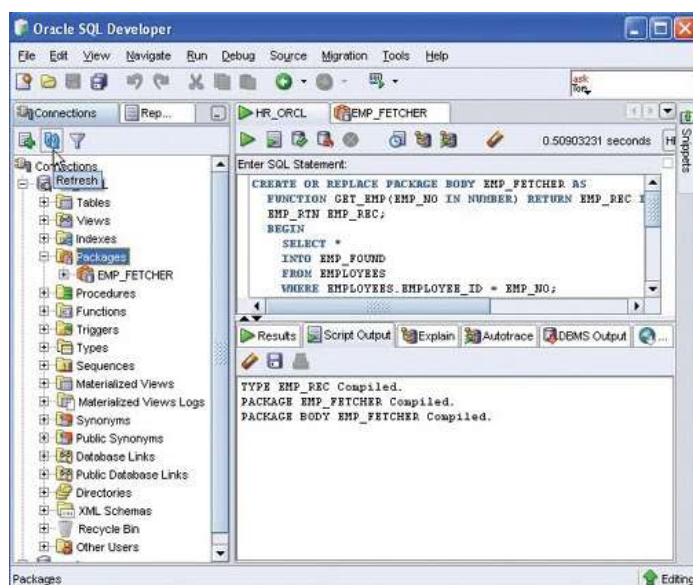


Рис. 2. Редактирование PL/SQL-кода в Oracle SQL Developer

Заранее заготовленные блоки программного кода доступны из окна Snippets (это окно можно открыть при помощи меню View). Такие блоки заготовлены как для встроенных функций Oracle SQL, так и для других синтаксических конструкций (операторов условного перехода, циклов, курсоров и т. п.). Набор сниппетов является расширяемым: Вы вполне можете добавлять в него свои блоки программного кода.

Для того чтобы было проще переносить программный код, написанный в расчете на выполнение в SQL Plus, в Oracle SQL Developer предусмотрена поддержка многих синтаксических конструкций SQL Plus (например, обращение к внешним файлам скриптов при помощи символов @ и @@). Однако поддерживаются далеко не все специальные команды SQL Plus.

Если объектов в базе данных много (обычная ситуация для реальных баз данных) и просматривать их все в окне Connection трудно, можно воспользоваться специальным окном поиска объектов. Для этого нужно в меню View воспользоваться командой Find DB Object, затем в открывшемся окне выбрать соединение, ввести имя объекта (можно использовать подстановочные символы) и нажать Enter. Для объектов в списке результатов из контекстного меню доступны те же возможности, что и для объектов в окне Connections.

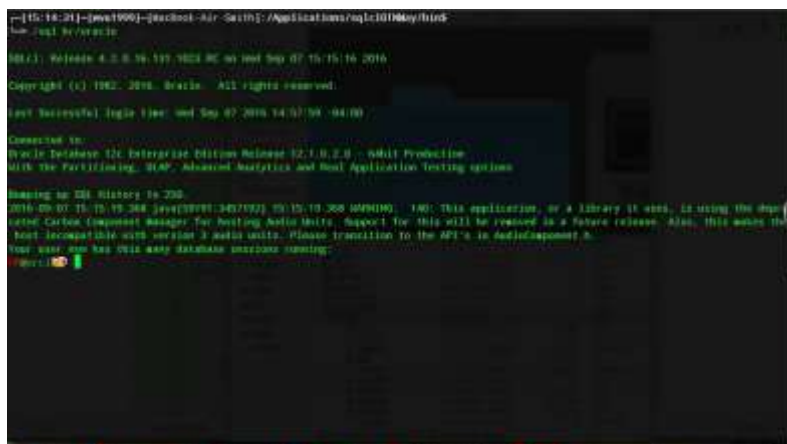
Во второй вкладке окна Connections предусмотрены средства работы с отчетами. В основном, правда, эти отчеты предназначены для администраторов баз данных. Можно использовать готовые отчеты и создавать свои (из контекстного меню для контейнера User Defined Reports). В отчетах предусмотрено использование параметров (например, для отчета, который возвращает все отчеты, принадлежащие какому-либо пользователю, нужно указать его имя).

В Oracle SQL Developer предусмотрены средства переноса информации между различными источниками данных, например из базы данных SQL Server в Oracle. Средства для выполнения такого переноса сосредоточены в меню Migration.

Oracle SQL Developer Command Line (SQLcl) — это утилита командной строки для Oracle Database.



SQLcl позволяет интерактивно или пакетно выполнять скрипты с командами SQL. SQLcl обеспечивает встроенное редактирование, автозавершение операторов и историю команд, также обеспечивается поддержка ранее написанных скриптов SQL*Plus.



```
--[15:14:21]--[java1900]--[oschod--Aur-Serth]--[Applications/sqlcl\IMay\link-  
-...  
SQLcl: Release 4.0.0.197.000 RC on Wed Sep 07 15:15:14 2016  
Copyright (c) 1982, 2016, Oracle. All rights reserved.  
Last Successful login time: Wed Sep 07 2016 14:57:59 +04:00  
Connected to:  
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production  
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options  
Warning: DB History is Disabled  
2016-09-07 15:15:19.364 [java1900:1347/192] 15:15:19.308 (MNNING) 190: This application, or a library it uses, is using the deprecated Carbon Component Manager for loading native libraries. Support for this will be removed in a future release. Also, this means the host incompatible with version 3 java units. Please transition to the API's in ApplicationManifest.  
Your user now has this many database sessions running:  
1 user.
```

Рис. 3. Утилита Oracle SQL Developer SQLcl

Помимо предоставления более современного способа работы с командной строкой, SQLcl также вводит новые команды и функции, отсутствующие в самом SQL * Plus. Например, в скриптах SQLcl можно использовать не только языковые конструкции SQL и PL/SQL, но также и JavaScript, Jython, LUA.

Oracle SQL Developer — полноценный инструмент для разработчиков, поддерживающий все самые новые технологии Oracle Database. Он позволяет работать как со обычными БД, так и с БД в облаке и с автономными БД.





Продукты Oracle

Встроенные и мобильные СУБД

Oracle Mobile Server

Oracle Database Mobile Server обеспечивает создание инфраструктуры систем мобильных приложений, которые предоставляют мобильным и автономно работающим пользователям постоянный доступ к данным независимо от наличия действующего подключения к сети. Целевая аудитория для данного решения — компании, имеющие удаленные офисы или предполагающие удаленную работу (страховые компании, службы доставки и т.п.). Oracle Database Mobile Server позволяет организовать автоматическую синхронизацию данных и приложений между мобильными устройствами и основным сервером базы данных. Включает средства для централизованного управления пользователями и устройствами, а также средства для разработки мобильных приложений.

Oracle Database Mobile Server состоит из трех основных компонентов:

Mobile Client — программное обеспечение, устанавливаемое на мобильных устройствах для обеспечения синхронизации данных и удаленного управления;

Sync Server — вместе с управляющей консолью Mobile Manager обеспечивает централизованное надежное и безопасное управление данными, приложениями и устройствами;

Mobile Development Kit (MDK) — набор инструментов и средств для разработки мобильных приложений.

Основной задачей предлагаемой инфраструктуры является обеспечение надежной и безопасной синхронизации данных между корпоративной базой данных Oracle Database и мобильными клиентами. При сеансах связи пользователям передается только измененная информация.

Oracle Database Mobile Server поддерживает различные мобильные платформы, такие как Java, Android, Blackberry, Windows Desktop & Mobile, Linux. В качестве локальной базы данных на мобильных устройствах используется Oracle Berkeley DB или SQLite database. Oracle Berkeley DB — небольшая, но полнофункциональная реляционная база данных, специально спроектированная для работы на мобильных устройствах, в которой полностью реализованы механизмы транзакций, ссылочной целостности и спецификации языка SQL. Berkeley DB совместима с SQLite Database на уровне SQL API и в тоже время значительно превосходит ее по масштабируемости и производительности.

Oracle Database Mobile Server обеспечивает взаимодействие мобильных приложений с Oracle Database и с различными интернет-приложениями. При синхронизации данных в случае разрыва соединения передача информации на мобильные устройства возобновится после восстановления связи именно с той точки, где она прервалась. Применение Mobile Server обеспечивает гарантированную доставку данных. Информация, которая передается по сети и хранится в базе данных, может быть зашифрована по алгоритмам FIPS-140, удовлетворяющим стандартам AES. Синхронизация данных между базой данных Oracle Berkeley DB и Oracle Database сервером происходит по протоколам TCP/IP, HTTP, CDPD, 802.11b Wireless LAN, PPP, GPRS, HotSync, ActiveSync. Программный интерфейс Open Transport API дает возможность использовать любой беспроводной транспортный протокол для синхронизации.

Мобильные приложения разрабатываются с помощью Mobile Development Kit на языках программирования C, C++, Java, Visual Basic, с использованием ActiveX Data Objects (ADO), в инструментальных средах Oracle JDeveloper, Microsoft Visual Studio.Net, Microsoft EVT 3.0, Borland Delphi, Sybase Power Builder, Metroworks CodeWarrior 8+, Rrapid Software Formation. Приложения, работающие на мобильных устройствах, имеют доступ к БД через различные программные интерфейсы (JDBC, ODBC, ADO.Net).

Berkeley DB

Berkeley Database (Berkeley DB) — это набор программных библиотек, которые включаются в приложения (на этапе сборки приложения — *linking*), нуждающиеся в высокопроизводительном механизме хранения



и извлечения данных в виде пар «ключ-значение». Ключи используются для обнаружения элементов и могут представлять собой данные любого типа или любые структуры, поддерживаемые применяемым языком программирования. Значение записи также может быть представлено составным типом произвольной структуры.

Berkeley DB распространяется в виде набора библиотек с исходными текстами, которые включаются в код приложения. Функции библиотеки доступны разработчикам через ряд API-интерфейсов для языков программирования Си, C++, Java. Имеются также интерфейсы для работы с *Berkeley DB* при помощи языков сценариев Perl, Python и Tcl. Указанные интерфейсы представляют собой функции-оболочки над C-функциями API *Berkeley DB*. Также доступны интерфейсы, которые делают возможным вызов функций API *Berkeley DB* в программах на языке C#, в среде выполнения Microsoft .Net.

Berkeley DB включается в код приложения на этапе его компоновки (linking). Также *Berkeley DB* можно оформить в виде динамической библиотеки (dll — в *MS Windows*, *so*, *sl* — в *Unix*), и подключать к приложению динамически (в исходном коде программы) в процессе его работы.

В комплект поставки *Berkeley DB* включены make-файлы для сборки проектов под различными операционными системами с использованием соответствующих компиляторов C/C++. Для компиляции проектов в операционной системе *MS Windows* поставляются файлы настроек для компилятора *Visual C++* компании *Microsoft*.

Программные интерфейсы, входящие в комплект поставки *Berkeley DB*, позволяют реализовать управление записями баз данных в виде «ключ-значение» и предоставляют доступ к множеству расширений, в числе которых механизм поддержки многопользовательского доступа и транзакций. Последнее расширение позволяет параллельно завершать по несколько транзакций (с перманентной модификацией данных) или откатывать их (с восстановлением баз данных до состояния перед началом транзакции).

Стоит отметить, что имеется дополнительное расширение SQLite API, которое представляет SQL-доступ к данным, хранящимся в *Berkeley DB*.

Программист может указать *Berkeley DB*, что нужно использовать написанные им самим функции выполнения операций над ключами. Например, метод доступа B+tree может использовать произвольную функцию сравнения, а Hash — произвольную хеш-функцию. Если пользовательские функции не определены, *Berkeley DB* использует собственные. Значения ключей могут быть произвольной длины.

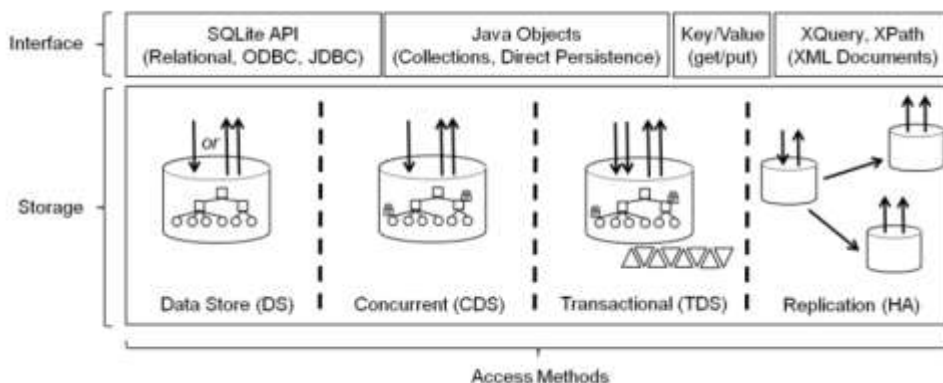


Рис.1. Методы доступа к Berkeley DB

Библиотека *Berkeley DB* поддерживает возможность многопользовательского доступа к базам данных. Под многопользовательским доступом понимается возможность чтения/записи из одной базы данных *Berkeley DB* из нескольких потоков (threads) или процессов внутри одного приложения.

Важно понимать, чем *Berkeley DB* не является.



Berkeley DB — это не сервер баз данных, обрабатывающий запросы, поступающие по сети. Это также не SQL-сервер, выполняющий запросы. Не является *Berkeley DB* и реляционной или объектно ориентированной СУБД. *Berkeley DB* является всего лишь встраиваемой в приложения библиотекой функций для реализации механизма баз данных. Разработчики постарались сделать ее переносимой, компактной, быстрой и надежной.

Варианты *Berkeley DB* по способу хранения информации

Ядром баз данных *Berkeley DB* является так называемое *Data Store*. *Data Store* представляет собой базовое хранилище информации БД *Berkley DB*. *Data Store* имеет в своем составе *API*, который обеспечивает однопользовательский доступ для записи и многопользовательский доступ на чтение данных для многопоточных приложений.

В *Berkeley DB* имеется две реализации *Data Store*: первая написана на языке *C/C++* и доступна из программ разработанных на *C/C++* или, с помощью интерфейсов-оболочек, из скриптов на языках *Perl* или *Tcl*. Вторая реализация полностью реализована на языке *Java* (*Data Store Java Edition*) и предоставляет в распоряжение разработчиков все преимущества платформы *Java*:

- платформенную независимость;
- масштабируемость;
- поддержку серверов приложений *J2EE*.

Дополнительно к обработке пар «ключ-значение» *Data Store Java Edition* реализует поддержку *Persistence API* (часть стандарта *Enterprise Java Beans 3.0*). Данный *API* позволяет записывать и считывать из БД *Berkeley DB* экземпляры классов *Java* напрямую, не прибегая к операциям обработки пар «ключ-значение».

Также *Berkley DB Java Edition* предоставляют в своем составе программные интерфейсы *Java Transaction API* (*JTA*), *J2EE Connector Architecture* (*JCA*) и *Java Management Extensions* (*JMX*). Данные *API* позволяют легко интегрировать *Berkeley DB* в *J2EE*-совместимые сервера приложений, в том числе и с *Oracle Application Server*.

Необходимо отметить, что вышеописанные реализации *Data Store* НЕ совместимы между собой на двоичном уровне, то есть невозможно базу данных, созданную с помощью *Berkeley DB Java Edition*, перенести для работы с приложением, использующим *C/C++* вариант *Data Store*.

Варианты *Berkeley DB* по способу доступа к информации

Исторически процедурный доступ является основным методом доступа к *Berkeley DB*. В случае использования процедурного метода доступа для манипуляции с данными используются функции или объектные методы *Berkeley DB*.

Помимо процедурного доступа к *Berkeley DB* с помощью функций *API*, возможно также для манипуляции с данными использовать *XQuery 1.0*, *XPath 2.0* и *XML Namespaces*. Способ доступа с помощью *XML* реализован в виде отдельного продукта и называется *Berkeley DB XML*.

С точки зрения реализации, *Berkeley DB XML* использует стандартное хранилище *Data Store* и реализован на языке *C/C++* как надстройка над обычными процедурными функциями доступа.

Berkeley DB XML, наряду с *XML*-документами, позволяет хранить и обрабатывать информацию в отличном от *XML* формате. Это обеспечивает гибкую поддержку приложений, которые помимо работы с *XML*-данными, обрабатывают информацию в различных форматах: текст, графика, видео, звук и т. д.

Редакции *Berkeley DB*

Редакции *Berkeley DB* делятся на четыре вида.

Berkley DB — *Data Store*



Как уже отмечалось ранее, *Data Store* — это ядро *Berkeley DB*. Представляет собой минимальный вариант использования *Berkeley DB* и обеспечивает функции записи данных из одного потока выполнения (thread) и чтения данных одновременно из нескольких потоков приложения.

Data Store входит во все редакции *Berkley DB*.

Concurrent Data Store

Предоставляя функции для работы с базовым хранилищем информации (*Data Store*), редакция *Concurrent Data Store* обеспечивает поддержку многопользовательской записи (multi-writers mode). То есть в многопоточном приложении несколько потоков одновременно могут записывать информацию в одну базу данных *Berkeley DB*.

Необходимо обратить внимание, что, в отличие от большинства СУБД, многопользовательский доступ в *Berkeley DB* может осуществляться и без поддержки транзакций!

Berkley DB — Transactional Data Store

Включает в себя всю функциональность видов редакций *Data Store* и *Concurrent Data Store*, то есть обеспечивает базовый набор функций и многопользовательскую запись в *Berkley DB*. Также дополнительно обеспечивает поддержку транзакций. В указанной редакции реализованы следующие возможности по работе с транзакциями:

- откат транзакции (rollback);
- фиксация транзакции (commit);
- автоматическая фиксация транзакции после каждой операции с БД (auto commit);
- вложенные транзакции.

Berkley DB — High Availability

Предоставляемые данным видом редакций возможности включают в себя весь набор функций трех предыдущих видов (*Data Store*, *Concurrent Data Store* и *Transactional Data Store*), но дополнительно предоставляется *API* для организации репликации между базами данных *Berkley DB*.

Вышеописанные виды редакций в зависимости от способа хранения информации и способа доступа к ней определяют конкретные редакции, которые предоставляются заказчикам.



На приведенном ниже рис.1 показана взаимосвязь редакций *Berkeley DB*.

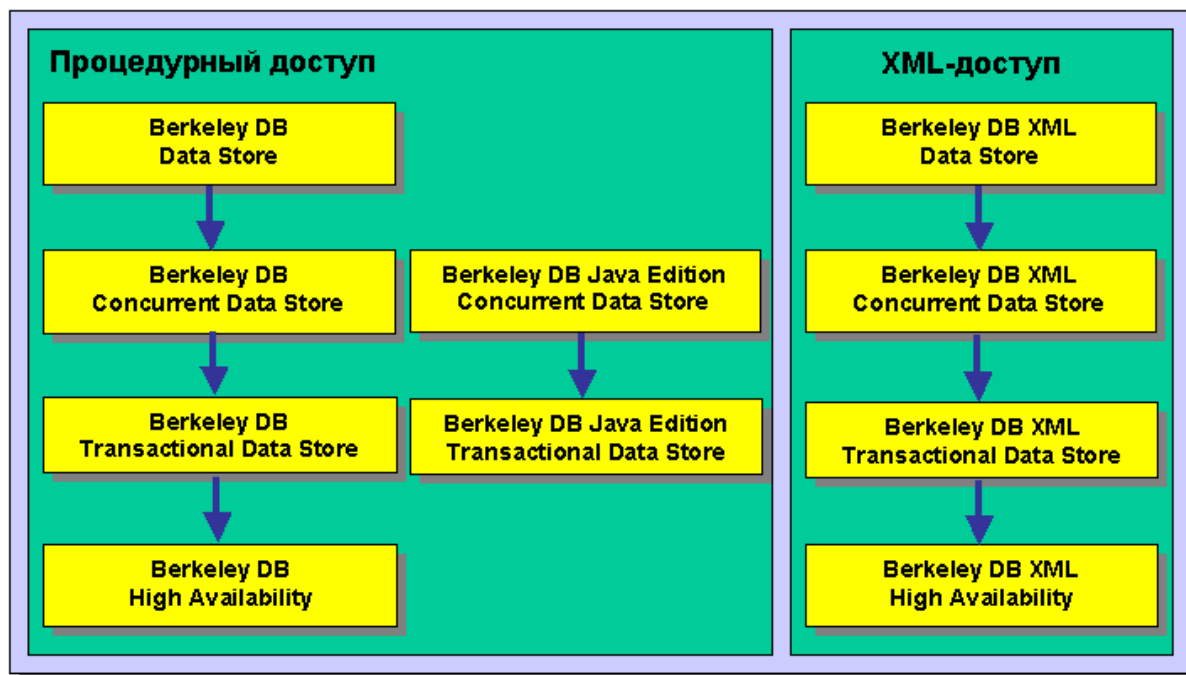


Рис.2. Редакции Berkley DB

Стрелками указаны связи между редакциями: каждая редакция включает в себя функциональные возможности всех предыдущих по иерархии сверху вниз.

Следует отметить, что в *Berkley DB Java Edition Data Store* в виде отдельной редакции не предоставляется и включена в состав всех редакций *Berkley DB Java Edition*. Также в настоящий момент функциональность *High Availability* для *DB Java Edition* недоступна.

NoSQL Database

Oracle NoSQL Database позволяет с легкостью управлять огромными объемами данных с динамическими схемами, поступающими из блогов, датчиков, веб-серверов, интеллектуальных счетчиков, социальных сетей, и данных, собранных для персонализации.

Oracle NoSQL Database, использующая простую модель данных типа «ключ-значение» (key-value), является масштабируемой СУБД высокой готовности, которая облегчает эффективное хранение данных в простом и гибком формате и идеально подходит для быстрой разработки и развертывания приложений.

Многие современные решения, основанные на технологии NoSQL, усложняют написание приложений, обеспечивая только итоговую согласованность данных (eventual consistency), то есть итоговое обновление ряда копий. Oracle NoSQL Database, напротив, управляет согласованностью и предлагает разработчикам ряд соответствующих настроек для удовлетворения различных потребностей приложения.

Oracle NoSQL Database, которая использует экспертизу и опыт Oracle в управлении корпоративными данными и поддерживается ведущей службой технической поддержки Oracle, проще устанавливать, настраивать и управлять, чем многие конкурирующие решения.



База данных Oracle NoSQL интегрирована со следующими продуктами Oracle: Big Data, Internet of Things, Spatial and Graph, Golden Gate, Relational Database, BerkeleyDB, Fusion Middleware, Enterprise Manager и Communication Elastic Charging Engine.

Быстрое и эффективное хранение больших объемов динамических данных

Ключевые преимущества Oracle NoSQL Database:

Программное обеспечение и техническая поддержка от Oracle — Oracle NoSQL Database является простой в установке, быстрой, масштабируемой и надежной СУБД. Поддерживаемая службой Oracle Support, Oracle NoSQL Database предлагает клиентам единую «точку контакта» для поиска и устранения неисправностей и обеспечения постоянной высокой готовности критически важных экземпляров баз данных.

Масштабируемая пропускная способность и ограниченное время ожидания — Oracle NoSQL Database масштабируется почти линейно, обеспечивая клиентам высокую надежность и производительность для широкого диапазона рабочих нагрузок и вариантов использования.

Простая модель программирования и работы — простая модель «основной + второстепенный ключ данных / значение» и облегченная поддержка ACID-транзакций обеспечивают исключительную гибкость программирования для больших наборов данных в горизонтально масштабируемых средах.

Простое управление — простота управления Oracle NoSQL Database обеспечивается благодаря консоли с веб-интерфейсом и открытыми API. Решение автоматически управляет и осуществляет мониторинг топологии, балансировки нагрузки, производительности, событий и уведомлений.

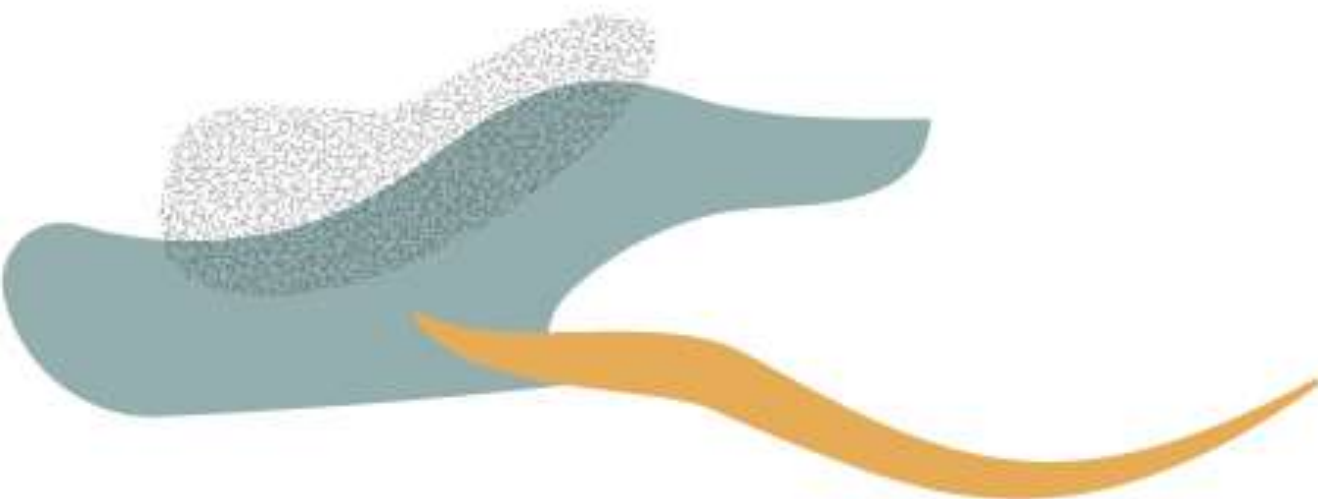




Продукты Oracle

—

Шлюзы



Шлюзы (Oracle Gateways)

Иногда в организации кроме СУБД Oracle имеются приложения на основе СУБД других производителей (MS SQL Server, Informix, DB2 и т. д.). Приложениям, работающим с Oracle, часто бывает необходимо использовать информацию не только из БД Oracle, но и из этих чужих БД, т. е. работать в гетерогенной среде. Решить эту задачу помогают шлюзы (Oracle Database Gateways).

