Статья Oracle,
июнь 2013 г.

Oracle Real Application Clusters (RAC)

Краткий обзор 3
Oracle Real Application Clusters – обзор 2
Непрерывность бизнеса и высокая доступность 5
Масштабируемость и гибкость 6
Эффективное по стоимости управление балансировкой нагрузки 7
Стандартизация развертывания и управление системами 11
Новые возможности Oracle RAC 12
Поддержка Oracle Multitenant в Oracle RAC 12
Application Continuity и Transaction Guard 13
Oracle Flex ASM 15
Функция «What-If» для эмуляции выполнения команды 17
Управление базами данных и кластером на основе политик 18
Итоги и заключение 20

Краткий обзор

Непрерывность бизнес-процессов, высокая доступность, масштабируемость, гибкость и адаптивность в сочетании с простотой управления — основы успешного развертывания облачной ИТ-инфраструктуры. Уже в течение более чем десяти лет решение Oracle Database with Real Application Clusters (RAC) — предпочтительное решение для баз данных тысяч клиентов Oracle. Oracle Real Application Cluster 12*c* продолжает этот успех, обеспечивая значительные дополнительные возможности во всех областях, важных для успешного бизнеса.

Решение Oracle RAC, исходной целью которого было улучшение работы базы данных, с годами развивалось и теперь основано на стеке продуктов высокой доступности (High Availability), который может использоваться в качестве основы для облачной СУБД, а также для общей инфраструктуры, способной обеспечить высокую доступность, масштабируемость, гибкость и адаптивность любого приложения в вашем Центре Обработки Данных.

В этой статье обсуждаются преимущества Oracle RAC и основные новшества на каждом уровне последней версии стека Oracle RAC, которые обеспечивают еще более высокий уровень доступности приложений, использующих базу данных Oracle RAC. Вот лишь некоторые основные преимущества Oracle RAC, обсуждаемые в этой статье:

- Непрерывность бизнеса и высокая доступность
- Масштабируемость и гибкость
- Эффективное по стоимости управление балансировкой нагрузки
- Стандартизация развертывания и управление системами

Oracle Real Application Clusters – обзор

СУБД Oracle Database с опцией Oracle Real Application Clusters (RAC) дает возможность запускать несколько экземпляров баз данных на разных серверах в кластере для одного и того же набора файлов данных, именуемых также базой данных. База данных обслуживается на нескольких аппаратных системах, но воспринимается приложением как единая база данных. Это дает возможность использовать недорогое универсальное оборудование для снижения совокупной стоимости владения (TCO) и создания масштабируемой вычислительной среды для различных типов приложений. Oracle RAC — технология номер один Oracle для кластеризации базы данных с общими дисками.

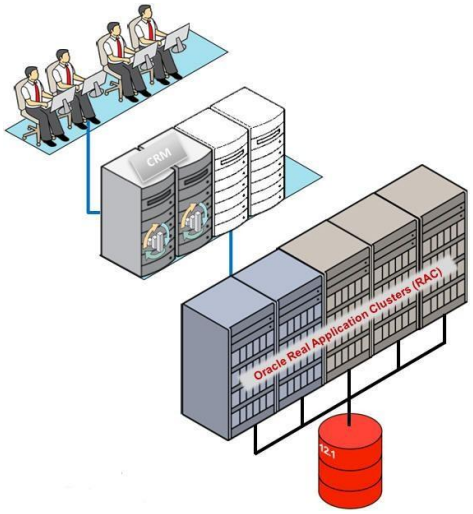

Рис. 1. Общий вид Oracle RAC 12*c* и Oracle RAC Stack

Стек Oracle RAC

До установки каталога базы данных Oracle RAC необходимо установить в системе ПО Oracle Clusterware (OCW). Oracle Clusterware (OCW) — неотъемлемый компонент пакета продуктов Oracle Grid Infrastructure (GI), который включает Automatic Storage Management (ASM) и Oracle Cloud File System (CloudFS).

Oracle Grid Infrastructure с Oracle ASM/CloudFS и Oracle Clusterware, а также СУБД Oracle Database с опцией Oracle Real Application Clusters (RAC) составляют стек Oracle RAC.

Oracle Grid Infrastructure — это основа баз данных Oracle RAC, использующих ПО Oracle Clusterware для автоматического размещения ресурсов, их настройки и управления, а также Oracle Automatic Storage Management для эффективного и надежного управления хранением данных.

Oracle Clusterware

Oracle Clusterware — технология | обеспечивающая совместную работу фермы серверов в виде единого кластера. Кластер определяется как группа независимых, но соединенных между собой серверов, работающих совместно как единая система. Oracle Clusterware — это «интеллект» такой системы, обеспечивающий совместную работу.

Программное обеспечение Oracle Clusterware впервые появилось в Oracle Database 10g Release 1 как базовое ПО кластеризации, необходимое для работы Oracle Real Application Clusters (RAC). Как часть стека Oracle RAC, ПО Oracle Clusterware также используется кластерной версией Oracle ASM и тесно интегрировано в стек Oracle RAC.

Oracle Clusterware — комплексное, бесплатное решение кластеризации, которое может быть использовано и без Oracle RAC. В таких средах ПО Oracle Clusterware обычно используется для автоматического размещения ресурсов, корректировки и автоматического управления ресурсами для приложений любого типа. В обеих средах ПО Oracle Clusterware отвечает за поддержание вхождения узлов в кластер и механизм исключения «поврежденного» узла из кластера (fencing).

Новое в Oracle Clusterware 12*c*

В зависимости от требований к развертыванию кластера и управлению ПО **Oracle Clusterware 12*c*** может быть установлено одним из следующих трех способов. 1) Стандартный кластер (типовой кластер Oracle RAC, сконфигурированный как в предыдущих версиях до Oracle Clusterware 12*c*); 2) Oracle Flex Cluster (новая кластерная топология, доступная с Oracle Clusterware 12*c* и сочетающая в едином кластере традиционные, жестко связанные серверы и новые, слабо связанные серверы) и 3) Application Cluster (установка, адаптированная для приложений, не являющихся приложениями базы данных)[1].

Oracle Flex Cluster — инновационное решение для обеспечения высокой доступности на всех уровнях, предназначенное для глобального управления ресурсами для комплексных облачных окружений, критически важных для бизнеса.

ПО Oracle Automatic Storage Management (ASM)

Поскольку Oracle RAC использует архитектуру с общим диском, средства управления томами и файловую систему, используемые для хранения данных базы данных, должны быть доступны всем узлам кластера. Oracle Automatic Storage Management (ASM) — рекомендуемый (кластерный) менеджер томов для СУБД Oracle Database.

Oracle ASM управляет всеми типами данных: файлами баз данных Oracle, файлами Oracle Clusterware и такими неструктурированными данными, как двоичные файлы, внешние файлы и текстовые файлы (через Oracle CloudFS). Благодаря низкой стоимости, простоте администрирования и высокой производительности решение Oracle ASM является предпочтительной технологией для управления хранением данных в базах данных Oracle.

