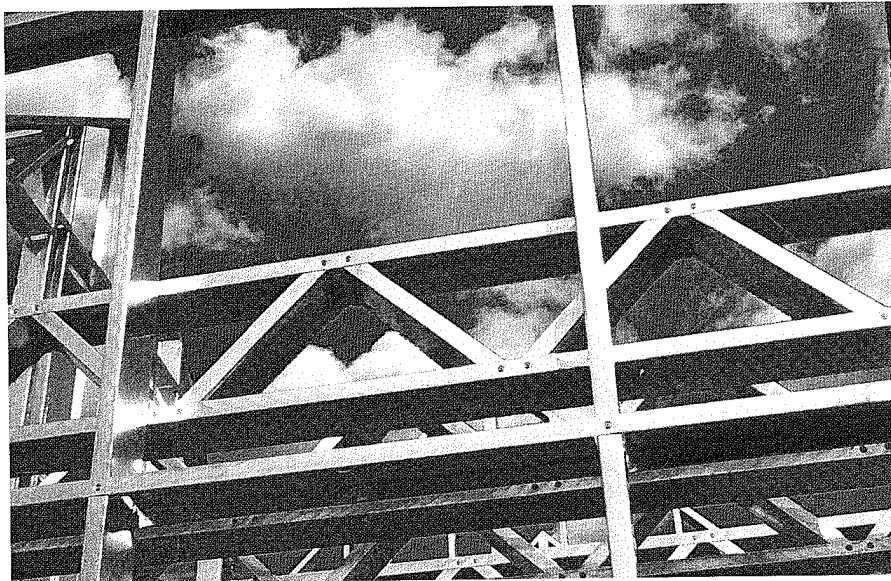


# СУБД для облаков

В июле 2013 года компания Oracle выпустила очередную версию СУБД Oracle Database 12c, за новым суффиксом в названии которой скрывается около 500 возможностей, открывающих перед пользователями и разработчиками дополнительные перспективы при работе с облаками.

Ключевые слова: мультиарендность, оптимизация хранения, безопасность данных, управляемость, облачная платформа



Марк Ривкин, Игорь Мельников

При обозначении версий СУБД Oracle обычно применяются суффиксы для указания их предназначений. Например, (i.g) — платформа для интернет- и грид-приложений, а недавно появился суффикс (c) в версии Oracle Database 12c, указывающий на «облачность» версии (cloud). Одновременно оказалось, что интересные новшества этой СУБД тоже начинаются с буквы «с»: консолидация баз данных (consolidation), средства автоматического сжатия данных (compression) и повтора прервавшихся транзакций (continuity).

## КОНСОЛИДАЦИЯ И МУЛЬТИАРЕНДНОСТЬ

Сегодня чаще всего каждая база размещается на отдельном компьютере, и сложность администрирования их множества растет пропорционально увеличению их количества. Консолидация баз данных на меньшем числе компьютеров увеличит эффективность и, возможно, облегчит проблему администрирования.

Самый простой и популярный сегодня способ консолидации — виртуализация,

например, за счет замены физической машины на виртуальную, когда несколько виртуальных машин развертываются на одном мощном сервере. Однако количество администрируемых объектов (виртуальных машин, баз данных) при этом не уменьшается, а при размещении СУБД внутри виртуальной машины, которая работает на физической машине, возникают дополнительные накладные расходы на слой виртуализации между СУБД и оборудованием. Кроме того, каждая виртуальная машина требует выделения большого количества ресурсов (память, процессоры, ввод-вывод) физического компьютера.

Другой вариант консолидации реализуется при развертывании на одной физической машине нескольких баз данных, работающих одновременно. Этот подход используется для облаков, и в нем нет накладных расходов, присущих виртуализации, степень консолидации выше, но и тут есть свои недостатки.

Третий вариант — консолидация схем из разных баз данных в одной. Это пока самый эффективный способ, но он и сложнее, так как схемы конфликтуют между собой по имени объектов, используют разные

варианты процедур, требуют разных параметров настройки СУБД и т. д.

В Oracle Database 12c реализован еще один вариант консолидации, потребовавший изменения всей архитектуры СУБД, и теперь у пользователя есть выбор — вернуть СУБД по старой традиционной архитектуре или реализовать мультиарендную (multitenant) архитектуру.

Каждая база данных Oracle — это набор файлов на дисках; для работы с базой запускается один (в случае кластера — несколько) экземпляр СУБД, который получает SQL-запросы, выполняет их и возвращает результат. При запуске каждый экземпляр забирает себе область оперативной памяти и запускает набор процессов ОС, использующих эту общую разделяемую область памяти и обеспечивающих работу базы. Однако, в отличие от того варианта, когда на одном физическом компьютере размещается несколько баз, в мультиарендной архитектуре (рис. 1) для консолидации создается одна *контейнерная база данных* (Container Database, CDB). Все консолидируемые базы помещаются в этот контейнер, а один экземпляр Oracle обслуживает все базы в нем. При таком подходе ресурсы компьютера используются эффективнее, причем базы можно не только помещать в контейнерную, но и извлекать из нее и переносить в другие контейнерные базы (*подключаемые базы* назвали Pluggable Database, PDB). После помещения в контейнерную базу все PDB остаются изолированными и независимыми — приложения, которые работают с ними, менять не надо.

Мультиарендная архитектура позволяет управлять множеством баз внутри контейнера как единым целым. Если, например, у вас было 100 разных баз, которые консолидированы в одну контейнерную, то теперь можно делать не 100 резервных копий, а одну для контейнерной базы, а если надо обеспечить надежность работы приложений за счет создания резервных баз, то можно создать одну резервную базу для контейнерной.