Основная задача шлюза к конкретной СУБД — «спрятать» эту СУБД. То есть связка «шлюз + чужая СУБД» выглядит для приложений Oracle и СУБД Oracle как еще одна СУБД Oracle, понимающая предложения Oracle SQL и умеющая возвращать результаты запроса в формате, понятном Oracle. После этого мы уже можем работать в распределенной среде множества баз Oracle, что сервер Oracle делает очень хорошо.

Для того чтобы «прятать» чужую СУБД, шлюз должен выполнять три основные функции:

- преобразовывать запросы SQL Oracle в синтаксис запросов этой чужой СУБД;
- преобразовывать запросы к словарию Oracle (к метаданным) в запросы к объектам словаря чужой СУБД;
- преобразовывать типы и форматы данных чужой СУБД в форматы и типы данных Oracle.

Кроме того, шлюзы позволяют поддерживать выполнение распределенных транзакций (2 phase commit), не только запрашивать данные из чужих СУБД, но и модифицировать их, оптимизировать распределенные запросы с учетом возможностей чужой СУБД. Шлюз также позволяет вызывать хранимые процедуры чужой СУБД и в явном виде писать и посылать на выполнение команды на языке чужой СУБД.

В отличие от функции Oracle Generic Connectivity, реализуемой сервером Oracle на той же машине, где он установлен, шлюзы используют для доступа к чужим СУБД не ODBC или OLE DB-драйверы, а специальные адаптеры, написанные на API этих СУБД. Это позволяет расширить их функционал и ускорить их работу.

Шлюз можно установить на отдельном компьютере и обращаться через него к нескольким СУБД одного типа, его можно ставить на стороне чужой СУБД или на машине, где установлена Oracle Database.

Доступ приложений к данным чужих СУБД прозрачен. Для этого в БД Oracle создается обычный Database Link к этой СУБД (через шлюз) и далее пользователь работает так, как будто объекты чужой СУБД являются объектами локальной БД Oracle. Механизм синонимов позволяет еще больше упростить работу, присвоив этим объектам простые имена.

В семейство Oracle Database Gateways входят:

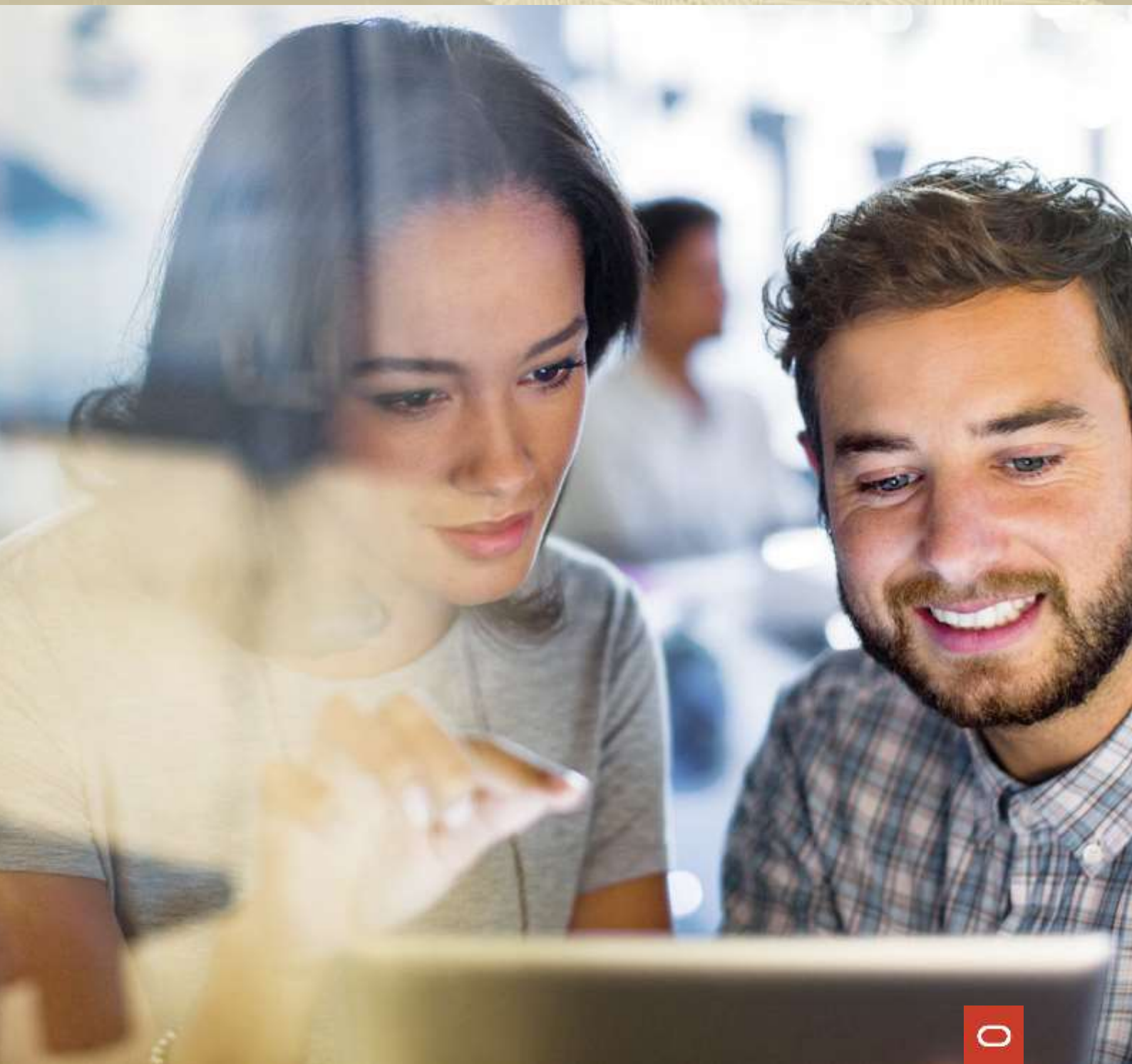
- Шлюзы, обеспечивающие доступ (посредством использования языка запросов SQL) к данным, хранящимся в отличных от Oracle базах данных на всех платформах открытых систем. В настоящий момент поддерживается доступ к следующим базам данных:
 - Microsoft SQL Server
 - Sybase
 - Informix
 - Teradata
 - Db2 (Gateway to DRDA)
- Группа продуктов, включающая шлюз к системам IBM DRDA, а также **Procedural Gateways** (*процедурные шлюзы*), обеспечивающие обработку вызовов удаленных процедур, причем удаленные процедуры определены и выполняются в отличной от Oracle программной системе. Доступны Gateway for WebSphere MQ и Procedural Gateway for APPC.



Продукты Oracle



Новые возможности релизов СУБД Oracle



Новые возможности Oracle Database 12.1

В июле 2013 года компания Oracle объявила о выпуске новой версии своей флагманской СУБД Oracle Database 12c. Эта версия разрабатывалась более 5 лет и длительное время проходила функциональное и нагрузочное тестирование. В ней реализовано более 500 новых возможностей, о наиболее важных и интересных из них мы и поговорим.

Более ранние версии СУБД Oracle имели префиксы i (8i и 9i) и g (10g и 11g), т. е. были платформами для интернет-вычислений (i) и грид-вычислений (g). Oracle Database 12c имеет префикс c, потому что в первую очередь ее можно назвать платформой для облачных (Cloud) вычислений. Действительно, Oracle Database 12c очень удобна для развертывания сервисов баз данных (DBaaS — Database as a Service). Но так уж вышло, что самые интересные новшества Oracle Database 12c, о которых и пойдет речь далее, тоже начинаются с буквы C. Это и уникальные свойства для консолидации баз данных (Consolidation), и средства автоматического сжатия данных (Compression), и уникальные средства автоматического повтора прервавшихся транзакций (Application Continuity).

Как всегда, в новой версии СУБД улучшены многие алгоритмы, что привело к ускорению выполнения отдельных операций. Например, создание новых БД происходит в тысячу раз быстрее, апгрейд ускорится на 40 %, работа с геоинформацией ускорится в 40 раз, операции аудита — в 5 раз и т. д., и т. п. Поэтому пользователи Oracle Database 12c сразу получают ускорение работы многих своих приложений.

Все новые возможности СУБД Oracle Database 12c можно разделить на несколько групп.

- Консолидация и мультиарендная архитектура
- Оптимизация хранения данных
- Высокая надежность
- Безопасность данных
- Управляемость
- Улучшения для разработчиков приложений БД

Рассмотрим эти новые возможности подробнее.

I. Консолидация и мультиарендная архитектура

Задача консолидации баз данных стоит перед многими организациями. Сейчас каждая БД размещается на отдельном компьютере, и сложность администрирования множества БД растет пропорционально увеличению их количества. Понятно, что консолидация баз данных на меньшем числе компьютеров увеличит эффективность использования компьютеров и, возможно, снизит сложность администрирования (т. е. уменьшатся CAPEX и OPEX). Вопрос состоит в том, как консолидировать базы данных.

Самый простой и популярный сегодня способ консолидации — виртуализация. Если у Вас есть 100 компьютеров с базами данных, Вы можете сделать 100 виртуальных машин (образов этих физических машин) и помещать на одной мощной физической машине несколько виртуальных машин. Например, поместить 100 виртуальных машин на один физический компьютер. Этот вариант консолидации самый простой и быстрый, но он имеет множество недостатков. Во-первых, количество администрируемых объектов (виртуальных машин, баз данных) не уменьшилось. Во-вторых, при работе СУБД внутри виртуальной машины, которая работает на физической машине, мы получаем дополнительные накладные расходы (слой виртуализации между СУБД и железом). В-третьих, каждая виртуальная машина требует выделения большого количества ресурсов (память, процессоры, ввод-вывод) физического компьютера. То есть такой вариант консолидации является наименее эффективным.



Второй вариант консолидации баз данных реализуется при развертывании на одной физической машине нескольких баз данных, работающих одновременно. Этот подход часто используется в облачных вычислениях DBaaS. Здесь нет накладных расходов от виртуализации, степень консолидации выше, но и этот подход имеет ряд недостатков, о которых мы поговорим ниже.

Третий вариант консолидации — консолидация схем из разных БД в одной БД. Это самый эффективный вариант консолидации, но он является самым сложным. В этом варианте мы помещаем схемы разных БД в одну единую БД. Это очень сложная операция, так как схемы конфликтуют между собой по имени объектов, используют разные варианты процедур, требуют разных параметров настройки СУБД и т. д. Несмотря на свою эффективность, такая консолидация занимает очень долгое время и используется редко. Это большой и дорогой проект.

В СУБД Oracle Database 12c реализован новый вариант консолидации баз данных, сохраняющий эффективность консолидации схем, но лишенный ее недостатков. Для его реализации была изменена архитектура СУБД. Теперь у заказчика есть выбор: развернуть СУБД по старой традиционной архитектуре или реализовать новую мультиарендную архитектуру (Multitenant architecture).

Мультиарендная архитектура

Каждая БД Oracle представляет собой набор файлов на дисках. Для работы с этой БД запускается экземпляр (instance) Oracle, который получает SQL-запросы, выполняет их и возвращает результат. Для работы с каждой БД необходим один (в случае кластера несколько) экземпляр Oracle. При запуске каждый экземпляр «откусывает» для себя большую область оперативной памяти и запускает набор процессов операционной системы, которые и реализуют его работу.

Поэтому в случае варианта 2 консолидации, когда мы размещаем на одном физическом компьютере несколько БД (например, БД для ERP-системы, БД для CRM-системы и БД для аналитической системы), у нас получается три набора файлов баз данных, три области памяти (1 для каждого экземпляра) и три набора процессов операционной системы. Если мы консолидируем 100 баз на одном компьютере, то число 3 надо заменить на число 100, причем и процессы, и области памяти частично дублируют друг друга. Это ведет к неэффективному использованию оборудования.

В новой мультиарендной архитектуре для консолидации баз данных создается одна база данных, называемая контейнерной БД (Container DB — CDB), и ее обслуживает один экземпляр Oracle с одной областью памяти и одним набором процессов. Все консолидируемые БД помещаются в этот контейнер. И этот один экземпляр Oracle будет обслуживать все БД контейнера. Мы видим, что при таком подходе резко возрастает эффективность использования ресурсов компьютера. Более того, базы можно не только помещать в контейнерную БД, но и легко извлекать из нее и переносить в другие контейнерные БД. Это похоже на подключение/отключение USB-устройств к компьютеру, поэтому подключаемые БД назвали Pluggable Database (PDB).

Самое интересное, что после помещения в контейнерную БД эти базы (PDB) остаются изолированными и независимыми, т. е. приложения, которые работают с ними, менять не надо, они даже не знают, что БД переехала в контейнерную БД, а PDB контейнерной БД не знают друг о друге и при подключении новых баз в контейнерную БД никаких конфликтов не возникает.

Поскольку один экземпляр Oracle обслуживает множество подключенных баз, то на одном физическом компьютере помещается больше консолидированных БД. Oracle провел эксперимент. Вначале на компьютере размещали БД по старой традиционной технологии. На компьютер «влезло» 50 БД, после чего ресурсы закончились. Когда же на тот же компьютер стали размещать БД по новой технологии, на нем уместилось в 5 раз больше (т. е. 250) БД. Очевидно, что новая архитектура упрощает консолидацию и повышает ее эффективность.



Но на этом ее преимущества не заканчиваются. Она позволяет управлять множеством БД контейнерной базы как единым целым. Допустим, у Вас было 100 разных БД, которые Вы консолидируете в одну контейнерную БД. Раньше для каждой БД Вы должны были иметь администратора БД, создавать резервные копии, выполнять обновления, устанавливать исправления, создавать резервную (standby) БД и т. д. Теперь можно создавать не 100 резервных копий, а одну (для контейнерной БД) и из нее при необходимости восстанавливать нужную PDB, можно обновлять и исправлять не 100 разных БД, а ограничиться обновлением и исправлением одной контейнерной БД. Если надо обеспечить надежность работы приложений за счет создания резервных (standby) БД, то можно создать одну резервную БД для всей контейнерной БД, а не 100 индивидуальных резервных БД. Новая мультиарендная архитектура приведена на рисунке 1.

Кстати, в случае обновления у администратора теперь есть выбор. Он может за один прием обновить экземпляр Oracle и, следовательно, все PDB, обслуживаемые этим экземпляром контейнерной БД. Но если надо обновить не все PDB контейнерной БД, то проще создать рядом другую контейнерную БД более свежей версии (например, 12.2) и просто перенести туда требуемые PDB. При их подключении автоматически производится их обновление до последней версии.

Еще одним преимуществом новой архитектуры является быстрое клонирование баз данных. В каждой контейнерной БД при ее создании создается управляющий контейнер (ROOT) и контейнер заготовка-пустышка из которого можно быстро клонировать новые PDB. Создавая новую PDB, Вы можете указать, что она создается на основе этой пустышки, либо на основе одной из PDB этой контейнерной БД, либо на основе PDB другой контейнерной БД. При создании PDB используется много процессоров (параллельное выполнение команд). Поэтому оно выполняется быстро.



Рисунок 1. Мультиарендная архитектура

Если система хранения, где расположена контейнерная БД, поддерживает механизм снимков (snapshot) (это поддерживается в ASM, ZFS storage appliance, Pilar Axiom), то он будет использован при клонировании, а это означает, что клон терабайтной БД создается за секунды.

Облегчается при новой архитектуре и передача БД заказчикам. Достаточно выполнить операцию отключения PDB от контейнерной БД и передать получившийся XML файл и файлы отключенной БД заказчику. Он выполнит операцию подключения и начнет работать с новой БД. Конечно, PDB одной контейнерной БД будут конкурировать за ресурсы экземпляра Oracle и компьютера. Resource manager позволяет определить приоритеты использования ресурсов.

Консолидированный DB Replay

Чтобы понять, как повлияет консолидация БД на их работу, желательно эту консолидацию предварительно протестировать. У Oracle есть опция RAT (Real Application Testing), которая позволяет захватить нагрузку



на реально работающей БД, а затем проиграть ее на тестовой БД и проанализировать влияние изменений на функционирование БД. В Oracle Database 12c этот функционал был расширен. Теперь Вы можете захватить нагрузку с баз данных до консолидации, консолидировать эти БД, консолидировать захваченную нагрузку и проиграть ее на консолидированной системе, чтобы убедиться, что в результате консолидации не появилось ошибок в работе приложений и что производительность приложений после консолидации баз не ухудшилась.

Кстати, для повышения надежности и скорости работы консолидированных баз лучше использовать не один экземпляр Oracle, а несколько (т.е. использовать RAC — Real Application Cluster для работы с консолидированной БД и ее PDB). При этом механизм сервисов позволит определить, какие узлы кластера с какими PDB будут работать. Для реализации мультиарендной архитектуры с более чем 1 PDB надо купить новую опцию Multitenant option.

II. Оптимизация хранения данных

Объемы данных, хранимых в базах данных, постоянно растут и справляться с ними становится все труднее. Все больше бюджета ИТ уходит на покупку дисков и систем хранения. Справится с этим можно только двумя способами: либо сжимать данные (но желательно без ущерба для производительности), либо перемещать редко используемые данные на более дешевые и медленные диски.

В Oracle давно существует механизм управления жизненным циклом информации (ILM). ILM Manager подсказывал администратору, какие данные можно сжимать/перемещать и какая выгода будет от этого. Но выполнять эти операции приходилось вручную, и ориентировались они не на реальную информацию об использовании данных, а на сами значения данных (например, сжимались данные о продажах прошлого года или данные о конкретных регионах).

Известно, что данные, поступающие в БД, сначала очень активны (горячие данные). Их часто запрашивают и изменяют. Затем данные начинают остывать. Они еще часто запрашиваются, но редко обновляются, т.е. переходят в разряд «теплых» данных. Постепенно количество запросов и изменений сводится к минимуму, а затем данные переходят в разряд «холодных» данных, с которыми работают очень редко. Холодные данные обычно используются для соответствия законодательству (например, хранить 10 лет) или для долговременной аналитики.

Температурная карта и автоматическая оптимизация данных

Oracle Database 12c вводится понятие температурной карты (Heat Map). В ней можно видеть температуру данных (горячие, теплые, холодные), т.е. она показывает, когда данные (блоки, секции, таблицы) последний раз использовались и как (читались, создавались, изменялись). Температурная карта уровня БД показывает использование таблиц и секций (partitions). Температурная карта уровня блока показывает, когда блок данных последний раз изменялся.

Имея эту информацию, мы можем указать Oracle Database 12c, когда он автоматически должен выполнять те или иные действия по сжатию/перемещению данных. Например, блоки данных, которые не менялись более недели, можно сжать в 2–3 раза, оставляя доступными для быстрой модификации данных, а таблицы/секции, которые не менялись более года, можно сжать в 15–50 раз для архивного хранения. Политика такой автоматической оптимизации данных (Automatic data Optimization — ADO) описывается декларативно и затем сервер БД будет сам ее выполнять для указанных таблиц или для всего табличного пространства.

В Oracle существует много вариантов сжатия данных. Это и Advanced Compression (сжатие в 2–3 раза строк внутри блока с минимальным влиянием на скорость операций DML), и гибридное колоночное сжатие в 10 раз (Query HCC), и максимальное гибридное колоночное сжатие в 15–50 раз (Archive HCC). Причем и в Query HCC и в Archive HCC можно использовать один из 2 алгоритмов сжатия (слабое и сильное). Т.е. мы имеем 5



вариантов сжатия, причем колоночное сжатие применимо только на специальном оборудовании (Exadata, ZFS storage Appliance, Pilar Axiom). Все эти операции сжатия прозрачны для приложений, т. е. переписывать приложения не надо. Оно не знает, сжаты ли данные, с которыми оно работает.

Остается только связать действие (вид сжатия или перемещение в другое табличное пространство) с температурой данных. Это делается за счет задания политик ILM. Например, политика

... ROW STORE COMPRESS ADVANCED ROW

AFTER 1 DAY OF NO MODIFICATION

сожмет все блоки таблицы, которые не модифицировались более 1 дня, используя механизм Advanced compression, а политика

... COMPRESS FOR QUERY LOW

AFTER 3 WEEKS OF NO MODIFICATION

сожмет все секции таблицы, которые не модифицировались более 3 недель, используя колоночное сжатие Query Low. Политика

... TIER TO cheap_disk TABLESPACE

переместит холодные секции таблицы с быстрых дисков на дешевые диски табличного пространства cheap_disk, когда в первичном табличном пространстве останется мало места. Сжатие/перемещение выполняется автоматически в фоновом режиме. Сжатие внутри блока — раз в день (частоту можно поменять), сжатие/перемещение таблиц и секций — во время выполнения операций поддержки (maintenance window).

Следует отметить, что, в отличие от всех существующих механизмов автоматизации ILM, этот знает об операциях работы с БД. Т. е. если данные читались не пользовательскими запросами, а для резервного копирования, сбора статистики, DDL операций, перемещения таблиц, Oracle поймет, что температуру данных менять не надо. Oracle отличает выборку по индексу от операций full scan при определении температуры.

Как видно из примеров, политики состоят из 2 частей: действие (сжатие, перемещение) и условие. Для блоков данных учитываются только операции модификации записей, для таблиц и секций — чтение, создание, редкое чтение (Low Access) и модификация данных. Для перемещения секций можно также учитывать количество свободного места в табличном пространстве. Пользователь может также написать свою функцию проверки температуры данных и использовать ее в условии. Она может, например, учитывать значения самих данных. Можно также указать, что после перемещения данные переходят в режим только для чтения. Это позволит реже выполнять операции резервного копирования.

Valid-Time Temporal

В Oracle давно существует механизм FLASH BACK QUERY, позволяющий на текущей БД сделать запрос данных в прошлом. Например, указав в операторе SELECT фразу: «... AS OF '1-JAN-2013'», я получу результат, который не учитывает изменения данных, произошедшие после 1 января 2013 года. Но этот механизм учитывает только информацию о реальном изменении данных и называется Transaction-Time Temporal.

Теперь представим, что у меня есть таблица сотрудников и я планирую нанять нового сотрудника. Но он выйдет на работу не сегодня, а только 1 сентября 2014 года. Более того, 3 месяца (по 30 ноября) у него будет небольшая зарплата (пробный период), а с 1 декабря зарплата станет нормальной. Если я сейчас введу информацию об этом сотруднике, то запрос к таблице покажет ее прямо сейчас, поэтому я вынужден ждать до 1 сентября. Новый механизм Valid-Time Temporal позволяет мне ввести данные о сотруднике прямо сейчас, но указать в отдельных полях дату начала и конца актуальности строки. Поэтому, указав для строки с низкой зарплатой срок действия с 1 сентября по 30 ноября, а для строки с нормальной зарплатой дату начала действия 1 декабря, я получу ситуацию, когда запрос данных к таблице сотрудников, выполненный



до 1 сентября, мне этого сотрудника не покажет, в период с 1 сентября по 30 ноября покажет его с низкой зарплатой, а в период после 30 ноября покажет его с нормальной зарплатой.

Умный протокол хранения (OISP)

В Oracle Database 12c вводится новый протокол обмена данными с системой хранения ZFS Storage Appliance, называемый Oracle Intelligent Storage Protocol (OISP). Он ускоряет работу с системой хранения ZFS и упрощает администрирование ZFS.

Улучшения при работе с секциями (partitions)

Асинхронная перестройка глобальных индексов после операций TRUNCATE или DROP секции. Теперь после удаления или очистки секции таблицы глобальные индексы не переводятся в недействительное состояние, а производится их перестройка в фоновом режиме. В операторах изменения секций (слияние, разделение, удаление и очистка) теперь можно обрабатывать сразу несколько секций. Поддерживается смешанное секционирование Interval+Reference. Перемещение секций (ALTER TABLE... MOVE PARTITION) можно выполнять в режиме онлайн, без остановки. DML-операции при этом не блокируются и продолжают свою работу.

III. Высокая надежность

Пожалуй, больше всего нововведений в Oracle Database 12c связано с обеспечением еще большей надежности работы СУБД и приложений.