[1] Дополнительные сведения об Oracle Clusterware доступны по адресу www.oracle.com/goto/clusterware

В целях увеличения производительности и высокой доступности, в ПО Oracle ASM используется принцип полного чередования и зеркалирования данных (stripe and mirror everything). Интеллектуальные средства зеркалирования дают возможность администраторам определять двойное, либо тройное «зеркало» — для защиты важных данных. Если во время операции чтения обнаруживается поврежденный блок, Oracle ASM автоматически переносит правильный блок с зеркальной копии в неповрежденную часть диска.[2]

Новое в Oracle ASM 12*c*

В Oracle ASM 12*c* представлена технология Oracle Flex ASM — новая модель развертывания Oracle ASM, которая увеличивает доступность экземпляров базы данных и снижает потребление ресурсов, связанное с Oracle ASM. Oracle Flex ASM упрощает консолидацию баз данных на основе кластера, так как гарантирует продолжение работы экземпляров Oracle Database 12*c*, работающих на конкретном сервере, в случае отказа экземпляра Oracle Flex ASM на этом сервере.

Файловая система Oracle Cloud File System (CloudFS)

Файловая система Oracle Cloud File System упрощает и автоматизирует функции управления хранением данных и повышает коэффициент использования системы хранения, работоспособность и гибкость системы, обеспечивая предсказуемую производительность и доступность для всех обычных файлов и файлов базы данных. CloudFS включает менеджер томов и кластерную файловую систему.

- Oracle ASM Dynamic Volume Manager (ADVM) — универсальный кластерный менеджер томов для Oracle ACFS, а также сторонних файловых систем.
- Oracle ASM Cluster File System (ACFS) — универсальная кластерная файловая система, совместимая с POSIX и Windows и обеспечивающая такие современные функции, как моментальные копии, репликация, шифрование, безопасность и т.д.

Новое в Oracle Cloud File System 12*c*

С Oracle Grid Infrastructure 12*c* файловая система Oracle Cluster File System, известная также как Oracle ACFS, обеспечивает еще большую гибкость. Помимо использования Oracle ACFS для хранения неструктурированных данных (файла конфигурации и файлов данных для конкретных приложений), эта файловая система может теперь использоваться для файлов СУБД Oracle RAC, а также для разделяемых каталогов выполняемых файлов СУБД Oracle Database (каталог ORACLE_HOME) и наборов резервных копий.

[2] Дополнительные сведения об Oracle ASM и файловой системе Oracle Cloud File System доступны по адресу www.oracle.com/goto/asm

Непрерывность бизнеса и высокая доступность

Oracle Real Application Clusters (RAC) является основой для высокой доступности в ЦОД. На основе инфраструктуры Oracle Grid Infrastructure, стек Oracle RAC обеспечивает среду для управления плановыми и внеплановыми простоями и позволяет добавлять ресурсы по требованию, гарантируя бесперебойную работу приложений любого типа.

Oracle RAC является неотъемлемым компонентом Архитектуры максимальной доступности Oracle (Oracle MAA), включающей в себя лучшие практики для обеспечения высокой доступности ЦОД. В этом контексте, одним из основных преимуществ Oracle RAC является изначально заложенная отказоустойчивость, которая обеспечивается тем, что база данных использует несколько физических серверов. Серверы в кластере работают независимо друг от друга, поэтому отказ одного или нескольких серверов не затрагивает другие серверы. Эта архитектура также позволяет прозрачно перевести группу серверов в offline-режим, в то время как остальная часть системы продолжает бесперебойную работу и предоставление сервиса базы данных.

Oracle RAC обеспечивает следующие основные характеристики, важные для управления данными с высокой доступностью.

- *Надежность.* СУБД Oracle Database известна своей надежностью. Oracle RAC идет на шаг дальше: сервер базы данных перестает быть компонентом, отказ которого приводит к отказу всей системы. В случае отказа одного экземпляра остальные экземпляры в кластере остаются открытыми и активными. Oracle Clusterware отслеживает все процессы Oracle и немедленно перезапускает любой отказавший компонент.
- *Выявление ошибок.* Oracle Clusterware автоматически отслеживает базы данных Oracle RAC, а также все остальные процессы Oracle (Oracle ASM, экземпляры, прослушивающие процессы и т.д.) и обеспечивает быстрое обнаружение проблем внутри этой среды. Это ПО также автоматически выполняет восстановление после сбоев — часто даже раньше, чем этот сбой почувствуют пользователи.
- *Восстанавливаемость.* В СУБД Oracle Database содержится много функций, облегчающих восстановление базы данных после всех видов сбоев. В случае отказа одного экземпляра в базе данных Oracle RAC, об этом становится известно другому экземпляру в кластере, и восстановление запускается автоматически. Технологии Fast Application Notification (FAN), Fast Connection Failover (FCF) и особенно Application Continuity в Oracle RAC 12*c* дают возможность скрыть сбой любого компонента от пользователя.
- *Непрерывность операций.* Oracle RAC обеспечивает непрерывное обслуживание как при запланированных простоях, так и при аварийных отключениях. В случае отказа сервера (или экземпляра) база данных остается открытой и приложения по-прежнему имеют доступ к данным, что дает возможность завершить важные задачи для бизнеса, чаще всего без задержки в оказании услуг.
    - Большинство операций по обслуживанию баз данных могут выполняться без простоев и незаметно для пользователя. Многие задачи обслуживания могут выполняться поочередно (rolling), благодаря чему сокращается или полностью исключается время простоя приложений.

Новое в Oracle Database 12*c*

Новшеством в Oracle Database 12*c* является возможность внести вклад в успех бизнеса, обеспечив работу критически важных бизнес-приложений даже при отказах на системном уровне.

Application Continuity — новая технология, защищающая приложения от сбоев экземпляров и сессий путем повтора транзакций, которые выполнялись во время сбоя, на другом экземпляре базы данных.

Технология Application Continuity использует новую функцию **Oracle Database 12*c* Transaction Guard**, благодаря которой приложения всегда получают из базы данных «известный результат» (ожидаемый результат) даже при сбое на уровне базы данных и в случае, если пользователь запустил «неправильную операцию» — дважды щелкнул на кнопке «Купить» в web-магазине при покупке или нажал кнопку «Назад» в браузере при оформлении заказа. Без этой функции результат таких операций может быть неожиданным. Application Continuity требует драйвер клиентского доступа Oracle Database 12*c* , а API-интерфейс Transaction Guard может использоваться разными приложениями напрямую.

Масштабируемость и гибкость

Решение Oracle Real Application Clusters располагает уникальной технологией для масштабирования приложений. В прошлом, когда мощности серверов баз данных не хватало, эти серверы заменялись новыми, более мощными. С ростом мощности серверов, также увеличивается цена серверов.

Для баз данных, использующих Oracle RAC, существуют альтернативные способы увеличения вычислительной мощности. Приложения, которые традиционно эксплуатировались на больших серверах SMP, могут быть перенесены для выполнения в пулах небольших серверов в кластере. Возможен и другой вариант, когда можно по-прежнему использовать уже имеющееся оборудование и добавить новые серверы в пул (или сформировать новый пул) без простоев.

Все серверы в кластере должны работать с одной операционной системой и версией Oracle, но их мощность не обязательно должна быть одинаковой. В настоящее время пользователи Oracle RAC часто используют пулы серверов, соответствующие их потребностям, и серверы в этих пулах имеют (слегка) отличающиеся характеристики.