Автоматический повтор прерванных транзакций (Application Continuity)

Наверное, многие из вас сталкивались с такой ситуацией при заказе билетов или товаров через интернет. Вы выбрали товар, ввели данные кредитной карточки для оплаты, нажали кнопку «купить» и ждете подтверждения. А подтверждения нет, т. к. произошел сбой либо на компьютере с БД, либо в системе хранения, либо в сети, либо в экземпляре СУБД. Подождав немного, Вы повторяете попытку (нажимаете кнопку) и, о чудо, все сработало, подтверждение получено. Сделав так однажды, я потом случайно заглянул в корзину покупок и обнаружил, что я купил (и, следовательно, оплатил) товар два раза. Происходит это потому, что пришедшая в БД транзакция выполнялась первый раз успешно, а вот подтверждение получить сервер приложения или клиент не успел. В такой ситуации пользователь может потерять данные и деньги, или ввести данные дважды, или начать перезагрузку сервера приложений.

Чтобы избежать такой ситуации, Oracle встроил в Oracle Database 12c компоненту Transaction Guard. Она работает на сервере БД и знает, зафиксирована или нет транзакция. Например, при работе в кластере БД мы можем выполнять транзакцию через первый узел, в случае его сбоя клиент или сервер приложений получают информацию об ошибке при выполнении транзакции, и тогда этот клиент / сервер приложений через другой живой узел кластера сможет спросить у Transaction Guard, а была ли зафиксирована эта транзакция. Если она была зафиксирована, то повторять ее не надо и клиент получит информацию об успешном завершении транзакции. Пример работы Transaction Guard приведен на рисунке 2.





Рисунок 2. Пример работы Transaction Guard

Transaction Guard свой API, но в большинстве случаев JDBC драйвер все сделает сам, автоматически. Теперь, когда мы знаем, что транзакция точно не завершилась и знаем тип сбоя, и если это восстановимый сбой, можно попытаться автоматически повторить прерванную транзакцию. Делает это новый уникальный механизм Application Continuity.

Application Continuity

Многие заказчики, работающие с кластером Oracle, знакомы с механизмом TAF (Transparent Application Failover). Он заключается в том, что если через один узел кластера выполняется оператор SELECT, который выводит данные из БД, и происходит сбой этого узла кластера, то клиент или сервер приложений получает сообщение о сбое (FAN — Fast Application Notification) и повторяет этот SELECT через другой узел кластера. Вывод данных прерванного оператора продолжается с той точки, где произошел сбой, так что пользователь даже не замечает, что был сбой узла кластера.

Похожий механизм, но уже для транзакции реализует Application Continuity. Когда драйвер JDBC получит сообщение о сбое при выполнении транзакции, он проверит, что это восстановимый сбой (recoverable error), проверит через живой узел, действительно ли транзакция не зафиксирована, и если это так, повторит выполнение транзакции. Т. е. для конечного пользователя сбой при выполнении транзакции не приводит к ошибке приложения.

Конечно, не все транзакции можно выполнять повторно. Разработчик приложения может указать, какие транзакции автоматически повторять нельзя, а какие можно, но с предварительным восстановлением контекста сеанса (для этого нужно написать и указать процедуру восстановления контекста). Если транзакция использовала последовательности, функции SYSGUID (получение уникального глобального идентификатора) и SYSDATE то replay драйвер JDBC восстановит их значения. Он также проверит, что успевшие выполниться до сбоя шаги транзакции при проигрывании дают тот же результат. Транзакции, которые пишут данные во внешние файлы или модифицируют другие БД, лучше не повторять (будет дублирование операций). Таким образом, уникальный механизм Application Continuity без сложного программирования маскирует большинство сбоев ПО, систем хранения, оборудования и сети.



Active Data Guard Far Sync

Для повышения надежности работы приложений и борьбы с катастрофическими сбоями пользователи Oracle часто используют механизм создания резервных баз данных (Standby) в удаленных резервных центрах. В случае сбоя основной БД пользователей можно переключить на резервную БД. Синхронизацией основной и резервных БД занимается компонента Data Guard. Опция Active Data Guard (ADG) позволяет открыть резервную БД на чтение, не останавливая синхронизацию данных. Это позволяет вынести часть функционала приложений с основной на резервные БД. В Oracle Database 12c эти возможности расширены.

Резервные БД могут синхронизироваться с основной БД либо в синхронном, либо в асинхронном режиме. При синхронном режиме работа основного узла продолжается только после того, как получена информация о том, что изменения основного узла получены и записаны резервным узлом. Это влияет на производительность основной БД, но защищает от потери данных. В случае асинхронной передачи, основная БД не ждет подтверждения, но в этом случае нет гарантии, что изменения достигли резервной БД и применились там. Задержка работы основной БД при синхронном режиме Data Guard прямо пропорциональна расстоянию между основной и резервной БД. Сегодня резервный центр может находиться в другой стране и даже на другом континенте. В этом случае, чтобы не потерять данные (как при синхронном режиме) и не снижать производительность основной БД, делалось следующее.

Рядом с основной БД ставилась резервная БД и ее Data Guard работал с основной БД в синхронном режиме, т. к. расстояние было невелико. Вторая резервная БД создавалась далеко от первой и ее Data Guard работал с первой резервной БД уже в асинхронном режиме. Т. е. первая резервная БД выполняла роль посредника, хотя и представляла из себя полномасштабную резервную БД с данными. Новый механизм Far Sync позволяет заменить этого посредника на облегченный вариант экземпляра Oracle, называемый ADG Far Sync. Он не содержит файлов данных, а только управляющие файлы и журнальные файлы, занимает мало места. Посредник передает изменения даже после выхода из строя основной БД. Кстати, если после переключения на удаленную резервную БД нам надо перевести бывшую основную БД в режим Standby и организовать синхронизацию в обратную сторону, мы можем разместить еще один ADG Far Sync возле бывшей удаленной резервной БД и заставить его работать с ней в синхронном режиме. А он будет в асинхронном режиме передавать изменения своему двойнику, размещенному рядом с бывшей основной БД. Так решается дилемма между скоростью работы и потерей данных.

Глобальные сервисы (Global Data Services)

При работе с кластером БД, где несколько экземпляров Oracle работают с одной БД, мы при открытии сеанса связываемся с любым прослушивателем (listener) и он перебрасывает сеанс (connection) на наименее загруженный узел кластера. Тем самым осуществляется балансировка нагрузки в кластере. При выходе узла из строя сеанс переключается на другой живой узел кластера.

Сегодня, когда мы имеем много неравномерно нагруженных копий одной и той же БД (резервные узлы, открытые на чтение, копии БД, синхронизирующиеся через Golden Gate и открытые для изменений), хотелось бы осуществлять такую балансировку нагрузки и обеспечить высокую надежность не между узлами кластера одной БД, а между копиями одной БД, возможно, удаленными очень далеко друг от друга. Сделать это позволяет механизм Global Data Services (GDS). Приложение устанавливает связь не с конкретным экземпляром БД или кластером, а с глобальным сервисом, который затем переключает эту связь на одну из баз глобального сервиса.

Вводится понятие пула GDS, т. е. набора одинаковых БД, которые могут предоставлять один сервис, и GDS региона (группа БД и клиентов, отдельной сети). Это позволяет менеджеру глобальных сервисов выбрать для назначения БД с учетом ее загруженности и удаленности (регион) от клиента. Учитывается также время отклика каждой БД и отставание резервной копии от основной БД. Можно задать допустимое время отставания, работает ли сервис на изменение (или только на чтение) и т. д. Вся эта информация хранится в каталоге глобального менеджера сервисов и используется при выборе наилучшей БД для подключения. Если одна из баз данных пула GDS выходит из строя, сервис может мигрировать на другую БД пула



по возможности того же региона. Таким образом приложения всегда работают с наименее загруженными серверами БД (рисунок 3).

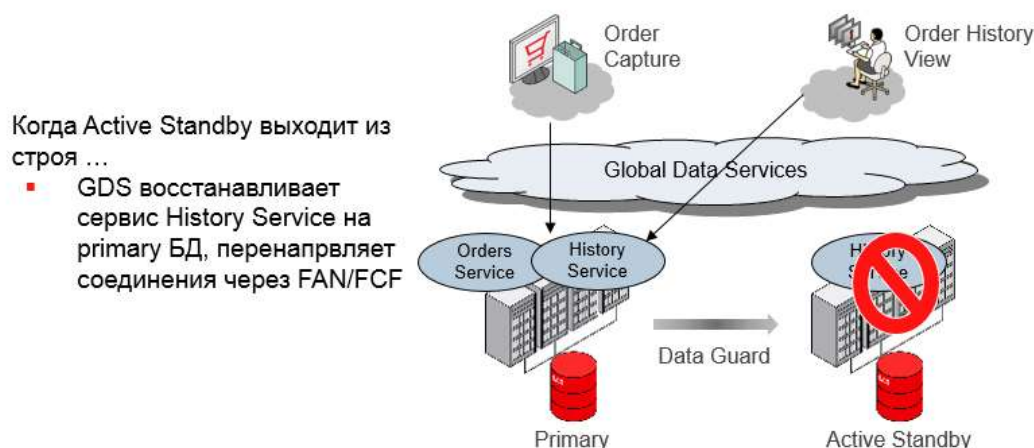


Рисунок 3. Восстановление после сбоя в глобальном сервисе

Прочие изменения в Data Guard

Каскадный Data Guard в режиме реального времени. Если у нас есть основная БД, ее первая резервная копия и вторая резервная копия, которая получает изменения из первой резервной копии, то в Oracle 11.2 первая резервная копия ожидала переключения журналов, прежде чем отправить архивный журнал с изменениями второй резервной БД. Это увеличивало отставание второй резервной БД от основной БД. В Oracle Database 12c первая резервная БД пересылает информацию об изменениях на вторую резервную БД, не ожидая переключения журналов.

Data Guard Fast Sync. В Oracle 11.2 при работе Data Guard в режиме синхронной передачи изменений, резервная БД сообщала о получении изменений только после их записи в свой журнал. Это замедляло работу. Теперь она сообщает о получении, до того как начинает запись в журнал.

Проверка готовности к переключению основной и резервной БД. Команда Validate Database проверит, готовы ли базы к переключению, отсутствует ли отставание, выявит несоответствия.

Уникальные последовательности и временные таблицы на резервной БД с Active Data Guard. Если резервная БД открыта на чтение, то работающим с ней приложениям иногда надо запрашивать уникальные значения у последовательностей (Sequence) и записывать данные во временные таблицы. Раньше это было невозможно. Теперь ADG позволяет это делать. Причем резервная БД запрашивает и получает от основной БД диапазоны уникальных значений последовательностей, поэтому значения на основной и резервной БД не дублируются. На временных таблицах резервной БД теперь можно выполнять операции языка DML. Все это упрощает вынос приложений с основной БД на резервную.

Flex ASM

В последних версиях СУБД Oracle рекомендуется хранить БД не на обычной файловой системе, а в ASM (Automatic Storage Manager). Это обеспечивает балансировку ввода-вывода, защиту от сбоев дисков и сбоев ввода-вывода за счет автоматической избыточности хранения блоков данных, увеличение скорости ввода-вывода, простоту конфигурирования и управления системой хранения, возможность ее наращивания на лету и т. д. Но для того, чтобы использовать ASM в Oracle 11.2, необходимо было на каждом сервере БД и на каждом узле кластера запускать не только экземпляр Oracle, но и экземпляр ASM, обслуживающий этот экземпляр Oracle. Это требовало дополнительных ресурсов компьютера, и при выходе из строя экземпляра ASM весь сервер или узел кластера, где он находился, выходил из строя. Flex ASM позволяет разнести экземпляры ASM



и Oracle по разным узлам кластера, более того, один экземпляр ASM теперь может обслуживать несколько экземпляров Oracle, причем они могут находиться на разных узлах. Это увеличивает количество вычислительных ресурсов, которые может использовать экземпляр Oracle, а при выходе из строя одного из экземпляров ASM экземпляры Oracle, работавшие с ним, переключаются на оставшиеся в живых экземпляры ASM, что повышает надежность и производительность СУБД.

Перемещение файлов данных БД в онлайн, дополнительные операции в онлайн

Теперь можно перемещать файлы данных БД на другие диски и в папки не останавливая работу с БД. Команда ALTER DATABASE MOVE DATAFILE ... позволяет это сделать. Кроме того, не останавливая работу приложений, теперь можно выполнять и такие операции, как DROP INDEX, ALTER INDEX UNUSABLE, ALTER INDEX VISIBLE/UNVISIBLE, DROP CONSTRAINTS, MOVE PARTITION

Новые возможности Recovery Manager (RMAN)

Во-первых, теперь из бэкапов, сделанных с помощью RMAN, можно восстанавливать в БД отдельную таблицу или группу таблиц. Это делается одной командой. Ранее восстанавливать таблицы можно было только из файлов экспорта. Во-вторых, теперь RMAN позволяет делать кроссплатформенный бэкап. Очень часто надо переносить БД с одной вычислительной платформы на другую, возможно даже имеющую другой эндиан (формат блоков). Например, перенос БД с AIX на Exadata (Linux, ASM) был непростым делом.

Теперь RMAN, делая бэкап БД или восстановление БД из бэкапа, может перекодировать файлы БД под другую платформу на лету. Процедура миграции БД при этом сильно упрощается. Можно перекодировать файлы в момент создания бэкапа, указав в команде BACKUP предложение TO PLATFORM = 'имя_платформы'. Можно создавать резервные копии файлов без преобразования, указав в команде BACKUP предложение FOR TRANSPORT, а потом при восстановлении из этого бэкапа на новой платформе указать в команде RESTORE, на какой платформе бэкап делался, с помощью предложения FROM PLATFORM 'имя_старой_платформы'. Перекодировка выполняется в момент восстановления БД из бэкапа.

RMAN в Oracle Database 12c позволяет уменьшить время нахождения табличного пространства в режиме только для чтения при перемещении табличных пространств (Transportable Tablespace). Используя кросс платформенный бэкап, Вы можете, оставляя табличное пространство открытым для изменений, сделать полный его бэкап уровня 0, затем инкрементальный бэкап уровня 1 (чтобы захватить изменения, произошедшие во время полного бэкапа), и только после этого перевести на короткое время табличное пространство в режим только для чтения, чтобы выполнить последний инкрементальный бэкап уровня 1 и выгрузить метаинформацию, необходимую для подключения транспортируемого табличного пространства к новой БД. Это позволит значительно сократить простои приложений, работающих с БД.

IV. Безопасность данных

Искажение секретных данных на лету (Data Redaction)

В БД могут содержаться конфиденциальные данные, которые не всем можно показывать, хотя приложения их используют. Если нам надо передать такую БД в другую организацию (разработчикам, тестировщикам), то мы можем замаскировать конфиденциальную информацию на постоянной основе с помощью пакета OEM Masking Pack. Но что делать, если части пользователей БД надо показать незамаскированную информацию, а части пользователей — искаженную или замаскированную информацию, причем одновременно? Например, банковские программы должны видеть весь номер банковского счета, а служба работы с заказчиками —



только последние 4 цифры номера счета. Или врачам должен быть виден весь диагноз, а больному только успокаивающее описание (не показать ничего нельзя — будет волноваться).

Чтобы сделать это прозрачно на уровне БД, а не приложений, надо уметь искажать данные на лету в момент выполнения запроса. И неважно, пришел запрос от приложения или из SQL PLUS. Такое искажение на лету выполняет механизм Data Redaction (смотри рисунок 4, где Call center и Processing видят разные данные).

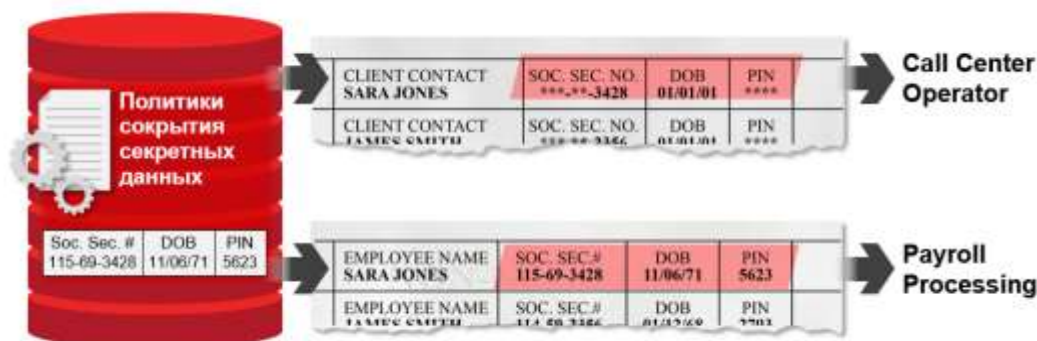


Рисунок 4. Искажение данных на лету

Он работает на основе политик, в которых мы можем декларативно описать, для кого и как искажаются данные. Тип искажения данных может зависеть от имени пользователя, приложения, IP-адреса, контекста приложения и т. д. Причем для приложений все это прозрачно (их не надо менять), просто они начинают получать и отображать искаженные данные. Поддерживается 4 вида искажений: полное (подмена значения поля), частичное (подмена части значения, например замена части символов звездочками), случайное (изменение символов на другие символы, но с сохранением формата), регулярные выражения (которые описывают правила искажения). Влияние искажений на производительность минимально.

Анализ привилегий (Privilege Analysis)

Не секрет, что у многих пользователей БД есть избыточные привилегии на выполнение операций и работу с объектами БД. Эта ситуация складывается исторически, т. к. никто не занимается детальным анализом, какие привилегии нужны конкретному пользователю. Обычно привилегии назначаются пачками (ролями) или срочно выдаются пользователю выполнения для конкретных работ, а потом не отбираются. Все это увеличивает риски и снижает защиту данных в БД. Поэтому в Oracle Database 12c был введен новый механизм анализа привилегий. Включив его, администратор может посмотреть, какие же привилегии реально использовал пользователь за определенный промежуток времени, и сравнить их со списком имеющихся у пользователя привилегий. На основе такого анализа можно отобрать у пользователя лишние привилегии и повысить защищенность БД.

Изменения в аудите

В Oracle Database 12c изменилась архитектура аудита. Теперь вся информация аудита хранится в единой открытой только на чтение таблице. Можно задать политики аудита, основываясь на времени дня, IP-адресе, имени программы, имени пользователя. Множество различных таблиц аудита в Oracle 11.2 заменено на единую таблицу аудита, что облегчает работу администратора.

Новые роли

В Oracle Database 12c появились новые стандартные роли, ограничивающие возможности администратора, выполняющего конкретный набор работ. Это роль SYSDBG для администратора Data Guard и роль SYSBACKUP для выполнения операций резервного копирования и восстановления БД.



Шифрование в SQL*Net

Если раньше шифрование данных, передаваемых через SQL*Net между клиентом и сервером, сервером приложений и сервером или серверами БД, выполнялось с помощью опции Advanced Security option, доступной только для пользователей редакции Enterprise Edition СУБД Oracle, то теперь эта возможность входит в состав ядра СУБД и доступна другим редакциям СУБД (SE и SE One) без покупки дополнительных опций.

V. Управляемость

Мы уже упоминали выше о возможности консолидировать результаты захвата нагрузки в DB Replay для ее консолидированного проигрывания на консолидированной контейнерной БД. Еще одна интересная возможность Oracle Database 12c — возможность оптимизатора запросов на ходу перестраивать план запроса (Adaptive Plans).

Адаптивное изменение планов запросов

Динамическое изменение отдельных шагов плана (например, замена Nested Loops на Hash Join), если после первого выполнения оптимизатор распознал ошибочный план SQL-запроса.

Расширенный Real Time ADDM

Для реализации самоуправления в Oracle используется механизм ADDM (Automatic Data Diagnostic manager). Он раз в час анализирует статистику о работе СУБД, собранную в таблицах AWR (Automatic Workload Repository) на диске, и выявляет проблемы и негативные тенденции в работе СУБД. Выявив проблемы, он либо исправляет их, либо сообщает о них администратору БД. Появившийся отчет Compare Period ADDM позволяет сравнить статистику за указанный период времени, выявить возникшие проблемы и изменения, произошедшие за этот период, и указать на причины возникновения проблем.

Появившийся в Oracle Database 12c Real Time ADDM работает в режиме реального времени, т. е. оживает не каждый час, а каждые 3 секунды и, используя оперативную информацию о работе СУБД, хранящуюся в памяти компьютера, оперативно обнаруживает проблемы, пики нагрузки, провалы производительности и т. д. Он позволяет выполнять проактивный анализ. Кроме того, если СУБД «повисла» или очень медленно работает, Real time ADDM может напрямую подключиться к областям памяти и обнаружить причины проблем.

OEM Database Express

В Oracle Database 12c широко использовавшаяся администраторами БД программа DB Control заменена на встроенную в БД компоненту OEM Database Express 12c. Она имеет простой графический интерфейс, похожий на интерфейс приложений APPEX, устанавливается вместе с СУБД и доступна через веб-браузер. Интерфейс ее виден на рисунке 5.



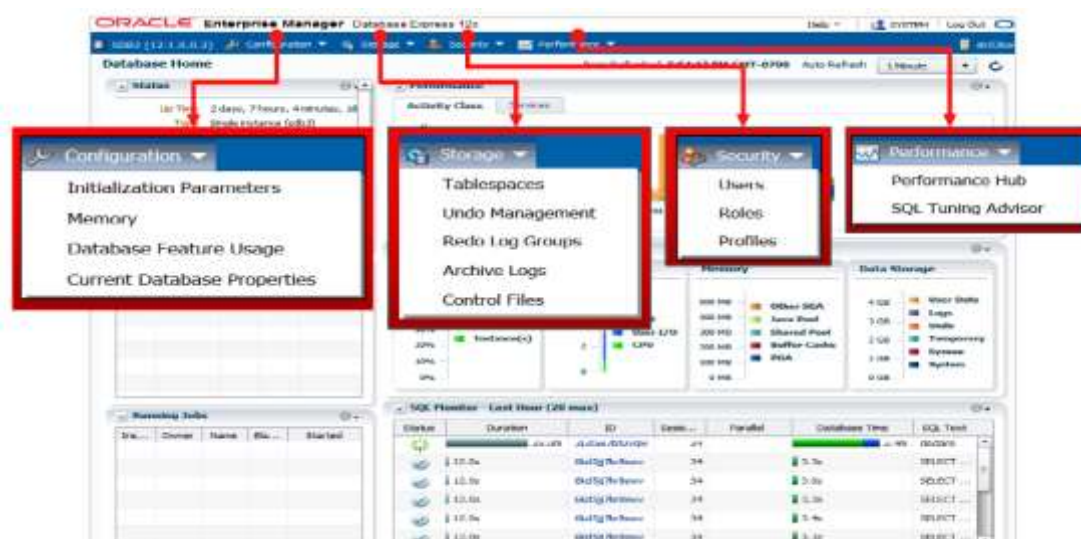


Рисунок 5. OEM Database Express 12c

Программа позволяет администратору БД управлять элементами БД (табличные пространства, undo, журналы, управляющие файлы), конфигурациями (параметры, используемые опции, память), пользователями и ролями, а также выполнять диагностику и настройку СУБД. Для выполнения более сложных операций надо использовать OEM Cloud Control 12c, Release 3, который поддерживает новые возможности Oracle Database 12c.

VI. Улучшения для разработчиков приложений БД

СУБД Oracle Database 12C предлагает множество новых возможностей для разработчиков. Вот лишь некоторые из них:

Встроенное определение PL/SQL-процедуры в SQL-запросе. Непосредственно в тексте SQL-запроса можно указывать текст PL/SQL-функции, которая вызывается внутри запроса. Цель этого нововведения — уменьшить время переключения между модулем выполнения SQL и виртуальной машиной PL/SQL в процессе выполнения запроса. В результате значительно повысилась скорость выполнения запросов, внутри которых вызываются PL/SQL-функции!

Расширенные возможности значений по умолчанию в столбцах таблиц. Включает в себя две новые возможности: автоинкрементные столбцы (**identity columns**) (значение которых вставляются автоматически из последовательности); установка значения по умолчанию при вставке NULL в поле, при этом в строке таблицы фактически НЕ происходит замена NULL на значение по умолчанию, а устанавливается ссылка на словарь.

В предыдущих версиях СУБД Oracle Database единственной возможностью создания автоинкрементных столбцов (столбцов, значения которых автоматически генерируются увеличением счетчика при вставке новой записи в таблицу) была самостоятельная разработка триггера на операцию вставки записи в таблицу (ON-INSERT TRIGGER), который получал очередное новое значение столбца из последовательности. В Oracle Database версии 12c появилась полноценная поддержка автоинкрементных столбцов, которая позволяет при создании таблицы указать, что конкретный столбец имеет тип identity.

Если столбец имеет автоинкрементный тип, то его значения будут генерироваться автоматически начиная с единицы и с шагом 1. Важно отметить, что в такой автоинкрементный столбец запрещена явная вставка значений. То есть вставка значений в автоинкрементный столбец возможна только по умолчанию, со значением из счетчика. Также есть возможность указать, что значение автоинкрементного поля начинается не с единицы, при этом можно задать произвольное приращение значения для следующего



значения столбца. Для автоинкрементного поля также можно указать, что его значение будет браться из предварительно созданной последовательности. В этом случае возможна вставка в автоинкрементное поле явно заданных значений, а не только получение их из последовательности.

Поддержка длины полей таблиц типов varchar2 до 32 КБ. Поля типа varchar2, nvarchar2, raw теперь могут иметь размер до 32 КБ. Добавлен новый параметр Max_SQL_String_Size в init.ora для включения этой возможности.

Поддержка фразы Top-N в запросах. В SQL-запросах теперь поддерживается фраза TOP, FETCH NEXT и OFFSET для получения необходимого количества строк в произвольном окне курсора (например, вывести только с 10-й по 100-ю строки результата запроса).

Транслятор SQL-запросов из синтаксиса других СУБД (SQL Translation Framework)

При миграции приложений из других СУБД на платформу Oracle Database часто бывает необходимо переписать тексты SQL-запросов. Большой объем работ, связанный с этой переработкой приложения, значительно затрудняет переход на СУБД Oracle Database. Для решения этой проблемы, в Oracle Database 12C появился механизм автоматической трансляции текста SQL-запросов из синтаксиса других СУБД в формат Oracle Database. В настоящий момент доступны модули для трансляции текста SQL-запросов из синтаксиса СУБД Sybase, SQL Server и DB2. В случае необходимости программист может разработать собственный модуль SQL-трансляции для диалекта языка SQL той СУБД, с которой производится миграция на Oracle Database.

Невидимые столбцы (invisible columns). В Oracle Database 12C появилась новая интересная возможность: невидимые столбцы (Invisible columns). Невидимость столбцов заключается в том, что их не видно в результатах выполнения оператора SELECT к таблице в случае отсутствия явного указания имен столбцов. Следующие операторы не возвращают скрытые столбцы в своем результате: SELECT*FROM; команда DESCRIBE в утилите SQL*Plus; тип записи PL/SQL основанный на структуре таблицы с помощью атрибута %ROWTYPE; функции определения структуры курсора в OCI (OCIDescribe). Для определения невидимости столбца, при создании таблицы указывается модификатор INVISIBLE. Уже существующий видимый столбец можно сделать невидимым с помощью оператора ALTER TABLE ... INVISIBLE. Точно так же невидимый столбец можно сделать видимым с помощью команды ALTER TABLE ... VISIBLE. Если необходимо вставить значение в невидимый столбец, то в операторе INSERT нужно явно указывать его имя.

Невидимый столбец можно вывести только явно указав его имя в запросе. Как и обычные столбцы, невидимые столбцы: могут быть ключом секционирования в таблице; могут участвовать в вычисляемых выражениях в виртуальном столбце; сами могут быть виртуальными столбцами. Невидимые столбцы полезны при плавных обновлениях (patching) приложения, когда в новой версии приложения нужно удалить некоторые столбцы. В сочетании с технологией обновления приложения Edition Base Redefinition, невидимые столбцы обеспечивают большую гибкость при переходе на новую версию прикладного приложения.

Упрощенная загрузка данных (SQL*Loader Express Mode) Утилита SQL*Loader предназначена для быстрой загрузки данных в СУБД Oracle Database из текстовых файлов и присутствует в составе СУБД начиная с самых ранних версий этой СУБД. В предыдущих версиях СУБД Oracle Database (до 12c) было необходимо вручную создавать так называемый управляющий файл, который описывал поля загружаемого текстового файла и их соответствие столбцам таблицы СУБД. В Oracle Database 12C утилита SQL*Loader поддерживает быстрый режим загрузки (Express Mode). При быстром режиме управляющий файл не используется, а достаточно в параметрах утилиты указать имя файла с данными и имя таблицы в СУБД, куда эти данные будут загружаться. При этом должны соблюдаться следующие условия: столбцы таблицы должны иметь только скалярные типы (строка, число, дата); текстовый файл должен состоять из текстовых полей фиксированной длины. Для описания структуры файла и соответствия его полей столбцам таблицы утилита SQL*loader использует структуру таблицы из СУБД.

Поддержка сопоставления записей по образцу (Row Pattern Matching). Часто в приложениях бизнес-аналитики необходимо формировать отчет на основе последовательности записей, удовлетворяющих



некоторому правилу изменения значений. Например, если мы имеем таблицу с упорядоченными по времени ценами на акции, аналитикам часто приходится искать в этой последовательности значений некоторую модель поведения. Например, модель «W», когда цены падали, потом росли, потом падали и потом опять росли. Здесь важно отметить, что выбираются не просто записи, удовлетворяющие определенному фильтру, а выбирается группа записей, удовлетворяющих определенному правилу распределения данных. Данная задача называется сопоставлением записей по образцу — Row Pattern Matching. До версии 12C для построения таких отчетов приходилось на уровне приложения кодировать сложную логику обработки данных. В Oracle Database 12C появилась встроенная поддержка поиска записей по образцу: в языке SQL появился оператор `MATCH_RECOGNIZE`, который позволяет задать модель изменения данных для выбираемых записей.

Новые возможности СУБД Oracle Database 12.1 — 20c

Компания Oracle непрерывно совершенствует свой флагманский продукт СУБД Oracle Database. Постоянно выходят новые версии, содержащие огромное количество новых функций и алгоритмов. Однако недавно порядок выхода новых версий изменился. Если раньше новая версия выходила каждые 3–4 года, а в течение этого отрезка времени выходили два релиза версии, то теперь, начиная с версии 18c, новая версия СУБД будет выходить каждый год и ее номер будет совпадать с названием года. Таким образом, теперь у нас существуют версии 18c, 19c, в 2020 году выходит версия 20c. Более частый выход версий подразумевает, что количество новых возможностей в каждой версии будет меньше, но заказчикам надо освоить технологию более частого перехода на новую версию.

Все новые версии теперь делятся на промежуточные и долговременные. Промежуточные позволяют попробовать новые возможности и хороши для разработчиков и тех, для кого новые функции важны. Долговременные версии более стабильны, имеют долгий срок технической поддержки (4 года + 2 года расширенной поддержки) и хороши для промышленной эксплуатации. Первый долговременный релиз — Oracle 19c, на него и рекомендуется переводить свои промышленные системы. Если раньше ДБА ждали финального релиза версии (например, 11.2, 10.2 и т. д.), то теперь таким устойчивым релизом является СУБД Oracle 19c. Поэтому нет смысла отдельно говорить о новых возможностях в СУБД Oracle 12.2, 18c, 19c. Мы рассмотрим новые возможности в Oracle 19c по сравнению с Oracle 12.1 и коротко упомянем о важных новшествах Oracle 20c. С новыми возможностями версии 12.1 можно ознакомиться здесь [1].

Кроме того, начиная с версии 19c, произошло 2 важных лицензионных изменения: опции Spatial&Graphs (работа с геоинформацией, семантическими сетями и графами) и Advanced Analytics (Data Mining, Machine Learning, язык R) теперь бесплатно входят в состав любой редакции СУБД (Standard Edition и Enterprise Edition). С появлением опции Multitenant (см. ниже) изменилась архитектура СУБД и БД Oracle, то есть с версии 20c Oracle прекращает поддержку старой архитектуры, но позволяет создавать до 3 PDB (pluggable database) в новой архитектуре бесплатно. Если надо создавать более 3 PDB в одной контейнерной БД, то требуется приобрести опцию Multitenant.

Конечно, с 2013 года, когда вышла версия 12.1, в СУБД было добавлено огромное количество новых возможностей, повысилась надежность, производительность, безопасность, масштабируемость и управляемость СУБД, появились автономные БД и механизмы самоуправления. Но наиболее важными являются следующие:

- Multitenant (мультиарендная) архитектура СУБД (опция Multitenant)
- Механизмы векторной обработки данных в оперативной памяти (опция In-Memory)
- Шардинг (параллельное хранение и обработка частей таблиц / групп таблиц на различных компьютерах)
- Автономные базы данных [2]
- Работа с энергонезависимой памятью (persistent memory)



Рассмотрим эти новые возможности подробнее.

I. Мультиарендная архитектура СУБД

Новая архитектура СУБД Oracle позволяет снизить сложность сопровождения множества БД Oracle и повысить эффективность использования оборудования. Она была введена в версии 12.1 (см. выше) и стала очень популярной среди пользователей. Архитектура и опция Multitenant, необходимая для ее реализации, постоянно совершенствуются, снимаются ограничения. Теперь практически все функции СУБД старой архитектуры реализованы в PDB и пользователь БД не различает, с какой архитектурой он работает.

Для повышения надежности работы группы PDB конкретной CDB можно сконфигурировать кластер (RAC) для этой CDB, и закрепить отдельные PDB за конкретными узлами RAC, чтобы обеспечить изолированность и масштабируемость. В CDB очень просто делать клоны PDB, существующих в этой или другой CDB, причем клоны могут обновляться по мере обновления исходных PDB.

Ну и конечно, упрощается обновление БД и установка исправлений. Вместо 10 апгрейдов для 10 БД мы можем обновить CDB, и все ее PDB автоматически обновятся до новой версии. Если же обновление и исправление нужны не для всех PDB, а лишь для части, то их просто надо перенести в CDB новой версии. Перенос PDB из одной CDB в другую очень прост. Одной командой (или мышкой в Enterprise Manager) выгружается в xml-файл метаинформация о PDB, и затем эта метаинформация загружается в другую CDB. Если эти CDB размещаются на одном компьютере, то даже копировать файлы БД PDB не требуется. Можно не только переместить PDB в другую CDB, но и делать в новой CDB клон PDB из другой CDB. При этом можно делать так называемый hot refreshable клон, т. е. во время клонирования исходная PDB открыта для изменений, а после окончания клонирования эти изменения применяются к клону, т. е. клон и исходная PDB синхронизируются. И далее эта синхронизация периодически повторяется в автоматическом или ручном режиме. Т. е. мы всегда имеем свежую копию мастер-клона в новой CDB, открытую на чтение, и можем делать открытые для изменения новые клоны PDB в этой CDB. Это очень удобно для разработчиков приложений.

Итак, у нас есть исходная PDB и ее синхронизирующийся клон в другой CDB. Это некоторый вариант резервной БД, похожий на standby. И так же, как при standby БД, мы можем переключить PDB-клон в режим основной открытой для изменений PDB, а ее исходная PDB превратится в доступный на чтение обновляемый мастер-клон. Т. е. происходит переключение (**PDB switchover**) — смена ролей.

Разработчикам и тестировщикам приложений часто бывает нужно восстанавливать свою БД на момент времени в прошлом, это можно делать с помощью механизма flashback (можно делать flashback отдельной PDB), но можно применить новый механизм — **карусель снэпшотов** (snapshot carousel). Если для PDB включен режим карусели, Oracle автоматически ежедневно будет делать копию этой PDB и хранить ее в архивном файле (такая копия называется снэпшотом). По умолчанию хранятся последние 8 снэпшотов. Если, например, в среду необходимо восстановить PDB на состояние 5 часов вечера понедельника, то мы просто восстанавливаем PDB из снэпшота за понедельник и далее накатываем архивные журналы, чтобы применить изменения, сделанные до 5 вечера.

Еще один интересный механизм мультиарендной БД — **Application Container (AC)**. Если несколько PDB имеют одинаковые объекты (таблицы, процедуры, функции и т. д.), то их можно поместить в отдельную PDB, называемую application container. Все PDB, наследующие объекты этого контейнера, будут видеть эти объекты. Таким образом, мы убираем дублирование и облегчаем сопровождение этих объектов (они изменяются в одном месте — application container). Причем, если таким разделяемым объектом является таблица, то у Вас есть 3 варианта:

- Хранить таблицу со всеми данными в AC (тогда PDB будут видеть ее как таблицу, открытую на чтение)
- Хранить таблицу и часть ее данных в AC (тогда каждая PDB будет видеть эти данные в режиме чтения, но может иметь свою открытую на изменение часть этой таблицы)



- Хранить только описание структуры таблицы в AC (тогда каждая PDB будет иметь свой, скрытый от других, открытый для изменений вариант этой таблицы).

PDB-базы изолированы и не видят друг друга, администратор БД может использовать менеджер ресурсов, чтобы управлять разделением ресурсов компьютера (память, процессоры, ввод-вывод, параллелизм) между БД. С помощью механизма lockdown profiles можно ограничить для PDB выполнение отдельных команд SQL и их частей, то есть запретить выполнение некоторых потенциально опасных команд (например, alter system), запретить выполнение команд операционной системы (ОС) и даже запретить прямой доступ к этой PDB.

Однако серверные процессы ОС имеют доступ к файлам БД и могут читать или модифицировать файлы чужих PDB. Чтобы избежать этого, в версии 20c вводится механизм **DB Nest**. Все процессы ОС экземпляра Oracle можно разделить на две группы: фоновые процессы и серверные процессы (они обслуживают сессии и SQL отдельных пользователей). DB Nest позволяет запретить серверным процессам конкретной PDB доступ к файлам БД, файлам трассировки, файлам настройки других PDB, им также можно запретить доступ к областям памяти (pga) других PDB и выполнение команд ОС. Механизм очень похож на механизм контейнеризации в ОС, каждая PDB со своими серверными процессами и файлами как бы работает в отдельном контейнере и изолирована от других PDB.

Поскольку теперь с PDB можно работать как с обычной традиционной БД, у нее есть и традиционные средства настройки БД. Она имеет свой AWR, можно запускать ADDM и строить AWR-отчеты уровня PDB, получать рекомендации по настройке БД. Можно запускать захват нагрузки RAT на отдельной PDB, чтобы захватить, а потом и воспроизвести нагрузку и оценить влияние изменений на эту PDB.

Мультиарендная архитектура доказала свои преимущества, на ней построены все автономные БД Oracle, и с версии 20c Oracle будет поддерживать только эту новую архитектуру. Те, кто хочет по-прежнему иметь один экземпляр Oracle для каждой БД, могут создавать CDB с одной PDB. Бесплатно можно создавать до 3 PDB в одной CDB, если надо больше — следует лицензировать опцию Multitenant.

II. Опция In-Memory, memory optimized таблицы

Опция Database In-Memory

Эта опция позволяет резко ускорить выполнение аналитических запросов. Ее использование несложно и требует лишь задать размер In-Memory кэша и указать таблицы, которые можно помещать в этот кэш. Сегодня в СУБД различных производителей используются два основных метода хранения данных — традиционный построчный и поколоночный. Известно, что аналитические запросы, подсчет агрегатов быстрее выполняются при поколоночном хранении, но обычные OLTP запросы, требующие доступа к небольшому набору строк, либо модифицирующие одну строку таблицы, на таком представлении работают медленно.

Наоборот, при построчном хранении OLTP работает быстро, но аналитика выполняется медленнее, чем при поколоночном хранении. А в обычной жизни большинство приложений — смешанные, т. е. есть и OLTP, и аналитические запросы, и сложные вычисления и т. д. Технология Oracle Database In-Memory позволяет решить эту проблему. Данные в БД на диске остаются в построчном формате. Это позволяет сохранить все механизмы хранения и ввода-вывода. Но в оперативной памяти для заданных таблиц (или частей таблиц) будет храниться два представления этих данных — построчное в буферном кэше и поколоночное в новом In-Memory кэше. (рис. 1).



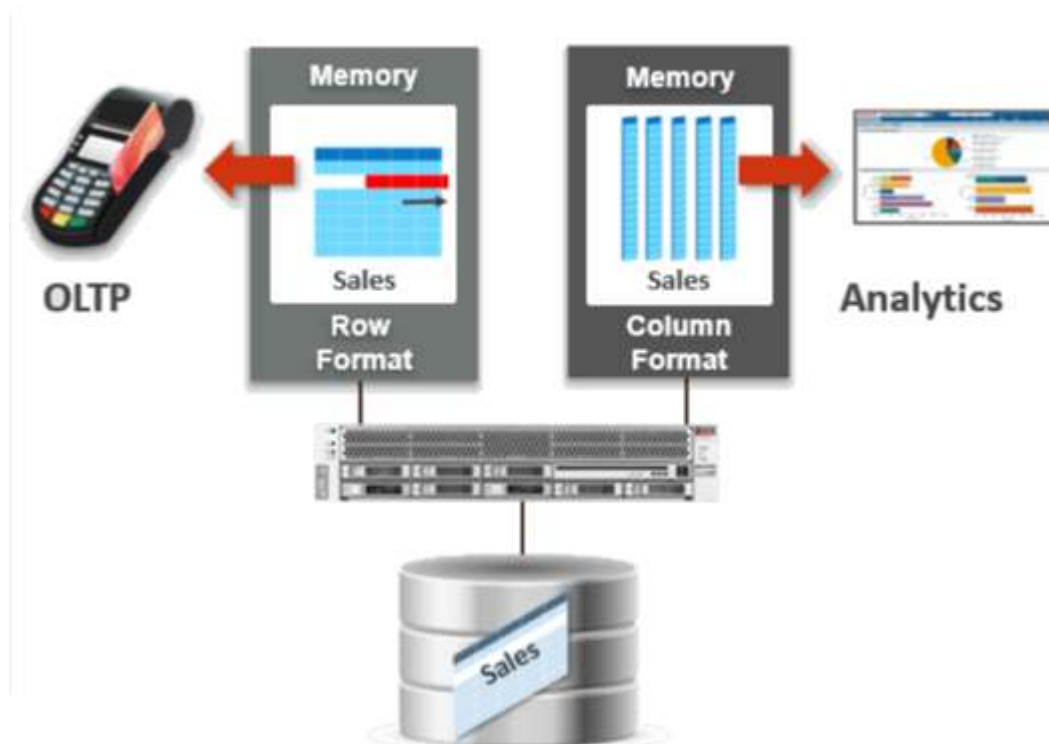


Рисунок 1. Архитектура Oracle Database In-Memory

Когда в СУБД поступает новый запрос, SQL-оптимизатор Oracle определяет, как он будет лучше выполняться, на каком кэше (на построчном или колоночном), и строит соответствующий план выполнения запроса. Работа аналитических запросов при этом может ускоряться в десятки раз. Это происходит не только за счет поколоночного представления данных и кэширования данных в памяти, но и за счет того, что операции с колонками (векторами значений) Oracle выполняет с использованием векторных операций процессора (большинство современных процессоров поддерживает векторные операции (Single Instruction Multiple Data — SIMD). Они позволяют одному ядру процессора сканировать миллиарды строк в секунду. Т. е. за одну команду процессора можно, например, провести поиск значений в колонке.

Более того, даже тяжелые и сложные операции соединения таблиц Oracle умеет преобразовать в набор векторных операций. В версии 20c появилась возможность и некоторые простые операции соединения производить с использованием векторных операций без преобразования. Это называется **In-Memory Vector Join**. Например, если нужно соединить таблицу продаж и таблицу пунктов продаж по ключу кода заказа, значения колонок кода заказа этих таблиц загружаются в два векторных регистра и за один такт процессора Oracle получает матрицу соответствия этих столбцов (практически результат соединения таблиц). Это на порядок ускоряет многие операции соединения. При векторной обработке также реализуются новые алгоритмы ускорения построения агрегированных отчетов.

Поскольку размер оперативной памяти и, соответственно, In-Memory кэша ограничен, специальная утилита In-Memory Advisor проанализирует Ваши запросы и порекомендует, какие таблицы следует поместить в In-Memory кэш, а также поможет это сделать. Чтобы сэкономить память, кэшировать можно части таблиц (секции), не все колонки, можно сжимать кэшируемые данные. В последних версиях СУБД (с 19c) данные автоматически подкачиваются в кэш на место давно не используемых. А в версии 20c, даже если часть таблиц запроса хранится в In-Memory кэше, а часть только в буферном кэше, Oracle будет использовать алгоритмы In-Memory и преимущества обоих кэшей.

Обновление данных всегда сначала выполняется в буферном кэше, и затем специальный фоновый процесс синхронизирует кэши. Опция In-Memory может использоваться и на резервной БД, открытой на чтение (опция



Active Data Guard), где выполняются аналитические запросы и строятся отчеты. При этом в основной и резервной базах в кэше могут быть разные таблицы.

В ячейках Exadata используется флэш-память и энергонезависимая память (persistent memory). На них данные могут кэшироваться в том же формате, что и в In-Memory кэше. Поэтому на ячейках хранения Exadata тоже применимы алгоритмы векторной обработки.

В Oracle 18с значительно повышена производительность многих операций In-Memory. Например, сканирование нескольких колонок таблиц может выполняться одновременно, значения числовых (number) колонок преобразуются в памяти в бинарный формат процессора, ускоряющий выполнение векторных вычислений, часто используемые выражения на основе колонок таблицы могут автоматически вычисляться заранее и храниться в памяти в виде дополнительных колонок, т. е. вычисления не производятся каждый раз.

В последних версиях СУБД Oracle Database технология In-Memory используется не только для реляционных данных, но и для колонок с геоинформацией, текстом и JSON. Она используется и при работе с внешними таблицами (external table).

Побочным эффектом опции In-Memory является ускорение в 2–3 раза OLTP запросов. Обычно для ускорения аналитических запросов, мы строим множество дополнительных индексов для таблиц. Каждое изменение данных таблицы вызывает также и изменение всех этих индексов, что замедляет работу. Теперь же мы можем удалить все эти дополнительные индексы и обеспечить ускорение OLTP-операций за счет технологии Oracle Database In-Memory, поскольку отсутствуют накладные расходы на сопровождение дополнительных индексов.

Memory optimized таблицы

Опция In-Memory позволяет ускорить работу с аналитическими запросами. Но есть еще два типа приложений со специфическими запросами, которые теперь Oracle умеет ускорять. Это работа с таблицами типа ключ-значение, где обычно по ключу надо быстро получить запись, и интернет вещей (IoT), где идет непрерывная вставка записей в таблицу (например, поток телеметрии с датчиков). Для этого используются так называемые memory optimized таблицы, для которых Oracle применяет специальные алгоритмы обработки.

Если надо быстро извлекать данные типа ключ-значение из таблицы по ключу, то достаточно просто объявить таблицу **Memoptimize for read** и задать размер кэша Memoptimized для таких таблиц. При этом таблица будет загружена в кэш при первом обращении к ней и для нее в памяти будет построен хеш-индекс. При запросе записи по ключу, Oracle в обход традиционных механизмов, без блокировок, посмотрит этот индекс и сразу выдернет из памяти нужные строки. Новый алгоритм использует новый клиентский протокол с низким временем отклика и обеспечивает прямой доступ к ядру СУБД в обход SQL-уровня. Механизм позволяет получить ускорение до 4 раз.

Если же Вам нужно создать таблицу, например, для интернета вещей, куда нужно быстро и непрерывно вставлять новые записи, то достаточно объявить эту таблицу **Memoptimize for write**. При добавлении записей в такую таблицу строки вначале быстро вставляются в специальный буфер в памяти, а затем буферизованные данные пакетами пишутся на диск в фоновом режиме. При этом производительность добавления записей может повыситься в два раза.

III. Шардинг

Обычно таблица хранится в одной БД на одном компьютере и ее обработкой занимается этот компьютер или группа узлов RAC. Но если таблица велика, нагрузка очень большая, а обработку отдельных частей таблицы можно распараллелить, то разумно разрезать таблицу на части (шарды) и разместить их не в одной БД, а на разных компьютерах. На каждом таком компьютере будет своя БД, в которой будет лежать часть



таблицы (или группа связанных частей связанных таблиц). Каждый компьютер будет выполнять обработку своих шардов.

Это позволит осуществить горизонтальное масштабирование для высоконагруженных, хорошо распараллеливаемых задач (например, поисковые системы). Такой механизм назвали шардингом. Таблицы разделяются на части по ключу шардинга, этот же ключ шардинга может явно указываться в SQL-запросах или при открытии сессии, и существует координатор (шард-директор), который определяет по ключу шарда, какому компьютеру передать обработку. Если же в запросе ключ шарда не указан, координатор выполняет запрос на всех шардах (кросс-шардинг), но это будет работать медленно. Все DDL-операции с множеством узлов выполняются централизованно через координатор.

Большим достоинством шардинга Oracle является то, что новые узлы в такую систему можно добавлять легко, на лету, увеличивая ее мощность, и переносить на них часть шардов (делать рещардинг). Часть баз шардированного приложения может быть реализована в виде PDB или размещена в облаке. Таким образом, шардинг не только ускоряет обработку, но и повышает отказоустойчивость системы. Выход из строя узла делает недоступным лишь часть данных. БД каждого узла может иметь свою резервную БД для надежности.

В БД узла можно помещать не только часть одной таблицы, но и связанные по ключам части нескольких таблиц (семейство таблиц — table family tree) или несколько семейств. Кроме того, в эти БД для ускорения вычислений помещаются копии справочных таблиц, чтобы вся обработка выполнялась на узле (рис. 2).