Благодаря расширенному использованию пулов серверов и определенных функций Oracle Database 12*c* можно размещать нагрузку разной важности на серверах разной мощности и, тем самым, выполнять требования Соглашений об уровне услуг (SLA) даже в условиях колебаний нагрузки. Таким образом архитектура Oracle RAC быстро и автоматически адаптируется к новым бизнес-требованиям и соответствующим изменениям в уровне нагрузки без вмешательства пользователя.

Для динамического распределения рабочих нагрузок приложений, пользователи приложений или клиенты серверов приложений должны подключиться к СУБД Oracle Database с помощью динамических сервисов баз данных (Dynamic Database Services). Использование динамических сервисов баз данных дает возможность Oracle Database автоматически балансировать нагрузку между серверами в пуле, назначая сервисы, выполняющие нагрузку пользователей, одному или нескольким экземплярам базы данных.

Администратор базы данных (АБД) может гибко выбирать, назначить конкретную нагрузку от приложения всем экземплярам базы данных или какому-либо подмножеству экземпляров базы данных в кластере. Администраторы также могут легко добавлять вычислительную мощность по мере роста требований приложения.

Технология Oracle Cache Fusion, интегрированная в базу данных Oracle RAC, немедленно использует ресурсы ЦП и памяти новых узлов кластера. Это делается автоматически без необходимости повторно секционировать данные вручную.

Эффективное по стоимости управление балансировкой нагрузки

Oracle RAC включает инновационные технологии для управления нагрузками в кластере и обеспечивает наилучшую пропускную способность приложений благодаря конфигурации и высокой доступности приложений. Таким образом, в настоящее время Oracle RAC является единственным решением для баз данных на рынке, обеспечивающим постоянную эксплуатационную доступность.

Эксплуатационная доступность, помимо высокой доступности и управления производительностью, означает, что приложениям обеспечивается постоянный доступ к сервисам баз данных без заметного изменения времени отклика или общего качества сервиса.

Различные компоненты, интегрированные в стек Oracle RAC, помогают обеспечить эксплуатационную доступность СУБД Oracle. В отличие от других решений на рынке, эти компоненты и функциональность являются частью стека и могут использоваться без дополнительных затрат.

Введение в пулы серверов

Во Oracle Grid Infrastructure 11g Release 2 пулы серверов используются как метод логического разделения кластера на группы серверов, предоставляющих службы базы данных или приложений. Масштабируемость и доступность этих баз данных и приложений регулируются свойствами пулов серверов. Каждому пулу серверов может быть задан минимальный или максимальный размер (количества серверов в пуле) для управления масштабируемостью. Управлять распределением ресурсов между пулами серверов и доступностью ресурсов, размещенных в пулах серверов, можно назначая уровень приоритета.

Oracle Clusterware назначает серверы для пользовательских пулов во время изменения настроек кластера. Oracle Clusterware назначает серверы исходя из уровня приоритета. Количество экземпляров, поддерживаемое для базы данных, определяется мощностью пула серверов.

Введение в динамические службы базы данных

ПО управления нагрузками (Workload Management) использует функцию Services (сервисы) СУБД Oracle Database. Функция Services скрывает сложность базы данных Oracle RAC, обеспечивая единый образ системы для управления нагрузкой.

Чтобы управлять нагрузками или группой приложений, можно определить динамические сервисы базы данных (Dynamic Database Services) и назначить их конкретному приложению или поднабору операций приложения. Это позволяет разделить рабочие нагрузки от приложений на управляемые компоненты в зависимости от таких бизнес-требований, как уровни обслуживания и приоритеты. Например, для подключения к одной и той же базе данных пользователи, работающие в сети, могут использовать один сервис, для пакетной обработки может использоваться другой сервис, а для отчетности — третий.

Сервис может работать на одном или нескольких экземплярах базы данных Oracle, а один экземпляр может поддерживать несколько сервисов. Управлять количеством экземпляров, предлагающих определенный сервис, может администратор базы данных, или это делается автоматически на основе политик независимо от приложения. В случае сбоев, сервисы автоматически восстанавливаются на работающих экземплярах.

Сервисы интегрированы с большим количеством функций СУБД Oracle Database. Например, в зависимости от сервиса пользователи приложения могут быть автоматически приписаны к группе потребителей Resource Manager для ограничения использования ими таких ресурсов, как ЦП. Пакетные задания могут быть приписаны к определенному классу заданий в зависимости от их сервиса.

Использование сервисов также обеспечивает прозрачность местонахождения для очередей запросов, если используется технология Oracle Streams Advanced Queuing, а обработку параллельных запросов между узлами можно ограничить экземплярами, на которых размещена служба.

Oracle Database Quality of Service (QoS) Management можно использовать, чтобы рекомендовать или предоставить сервис другой группе потребителей в плане менеджера ресурсов Oracle Database Resource Manager — например, предоставить рабочей нагрузке, использующей сервис, более частый доступ к ресурсам ЦП.

Введение в SCAN

Функция Single Client Access Name (SCAN) выступает в качестве псевдонима кластера для баз данных в кластере и впервые появилась Oracle Grid Infrastructure 11g Release 2. Преимущество этой функции в том, что не нужно менять настройки для подключения клиента при добавлении или удалении узлов кластера. Если для доступа в кластер используется единое имя, приложения могут использовать клиенты, совместимые с EZConnect, и URL-адрес тонкого JDBC-драйвера для доступа к любой базе данных в кластере независимо от того, на каком сервере или серверах в кластере она работает в этот момент.

Балансировка нагрузки соединений

ПО Oracle Services обеспечивает балансировку нагрузки соединений при подключении баз данных на уровне Oracle Net. Балансировка нагрузки на стороне клиента, распределяющая запросы на подключение между всеми прослушивающими SCAN -листенерами в кластере, достигается путем указания SCAN-имени в списке адресов в строке подключения клиента. SQL*NET произвольно выбирает один из IP-адресов SCAN-листенера-. Если выбранный сервер недоступен, производится попытка использовать следующий сервер в списке.

Балансировка нагрузки на серверной стороне достигается с помощью SCAN-листенеров. Каждому SCAN-листенеру известны все экземпляры в кластере, предоставляющие каждый сервис. А зависимости от правила балансировки нагрузки (по количеству сессий, либо по загрузке ЦПУ), заданного для каждого сервиса, прослушивающий процесс выбирает экземпляр, наиболее удовлетворяющий этому правилу, и подключение направляется на этот экземпляр через локальный прослушивающий процесс.

Fast Application Notification

Компонент Fast Application Notification (FAN) обеспечивает интеграцию между базой данных Oracle RAC и приложением. Благодаря этому компоненту приложение всегда осведомлено о текущей конфигурации пула кластеров и подключается только к тем экземплярам, которые на данный момент способны отвечать на запросы приложений.

Стек Oracle RAC немедленно публикует каждое событие FAN, если изменяется состояние внутри кластера, например: один из узлов кластера выключился по аварийной ошибке оборудования. Клиенты использующие FAN, получают информацию об этих событиях и могут немедленно на них реагировать. В случае события DOWN простой работы приложения сокращается путем удаления соединений с экземпляром, в котором произошел сбой, а текущие операции прерываются с сообщением об ошибке и возвращаются приложению.