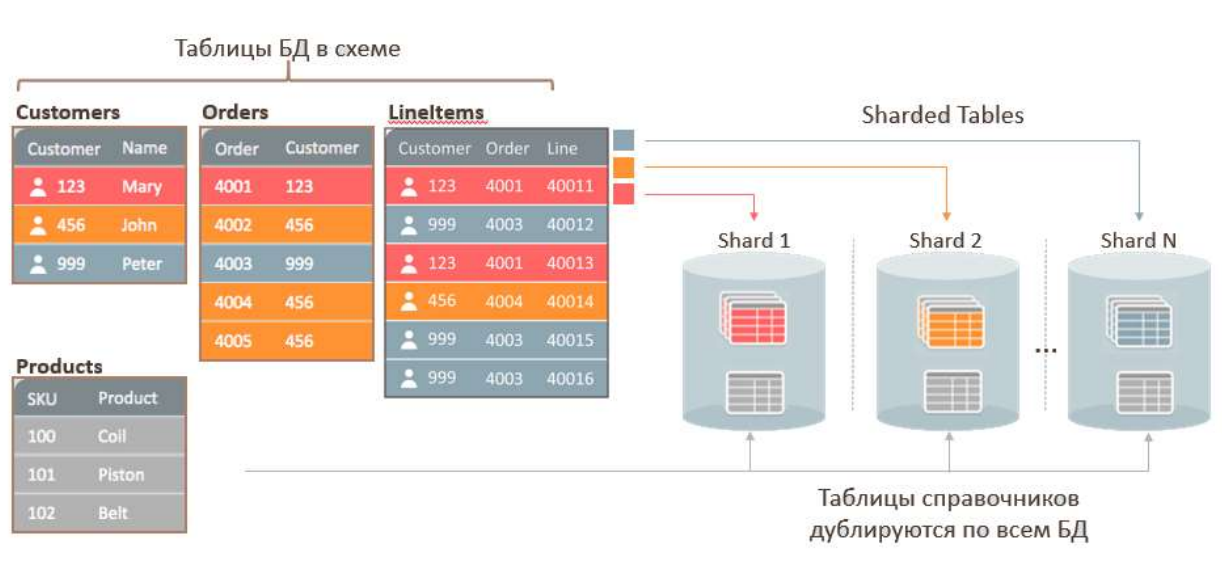


Рисунок 2. Шардинг

Начиная с версии 19с в одной CDB могут быть PDB для шардов одной или разных шардированных БД. Если при работе с шардами в узлах используются последовательности Oracle (sequence) для генерации уникальных значений, то можно обеспечить, что генерируемые значения будут уникальны как в шарде, так и для всей группы шардов. Утилита Sharding Advisor поможет Вам выбрать способ разбиения Вашей БД на шарды.

Sharded RAC. Сегодня у нас существует два способа горизонтального масштабирования БД: RAC и шардинг (и отчасти standby с Active Data Guard). RAC работает с одной БД (или CDB) и распараллеливает обработку (процессоры и память), шардинг делит одну БД на несколько БД со своими процессорами и областями памяти. Oracle с версии 18с предлагает еще один вариант: шардированный RAC (виртуальный шардинг). При этом БД не разделяется на части, а остается единым целым, но данные делятся на виртуальные шарды, привязанные к узлам кластера RAC. Запросы с ключом шарда перенаправляются к узлу/узлам кластера, обслуживающим этот шард. Это позволяет лучше использовать кэш БД и уменьшить пересылку блоков данных по интерконнекту. Запросы без ключа шарда обрабатываются как раньше (рис. 3).



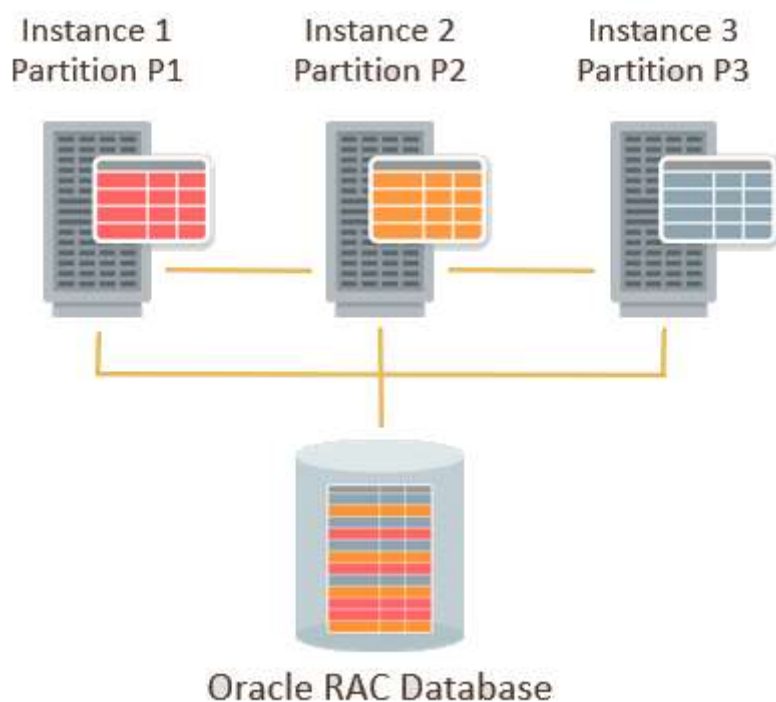


Рис. 3. Шардированный RAC

IV. Работа с энергонезависимой памятью

В традиционных СУБД данные БД хранятся на дисках и для обработки считываются в оперативную память компьютера. После выключения компьютера данные в оперативной памяти теряются. Оперативная память очень быстрая, но дорогая, поэтому поместить в нее всю БД могут немногие заказчики. В качестве паллиатива в ячейках Exadata между дисками и оперативной памятью ставится flash-память, которая автоматически кэширует часто используемые данные. К сожалению, она хоть и быстрее дисков, но намного медленнее оперативной памяти.

Появление энергонезависимой памяти (Persistent Memory — PMEM) революционно меняет правила игры и, очевидно, приведет к изменению архитектуры СУБД. PMEM по стоимости дешевле оперативной памяти, но по скорости близка к ней. И она не теряет данные после выключения питания. Т. е. в будущем можно будет всю БД поместить в PMEM и отказаться от дорогостоящих операций ввода-вывода, кэшей в памяти, подкачки блоков и т. д.

Oracle с версии 20c научился работать с PMEM. Он использует ее двумя способами. Во-первых, PMEM марки Intel Optain помещается в ячейки хранения машины БД Exadata (рис. 4).



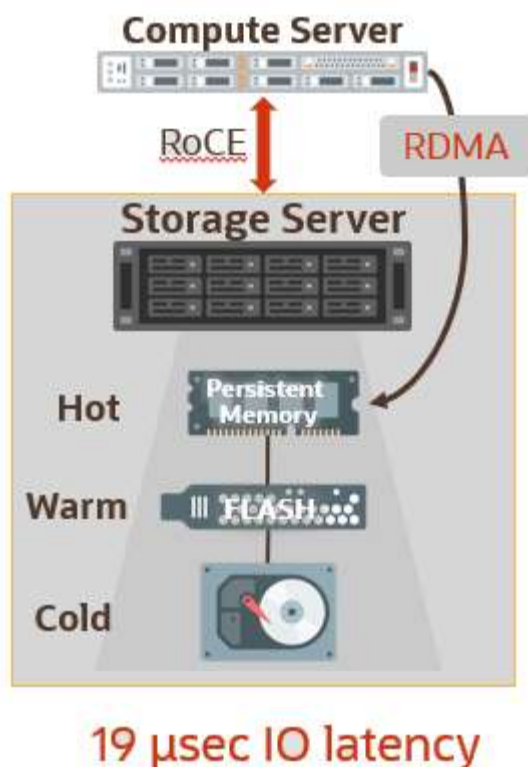


Рис. 4. Уровни хранения данных в Exadata

Теперь в Exadata реализовано четыре уровня хранения данных:

- Диски (холодные данные)
- Flash (кэширование теплых данных)
- PMEM (кэширование горячих данных)
- Оперативная память

При этом обращение экземпляра СУБД к данным PMEM происходит не по стандартному протоколу ROCE (замена Infiniband), а по специальному протоколу RDMA напрямую, что исключает сетевой стек и стек ввода-вывода и снижает задержки в десятки раз. Кроме того, Exadata пишет журналы транзакций (redo log) на PMEM. Все это значительно ускоряет операции ввода-вывода.

Но и пользователи СУБД Oracle 20c, работающие на обычном (не Exadata) оборудовании, могут использовать преимущества PMEM. На PMEM можно размещать журналы транзакций (redo logs) и файлы данных. Это обеспечивает большую скорость работы, чем флэш- и SSD-диски. Причем данные из PMEM не считываются в буфера оперативной памяти. SQL-запросы выполняются напрямую над данными файловой системы PMEM по специальным алгоритмам. Накладные расходы, связанные с сопровождением буферного кэша, здесь исключены. Это позволяет значительно ускорить выполнение важных запросов и повысить производительность всей СУБД.

V. Хранилища данных и большие БД

В Oracle 19c можно создавать **гибридные секционированные таблицы**. Это значит, что часть секций таблицы — это обычные секции в БД, а часть секций — внешние (external), лежащие в файлах ОС или на системе облачного хранения. SQL-операторы видят этот набор секций как единую внутреннюю таблицу БД. Поддерживаются все виды внешних секций, которые можно загружать с помощью: SQL Loader,



Data Pump, файлы Hadoop (HDFS и Hive формат). Внешние секции открыты только на чтение. Гибридное секционирование позволяет, например, легко и быстро перемещать архивные данные из БД в Hadoop и продолжать иметь к ним доступ, не переписывая приложения и снижая стоимость хранения.

Многие приложения используют временные таблицы, но до версии 18с мы могли создавать только глобальные временные таблицы. Их видели все сессии, и они существовали в БД постоянно, но каждая сессия помещала туда свои данные, работала с ними, и затем данные автоматически удалялись при завершении транзакции (commit) либо при завершении сессии. Теперь Вы можете упростить работу, создавая **частные временные таблицы** сессии (private temporary table). Они размещаются в оперативной памяти, другие сессии их не видят и не имеют доступа к их данным. Такая частная временная таблица будет автоматически удаляться либо по окончании сессии, либо по окончании транзакции. Можно создавать частные временные таблицы и на standby БД.

Oracle стремится к минимизации времени простоя и задержки при любых операциях с объектами БД. Теперь практически все **операции с секциями таблиц и их индексами** (создание, перемещение, удаление, изменение, расщепление, преобразование типа секционирования и т.д.) можно выполнять онлайн, без остановки работы с таблицами.

VI. Повышение надежности СУБД

Выше уже упоминались некоторые нововведения в области обеспечения архитектуры максимальной надежности и доступности (Maximum Availability Architecture — MAA), такие как Sharded RAC, Multitenant, Sharding, Application Continuity [1], автономные БД [2]. Сегодня автономная БД может обеспечить уровень надежности 99,995. Это 30 мин. плановых и неплановых простоев в год!!! Много изменений было сделано в механизме работы резервной (standby) БД.

DML в Standby. Первоначально резервная БД служила только для переключения на нее в случае сбоя основной БД. Она только «догоняла» основную БД. Затем появилась опция Active data Guard (ADG), которая позволила открыть резервную БД на чтение. Это дало возможность разгрузить основную БД и вынести на резервную БД операции построения отчетов, аналитику, бэкапы. Однако менять данные в standby БД было нельзя. Однако очень часто построение большого сложного отчета требует произвести небольшое изменение данных в таблицах. Например, записать диапазон дат, за которые строится отчет, или сохранить информацию о времени последнего построения отчета. Теперь, начиная с версии 19с, это можно сделать.

Standby БД с ADG открыта для небольших изменений. Осуществляется это следующим образом. Когда пользователь, работающий с резервной БД, выполняет на ней оператор DML (Update, Delete, Insert), этот оператор автоматически перенаправляется на основную БД, выполняется там и только потом изменения передаются на резервную БД. При этом сессия на резервной БД ждет, пока изменения будут доставлены. При этом транзакционность поддерживается, т. е. изменения до фиксации транзакции другим транзакциям не видны. Этот механизм полезен для приложений, которые часто читают данные, но редко изменяют данные.

Синхронный Flashback. Еще один полезный новый механизм — синхронный откат (flashback) основной и резервной БД. Это можно сделать одной командой. При откате назад основной БД, резервная БД также откатывается автоматически.

Multi Instance Recovery Apply (MIRA). При большом объеме изменений на основной БД поток изменений, передаваемых на резервную БД тоже очень велик, и ранее процесс наката изменений на резервную БД мог не справляться с такой нагрузкой и увеличивать отставание резервной БД от основной БД. Теперь, если на основной и резервной системах работает RAC, изменения от узла/узлов основной БД могут параллельно применять различные узлы RAC резервной БД. При этом сохраняется порядок выполнения транзакций,



но за счет распараллеливания нагрузки резко возрастает скорость синхронизации основной и резервной БД. Механизм называется MIRA — Multi Instance Recovery Apply.

Нежурналируемые операции (Nologging) и Standby. Большинство операций основной БД записываются в журнал (redo log), передаются на резервную БД и там применяются. Но журналирование замедляет работу, создает накладные расходы, увеличивает объем передаваемой на резервную БД информации. Поэтому для некоторых некритичных операций ДБА отключают журналирование (режим nologging). При этом происходит рассинхронизация объектов основной и резервной БД. Новая команда Rman — «validate/recover .. nologging block» позволяет выявить и синхронизировать такие рассинхронизированные объекты данных. А команда «**recover standby database from service**» позволяет быстро восстановить резервную БД не из бэкапа, а из основной БД. Это очень удобно при большом отставании резервной БД от основной. Можно также одной командой сравнить конфигурации основной и резервной БД и выявить отличия.

Начиная с версии 18с поддерживается передача изменений при nologging-операциях с основной БД на резервную прямо по сети — минуя фазу записи изменений в журнал транзакций. Это позволяет снизить нагрузку на дисковую подсистему и увеличить скорость передачи изменений с основной БД на резервную.

Сохранение сессий. Ранее при переводе резервной БД в режим основной все открытые сессии этой БД терялись и их надо было открывать заново. Теперь резервная БД при переводе в режим основной сессии и их контекст не теряет, просто они из режима чтения переходят в режим чтения/записи. Это уменьшает время переключения и упрощает процесс.

VII. Механизмы повышения производительности и автономности

В каждой новой версии СУБД Oracle происходит оптимизация алгоритмов, ускоряются многие операции, повышается производительность выполнения SQL и PL/SQL кода. Здесь рассмотрим самые интересные механизмы.

Статистика реального времени. Скорость выполнения SQL-запроса зависит от качества выбранного плана выполнения запроса. Cost based оптимизатор запросов строит планы, в том числе на основе собранной статистики (количество записей, наличие индексов, кардинальность и т. д.). При устаревшей, неверной статистике план будет плохим. Поэтому ДБА учат постоянно освежать статистику. Сейчас, например, это можно делать автоматически, в окне обслуживания (например, ночью). Но операция сбора статистики тоже влияет на производительность БД, а между двумя сборами статистики мы опять имеем дело с неактуальной статистикой.

Чтобы решить эту проблему, Oracle реализовал механизм Real Time Statistic — сбор статистики в реальном времени. Теперь часть наиболее важной статистики (базовая статистика) хранится в оперативной памяти, автоматически обновляется при выполнении каждого DML оператора и периодически сбрасывается на диск. Каждый новый запрос будет использовать свежую статистику (она формируется на основе статистики из памяти и из словаря БД) и выполняться с оптимальным планом запроса.

Поскольку статистика собирается на лету и не должна тормозить DML, то собирается только базовая статистика. Это значит, что полную статистику тоже надо собирать, но это можно делать реже и эта операция теперь меньше влияет на качество выполнения запросов.

Побочным эффектом статистики реального времени является то, что многие агрегационные функции (min, max, count), если в запросе нет предложения where, не читают данные с диска, а берут их из базовой статистики в памяти, что резко ускоряет выполнение запроса.

Автоматическое создание индексов. Самый интересный новый механизм — автоматическое создание индексов. Индекс в БД служит для ускорения выполнения SQL-запросов. Считается, что основное количество индексов строят разработчики приложения на этапе создания приложения и ДБА должен поддерживать эти



индексы. Однако на практике мы видим, что ДБА и разработчики, ответственные за обеспечение высокой производительности приложения, постоянно на этапе эксплуатации настраивают БД и создают все новые и новые индексы. Понятно, что некоторые индексы дублируют друг друга, индексы занимают место в БД и, главное, они замедляют операции вставки, удаления и изменения данных, которые требуют изменения индексов.

Создание оптимального набора индексов для БД — это искусство, которым владеет не каждый ДБА. Поэтому Oracle разработал механизм автоматического создания индексов. Механизм состоит из 6 шагов:

1. Захват (Capture). Во время работы приложения Oracle периодически (каждые 15 минут) захватывает выполняемые SQL-запросы и помещает их в специальный репозиторий (ASTS — Automatic SQL Tuning Set). Захватываются не только тексты SQL, но и планы выполнения, переменные привязки, статистика выполнения и т. д.

2. Идентификация кандидатов (Identify). На этом этапе Oracle анализирует вновь захваченные запросы и с использованием алгоритмов искусственного интеллекта (AI) определяет индексы-кандидаты, которые могут улучшить выполнение этих запросов. Эти индексы-кандидаты создаются в БД как неиспользуемые индексы (Unusable). Т. е. создаются не обычные индексы, а только их описания в словаре БД (метаданные). Кроме того, выявляются индексы — кандидаты на удаление, так как новые индексы-кандидаты могут перекрывать существующие автоматические индексы.

3. Верификация (Verify). На этом этапе оптимизатор запросов проверяет, что новые индексы влияют на планы выполнения запросов. Те кандидаты, которые влияют — уже реально создаются в БД, но помечаются как невидимые (они пока не видны сессиям БД). Захваченные запросы (Select) выполняются в отдельной сессии с учетом существования новых индексов и собирается статистика их выполнения.

4. Принятие решения (Decide). Если производительность всех запросов, использующих новый индекс, улучшилась, то индекс помечается как видимый для всех сессий и далее используется как обычно при построении планов запросов. Если производительность всех запросов, использующих новый индекс, ухудшилась, то индекс остается невидимым и потом удаляется. Если же производительность части запросов улучшилась, а части ухудшилась, то индекс помечается как видимый только для тех запросов, которые он улучшает. Остальные запросы его не видят и не используют.

5. Онлайн проверка (Online validation). Поскольку работа приложений продолжается, меняется статистика, нагрузка, появляются новые запросы, Oracle периодически повторяет работу по выявлению и построению новых автоматических индексов.

6. Мониторинг. СУБД постоянно мониторит использование автоматических индексов, и те из них, которые долго не используются, удаляются (политику хранения задает ДБА). Кроме того, в БД могут сосуществовать автоматические и ручные индексы, политика удаления для них может быть разной.

ДБА может включить/выключить автоматическое создание индексов, указать схемы БД, для которых этот режим не работает, указать, в каком табличном пространстве будут создаваться эти индексы, попросить СУБД не создавать автоматические индексы, а просто построить отчет о полезных рекомендуемых индексах.

Конечно, реализация вышеописанного механизма требует дополнительных вычислительных ресурсов. Поэтому если Вы удалили все индексы из БД и включили режим автоматического построения индексов, то вначале нагрузка на процессоры резко возрастет. Это допустимо на этапе первичной оптимизации, но лучше делать это на машине с большим количеством процессоров (например, на Exadata). Далее, когда основной массив индексов построен, накладные расходы на мониторинг, построение новых индексов и т. д. будут уже не так велики.

Применение механизма автоматического построения индексов на реальных приложениях показало, что он позволяет использовать меньше индексов, чем при ручной настройке, при сохранении производительности



приложения. Можно также запускать автоматическое построение индексов на копии промышленной БД, строить отчет, а затем уже вручную создавать рекомендованные индексы на промышленной БД.

Карантин для ресурсоемких запросов. Известно, что большие сложные запросы сильно нагружают систему и мешают работать другим пользователям. Причем запрос, который раньше работал быстро, при изменении плана выполнения или объема данных может стать помехой для других запросов. Поэтому в менеджере ресурсов СУБД появилась возможность отправлять в карантин запросы, использующие больше ресурсов, чем позволено. При этом работа такого запроса прерывается и при последующих запусках он не выполняется. Когда ситуация изменится или планка ограничения ресурсов будет поднята и запрос станет укладываться в ограничения, он уйдет из карантина и сможет выполняться. В качестве пороговых значений для отправки запроса в карантин могут использоваться следующие: cpu time, elapsed time, объем ввода-вывода, число физических или логических операций ввода-вывода.

VIII. Новые возможности Oracle 20c

В 2020 году Oracle анонсировал версию 20c. Это промежуточная версия с множеством новых возможностей. Отметим лишь самые интересные из них (рис. 5).



Рис. 5. Инновации в СУБД Oracle 20c

DB Nest, In-Memory Vector Join и Persistent memory мы уже обсудили.

Блокчейн-таблицы. В БД можно создавать, кроме обычных, блокчейн-таблицы (blockchain). Это обычные таблицы БД, открытые только на чтение и добавление строк. Изменить данные в этих таблицах нельзя. Удалить строки и всю таблицу можно будет только через определенный срок (или никогда). Сроки задает ДБА. Строки таблицы связаны в цепочки (как в обычном блокчейн). В каждой строке хранится хеш-значение



предыдущей строки цепочки. Это дает пользователю гарантию, что строки цепочки не изменились, т. к. для изменения одной строки придется перестроить всю цепочку. Кроме того, пользователи могут держать у себя копии этих хеш-значений и проверять, не изменились ли их значения в таблице. Это простой и удобный способ организовать централизованный блокчейн в БД для приложений/пользователей, не доверяющих друг другу.

Native JSON в БД. В таблицах БД могут быть колонки, в которых хранятся JSON документы. С ними можно работать, можно обновлять элементы данных в JSON документах. Теперь эти документы хранятся не в виде текста, а в бинарном представлении, что сильно ускоряет работу с этими данными. Кроме того, и опция In-Memory и ячейки Exadata умеют работать и с колонками, содержащими JSON документы.

SQL макросы в запросе. Чтобы упростить написание SQL-запроса, можно использовать SQL-макросы. SQL-макрос — это функция, которая на выходе возвращает текст, который подставляется в текст исходного SQL-запроса. Например, это может быть динамически сформированный текст подзапроса или предикат условия для WHERE. Макросы делятся на табличные (подставляются в фразе FROM и формируют имя таблицы/таблиц или подзапрос для формирования таблицы) и скалярные (подставляются в WHERE/HAVING, GROUP/ORDER BY и формируют имя колонки/группы колонок или целого предиката для WHERE. SQL-макросы позволяют проще писать то, что раньше реализовывалось с помощью сложного динамического SQL.

Автоматизация машинного обучения (AutoML в OML4Py). В состав опции Advanced Analytics, которая теперь является бесплатной компонентой СУБД Oracle, входят средства создания, обучения и использования моделей для машинного обучения (Machine Learning — ML). Опция имеет большую библиотеку готовых моделей и средств их построения и настройки (рис. 6). Как правило, для разработки моделей машинного обучения используется язык Python. Однако выбор, построение и обучение таких моделей — сложная операция и требует высокой квалификации в области машинного обучения.

Чтобы упростить работу по созданию ML моделей, Oracle разработал компоненту AutoML. Она помогает сделать три вещи:

- Выбрать правильную и наиболее подходящую модель для Вашей задачи (дает рекомендации и список подходящих моделей)
- Выбрать необходимые атрибуты (колонки таблицы), являющиеся входом для модели (рекомендует, какие атрибуты и как сильно влияют на результат. Чем меньше значимых атрибутов, тем быстрее и проще строится модель)
- Выбирать параметры настройки для моделей (features)

Сегодня алгоритмы машинного обучения очень широко используются в приложениях. Механизм AutoML в OML4Py (Oracle Machine Learning for Python) помогут неспециалистам в области машинного обучения создавать и использовать модели.



CLASSIFICATION

- Naïve Bayes
- Logistic Regression (GLM)
- Decision Tree
- Random Forest
- Neural Network
- Support Vector Machine
- Explicit Semantic Analysis

CLUSTERING

- Hierarchical K-Means
- Hierarchical O-Cluster
- Expectation Maximization (EM)

ANOMALY DETECTION

- One-Class SVM

TIME SERIES

- State of the art forecasting using Exponential Smoothing.
- Includes all popular models e.g. Holt-Winters with trends, seasons, irregularity, missing data

REGRESSION

- Linear Model
- Generalized Linear Model
- Support Vector Machine (SVM)
- Stepwise Linear regression
- Neural Network
- LASSO *

ATTRIBUTE IMPORTANCE

- Minimum Description Length
- Principal Comp Analysis (PCA)
- Unsupervised Pair-wise KL Div
- CUR decomposition for row & AI

ASSOCIATION RULES

- A priori/ market basket

PREDICTIVE QUERIES

- Predict, cluster, detect, features

SQL ANALYTICS

- SQL Windows, SQL Patterns, SQL Aggregates

FEATURE EXTRACTION

- Principal Comp Analysis (PCA)
- Non-negative Matrix Factorization
- Singular Value Decomposition (SVD)
- Explicit Semantic Analysis (ESA)

TEXT MINING SUPPORT

- Algorithms support text type
- Tokenization and theme extraction
- Explicit Semantic Analysis (ESA) for document similarity

STATISTICAL FUNCTIONS

- Basic statistics: min, max, median, stdev, t-test, F-test, Pearson's, Chi-Sq, ANOVA, etc.

R PACKAGES

- CRAN R Algorithm Packages through Embedded R Execution
- Spark MLlib algorithm integration

EXPORTABLE ML MODELS

- REST APIs for deployment

Рис. 6. Алгоритмы машинного обучения СУБД Oracle

IX. Прочие полезные улучшения

Безопасность. Появилась возможность прямой работы БД с MS Active Directory, без использования Oracle Internet Directory. Появилась возможность создавать пользователей БД без схемы (т. е. они имеют доступ только к объектам чужих схем) и схемы без пользователей (т. е. к такой схеме нельзя присоединиться извне, но с ее объектами могут работать другие пользователи, если им дали соответствующие привилегии). Появилась в БД и возможность анализа использования пользователями привилегий (Privilege Analysis). Не секрет, что часто ДБА дает пользователю избыточные привилегии и забывает об этом. Это снижает уровень защиты БД. Теперь можно проанализировать, какие привилегии пользователь реально использует, а какие нет, и отобрать у него излишние привилегии.

Oracle XE 18c. Бесплатная версия СУБД — Oracle XE (Express Edition), удобная для изучения возможностей СУБД и разработки некритичных приложений, теперь обновлена до версии 18c.

SQL монитор для разработчиков. Ранее смотреть в SQL-мониторе выполняющиеся запросы, их статус, план, параллелизм и т. д. мог только ДБА. Он видел все запросы БД. Теперь это может делать и пользователь с правами разработчика. Но он видит только свои запросы.

Web SQL Developer. Очень полезный и популярный инструмент SQL*Developer теперь можно запускать не только в ОС своей машины, но и через веб-браузер. Появился удобный веб-интерфейс. Web SQL Developer с ORDS (Oracle Rest DB service) может быть установлен на Вашей машине или на другой машине или в облаке и вызван через веб-браузер.

Отчет об использовании подсказок (хинтов). Для управления планами запросов ДБА и разработчики часто вставляют в текст запроса подсказки оптимизатору (hints). Но иногда они оптимизатором не используются (либо противоречат друг другу, либо содержат ошибки и т. д.). Часто мы даже не знаем, что подсказка не используется. Теперь в плане выполнения запроса мы видим какие подсказки не используются и почему.

Auto List Partitioning. Раньше новые секции таблицы при появлении записей с новым значением ключа секционирования автоматически создавались только для range partitioning. Теперь эта функция распространена и на List partitioning.



PL/SQL block code coverage (процент протестированного кода в PL/SQL). При тестировании PL/SQL-кода мы выполняем приложение в разных режимах, с разными входными параметрами и данными. Но, в зависимости от значения этих параметров и данных, выполняются те или иные части кода (например, разные секции оператора IF). А часть кода остается не протестированной. Теперь мы можем смотреть, какой процент кода протестирован, и подбирать исходные данные так, чтобы достичь полного тестирования кода. Это позволяет снизить вероятность возникновения ошибок на этапе эксплуатации и ускорить тестирование. Механизм позволяет построить отчет, в котором видно какие процедуры, функции, пакеты и т. д. выполнялись во время тестирования приложения и какой процент их кода был выполнен.

Онлайн шифрование табличных пространств. Для защиты данных БД от воровства рекомендуется шифровать данные в табличных пространствах. Более того, в автономных и облачных БД Oracle это делается автоматически в обязательном порядке. Но если у Вас есть БД с незашифрованными табличными пространствами, то Вы теперь можете их зашифровать, не останавливая работу с БД. Правда, для этого понадобится дополнительное дисковое пространство, т. к. зашифрованная и незашифрованная версии какое-то время будут сосуществовать.

ASM Flex Disk группы. При использовании ASM (Automatic Storage Manager) файлы разных БД хранятся на диске «россыпью», а хотелось бы объединить их в группы (например, группы файлов конкретной БД) и выполнять операции над группами, устанавливать ограничения (квоты на дисковое пространство для группы, уровень зеркалирования), делать расщепление зеркал (split mirroring) файлов группы. Группа ASM Flex Disk позволяет это сделать.

Shadow Lost File Protection (защита от сбойных блоков). Когда экземпляр Oracle пишет измененный блок данных на диск, он передает его файловой системе или ASM и считает, что блок в БД будет обновлен. Однако в момент записи может произойти сбой, и мы получим на диске старую версию блока. Для борьбы с этим используется standby БД, т. е. мы можем восстановить испорченный/устаревший блок из Standby БД. Новый механизм Shadow Lost File Protection позволяет заранее определять, что блок на диске старый и его надо обновить из резервной БД. Для этого Oracle в специально выделенное табличное пространство пишет SCN (номера) записываемых блоков. При считывании блока его SCN сравнивается с сохраненным SCN и при их расхождении выдается сообщение об ошибке.

ASM Parity Protected Files (программный аналог RAID-5 для ASM). Для восстановления испорченных или потерянных файлов и блоков обычно используют либо программное, либо аппаратное резервирование, т. е. все данные хранятся на дисках минимум в двух экземплярах. Однако существует более компактный механизм резервирования данных — RAID-5. В RAID-5 в дисковых подсистемах используется код Хемминга и хранятся не полные копии данных, а лишь коды коррекции, которые позволяют по сохранившимся данным восстанавливать потерянные данные. Это значительно снижает объем дискового пространства, необходимый для защиты от потери данных. Механизм ASM Parity Protected Files позволяет реализовать аналогичную RAID-5 защиту данных группы ASM. Он применяется для неизменяемых (write-once) файлов группы (архивные журналы, резервные копии и flashback-логи). ASM хранит не копии блоков, а лишь коды коррекции

Увеличение размера колонок типа Varchar2. Теперь размер поля колонки типа Varchar2 может достигать 32К.

Изменения в опции RAT. Уникальная опция RAT — Real Application Testing позволяет захватить реальную нагрузку на промышленной БД и потом проигрывать эту нагрузку на тестовой или промышленной БД и заранее выявлять влияние изменений в оборудовании, ПО, настройках на производительность БД и SQL-запросов и работоспособность БД. RAT широко используется заказчиками (особенно при миграции на новые версии СУБД) и постоянно развивается. Например, если раньше Database Replay (один из компонентов RAT) при исходной нагрузке с вызовами PL/SQL кода (процедур и функций) захватывал и проигрывал только вызовы верхнего уровня, а отдельные user calls внутри кода процедур не синхронизировались между сессиями, то теперь, с появлением режима Extended в Database Replay, вся нагрузка проигрывается более аккуратно, с полным соблюдением синхронности по транзакциям. Это делает воспроизводимую нагрузку еще более реалистичной.



Появилась возможность ремэпинга схемы БД, т. е. захват идет на одной схеме БД, а проигрывание — на другой идентичной схеме. Появился полезный режим проигрывания DB Replay Query only (только чтение), при котором исходные DML изменяющие данные заменяются на часть Select, таким образом, проигрывание позволяет воспроизвести нагрузку на чтение, но не менять данные в БД. Это во многих случаях ускоряет и упрощает тестирование, не надо откатывать БД, кроме того, режим Query Only можно использовать для «разогрева» кэша перед основным прогоном нагрузки.

Для экономии времени при сравнительной оценке производительности SQL-запросов, в SQL Performance Analyzer (SPA) теперь можно включить режим параллельного тестового запуска (trial). Т. е. группы запросов будут проигрываться одновременно в разных сессиях, не влияя друг на друга. Также в SPA появилась возможность сравнивать результаты запроса (Comparison of SQL result sets) между различными тестовыми запусками, таким образом мы увидим, что один и тот же запрос после изменений условий выдал различные результаты — а это предмет для расследования.

Следует заметить, что здесь перечислена только часть новых возможностей СУБД Oracle 12.2, 18с, 19с и 20с. Полный список можно найти в документации по конкретной версии СУБД. Кроме того, некоторые из рассмотренные новые возможности требуют больших дополнительных вычислительных ресурсов (например, автоматические индексы, онлайн статистика, memoptimized таблицы и т. д.), поэтому они реализованы только на платформе Exadata. Полный список возможностей на разных платформах также приведен в документации СУБД Oracle.

Новые версии СУБД могут работать как в Вашем ЦОД (on-premise), так и в публичном облаке Oracle. Они также доступны на Exadata Cloud Machine (ExaCC), которую Oracle ставит в Вашем ЦОД, но обслуживает сам. На базе этих СУБД (начиная с 18с) также реализованы автономные БД Oracle [2]. Они тоже могут работать как в публичном облаке Oracle, так и на ExaCC. Каждая автономная БД — это PDB.

Новая инициатива Oracle — Always Free — позволяет каждому желающему бесплатно заказать на неограниченный срок две автономные БД. На них удобно изучать новые возможности, вести разработку, тестировать свои приложения.



Продукты Oracle



Программно-аппаратные комплексы (Инженерные системы)

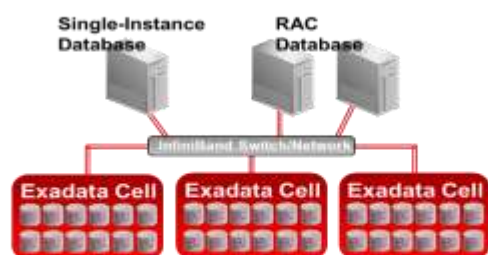
Машина баз данных Oracle Exadata

Машина баз данных Oracle Exadata является наиболее производительной, высокодоступной платформой для работы СУБД Oracle. За счет высокой «плотности вычислений», по сравнению с альтернативными платформами, позволяет на сопоставимом по характеристикам аппаратном обеспечении выполнять гораздо больше задач. Как результат оптимизации стека программного обеспечения и использования уникальных алгоритмов для реализации стандартного функционала СУБД Oracle, производительность прикладных систем на Exadata часто недостижима при работе на альтернативных платформах.

Машина баз данных Oracle Exadata разрабатывается и постоянно совершенствуется совместной работой экспертов СУБД Oracle и системных инженеров. Именно благодаря этому в состав программно-аппаратного комплекса входят не только оптимальные для производительной и бесперебойной работы СУБД аппаратные компоненты, но и новые программные уровни, вынесенные за пределы кода СУБД Oracle. Само ядро СУБД включает в себя доработки, которые позволяют с помощью специального API работать с программными слоями, расположенными на системе хранения данных — Exadata Storage Software.

В основе решений Exadata лежит современная облачная архитектура, которая сочетает высокопроизводительные горизонтально масштабируемые серверы баз данных, горизонтально масштабируемые интеллектуальные серверы хранения данных с современными флэш-накопителями PCI NVMe, передовой кэш СХД на базе энергонезависимой памяти (PMEM), а также сетевой фабрика RoCE (RDMA over Converged Ethernet), связывающая серверы и ячейки хранения. Уникальные сквозные алгоритмы обработки баз данных в Exadata реализованы на уровнях системы хранения, вычислительных серверах и сетевых компонентах, обеспечивая более высокую мощность и производительность при уменьшении затратах по сравнению с другими платформами.

Конфигурация системы с Exadata



- Каждая ячейка Exadata – самостоятельный сервер с установленными дисками и ПО Exadata
- Данные «размазаны» между многими ячейками Exadata
- Нет ограничения на количество ячеек в системе

Машины баз данных Exadata предназначены для любых задач баз данных, включая массированные транзакционные системы, корпоративные информационных хранилища, выполнение аналитики в оперативной памяти (In-Memory Analytics), Интернет вещей (IoT) и любые их комбинации. Простые и быстрые при введении в эксплуатацию машины баз данных Exadata обеспечивают обработку и надежную защиту важнейших баз данных клиентов.

На Oracle Exadata используется стандартная СУБД Oracle. Именно поэтому любое приложение, работающее сегодня с СУБД Oracle, можно легко перенести на машину баз данных Exadata без каких бы то ни было изменений.



Быстрое надежное развертывание и масштабирование

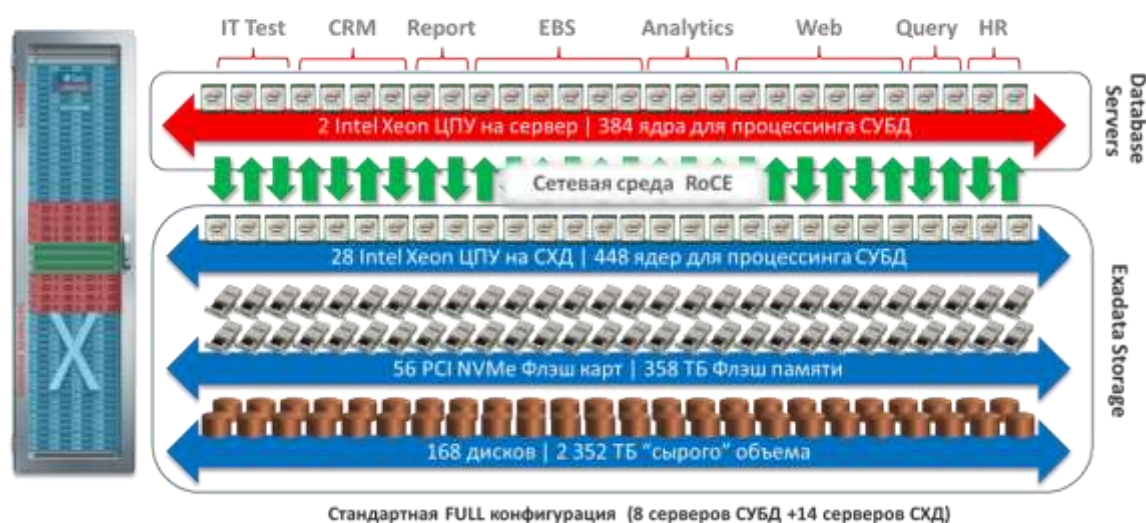
Программно-аппаратный комплекс поставляется, обслуживается и поддерживается как единая система, с повсеместной сквозной автоматизацией процессов диагностики и обновления — это позволяет существенно снизить накладные расходы.

Машина БД Exadata может быть куплена и развернута в ЦОД у заказчика как основа частного облака для баз данных, или приобретена как облачная среда через сервисную подписку (в этом случае вся инфраструктура управляется инженерами Oracle) при двух возможных вариантах размещения — в публичном облаке Oracle Public Cloud, или в ЦОД самого заказчика в формате Cloud at Customer.

Машина баз данных Exadata проста в развертывании, т.к. включает в себя все необходимые для работы с СУБД Oracle аппаратные компоненты. Сетевое оборудование, серверы баз данных и серверы хранения данных предварительно сконфигурированы, настроены и протестированы экспертами Oracle, что позволяет экономить недели и даже месяцы рабочего времени, которые обычно требуются для развертывания высокопроизводительных систем. Всестороннее комплексное тестирование гарантирует совместимость всех компонентов (ПО баз данных, операционной системы, микропрограммного обеспечения, драйверов и т.п.), а также отсутствие узких мест производительности и единых точек отказа, которые могут стать причиной выхода системы из строя.

Машины баз данных Exadata построены на горизонтально масштабируемой архитектуре, как на уровне серверов баз данных и серверов хранения, так и на уровне сетевой инфраструктуры. Расширение мощностей и емкостей Exadata происходит путем сбалансированного увеличения числа процессоров, ресурсов хранения и сетевых ресурсов, что обеспечивает масштабируемость без возникновения узких мест.

Горизонтально масштабируемая архитектура Exadata



В настоящий момент Oracle предлагает уже десятое поколение этого продукта, состоящее из двух моделей X8M-2 (сервера СУБД с двумя ЦПУ по 24 ядра каждый) и X8M-8 (сервера СУБД с восемью ЦПУ по 24 ядра каждый). Базовой минимальной комплектацией для обеих моделей является стойка с двумя серверами СУБД и тремя серверами СХД — используя принцип эластичности, заказчик может выбрать конфигурацию с любым необходимым количеством серверов выше минимального. Комплекс обеспечивает последовательное и сбалансированное масштабирование как добавлением компонент внутри одной стойки, так и с помощью объединения нескольких стоек в одну систему.

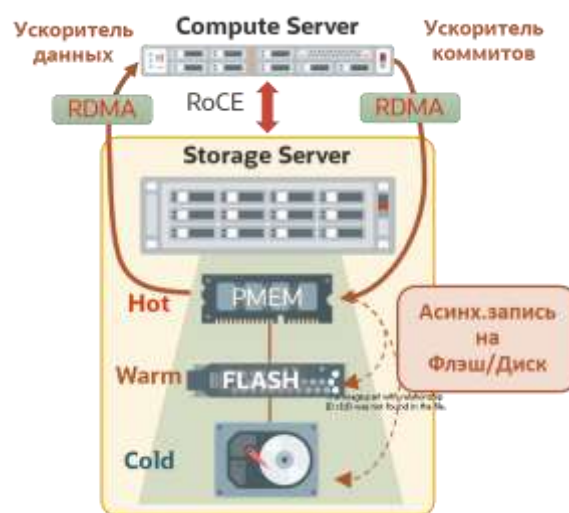


Память Intel Optane и революционный сетевой интерконнект RDMA

Модель Exadata X8M представляет собой следующее поколение программно-аппаратного комплекса с очередными инновациями - сверхскоростной облачной сетевой среды на базе технологии RDMA over Converged Ethernet (RoCE). RDMA (удаленный прямой доступ к памяти) позволяет одному компьютеру получать доступ к данным с другого компьютера без обращения к операционной системе и процессору, что обеспечивает высокую пропускную способность и низкие задержки. Сетевая карта напрямую выполняет операции чтения и записи данных в памяти без дополнительного копирования и буферизации. RDMA является неотъемлемой частью высокопроизводительной архитектуры Exadata. За последнее десятилетие она была улучшена и усовершенствована. В ее основе лежат несколько технологий СУБД Oracle, основанных только на разработках Exadata, таких как протокол Exafusion Direct-toWire и решение Smart Fusion Block Transfer.

В модели Exadata X8M реализована сетевая среда RoCE, работающая с производительностью 100 Гбит/с, что делает эту самую быструю в мире машину баз данных еще быстрее. Реальные рабочие нагрузки баз данных, выполняемые на Exadata X8M-2, используя новые ускорители разделяемой энергонезависимой памяти Intel Optane, побили предыдущий рекорд Exadata X8 в 2,5 раза, показав результат более чем 16 млн операций чтения OLTP (для запросов блоков базы данных 8 КБ).

Впервые в серии Exadata X8M серверы хранения могут кэшировать данные в постоянной памяти (PMEM) Intel Optane и использовать ускорители операций фиксации транзакций базы данных, что позволяет на порядок уменьшить задержки при доступе к удаленно хранимым данным. Постоянная энергонезависимая память — это новая аппаратная технология Intel Optane, добавляющая отдельный уровень производительности, по показателям емкости, объема и стоимости располагающаяся между DRAM и флэш-памятью. Поскольку постоянная память физически размещается на шине памяти ЦПУ сервера хранения, операции чтения выполняются со скоростью, намного более быстрой, чем флэш-память.



Ключевые программные технологии для высокой производительности

Современные многопроцессорные серверы обрабатывают данные со скоростью в десятки и даже сотни гигабайт в секунду. Традиционные системы хранения не могут обеспечить вычислительным серверам эквивалентную скорость передачи данных через контроллеры и сетевую среду системы хранения.

Беспрецедентная производительность Exadata и отсутствие узких мест, характерных для обычных массивов хранения, обеспечиваются за счет ПО серверов хранения Exadata. Это программное обеспечение включает алгоритмы расширенного взаимодействия с СУБД Oracle, что позволяет создать высокоэффективную инфраструктуру хранения, оптимизированную для баз данных.

Каждый сервер хранения Exadata оснащен двумя 16-ядерными процессорами x86, используемыми для выполнения массивной обработки данных БД Oracle, которая в альтернативных системах выполняется только на серверах СУБД. В стойке Exadata может быть до 576 ядер процессоров на уровне системы хранения. Процессоры серверов хранения не заменяют полностью ЦПУ серверов СУБД, но существенно разгружают их от рутинной работы. Они ускоряют интенсивную обработку данных подобно тому, как графические карты ускоряют обработку изображений.

Параллельная массивная обработка и фильтрация данных на серверах хранения Exadata позволяет многократно снизить объем информации, передаваемой по сети между СХД и вычислительными серверами,



значительно ускорить выполнение запросов и освободить процессорные мощности серверов СУБД для другой полезной работы.

В реальных комплексах с сетевой СХД время между началом и завершением запроса к системе хранения на уровне кода СУБД сейчас во многом зависит от времени выполнения полного стека ввода-вывода, а не скорости работы конечных устройств хранения на флэш. При использовании стандартных протоколов доступа установка на уровне общего СХД еще более быстрых накопителей уже лишено смысла, т. к. время обработки стека ввода-вывода значительно превышает время выполнения команды на уровне самого носителя.

В Exadata применен инновационный метод обращения к сверхбыстрой памяти — непосредственно сама СУБД производит обращение к блокам данным закешированным в энергонезависимой памяти Intel Optane через протокол прямого доступа к памяти другого компьютера — RDMA. Это позволяет добиться скорости чтения и записи, недостижимой при использовании стандартного стека ввода-вывода.

Интеграция СУБД Oracle с интеллектуальной системой хранения позволяет СУБД ускорять типовые операций за счет инновационных технологий поддерживаемых программным обеспечением Exadata:

- **Smart Scans** — ПО Exadata обеспечивает перенос массивных операций обработки данных на уровень системы хранения, и выполняет ее сразу после чтения данных с носителей. Благодаря многократному сокращению объемов данных, возвращаемых на сервера СУБД, а также распараллеливания обработки между серверами хранения, скорость выполнения SQL может быть увеличена в десятки и сотни раз;
- **Storage Indexes** — структура, позволяющая исключить операции ввода-вывода и обработки данных последовательности блоков на диске, если известно, что там нет строк, соответствующих искомому диапазону запроса SQL. Такие индексы автоматически поддерживаются в памяти СХД во время выполнения Smart Scans. В Storage Index сохраняются минимальные и максимальные значения столбца базы данных для последовательности блоков на диске, принадлежащих одной таблице. Обычно совокупный размер каждой такой последовательности блоков соответствует оптимальному размеру одной операции ввода-вывода при сканировании;
- **Hybrid Columnar Compression** — инновационный механизм сжатия данных позволяет уменьшить объемы дискового пространства, требуемого для хранения информации, в десятки раз, в то же время многократно увеличивая скорость их сканирования. Технология предусматривает группировку данных таблиц по столбцам, что обеспечивает эффективность сжатия и способствует быстрому доступу к данным для аналитических запросов;
- **Smart Flash Cache** — флэш-память, реализованная на PCI-картах, реализует промежуточный уровень хранения данных между быстрой оперативной памятью с «горячими» данными и постоянным хранилищем на дисках. Флэш-кэш хранит активные наборы данных, предсказывая их востребованность в будущем за счет интеграции с СУБД Oracle. В каждый сервер хранения X8M HC установлено 4 PCI флэш-накопителя общим объемом 25,6 ТБ. Количество серверов хранения на каждой машине не может быть менее трех;
- **Smart PMEM Cache** — промежуточный уровень кэширования между кэшем серверов СУБД в DRAM и Smart Flash Cache серверов хранения, реализованный на сверхбыстрой энергонезависимой памяти Intel Optane. Благодаря использованию уникальной технологии RDMA обеспечивает впечатляющее время чтения данных из кэша до 19 микросекунд и производительность до 16 миллионов IOPS для конфигурации полной стойки. В каждом сервере хранения установлено по 1,5 ТБ энергонезависимой памяти. Количество серверов хранения на каждой машине не может быть менее трех;
- **Smart PMEM Log** — запись в журнальный файл через кэш, расположенный в энергонезависимой памяти Intel Optane серверов хранения.



Кроме того, только на Exadata, за счёт её уникального функционала, СУБД Oracle имеет ряд возможностей недоступных на альтернативных платформах.

На Exadata СУБД Oracle использует машинное обучение для автоматической индексации данных в таблицах. Автоматическая индексация непрерывно анализирует выполняемые SQL запросы и создает новые индексы для повышения производительности. Непрерывность анализа позволяет среагировать на изменение шаблона нагрузки или модели данных.

Только на Exadata СУБД Oracle собирает статистические данные в реальном времени при добавлении, изменении или удалении данных операциями DML. Сбор статистических данных в реальном времени позволяет механизму оптимизации SQL динамически адаптировать планы выполнения при существенном изменении данных.

Эффективная консолидация баз данных

С помощью Oracle Exadata предприятия могут сократить затраты на IT-инфраструктуру за счет эффективной консолидации, многократного сжатия данных, улучшения производительности всех приложений, сокращения времени ввода в эксплуатацию и повышения качества бизнес-решений в реальном времени.

Только на Exadata для СУБД Oracle реализовано сквозное разграничение системными ресурсами между базами данных и группами задач в них — ЦПУ, память, сетевой трафик, энергонезависимая память, флэш, сервисное время системы ввода-вывода. Для этого на Exadata интегрируются встроенный в СУБД ресурсный менеджер, сетевой менеджер ресурсов и менеджер ресурсов системы ввода-вывода. Данная связка обеспечивает исключительный контроль всех ресурсов требуемых для работы СУБД Oracle.

Гибкие квоты для баз данных и групп процессов позволяют устанавливать приоритеты на использование ресурсов и позволяют их использовать более эффективно. Это позволяет гораздо более высокую степень консолидации, чем на альтернативных платформах. Общее количество баз данных, использующих соизмеримое оборудование на Exadata может быть значительно выше — больше плотность вычислений, выше эффективность и рентабельность.

При необходимости изоляции на Exadata можно использовать виртуальные машины KVM.

Все это в совокупности с высокой производительностью позволяет говорить о том, что машина баз данных Oracle Exadata обеспечивает наиболее экономически выгодную среду для работы и консолидации множества баз данных Oracle.