Новый компонент Application Continuity использует FAN для повтора транзакций на которых был сбой в базе данных, после исправимой ошибки, из-за которой сеанс становится недоступным. Повтор производится быстро и без прерывания работы.

Операция пользователя в БД может содержать транзакционные и нетранзакционные вызовы базы данных. После успешного повтора приложение может продолжить работу с момента, когда сеанс базы данных был прекращен.

В случае событий UP создаются новые соединения, чтобы приложение могло немедленно воспользоваться дополнительными ресурсами. Клиенты Oracle JDBC, Oracle Universal Connection Pool, ODP.NET и OCI интегрированы с FAN. Приложения с собственными пулами соединений могут воспользоваться преимуществами FAN, используя API-интерфейс Oracle RAC FAN, доступный с JDBC -драйвером Oracle Database 11g Release 2, или используя функцию обратного вызова Oracle Call Interface. Как правило, Oracle рекомендует использовать Oracle Universal Connection Pool как стандарт для подключения к СУБД Oracle Database.

Консультативная справка по балансировке нагрузки (Load Balancing Advisory)

Механизм балансировки нагрузки распределяет работу между всеми доступными экземплярами базы данных Oracle RAC. Oracle рекомендует, чтобы приложения использовали пулы соединений с постоянными соединениями, охватывающими экземпляры с конкретной службой. Если используются постоянные соединения, соединения создаются нечасто и существуют в течение большого периода времени. Нагрузка поступает в систему часто, получает эти из пула соединения и существует в течение относительно малого периода времени.

Функция Load Balancing Advisory дает рекомендации о том, на какой узел кластера направить входящую нагрузку, обеспечивающий наилучшее качество обслуживания для такой нагрузки. Таким образом необходимость последующего перенаправления работы сводится к минимуму.

Благодаря использованию динамической балансировки нагрузки — Load Balancing Advisory и определению правил балансировки нагрузки приложение получает встроенный механизм обратной связи к изменениям распределения нагрузки между узлами кластера. Нагрузка направляется таким образом, чтобы обеспечить лучшее время обслуживания в рамках всего кластера с учетом изменения условий в системе. В устойчивом состоянии система близка к равновесию с улучшенной пропускной способностью на всех экземплярах Oracle RAC.

Load Balancing Advisory встроен в основное клиентское ПО Oracle, такие как: Listeners (прослушивающий процесс), Universal Connection Pool, Oracle WebLogic Server Active GridLink для Oracle RAC и пулы соединений ODP.NET. Сторонние приложения также могут быть подписаны на получение информации о событиях Load Balancing Advisory, используя для этого JDBC и API-интерфейс Oracle RAC FAN или с помощью определяемых на стороне приложения функций обратного вызова с интерфейсом Oracle Call Interface (OCI).

Новое в Oracle Database 12*c*

Технология SCAN в Oracle Grid Infrastructure 12*c* была расширена в следующих целях: 1) для поддержки IP-адресов IPv6; 2) для поддержки нескольких подсетей в кластере и 3) для ограничения регистрации служб.[3]

Начиная с Oracle Grid Infrastructure 12*c*, узлы кластера могут быть настроены для использования IP-адресов IPv4 или IPv6 для виртуальных IP-адресов (VIP) в общедоступной сети кластера

[3] Сведения о SCAN доступны по адресу www.oracle.com/technetwork/products/clustering/overview/scan-129069.pdf

можно определить более одной общедоступной сети. Клиенты и приложения базы данных могут подключаться к адресам IPv4 или IPv6 VIP. SCAN-листенер автоматически перенаправляет подключения клиентов к соответствующему прослушивающему процессу в данной подсети с учетом IP-протокола, запрошенного клиентом, и SCAN-листенер может быть определен для каждой подсети в кластере.

Прослушивающие процессы, управляемые Oracle Grid Infrastructure, могут быть настроены так, чтобы ограничить доступ клиентов к базе данных, которая зарегистрирована в прослушивающем процессе, с использованием различных условий — например, подсети, из которой подключаются эти клиенты. Та же функция может использоваться для ограничения регистрации экземпляров базы данных в определенном SCAN-листенере как части динамической регистрации экземпляра. Это делается для предотвращения случайной регистрации экземпляра базы данных в прослушивающих процессах в общедоступной среде.

Начиная с Oracle Database 12c, продукт Oracle Database Quality of Service (QoS) Management входит в стандартную лицензию Oracle RAC и может использоваться для управления на основе политик всеми базами данных Oracle RAC, которые поддерживают пулы серверов (для этого потребуются Oracle Grid Infrastructure и Oracle Database 11g Release 2 (или поздней версии)).

Oracle Database Quality of Service Management — автоматизированная функция управления на основе политик. Она отслеживает запросы рабочих нагрузок по всей системе, управляет ресурсами, общими для приложений, и корректирует настройки системы, чтобы приложения работали с требуемыми уровнями производительности. Эта функция гибко приспосабливается в изменениям конфигурации системы и уровня нагрузки, помогая избежать колебаний в уровнях производительности приложений.[4]

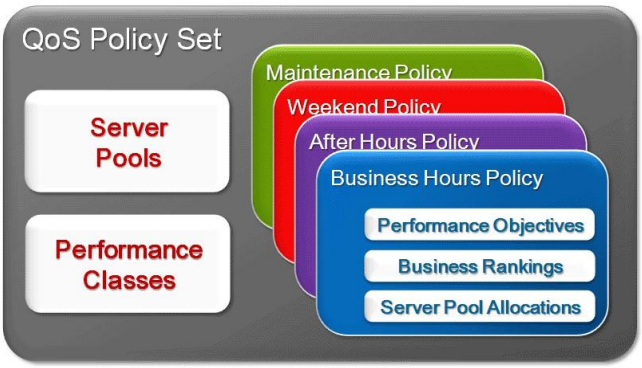

Рис. 2. Общий вид Oracle Quality of Service (QoS) Management

[4] Дополнительные сведения о Quality of Service (QoS) Management доступны по адресу http://www.oracle.com/technetwork/products/clustering/overview/qosmanageent-508184.html

Стандартизация развертывания и управление системами

Основным преимуществом Oracle RAC является эффективное управление рабочей нагрузкой в кластере. Для этого используется Oracle Grid Infrastructure в качестве структуры управления, обеспечивающей средства управления и защиты от плановых и внеплановых простоев, а также предоставлением ресурсов по требованию.

Oracle RAC использует Oracle Grid Infrastructure для **поддержки двух следующих типов управления СУБД**.

1. Базы данных Oracle RAC, управляемые администраторами — традиционный способ, при котором экземпляры приписываются к определенным серверам в кластере. Чаще всего это используется в выделенных системах баз данных.
2. Базы данных Oracle RAC, управляемые на основе политик, обеспечивают более гибкое управление, устраняя фиксированное распределение ресурсов (например, когда экземпляры могут работать только на определенных серверах в кластере).

Базы данных, управляемые на основе политик, используют пулы серверов, логическое секционирование кластеров и сервисы баз данных для гибкого распределения и управления нагрузками в одной или нескольких базах данных в кластере. При управлении на основе политик, распределение ресурсов для критичных нагрузок автоматически корректируется в зависимости от предварительно определенного пользователем плана. Эти политики определяют распределение ресурсов и корректировку вычислительной мощности в случае плановых и внеплановых отключений, при изменении нагрузки или программируемой перенастройке кластера.

Oracle Database Quality of Service (QoS) Management использует пулы серверов для создания логических пулов серверов в кластере в целях изоляции рабочих нагрузок. Oracle Database QoS Management может рекомендовать или реализовать перенос сервера из одного пула серверов в другой в зависимости от наблюдаемого или прогнозируемого уровня нагрузки в соответствии с требованиями производительности в данный момент.

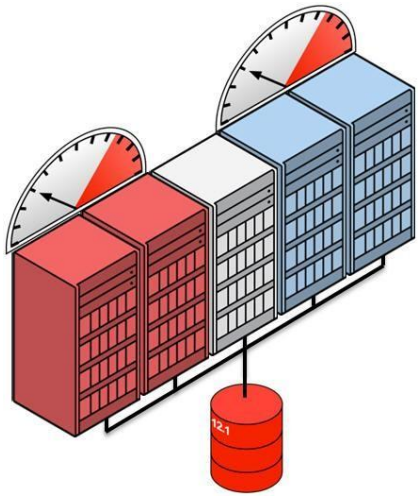

Рис. 3. Базы данных Oracle RAC, управляемые на основе политик, обеспечивают более высокую гибкость и доступность

Новые возможности Oracle RAC

Поддержка Oracle Multitenant в Oracle RAC

Oracle Multitenant — новый компонент Oracle Database 12*c* Enterprise Edition, с помощью которого клиенты могут сократить расходы на ИТ путем упрощения консолидации, развертывания , установки патчей и т.д.. Эти возможности основаны на новой архитектуре, в которой мультиарендная база данных контейнера (CDB) может содержать множество подключаемых баз данных (PDB).
Идея состоит в том, что существующая база данных может быть просто перенесена в контейнер в виде подключаемой базы данных без каких-либо изменений на уровне приложений. В этой архитектуре Oracle RAC обеспечивает локальную высокую доступность, необходимую при консолидации различных важных бизнес-приложений в одной системе.

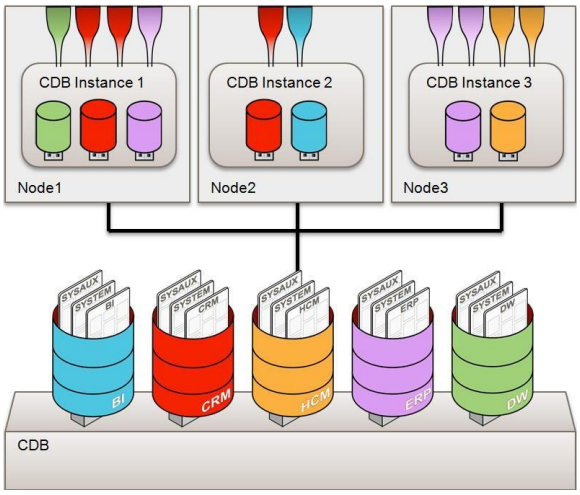

Рис 4. Oracle RAC для поддержки PDB — общий вид

Если подключаемые базы данных (PDB) используются с Oracle RAC, мультиарендная база данных контейнера (CDB) основывается на Oracle RAC. Каждая подключаемая база данных может быть доступна на каждом экземпляре RAC CDB или подмножестве экземпляров.
В любом случае доступ к подключаемым базам данных (PDB) и управление ими регулируется с помощью динамических сервисов баз данных (Dynamic Database Services). Эти сервисы также могут использоваться приложениями для подключения к соответствующей PDB — как в случае с единственным экземпляром СУБД Oracle Database, используя Oracle Net Services для подключения.

Application Continuity и Transaction Guard

Application Continuity — новая технология, которая защищает приложения от сбоев экземпляров и сессий путем повтора транзакций, которые выполнялись в момент сбоя, на другом экземпляре базы данных в кластере.

Эта технология позволяет повторить запрос к базе данных после исправимой ошибки, из-за которой сессия стала недоступной. Повтор производится быстро и без прерывания работы. Операция пользователя в БД может содержать транзакционные и нетранзакционные вызовы базы данных. После успешного повтора приложение может продолжить работу с момента, когда сеанс базы данных был прекращен.

Это дает преимущества пользователям, разработчикам и администраторам одновременно. Пользователям не приходится гадать, прошли ли успешно денежные переводы, бронирование билетов или другие транзакции, а администраторам не нужно перезагружать сервера приложений для восстановления после «шторма» сессий (всплеска запросов на вход в приложение) .

Transaction Guard — надежный протокол и инструмент, который возвращает состояние последней транзакции, прерванной из-за сбоя, сделавшего сеанс базы данных недоступным. Без инструмента Transaction Guard приложения и пользователи, пытающиеся повторно выполнить операции после сбоя, могут вызвать логические повреждения, зафиксировав транзакции-дубликаты или зафиксировав транзакции в неправильном порядке.

Transaction Guard предотвращает неопределенные ошибки, которые приводят к обеспокоенности пользователей, звонкам клиентов в службу поддержки и потере сделок с клиентами. Инструмент Transaction Guard более безопасен и работает лучше и с меньшими издержками, чем решения собственной разработки, цель которых — получение ожидаемого результата при однократном выполнении операций.

Потеря сеанса базы данных может повлиять на пользователя многими способами. Например:

- *Растерянность.* Пользователи не знают, что произошло с их запросами к базе данных (передачей средств, заказами, платежами, бронированием и т. д.).
- *Удобство использования.* Пользователь может потерять незафиксированные данные, в результате чего придется снова входить в систему и вводить или отправлять данные повторно.
- *Прерывание работы.* После сбоя администратору может потребоваться перезагрузка серверов приложений для балансировки нагрузки и устранения «шторма» сессий, что может привести к еще одному отключению пользователя.

Принципы работы — обзор

При использовании Application Continuity, приложению не нужно самому заниматься обработкой отдельных ошибок, но необходимо, чтобы обнаруженный сбой представлял собой «исправимую ошибку». Исправимая ошибка определяется следующим образом:

- *Исправимая ошибка* — это ошибка, возникшая из-за сбоя внешней системы независимо от логики в сессии приложения, выполняемого в данный момент. Исправимые ошибки могут возникнуть после плановых и внеплановых отключений пользовательских процессов, сетевых компонентов, серверов, компонентов системы хранения или экземпляров базы данных. При возникновении исправимой ошибки приложение обычно получает код ошибки, из которого невозможно понять состояние последней отправленной операции. В таких случаях Application Continuity, устраняет необходимость обработки и хранения кодов ошибок в приложении для индивидуального реагирования на соответствующие сбои.

Тем не менее, приложения должны по-прежнему включать обработку ошибок в следующих случаях:

1. Неисправимые ошибки, такие как неправильные входные данные.
2. Исправимые ошибки, когда повтор невозможен из-за какого-то документированного ограничения.

Итог — Application Continuity и Transaction Guard

Если сеанс базы данных становится недоступным из-за исправимой ошибки, Application Continuity пытается пересоздать сеанс и вернуть активные транзакции в правильное состояние. Если транзакция успешно завершена и выполнять ее повторно не требуется, приложение получает успешный статус транзакции.