Надежность работы системы

Конфигурации всех машин баз данных Exadata стандартны, что дает заказчикам возможность опираться на опыт тысяч других пользователей — банков, телекоммуникационных компаний и организаций розничной торговли, — которые уже внедрили Exadata для своих критически важных приложений. Системы заказчиков также идентичны системам, которые служба поддержки Oracle использует для выявления и разрешения проблем, системам, которые разработчики Oracle применяют при создании и тестировании СУБД Oracle, и системам, которые корпорация Oracle использует для реализации собственных публичных облачных сред SaaS (ПО как услуга) и PaaS (платформа как услуга).

Oracle Exadata — это наиболее тщательно протестированная и настроенная платформа для работы с СУБД Oracle, обеспечивающая высочайший уровень поддержки.



В комплексе реализована избыточность всех аппаратных и программных компонент, что обеспечивает работоспособность системы в случае сбоя любой из них. Кроме этого, высокая готовность и безопасность системы обеспечивается функционалом СУБД Oracle.

Машина баз данных Oracle Exadata поддерживается как единый программно-аппаратный комплекс: компания Oracle является производителем и поставщиком аппаратных и программных компонент системы.

Модели конфигурации Exadata X8M

Программно-аппаратный комплекс поставляется в двух моделях, X8M-2 и X8M-8, отличающимися входящими в их состав серверами СУБД, на которых работает основной код Oracle Database.

- **Сервера СУБД X8M-2:** системы, оснащенные **двумя** 24-ядерными процессорами Intel Xeon, всего 48 ядер на один сервер СУБД. Три варианта конфигурация DRAM: 384 ГБ, 768 ГБ и 1,5 ТБ. Сервера содержат два порта Ethernet 100 Гб/с для внутрисистемных коммуникаций и несколько сетевых портов (в зависимости от конфигурации) Ethernet стандартов 1/10/25 Гб/с для связи с пользователями и внешними системами.
- **Сервера СУБД X8M-8:** системы, оснащенные **восемью** 24-ядерными процессорами Intel Xeon, всего 192 ядра на один сервер СУБД. Два варианта конфигурация DRAM: 3 ТБ и 6 ТБ. Сервера содержат восемь портов Ethernet 100 Гб/с для внутрисистемных коммуникаций и несколько сетевых портов (в зависимости от конфигурации) Ethernet стандартов 1/10/25 Гб/с для связи с пользователями и внешними системами.

Минимальное количество серверов СУБД в одной стандартной стойке Exadata — два, для возможности обеспечения высокой готовности системы.

Все остальные компоненты комплекса идентичны, включая серверы хранения и интеллектуальной обработки данных Exadata и два коммутатора Ethernet 100 Гб/с, связывающих все серверные модули между собой.

Машина баз данных Exadata поставляется с тремя типами серверов хранения и обработки данных:

- **Exadata Extreme Flash (EF)** — для обеспечения стабильности времени доступа. Такие сервера оснащаются флэш-накопителями PCI (общая емкость 51,2 ТБ) и энергонезависимой памятью Intel Optane (общая емкость 1,5 ТБ). Флэш используется для постоянного хранения данных и их кэширования в колоночном формате. На PMEM реализован кэш для сверхбыстрого отклика OLTP операций чтения данных и записи в журнальный файл. Каждый сервер включает в себя два процессора с общим количеством в 32 вычислительных ядра для выполнения обработки данных сразу после их чтения с носителей.
- **Exadata High Capacity (HC)** — наиболее сбалансированный сервер, сочетающий в себе производительность флэш- и энергонезависимой памяти с большой емкостью жестких дисков. Такие сервера оснащаются двенадцатью HD-дисками (общая неформатированная емкость 168 ТБ), четырьмя флэш-накопителями PCI (общая емкость 25,6 ТБ) для кэширования, и энергонезависимой памятью Intel Optane (общая емкость 1,5 ТБ). Каждый сервер включает в себя два процессора с общим количеством в 32 вычислительных ядра для выполнения обработки.
- **Exadata Extended (XT)** — недорогие сервера, которые используются для расширения дискового пространства конфигураций с другими типами серверов хранения. Предназначен для хранения архивных данных, доступ к которым не требует высокой производительности. Может быть альтернативой выгрузки данных на внешние системы, например сервер Hadoop. Такие сервера оснащаются двенадцатью механическими дисками (общая неформатированная емкость 168 ТБ). Каждый сервер включает в себя один процессор с общим количеством в 16 вычислительных ядер для выполнения обработки.



Минимальное количество серверов хранения в стандартной стойке Exadata — **три** одного типа, для возможности обеспечения высокой готовности системы. При расширении системы предоставляется возможность добавлять сервера тех типов, которые необходимы для наиболее оптимального решения выполняемых прикладных задач.

Система может эластично расширяться по мере повышения требований к вычислительным ресурсам или дисковому пространству. В стойку можно добавлять необходимое количество серверов СУБД соответствующие модели машины и любые интеллектуальные сервера СХД требуемого типа. Несколько стоек можно связывать между собой, объединяя их в один программно-аппаратный комплекс.

Преимущества Exadata для бизнеса

Exadata ускоряет вывод новых бизнес-приложений на рынок за счет более быстрого конфигурирования, настройки и тестирования. Сроки внедрения сокращаются с нескольких месяцев до нескольких дней. Значительно снижается риск возникновения непредвиденных проблем системного уровня после ввода в эксплуатацию – Exadata является надёжной основой для обеспечения непрерывности бизнеса.

Высокая производительность Exadata способствует повышению эффективности бизнес-процессов, и помогает коммерческим пользователям принимать более взвешенные решения, исследовать возможности для роста и снижать затраты. Exadata обеспечивает значительное сокращение времени отклика на запросы пользователей, что, в свою очередь, повышает эффективность работы сотрудников и способствует благоприятному впечатлению у заказчиков.

Заключение

Exadata — это полностью интегрированная платформа для баз данных, использующая новейшие аппаратные технологии и уникальное программное обеспечение, что позволяет добиться высочайшего уровня производительности, готовности и безопасности. Все эти преимущества в сочетании с экономией расходов, простотой управления и улучшенной поддержкой помогают повысить гибкость и эффективность бизнеса. Машина БД Exadata является новым глобальным стандартом для СУБД Oracle независимо от того, где она работает — в публичном облаке Oracle или в ЦОД у заказчика.

Облачная машина — Exadata Cloud at Customer Gen 2

Заказчики выбирают Oracle Exadata по целому ряду причин. Это экстремальная производительность для всех типов нагрузок, замечательная масштабируемость, высокая надежность на уровне 99,999 %, комплексная поддержка от одного поставщика, широкие возможности консолидации — вот лишь часть из множества факторов, которые определяют выбор Oracle Exadata со стороны заказчиков. Для работы приложений на Exadata нет необходимости переделывать приложения, СУБД Oracle, работающая на Exadata в облаке, на 100 % совместима с существующими приложениями.

Достоинства публичного облака общеизвестны: это цена, это эластичность — т. е. возможность увеличивать и уменьшать потребление ресурсов облака по мере необходимости и, следовательно, уменьшать свои расходы, это обслуживание облака специалистами компании Oracle и т. д. Однако не все приложения заказчиков могут быть перенесены в облако. Переносу приложений в облако могут препятствовать законодательные ограничения, политики безопасности самих компаний, технические причины (приложения и базы тесно переплетены между собой, приложения требуют низких сетевых задержек).

Exadata Cloud at Customer предназначена для пользователей СУБД Oracle, для которых перенос данных в облако Oracle невозможен по каким-либо причинам. Для таких пользователей компания Oracle предлагает сервис Exadata Cloud at Customer: это оборудование Exadata, предоставляемое заказчикам по облачной модели подписки, установленное в ЦОД пользователя. Иными словами, Exadata Cloud at Customer —



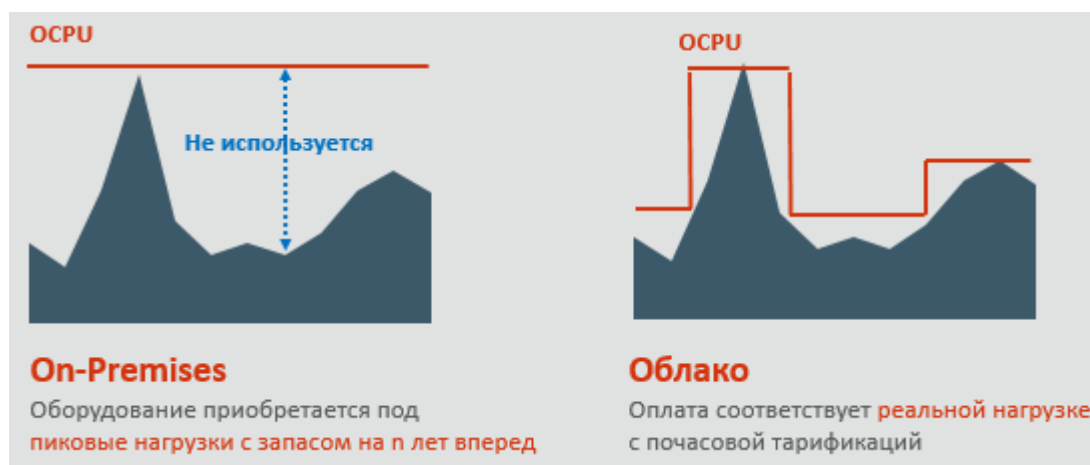
это Exadata, находящаяся в ЦОД заказчика, с интерфейсами облака, оборудованием облака, программным обеспечением из облака, с оплатой по модели облака (за фактически использованные ресурсы), с настройками безопасности по модели облака.

В Exadata Cloud at Customer всегда используется самое новое оборудование, на сегодня это поколение X8. Минимальная поставляемая конфигурация — это **Quarter Rack**: два сервера БД по 50 физических ядер каждый (всего 100 физических ядер), 720 ГБ RAM в каждом сервере, 3 ячейки хранения по 48 физических ядер в каждой, 192 ГБ RAM, 168 ТБ дисковой емкости (raw) и 25 ТБ флэш-кэша, максимальный объем файлов данных в такой конфигурации — 60 ТБ. Версии СУБД Oracle, которые поддерживаются в Exadata Cloud at Customer: 11.2.0.4, 12.1, 12.2, 18с, 19с. При необходимости объем ресурсов в Exadata Cloud at Customer можно как наращивать, добавляя как серверы БД, так и ячейки хранения, так и уменьшать, отключая часть ядер в серверах БД. Максимальный объем ресурсов в одной стойке: 8 серверов БД (400 физических ядер, 5760 ГБ RAM в сумме), 12 ячеек хранения: 307 ТБ флэш-кэша, максимальный объем БД 479 ТБ.

К Exadata Cloud at Customer Oracle предлагает две лицензионных модели:

Enterprise Edition Extreme Performance = «все опции СУБД Oracle + все Enterprise Manager Pack» — когда заказчик подписываясь на аренду оборудования подписывается вдобавок лицензии СУБД Oracle на это оборудование либо **Bring Your Own Licenses** — когда заказчик переносит на Exadata Cloud at Customer имеющиеся у него лицензии СУБД Oracle. Опция BYOL позволяет значительно снизить периодические платежи.

Оплата по мере использования. Нагрузка от приложения может значительно изменяться в течение суток, в течение недели (низкая нагрузка в выходные дни), в течение дней месяца (повышенная нагрузка при закрытии месяца и подведении месячной отчетности). В этих условиях владельцы оборудования вынуждены приобретать оборудование с таким расчетом, чтобы покрыть самые высокие потребности на несколько лет вперед. В остальные периоды оборудование недозагружено (левая половина рисунка).



Подписываясь на облачные сервисы, заказчик платит только за реально использованный объем ресурсов. Благодаря эластичности облака при возникновении потребности в ресурсах заказчик может поднять свои лимиты, для того чтобы уложиться в отведенный период времени, а по окончании высокой нагрузки заказчик понижает лимиты, для того чтобы уменьшить свои платежи.

В Exadata Cloud at Customer все расчеты ведутся в ОСРPU (виртуальные ядра), минимально разрешается 2 ОСРPU, разрешается уменьшать и увеличивать количество ядер по мере необходимости. Оплата рассчитывается на основе количества и длительности использования ОСРPU. Дисковые объемы, флэш, оперативная память и сетевой трафик не тарифицируются. Абонентская плата складывается из двух составляющих: оплата оборудования и оплата ОСРPU.

Все манипуляции выполняются онлайн, т. е. не требуют остановки баз данных.



Преимущества ExaCC

- Оборудование и ПО находится на площадке заказчика
- Архитектура Exadata обеспечивает высокую производительность
- Высокая надежность и устойчивость к сбоям обеспечивается за счет аппаратной избыточности и ПО (ASM, Grid, RAC), минимальное время простоя
- Техническое обслуживание ПО и оборудования осуществляют специалисты Oracle (единая точка входа по всем проблемам)
- Широкий выбор опций СУБД Oracle (все опции включены)
- Масштабирование ресурсов в широком диапазоне

Oracle Zero Data Loss Recovery Appliance

Бизнесу требуется, чтобы системы никогда не теряли критичных для бизнеса данных, и чтобы при этом резервирование не влияло на работу бизнес-приложений, поэтому резервное копирование базы данных требует особого подхода, недостаточно рассматривать базу данных как набор файлов. И в этом заключается проблема традиционных решений для резервного копирования баз данных, поскольку они рассматривают базу данных не как транзакционную систему со специфическими требованиями к целостности данных. Просто скопировать файлы базы данных недостаточно — это не гарантирует целостности базы данных.

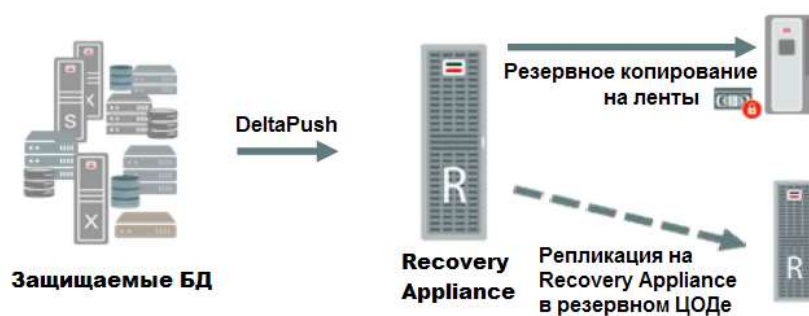
Компания Oracle решила этот вопрос просто и естественно. Был создан аппаратно-программный комплекс для бэкапирования и восстановления БД Oracle — Oracle Zero Data Loss Recovery Appliance (ZDLRA). Он может обслуживать множество БД Вашего ЦОД. После создания полного бэкапа Вашей БД последующие изменения в БД (инкрементальные бэкапы) и журналах (redo) постоянно передаются на ZDLRA. Механизм чем-то похож на синхронизацию основной и резервной БД при использовании Data Guard для standby БД.

Чтобы обеспечить надежное восстановление баз данных, Oracle Zero Data Loss Recovery Appliance постоянно проверяет целостность резервных копий. Ключевой термин при работе с Recovery Appliance — «виртуальная полная копия БД». Сначала только один раз делается полная резервная копия базы данных, а затем благодаря инкрементальным копиям («дельтам») Recovery Appliance может на лету воссоздать «виртуальную полную копию БД». При этом благодаря сохранению журналов транзакций в реальном времени обеспечивается нулевая потеря данных. Recovery Appliance может быстро воссоздать полную копию БД на любой момент времени, быстро сложив все «дельты», таким образом минимизируется объем передаваемой информации при необходимости восстановления базы данных в случае сбоя.



Recovery Appliance может автономно, в асинхронном режиме, копировать резервные копии Ваших баз данных на ленту или на резервный ЦОД.





Сжатие резервных копий тоже занимается Recovery Appliance, тем самым высвобождая Ваши промышленные серверы для основной работы.

Очень важно, что управление процессом резервного копирования Recovery Appliance, всеми базами данных, всеми репликами и копиями на ленту выполняется через единый интерфейс Enterprise Manager. Все копии регистрируются в Recovery Catalog, и администратор базы видит все способы, какими он может восстановить базу данных.

Базовая комплектация Zero Data Loss Recovery Appliance называется Base Rack и состоит из двух серверов, выполняющих резервное копирование и трех серверов хранения, на которых хранятся резервные копии. Объем дискового хранения для Base Rack составляет 155 ТБ. Расширять дисковое хранилище можно легко добавляя дополнительные сервера хранения, каждый сервер хранения добавляет 53 ТБ полезного дискового пространства. Для Full Rack (18 серверов хранения) объем хранения составляет 949 ТБ, а скорость резервирования до 24 ТБ/час. Основой для ZDLRA является Exadata.

В отличие от других средств бэкапирования, ZDLRA обеспечивает почти нулевую потерю данных при сбое БД, минимальную нагрузку на продуктивную БД, гарантированное восстановление БД, компактное и надежное хранение всех бэкапов организации, простую настройку и управление процессом регулярного бэкапирования, быстрое восстановление БД Oracle на любой момент времени.

Уникальные преимущества Recovery Appliance



Без потерь данных

Передача журналов изменений в режиме реального времени защищает текущие транзакции



Минимальное влияние на промышленные базы данных

Сжатие, проверка резервных копий, работа с лентами и т.п. выполняется уже не на промышленном сервере.



Надежное восстановление БД

Восстановление базы данных в согласованном состоянии, а не просто вернуть файлы БД на момент в прошлом



Облачная масштабируемость

Лёгкая защита всех баз данных в ЦОД, используя масштабируемый сервис



Oracle Database Appliance (ODA)

Oracle Database Appliance — оборудование из семейства Engineered System, предназначенное для работы с СУБД Oracle. Впервые появилось в 2011 году и на протяжении 7 поколений пользуется большой популярностью у пользователей. Это оборудование отличается низкой ценой, гибкими возможностями конфигурации и лицензирования. Все модели семейства ODA включают сервер БД (ЦПУ, память) плюс встроенная СХД, спроектированные специально для СУБД Oracle. Это оборудование идеально подходит для заказчиков, которым не требуется вникать в технические детали, но при этом необходимо быстро и дешево запустить базу данных для любых нагрузок: OLTP, Data Warehousing, In-Memory. В отличие от Exadata, ODA не имеет умных ячеек хранения (Exadata Storage Cell) и не выполняет вычислений на ячейках.

В настоящее время семейство ODA состоит из трех моделей:

- X8-2 **S** — самая маленькая модель в линейке, один сервер 16 ядер, 192 ГБ RAM, дисковое пространство 6Т
- X8-2 **M** — средняя модель, один сервер 32 ядра, 384 ГБ RAM, дисковое пространство 6Т
- X8-2 **HA** — самая мощная модель, на базе которой можно строить системы с высокой надежностью (High Available), два сервера, каждый: 32 ядра, 384 ГБ RAM (с возможностью расширения до 768 ГБ), 20 ТБ дискового пространства, при необходимости к этой модели можно подключить вторую дисковую полку.

Основное достоинство ODA — простота. Чтобы запустить это оборудование, распакуйте коробку, в котором оно прибыло, включите в электрическую розетку, подключите сетевые кабели и нажмите кнопку запуска на передней панели сервера. При первом включении сервера запустится процесс настройки: сервер попросит указать hostname, попросит указать IP-адрес, спросит название БД, которую предстоит создать, и задаст еще несколько технических вопросов. Обычно процесс первого запуска занимает около 60 минут, после чего сервер с развернутой базой данных готов к работе. Компания Oracle является единой точкой технической поддержки по любым вопросам с этим оборудованием и по работе СУБД Oracle на этом оборудовании.

В ODA используется специализированное ПО для управления базами данных и сервером — Appliance Manager. Посредством данного ПО выполняется обновление прошивок, установка обновлений на ОС, на СУБД. Использование встроенного ПО уменьшает риск человеческих ошибок. Appliance Manager также управляет ASM (Automatic Storage Management), следит за сбоями оборудования.

Система хранения

В базовой поставке модели S и M оснащены двумя флэш-дисками NVMe Express по 6,4 ТБ, общей емкостью по 12,8 ТБ. Использование NVMe-дисков позволяет обеспечить высокую производительность СУБД Oracle. Модель S не расширяемая, а в модели X8-2M имеется возможность добавить еще 10 NVMe дисков по 6,4 ТБ, чтобы расширить дисковое пространство до 76,8 ТБ.

В модели X8-2-HA используется отдельная полка на 24 диска. В базовой поставке она содержит 6 флэш-SSD-дисков по 7,68 ТБ общей емкостью 46 ТБ. В свободные слоты этой полки можно добавить еще 18 SSD-дисков по 7,68 ТБ и таким образом довести общую емкость до 369 ТБ SSD, либо можно добавить 92 ТБ SSD + 504 ТБ HDD-дисков. При необходимости к этой модели можно дополнительно подключить еще одну полку на 24 диска.

Все диски находятся под управлением ASM (Oracle Automatic Storage Management).

Программное обеспечение

Все модели семейства ODA X8 сертифицированы для Oracle Linux 7 и 6, Oracle VM, KVM, СУБД Oracle версий 11.2, 12c, 18c, 19c как Enterprise Edition, так и Standard Edition.



Модель ODA X8-2-NA — это оборудование, предназначенное для серьезных нагрузок, обладающее встроенными HA-решениями (RAC, RAC One Node), предоставляет высокую надежность и высокую производительность.

Разрешается устанавливать Oracle Database Standard Edition и Enterprise Edition на любую модель семейства. При установке Enterprise Edition заказчику предоставляется возможность выбрать необходимое ему рабочее количество ядер (иными словами, отключить неиспользуемые ядра - технология Capacity on demand) и таким образом регулировать количество лицензий СУБД Oracle. Позднее, при необходимости, заблокированные ядра всегда можно включить и таким образом увеличить производительность оборудования. Минимальное количество включенных ядер равно двум физическим ядрам процессора, шаг приращения — два физических ядра. При увеличении числа ядер заказчику следует побеспокоиться о приобретении дополнительных лицензий на СУБД Oracle. Разрешается только увеличивать количество ядер ЦПУ, уменьшать количество ядер не разрешается. Более подробно о лицензировании ODA смотрите в ODA License Guide.

Виртуализация

ODA позволяет пользователям употреблять средства виртуализации, сертифицированные для поддержки KVM. Все модели ODA обеспечивают работу СУБД Oracle и приложений, установленных в виртуальной машине в формате KVM. ODA модели X8-2-NA также поддерживает виртуальный стек основанный на Oracle VM. Поддержка виртуализации упрощает ее эксплуатацию, позволяя использовать готовые образы виртуальных машин с проинсталлированными приложениями, что придает дополнительной гибкости оборудованию семейства ODA.

Через Appliance Manager выполняется администрирование виртуальных машин в ODA. AM позволяет создавать новые VM, выполнять установку обновлений на Linux и СУБД Oracle, позволяет управлять ASM и системой хранения, мониторинг оборудования и в ряде случаев предлагает решения проблем.

Использование этой утилиты обеспечивает все стадии жизненного цикла: начальное развертывание, повседневное обслуживание, решение проблем.

Automatic Service Request

Для упрощения и ускорения решения технических проблем можно настроить Auto Service Request. ASR — это специализированное ПО, которое автоматически создает запрос в техническую поддержку Oracle при сбоях оборудования для замены вышедших из строя компонент. ASR сам соберет логи, трейсы и диагностическую информацию, относящиеся к произошедшему инциденту и отправит в техподдержку.



Продукты Oracle



Продукты для управления качеством приложений



Oracle Application Testing Suite

Oracle Application Testing Suite (OATS) — интегрированное решение Oracle для функционального и нагрузочного тестирования, а также для выполнения задач управления качеством современных бизнес-приложений.

В настоящее время бизнес-критичные веб-приложения получили широкое распространение. Они решают задачи автоматизации широкого спектра бизнес-функций. Развертывание новых веб-приложений сопряжено с определенными рисками для бизнеса. Приложения должны обеспечивать высокую производительность и доступность. В противном случае бизнес будет терять деньги из-за низкой производительности портала или из-за большого количества звонков в службу технической поддержки. Исследования показывают, что более 40 % программного обеспечения попадает в промышленную эксплуатацию с критичными дефектами. При этом исправление дефектов на этапе промышленной эксплуатации обходится до 80 раз дороже, чем на этапе разработки.

Благодаря использованию **OATS** заказчики Oracle могут развертывать веб-приложения в сжатые сроки, снижая при этом риски некорректной работы бизнес-критичных приложений.

Продукт **Oracle Application Testing Suite (OATS)** — относится к категории средств автоматизации «синтетического» тестирования т. е., в отличие от других инструментов (RAT, RUEI и т. д.), основным материалом для ATS является не реальная, а искусственно смоделированная нагрузка и имитация действий пользователей. «Синтетическое» тестирование позволяет обеспечить качество и выявить дефекты в системах еще до вывода их в реальную промышленную эксплуатацию, например: внедрение полностью нового бизнес-приложения, переход на новый релиз ERP, планируемое расширение штата сотрудников и т. д.