В случае успешного повтора транзакции, работа приложения продолжается без дублирования транзакции.

Если попытка повтора неудачна, база данных останавливает повтор и приложение получает исходную ошибку. Чтобы повтор считался успешным, он должен вернуть клиенту точно те же данные, которые клиент получил бы ранее в ответ на запрос, и на основании которых приложение могло корректно продолжить работу дальше.[5]

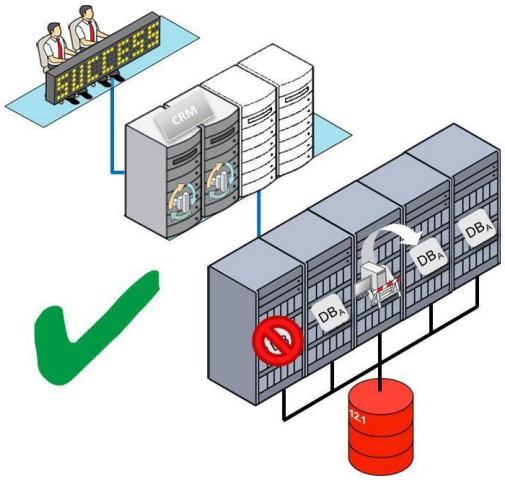

Рис. 5. Работа пользователя *с* Application Continuity

[5] Дополнительные сведения об Application Continuity доступны по адресу: http://www.oracle.com/technetwork/products/clustering/ac-overview-1967264.html

Oracle Flex ASM

Oracle Flex ASM — новая модель развертывания Oracle Automatic Storage Management (ASM), которая увеличивает доступность экземпляров базы данных и снижает потребление ресурсов, связанное с Oracle ASM. Oracle Flex ASM упрощает консолидацию баз данных на основе кластера, так как гарантирует продолжение работы экземпляров Oracle Database 12c, работающих на конкретном сервере, в случае отказа экземпляра Oracle Flex ASM на этом сервере.

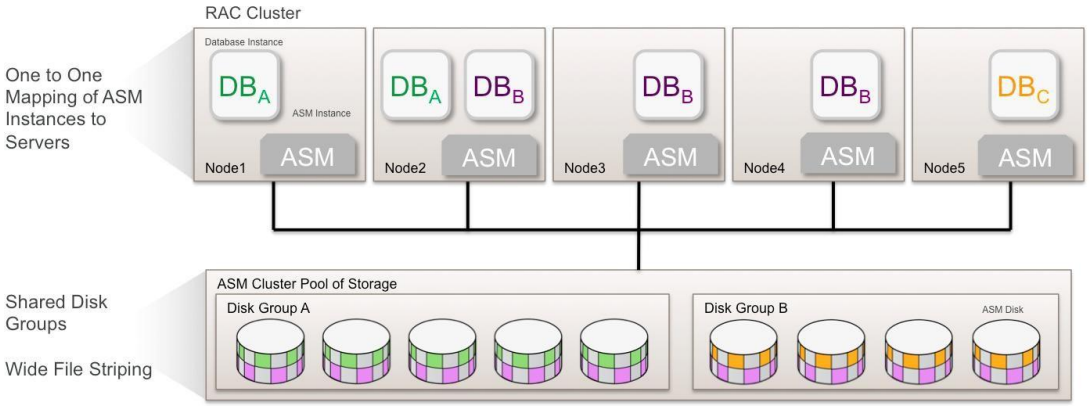

Рис. 6. Oracle ASM *до* Oracle Database 12*c*

До появления Oracle Database 12*c* экземпляр ПО Oracle ASM должен был работать на каждом узле кластера. Экземпляры базы данных должны были подключаться локально на том же самом сервере к экземпляру Oracle ASM, использующему аутентификацию на основе ОС. Сбой экземпляра Oracle ASM привел бы к сбою всех экземпляров базы данных, работающих на этом сервере.

В результате новой стратегии Oracle RAC 12*c,* разрешающей включение новых возможностей по мере необходимости, никакие изменения при обновлении до Oracle ASM 12*c* не требуются, а Oracle ASM может использоваться с предыдущими версиями СУБД в конфигурации, показанной на рис. 6.

Для полного использования новых возможностей Oracle ASM необходимо включить Oracle Flex ASM. Включить Oracle Flex ASM можно во время начальной установки Oracle Grid Infrastructure 12*c* или в отдельном шаге после установки. Необходимо включить Oracle Flex ASM для всего кластера.

Самый заметный эффект включения Oracle Flex ASM состоит в том, что устраняется жесткая («один к одному») привязка экземпляров Oracle ASM к серверам в кластере и в кластерах с тремя или более серверами будет размещаться набор всего из трех экземпляров Oracle Flex ASM, «плавающих» в кластере по его узлам (как кластерный ресурс). Двух узловой кластер серверов будет поддерживать сопоставление «один к одному».

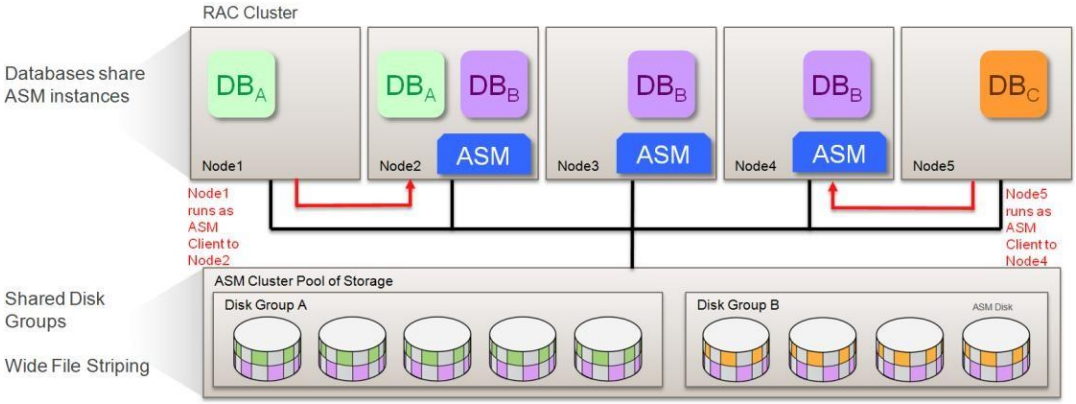

Рис. 7. Включение Oracle Flex ASM устраняет привязку экземпляров к узлам кластера «один к одному»

Включение Oracle Flex ASM также обеспечивает аварийное переключение экземпляра Oracle ASM с одного узла на другой, что вместе с Oracle Database версии 12*c* и выше обеспечит бесперебойную работу базы данных даже при отсутствии экземпляра Oracle ASM на локальном узле.

Oracle Flex ASM может защитить экземпляры базы данных на определенном сервере от внеплановых простоев (незапланированных), а также плановых простоев для профилактического обслуживания (например, установки патчей на ОС или поочередной установки патчей на серверы БД). Независимость экземпляра базы данных от локального экземпляра Oracle ASM значительно увеличивает доступность базы данных — особенно в консолидированных средах с несколькими экземплярами базы данных, работающими на одном сервере.

Принципы работы — обзор

В случае отказа экземпляра Oracle Flex ASM на одном узле этот экземпляр аварийно переключается на другой узел, и экземпляры СУБД Oracle Database 12*c* или более поздней версии будут продолжать работу, используя нелокальный экземпляр Flex ASM в кластере.

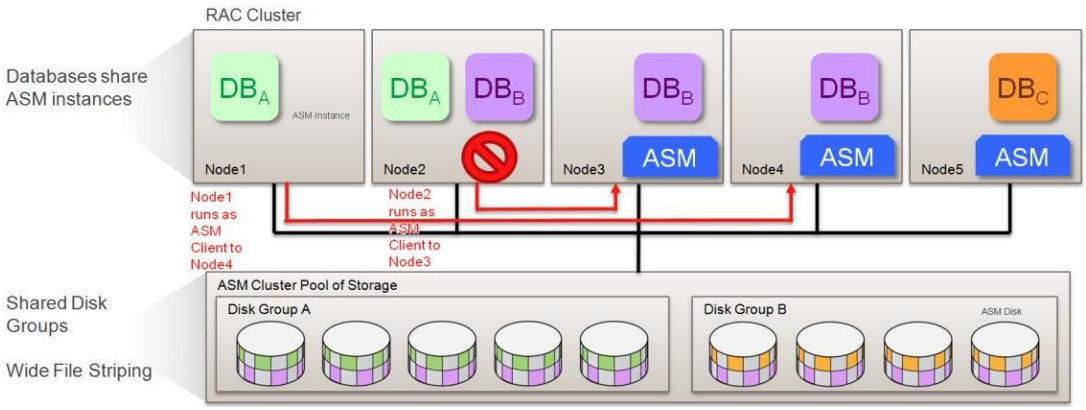

Рис. 8. Экземпляры Oracle Database продолжают работу после сбоя локального экземпляра ASM

Функция «What-If» для эмуляции выполнения команды

Для управления большими и динамическими средами чрезвычайно важно понимать механизм размещения ресурсов до наступления таких событий, как отключение сервера или динамическое перераспределение ресурсов. Функция оценки возможных последствий выполнения команды (функция «What-If») упрощает управление современными динамическими инфраструктурными средами.

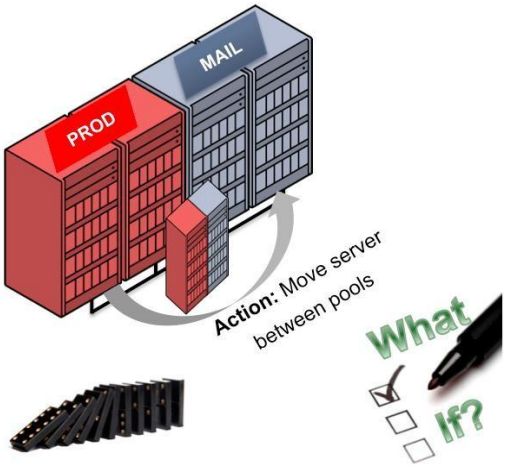

Рис. 9. Функция «What-If» для эмуляции выполнения команды

Как видно из названия, функция «What-If» предназначена для оценки возможного влияния выполнения команды в кластере до того, как команда будет действительно выполнена. Существует два уровня оценки возможных последствий выполнения команды: 1) представление для системных администраторов и 2) представление для администраторов баз данных. Разница между двумя представлениями — помимо того, что они предоставляются утилитой CRSCTL (для администратора кластера) или утилитой SRVCTL (для администратора базы данных) — состоит том, какая именно информация предоставляется.

Предполагается, что администраторы кластеров должны управлять кластером как единым целым и контролировать все приложения в кластере, в то время как администраторы баз данных скорее всего захотят узнать влияние выполнения команды на одну или несколько баз данных, которыми они управляют. Кроме того, команды, необходимые для выполнения ежедневных или разовых задач управления, отличаются для каждой группы администраторов, и для каждой команды должна быть доступна одна и та же функциональность. Командное средство CRSCTL также позволяет задать уровень детализации выходного результата.

Функция «What-If» может использоваться с разными параметрами в утилитах SRVCTL и CRSCTL, что позволяет оценить практически любые сценарии управления до их фактического выполнения. Специальные команды (напр., SRVCTL predict) или параметры команд (напр., параметр «–fail» команды CRSCTL) могут использоваться даже для прогнозирования сбоя компонента.

Функция «What-If» не обеспечивает транзакционную безопасность. Это означает, что оценка проводится по состоянию кластера, существующего на момент запуска оценки. В результатах будет учтено именно это состояние. Если предположить, что состояние кластера не менялось с момента запуска оценки до фактического выполнения команды, результат должен быть таким, как был предсказан функцией What-If. Если состояние кластера меняется, результат фактического выполнения может отличаться от прогноза.

Управление базами данных и кластером на основе политик

Управление базами данных и кластером на основе политик чрезвычайно важно, если речь идет об управлении системами управления базами данных высокой доступности в динамических средах. В средах, управляемых на основе политик, можно автоматически и динамически настраивать систему в зависимости от изменения нагрузки. Такие среды также обеспечивают доступность услуг в системах, где произошло несколько сбоев подряд. Большинство приложений не могут понять, является изменение в кластере следствием сбоя или плановой операции обслуживания, поэтому управление на основе политик может использоваться для улучшения плановых и внеплановых операций.

Расширенное использование и определение пула серверов в Oracle Grid Infrastructure 12*c*

Oracle Grid Infrastructure 12*c* расширяет использование пулов серверов с помощью новых атрибутов серверов, которые хранятся отдельно для каждого сервера, а также определения пулов серверов с учетом категорий серверов. Используя атрибуты серверов, ПО Oracle Grid Infrastructure 12*c* понимает и активно использует концепцию разнородных серверов в кластере. Для этого предварительно заданные атрибуты серверов, а также атрибуты, заданные пользователем, хранятся отдельно для каждого сервера.

Атрибуты серверов автоматически определяются внутри системы при запуске стека Oracle Clusterware на определенном сервере и хранятся, пока стек на сервере не будет перезапущен. Набор атрибутов сервера по умолчанию (к которому можно добавить атрибуты, заданные пользователем) включает, в частности, «SERVER NAME», «MEMORY_SIZE» и «CPU_COUNT».

Атрибуты серверов могут затем использоваться для объединения в категории серверов для определения пулов серверов. Категории серверов определяют пул серверов на основе атрибутов, а не просто по номеру или имени сервера, который будет частью пула. Категории серверов используют регулярные выражения для определения того, какой тип серверов может использоваться в конкретном пуле серверов. Это позволяет определить начальные требования ч для присоединения сервера к пулу, и поэтому может использоваться для гарантии того, что все серверы в пуле серверов обеспечивают минимальную мощность для выполнения нагрузки, назначенных этому пулу серверов через Dynamic Database Service.

Следует отметить, что стек Oracle RAC всегда позволял использовать серверы разной мощности в одном кластере. Пользователь должен был сам привязать нагрузку к серверам с разной вычислительной мощностью или другими конкретными атрибутами. Этот тип управления ресурсами вручную не только создает большой объем работы для администратора, но и по своей природе порождает большое количество ошибок при анализе требуемой вычислительной мощности, в случае сбоев и отключений. Определения пулов серверов с помощью категорий серверов, основанных на физических атрибутах или атрибутах, заданных пользователем, обеспечивают автоматический и экономичный способ управления средами с различными требованиями рабочих нагрузок и серверами разной мощности.