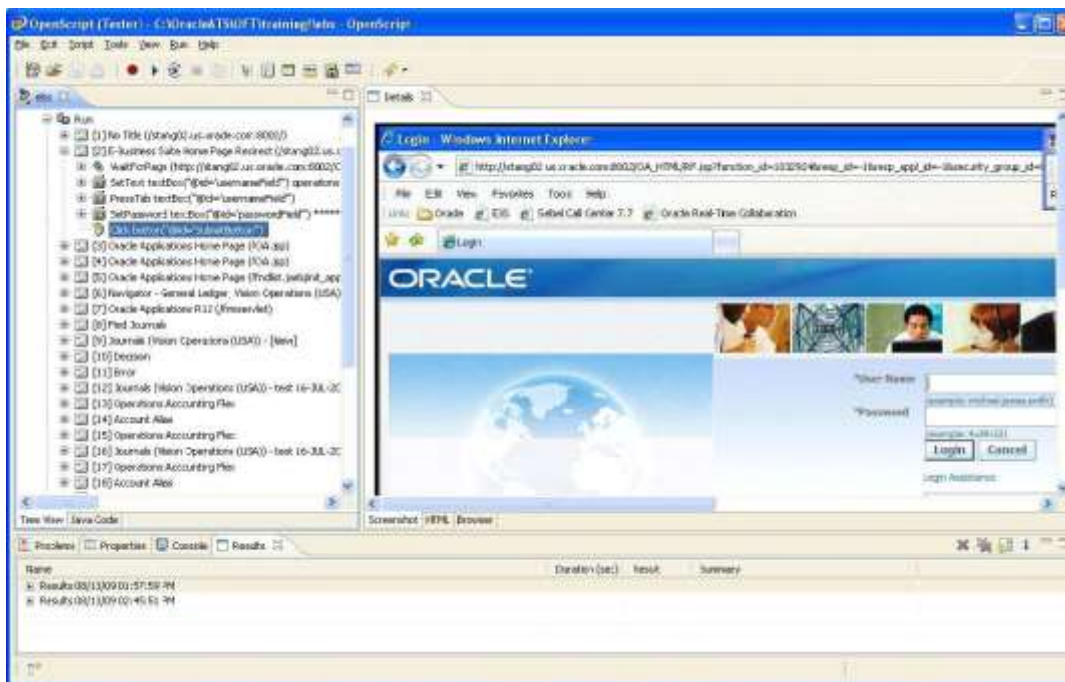
Основанное на открытых технологиях, решение OATS позволяет специалистам QA, тестировщикам, разработчикам тестов избавиться от рутинной работы при подготовке и выполнении тестов, повысить качество взаимодействия в команде и сократить сроки тестирования. В отличие от аналогичных конкурирующих решений, **OATS не** требует от тестировщиков освоения языков и навыков программирования.

Oracle Application Testing Suite (OATS) состоит из трех модулей, которые просто устанавливаются и имеют интуитивно понятный интерфейс — **Functional Testing, Load Testing и Test Manager.**

Основные возможности OATS:

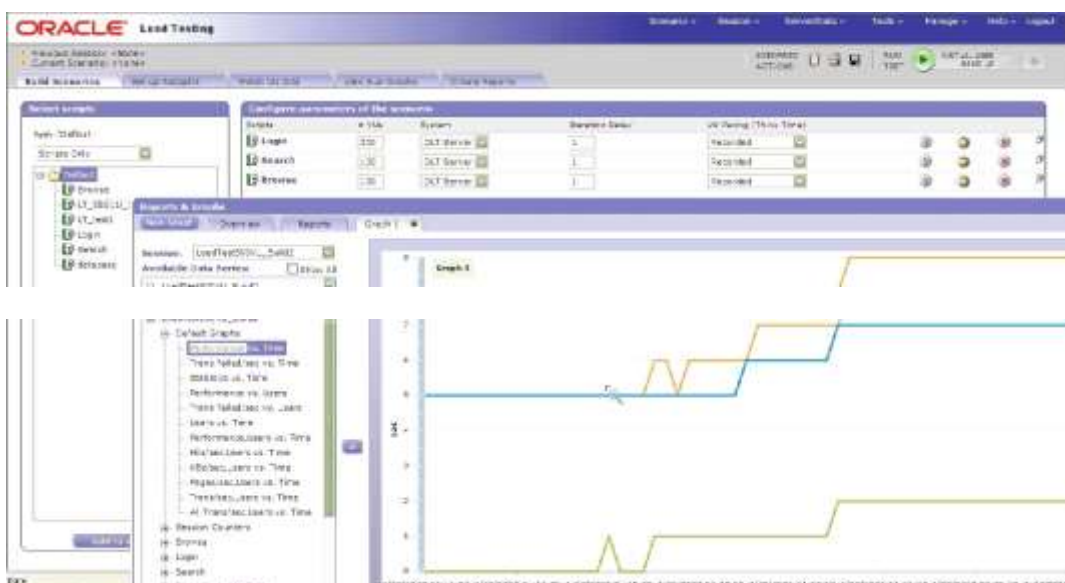
- Функциональное тестирование (**Oracle Functional Testing**)
 - Автоматизация функционального и регрессионного тестирования веб-приложений
 - Среда разработки функциональных и нагрузочных скриптов OpenScript на основе Java
 - Создание основы скрипта без программирования — генерация кода с помощью перехвата активности в браузере
 - Готовый скрипт полностью имитирует действия пользователя на странице в веб-браузере
 - Интуитивный визуальный интерфейс для построения скриптов
 - Параметризация и корреляция
 - Мощная интегрированная среда разработки Java обеспечивает расширяемость скриптов
 - Специальные библиотеки-акселераторы для готовых приложений OEBS, Siebel, технологии ADF, веб-сервисов и баз данных





- Нагрузочное тестирование (**Oracle Load Testing**)

- Генератор нагрузки
- Реалистичное нагрузочное тестирование Web, SOA, баз данных и готовых приложений Oracle
- Гибкие возможности моделирования нагрузки при помощи среды AutoPilot
- Параметризация и контрольные точки
- Одновременный мониторинг нагрузки на инфраструктуру
- Масштабируемость нагрузочного тестирования для имитации пиковых нагрузок на систему
- Консоль для отслеживания результатов совместного тестирования
- Встроенные мониторы для идентификации и устранения «узких мест» приложения
- Акселераторы для OEBS, Siebel, ADF и веб-сервисов для эффективного и оптимизированного тестирования



- Oracle Test Manager

- Управление проектом тестирования из единой консоли
- Ведение и документирование плана тестирования
- Структура тестирования: требования-тесты-проблемы
- Управление и приоритизация дефектов
- Настройка под методику тестирования, используемую у заказчика
- Простое развертывание



Преимущества OATS:

- Обеспечение успешного внедрения веб-приложений и снижения риска сбоев на этапе эксплуатации
- Повышение производительности тестировщиков за счет автоматизации рутинных операций
- Простота освоения. Использование простой унифицированной платформы тестирования OpenScript позволяет тестировщикам заниматься своими прямыми задачами, а не программированием
- Повышение качества и сокращение сроков тестирования

Функциональные лицензионные модули в составе продукта OATS:

продукт **Oracle Application Testing Suite** имеет гибкий состав и схему лицензирования, она позволяет использовать именно те функциональные модули, которые необходимы для решения конкретной задачи:

Load Testing Controller

– общая платформа для нагрузочных тестов и управления тестовыми сценариями

Load Testing

– количество «виртуальных пользователей», создающих синтетическую нагрузку

Load Testing Accelerator for Web Services

– дополнительная библиотека, используется для нагрузочных тестов веб-сервисов

Load Testing Accelerator for Oracle Database

– дополнительная библиотека, используется для нагрузочных тестов БД Oracle

Functional Testing

– среда разработки функциональных и нагрузочных скриптов OpenScript

Functional Testing Accelerator for Web Services

– дополнительная библиотека, используется для разработки тестов веб-сервисов



Test Manager

– среда для управления проектом тестирования

Таким образом, можно сформировать состав продукта под свои задачи.

Например, нагрузочный тест веб-приложения, которое использует БД Oracle и веб-сервисы, потребует лицензирования следующих модулей:

1. Load Testing Controller
2. Load Testing
3. Load Testing Accelerator for Web Services
4. Load Testing Accelerator for Oracle Database

А если стоят задачи только функционального тестирования, то достаточно лицензии на Functional Testing.

Real User Experience Insight

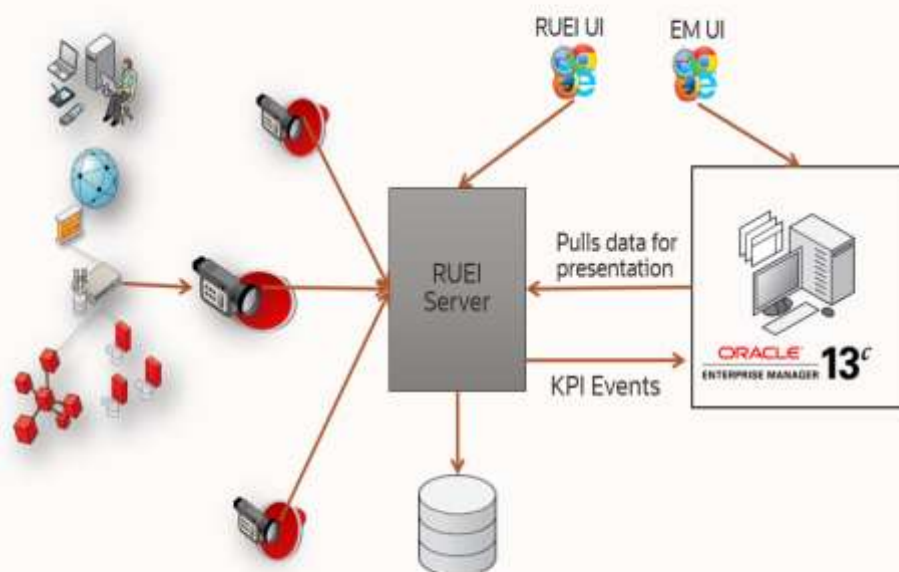
Сегодняшние бизнес-приложения используются для автоматизации и упрощения практически всех бизнес-функций современной организации. Эти приложения должны обеспечивать превосходные производительность, доступность и положительный опыт взаимодействия (user experience). В противном случае бизнес будет терять деньги из-за разочарованных пользователей, низкой производительности сотрудников и, возможно, возросших расходов на техническую поддержку, т. к. пользователи будут чаще обращаться в службу технической поддержки после прерывания их онлайн-действий. **Oracle Real User Experience Insight** — ключевой продукт в семействе решений Oracle Enterprise Manager для нисходящего управления приложениями (Top-Down Application Management), который может помочь в решении этих задач.

Oracle Real User Experience Insight (RUEI) использует передовую технологию для сбора данных, которая не требует каких-либо изменений, модификации или инструментирования контролируемых приложений. Пассивный мониторинг позволяет разворачивать RUEI в производственной среде, не требуя дорогостоящих тестов и проверки окружения.

В качестве технологии сбора исходных данных **Oracle Real User Experience Insight (RUEI)** использует «неинвазивные» методы сбора трафика сессий веб-пользователей. Компонент **RUEI Collector** использует аппаратную копию сетевого трафика, таким образом, накладные расходы на производительность минимальны. Он измеряет время отклика страниц и действий пользователей в наиболее критических точках сетевой инфраструктуры. Собранный трафик попадает на **сервер RUEI** и используется компонентом **Reporter для визуализации**. Мощные функции диагностики позволяют администраторам приложений и ИТ-специалистам выполнять анализ первопричин. RUEI имеет собственную графическую консоль, также полную интеграцию с Oracle Enterprise Manager:

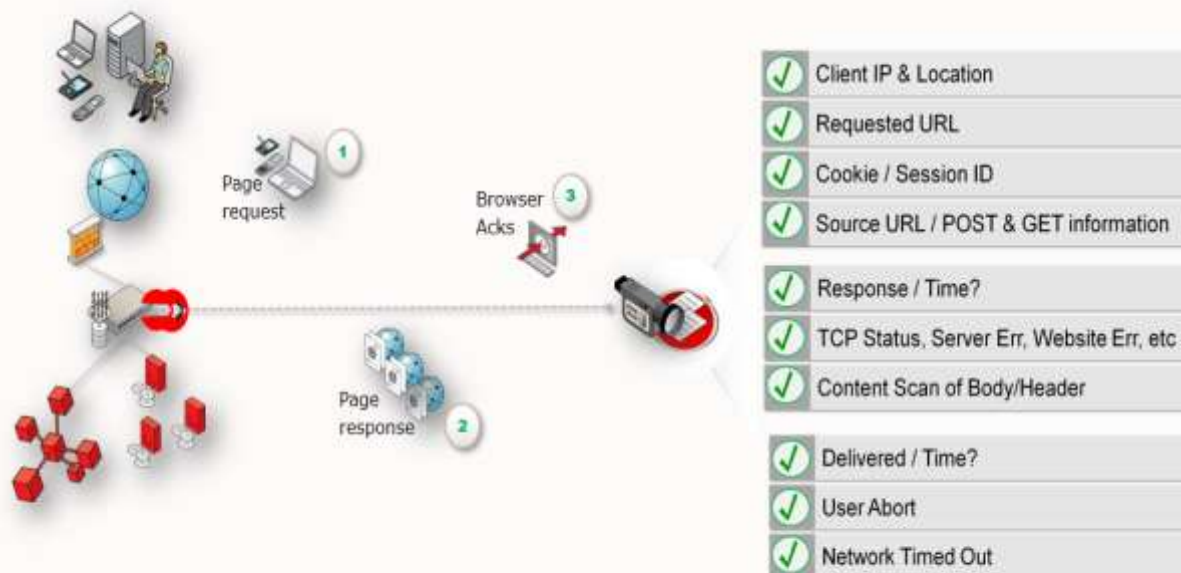


RUEI Architecture



7 Copyright © 2020, Oracle and/or its affiliates. All rights reserved.

Real User Monitoring - Network Session Capture



5 Copyright © 2020, Oracle and/or its affiliates. All rights reserved.

Oracle Real User Experience Insight предоставляет широкий набор возможностей для создания информационных панелей (**Dashboards**). Панели могут быть заранее построены администраторами и представлены определенным группам пользователей. Любая рабочая панель может быть полностью настроена для отображения информации из одного приложения. Доступ к информации, хранящейся

в интегрированном BI-хранилище данных, может быть назначен на основе комбинации пользователь/приложение.



Рис.1. Пример информационной панели приложения

В Oracle Real User Experience Insight уже встроена готовая поддержка Oracle Fusion Applications, Oracle E-Business Suite, Siebel, PeopleSoft, JD Edwards EnterpriseOne и FLEXCUBE, включая специальные для каждого приложения наборы данных и отчетность. Поддерживается все семейство продуктов Oracle Fusion Applications.

Oracle Real User Experience Insight содержит готовые модули и настройки для Oracle Applications, которые позволяют автоматически обнаруживать работающие модули приложения и транслировать сетевые объекты в удобочитаемые имена бизнес-функций. Например, модули Siebel представляются в терминах представлений, экранов и методов, а Oracle E-Business Suite представляется в терминах имен форм, блоков форм, фреймворка и имени модуля.

Oracle Real User Experience Insight позволяет Вам полностью просматривать сессии пользователей, основываясь на идентификаторе пользователя или IP-адресе за заданный период времени. Вы можете легко идентифицировать любую сессию пользователя и просмотреть все его взаимодействие с приложением с функциональной или объектной точки зрения. Возможность диагностики сессии позволяет Вам просмотреть и проиграть полную сессию так, как видел ее пользователь.



System Graph Values View Help					
Dashboard Reports Browse data KPI overview Configuration System					
» Browse data » Fusion framework suite » Session diagnostics					
Export session pages Export session data					
Session activity		Page load time (s)	Info	Object end-to-end time (ms)	Time
Applications Functional Core		8,7			00:44:39
none					
fetch					
no errors found					
Applications Functional Core		1,4			00:45:06
Manage Implementation Projects					
action					
no errors found					
Applications Functional Core		0,7			00:49:51
none					
query					
no errors found					
Applications Functional Core		0,4			00:50:54
none					
Query					

Рис. 2. Пример информационной панели приложения

Помимо просмотра всей относящейся к сессии информации в отчете, сессия может быть передана для воспроизведения в другие отделы, используя функцию экспорта. Это дает Вам возможность обмениваться реальной информацией о пользовательской сессии, например, с разработчиками, позволяя им учитывать реальное поведение пользователей и исправлять ошибки.

Full session replay

- Shop » Hitachi DH24PC3
- Shop » Cart
- Shop » UX Technology site 6
- Shop » UX Technology site 6
- Shop » Please select a Shipping Method!
- Shop » Please select a Payment Method!
- Shop » Please review the provided data and confirm the order! - UX Tech...
- Shop » UX Technology site 6
- Shop » Order Information :

Page information

Satisfied loading time

Page events

- no_name_1
- searchword :
- option : com_virtuemart
- itemid : 26
- addtocart
- product_id : 19
- quantity : 1
- submit : Add to Cart
- flypage : shop.flypage
- page : shop.cart
- manufacturer_id : 1

HTTP content

Home Blog Links Contact Us Search News Feeds FAQs Search

Home » Shop » Power Tools » Hitachi DH24PC3

Power Tools » Hitachi DH24PC3

View Full-Size Image

Hitachi DH24PC3
(Manufacturer)
Price per Unit (piece):
~~\$220.00~~ (including 8.25 % tax)
\$180.00 (including 8.25 % tax)
You Save: \$140.00

Availability

Usually ships in:

On Order

Quantity: 1

Add to Cart

Customer Reviews:

There are yet no reviews for this product.
Please log in to write a review.

You may also be interested in this/these product(s):

User id: abigailchan; Time: 04:01:11; Load time (sec): 0.7 Objects: 1; Source: Storage

VirtualMart Module

- Hand Tools
- Power Tools
- Outdoor Tools
- Indoor Tools
- Garden Tools

List All Products

Product Search

Search

Advanced Search

Show Cart

Your Cart is currently empty.

Main Menu

- Home
- Blog
- Links
- Contact Us
- Search
- News Feeds

Рис. 3. Пример воспроизведения сессии пользователя



Oracle Real User Experience Insight собирает, обрабатывает и позволяет наглядно отобразить полную и детальную информацию о взаимодействии пользователей с приложением. Захватывая пользовательский запрос к приложению и отслеживая все ответы приложения на него, Вы можете применить новый подход к управлению производительностью приложения. Он комбинирует мониторинг запросов реальных пользователей с мониторингом запросов синтетических пользователей. Отчеты и вычисление KPI могут делаться раздельно, позволяя видеть, как соотносится производительность Вашего приложения для реальных и синтетических пользователей.

Oracle Real User Experience Insight позволяет организациям получить максимальную отдачу от своих критически важных бизнес-приложений, обеспечивая детальное понимание опыта взаимодействия пользователей с приложениями. Этот продукт объединяет анализ производительности и анализ использования приложения, что позволяет бизнесу и ИТ выработать общее понимание качества работы приложения и как его улучшить.

Application Replay

Сегодняшние корпоративные приложения очень сложны и трудноуправляемы. Они состоят из многочисленных уровней, включая веб-серверы, серверы приложений и базы данных — все работающие на многих узлах. Их архитектура программного обеспечения комбинирует многочисленные независимые компоненты, такие как клиентские пользовательские интерфейсы, бизнес-логику и механизмы доступа к данным, в дополнение к протоколам клиент-сервер с сохранением состояния, работающим обычно поверх HTTP.

Так как эти структуры очень сложны, предсказать поведение всего стека в производственной среде чрезвычайно трудно. Учитывая сложность систем и отсутствие методов проверки их целиком, эффективное тестирование имеет решающее значение для успешного внедрения изменений инфраструктуры.

Oracle Application Replay Pack позволяет провести реалистичное тестирование запланированных изменений в любой части стека приложения от сервера приложений до диска, воссоздавая производственную нагрузку на тестовой системе. При помощи Oracle Application Replay Вы можете захватить нагрузку на производственной системе и воспроизвести ее на тестовой системе, сохраняя временные интервалы, параллелизм и транзакционные характеристики исходной нагрузки. Это позволяет полностью оценить влияние на работу приложения любого изменения, включая выявление новых конфликтных точек, регрессии плана выполнения SQL или нежелательных результатов. Предоставляемые всесторонний анализ и отчетность помогают идентифицировать любые потенциальные проблемы, такие как встретившиеся новые ошибки и отклонение производительности. Типы изменений, которые могут быть протестированы Oracle Application Replay Pack, включают в себя обновления серверов приложений, обновления аппаратного обеспечения, изменения операционной системы, конфигурационные изменения и т. д.

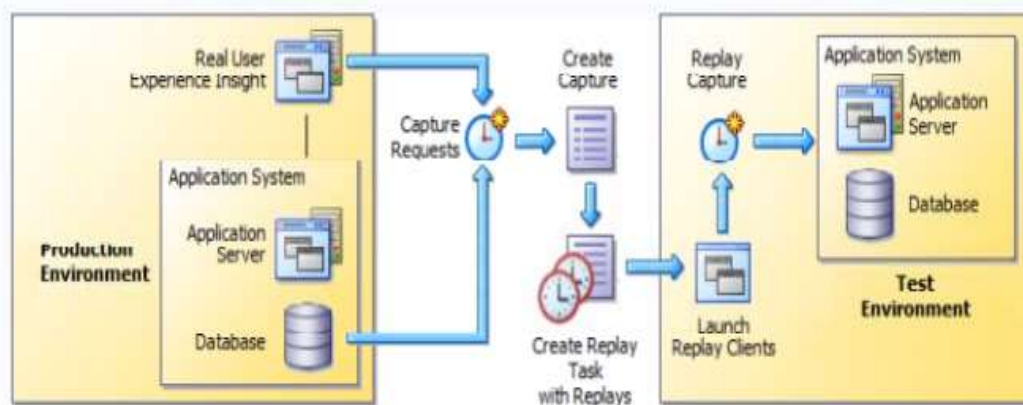


Рис. 1. Схема работы Oracle Application Replay



Захват реальной производственной нагрузки избавляет от необходимости разрабатывать скрипты для моделирования нагрузки, позволяя существенно снизить затраты и сэкономить время. При помощи Oracle Application Replay Pack реалистичное тестирование сложных приложений, на которое раньше уходили месяцы и которое делалось с применением традиционных средств моделирования загрузки, может быть проведено теперь за несколько дней. В результате Вы можете быстро протестировать запланированные изменения и внедрить новые технологии с более высокой степенью доверия и с меньшим риском.

Использование реальных нагрузок имеет несколько существенных преимуществ по сравнению с тестированием на основе синтетических нагрузок. В частности:

- Позволяет оценить, как влияет активность пользователей на всю систему целиком. Это отличается от традиционного тестирования отдельных компонентов по частям, которое предоставляет мало информации об их совместном поведении и производительности под реальной нагрузкой.
- Вместо того чтобы полагаться на predetermined сценарии, использование реальных нагрузок обеспечивает комплексное тестирование системы с помощью операций реальных пользователей. Для веб-приложений это позволяет не только исследовать все возможные способы взаимодействия пользователя с системой, но также и все возможные условия нагрузки на систему. Это необходимо, потому что системы ведут себя совершенно по-разному при разных характеристиках нагрузки (например, в зависимости от числа одновременно работающих пользователей).
- Более глубокое понимание возможных ошибок. Результаты тестов включают данные для каждого уровня стека, и они могут быть соотнесены между различными уровнями. Это помогает проверить корректность выполнения теста, позволяя выявить ошибки, проверяя на ошибки или неожиданные ответы сервера.

Oracle Application Replay предоставляет всесторонний анализ, сводные и детализированные отчеты с информацией об ошибках, производительности и расхождении данных, помогающие понять, как воспроизведение нагрузки вело себя по сравнению с захватом или с другим воспроизведением. Широкий набор отчетов позволяет быстро идентифицировать проблемные области и помогает определить план действий по их устранению.



СВЯЖИТЕСЬ С НАМИ

Позвоните по номеру +7 (495) 641-14-00 или зайдите на сайт oracle.com/ru.

За пределами Северной Америки ближайший к вам офис компании можно найти на странице oracle.com/contact.



blogs.oracle.com/russia/



facebook.com/Oracle.Russia



twitter.com/OracleRU

© Oracle и/или аффилированные компании, 2020. Все права защищены. Данный документ предоставляется исключительно в информационных целях, и его содержание может меняться без уведомления. Документ может содержать ошибки, и на него не распространяются никакие гарантии или условия, выраженные устно или предусмотренные законодательством, включая подразумеваемые гарантии товарного состояния и соответствия определенным целям. Oracle не несет никакой ответственности в связи с этим документом. Документ также не создает никаких договорных обязательств прямо или косвенно. Воспроизведение или передача этого документа в любой форме, любым способом (электронным или физическим) и для любой цели возможны только с предварительного письменного разрешения Oracle.

Oracle и Java являются зарегистрированными товарными знаками корпорации Oracle и/или ее дочерних компаний. Другие названия могут быть товарными знаками соответствующих владельцев.

Intel и Intel Xeon являются товарными знаками или зарегистрированными товарными знаками компании Intel Corporation. Все товарные знаки SPARC используются по лицензии и являются товарными знаками или зарегистрированными товарными знаками SPARC International, Inc. AMD, Opteron, логотипы AMD и AMD Opteron являются торговыми марками или зарегистрированными торговыми марками компании Advanced Micro Devices. UNIX является зарегистрированным товарным знаком компании The Open Group. 0120

Oracle Autonomous Database Strategy
Июль, 2020 г.