Использование сред, управляемых на основе политик, для масштабируемости и гибкости

Одним из основных примеров использования базы данных, управляемой на основе политик, является автоматическая корректировка сред баз данных с учетом изменения нагрузки. Еще один пример использования (особенно с Oracle Grid Infrastructure 12*c*) охватывает приложения, не являющиеся приложениями базы данных и функционирующие с наборами политик.

Изменение нагрузки может быть вызвано ожидаемым или неожиданным увеличением нагрузки на одно или несколько приложений, размещенных в кластере[6]. Но это также может быть результатом операции планового обслуживания (когда необходимо отключить сервер для профилактики) или периодической корректировки настроек кластера для круглосуточной работы (24x7) в разных географических регионах и часовых поясах или соответствия изменениям нагрузки в конце месяца или года. Среды, управляемые на основе политик, обеспечивают инфраструктуру для автоматического выполнения всех этих сценариев.

Использование сред, управляемых на основе политик, для повышения показателя высокой доступности

Среды, управляемые на основе политик, — самый простой способ адекватной и автоматической адаптации к множеству сценариев сбоев в кластере. В системах, управляемых администраторами, где такие ресурсы, как экземпляры баз данных и динамические сервисы баз данных, статически приписаны к определенным серверам в кластере, сбои обрабатываются путем определения «предпочтительных» и «доступных» узлов вручную.

В средах, управляемых администраторами, ресурсы определяются таким образом, что работают только на определенном экземпляре (сервисы Dynamic Database Services назначаются экземплярам базы данных) или на определенных серверах (в случае экземпляров базы данных), и только в случае недоступности такого экземпляра или сервера ресурс (напр., сервис базы данных) может быть размещен на другом экземпляре или сервере в кластере. Никакие решения на основе политик не принимаются, все серверы считаются равными, назначение рабочей нагрузки сервера статично и задается вручную.

Среды, управляемые на основе политик, устраняют необходимость таких статических определений. Используя политики и наборы политик, администраторы могут легко определять пулы серверов разной важности и привязывать критически важные сервисы к конкретным пулам серверов заранее установленной важности. С учетом доступных ресурсов в кластере, политики гарантируют, что пул серверов большей важности и критическая рабочая нагрузка, назначенная этому пулу серверов, получат необходимые вычислительные ресурсы для нужной производительности, доступности и выполнения соглашений об уровне обслуживания (Service Level Agreement).

Категории серверов гарантируют, что назначение серверов пулам серверов будет соответствовать необходимым минимумам. Если происходит сбой сервера, учитывается приоритет рабочей нагрузки (с учетом определения важности в пуле серверов) и применяются предсказуемые политики для соответствующего переноса ресурсов внутри кластера. Если оставшийся размер кластера недостаточен для размещения требуемой нагрузки, те же самые политики могут использоваться для возможного переноса вычислительных ресурсов из пула серверов с меньшим приоритетом в пул серверов с более высоким приоритетом для предоставления необходимых вычислительных ресурсов рабочим нагрузкам, важным для бизнеса.

[6] Для автоматического реагирования на изменения нагрузки, вызванные приложениями, требуется мониторинг приложений с помощью Oracle Enterprise Manager 12*c* или Oracle Quality of Service (QoS) Management.

Итоги и заключение

Непрерывность бизнеса, высокая доступность, масштабируемость, гибкость и оперативность в сочетании с простотой управления являются основой успешного развертывания ИТ-инфраструктур для облачных вычислений. Уже более десяти лет тысячи клиентов Oracle выбирают решение Oracle Database Real Application Clusters (RAC) для своих СУБД, важных для бизнеса.

Oracle Real Application Cluster 12*c* продолжает этот успех, обеспечивая значительные дополнительные возможности во всех областях, важных для успешного бизнеса. Такие новые функции, как Application Continuity и Oracle Flex ASM, существенные новые возможности в области управления базами данных и кластерами на основе политик, а также комплексная консолидация баз данных и поддержка опции Oracle Multitenant делают Oracle RAC наилучшим решением для виртуализации баз данных в вашей инфраструктуре.

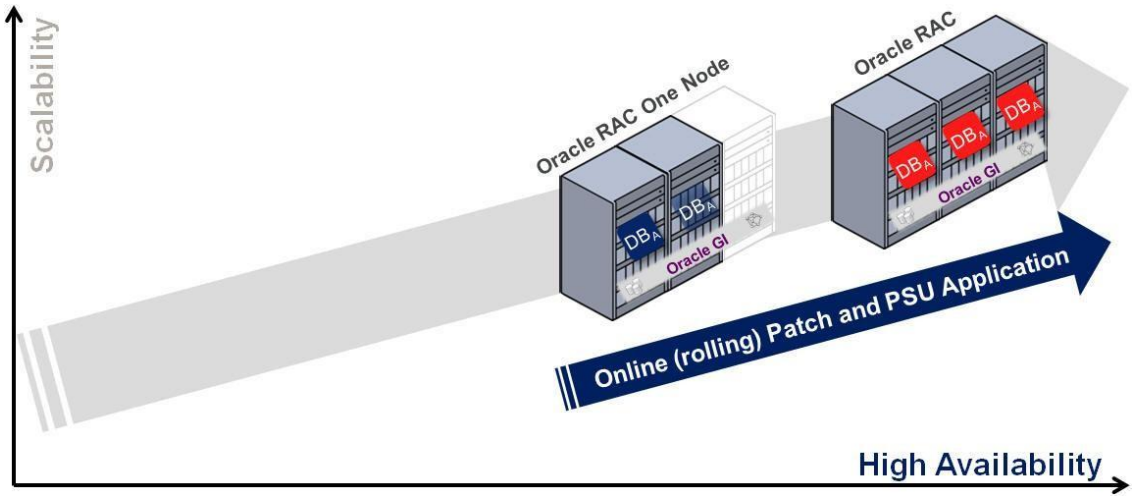

Рис. 10. Стандартизация на основе Oracle RAC для лучшей масштабируемости и доступности

ORACLE

Разрабатывая свои программы и продукцию, корпорация Oracle заботится об окружающей среде.

Oracle Real Application Clusters (RAC),
июнь 2013 г.
Автор: Маркус Михалевич
(Markus Michalewicz) Соавторы:
Берт Клоуз (Burt Clouse), Джон
Макхью (John McHugh)

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A. (США)

Для запросов со всего
мира: Телефон:
+1.650.506.7000
Факс: +1.650.506.7200

oracle.com

© Oracle и аффилированные компании, 2013. 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 и условия товарного состояния или соответствия определенной цели. Oracle не несет никакой ответственности в связи с данным документом. Документ также не создает никаких договорных обязательств ни прямо, ни 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 аффилированных 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 компании SPARC International, Inc. AMD, Opteron. Логотипы AMD и AMD Opteron являются товарными знаками или зарегистрированными товарными знаками компании Advanced Micro Devices. UNIX является зарегистрированным товарным знаком The Open Group. 0113

Оборудование и программное обеспечение, созданные для совместной работы