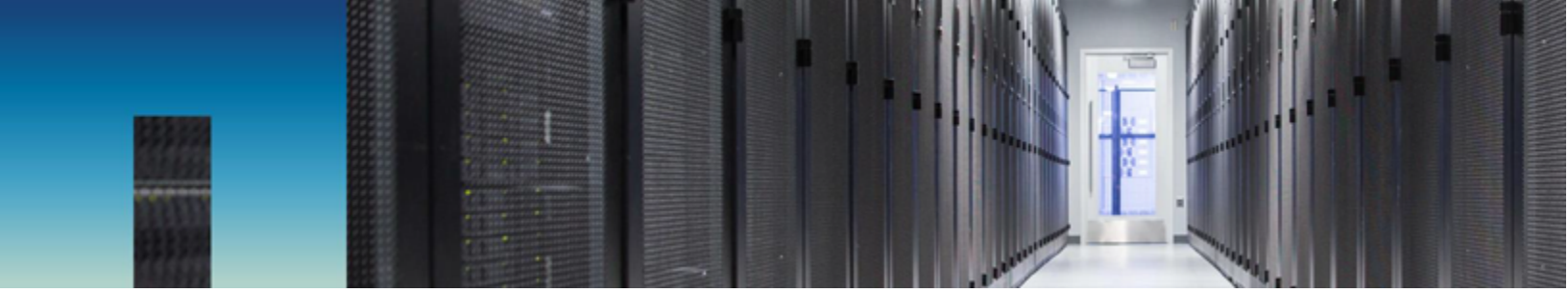

Технический отчет

Базы данных Oracle на платформе ONTAP

Джеффри Штайнер (Jeffrey Steiner),
NetApp, март 2020 г. | TR-3633

Важно!

Используйте инструмент Interoperability Matrix Tool (IMT), чтобы определить, поддерживают ли среда, конфигурации и версии, указанные в этом отчете, вашу среду.

NetApp®

СОДЕРЖАНИЕ

1 Введение

NetApp® ONTAP® – это мощная платформа управления данными со встроенными возможностями сжатия на лету, обновления оборудования без нарушения его работы и импорта логических дисков (LUN) из внешнего массива хранения. В кластер можно объединить до 24 узлов, одновременно работающих с данными по протоколам сетевой файловой системы (NFS), общей файловой системы Интернета (CIFS), iSCSI, Fibre Channel (FC) и Fibre Channel over Ethernet (FCoE). Кроме того, технология NetApp Snapshot® служит основой для создания десятков тысяч онлайновых резервных копий и полностью работоспособных клонов баз данных.

Помимо богатого набора возможностей ONTAP, доступна настройка под широкий спектр требований пользователей, включая размер базы данных, производительность и потребности в защите данных. Сегодня развернуты самые разные хранилища NetApp, начиная с виртуализированной среды примерно из 6000 баз данных под управлением VMware ESX и заканчивая хранилищем данных на базе одного экземпляра, объем которого сейчас составляет 996 ТБ и продолжает расти. Поэтому есть ряд четких практических рекомендаций по настройке БД Oracle в хранилищах NetApp.

В этом документе требования к работе с БД Oracle в хранилище NetApp описываются двумя способами. Во-первых, если существует четкая практическая рекомендация, то она указывается в явной форме. Во-вторых, в этом документе рассматриваются многие аспекты проектирования, которые должны учитывать архитекторы, строящие решения для хранения данных на базе Oracle, исходя из конкретных бизнес-требований.

В этом документе сначала рассматриваются общие соображения для всех сред, а затем – конкретные рекомендации, основанные на выборе виртуализации и ОС. Специальные темы, такие как выбор структуры файловой системы и снятие блокировок NFS, включены в приложения.

Дополнительные сведения приведены в следующих материалах:

- TR-4591: Защита данных в БД
- TR-4592: Oracle в кластере MetroCluster
- TR-4534: Перенос БД Oracle в СХД NetApp

2 Платформы ONTAP

Программное обеспечение ONTAP является основой передовой защиты данных и управления ими. Однако под «ONTAP» понимается только программное обеспечение. Предлагается несколько аппаратных платформ ONTAP на выбор:

- ONTAP на системах All Flash FAS (AFF) и FAS
- NetApp Private Storage (NPS) для облака
- ONTAP Select
- Cloud Volumes ONTAP

Основная идея состоит в том, что ONTAP – это ONTAP. Одни варианты обеспечивают лучшую производительность, другие – меньшие расходы, а третьи работают в облаках крупнейших провайдеров. Основные функции ONTAP остаются неизменными, и есть множество вариантов репликации, чтобы связать разные платформы ONTAP в одно решение. Поэтому стратегии защиты данных и аварийного восстановления можно строить, исходя из реальных потребностей (производительности, капитальных и эксплуатационных затрат) и общей облачной стратегии. Базовая технология хранения работает в любой среде.

2.1 ONTAP с контроллерами AFF и FAS

Если нужно обеспечить максимум производительности и возможностей контроля данных, то ONTAP на физическом контроллере AFF или FAS остается вне конкуренции. Это стандартный вариант, на который тысячи заказчиков полагаются уже больше 20 лет. ONTAP предлагает решения для любой среды – от систем с тремя критически важными БД и до систем поставщиков услуг, содержащих по 60 000 БД, – и позволяет мгновенно восстанавливать петабайтные базы и предлагать резервное копирование как услугу (DBaaS) для резервирования сотен клонов одной базы данных.

2.2 NetApp Private Storage (NPS) for Cloud

Компания NetApp предлагает данное решение для удовлетворения потребностей заказчиков, обрабатывающих большие объемы данных в публичном облаке. Хотя существует много вариантов хранения в публичном облаке, большинство из них имеют ограничения по производительности, возможностям управления и масштабированию. Если говорить о рабочих нагрузках баз данных, то основными ограничениями являются следующие:

- У многих публичных облачных хранилищ производительность не масштабируется – с учетом приемлемых затрат, уровней эффективности или возможностей управления – до значений IOPS, необходимых современным задачам баз данных.
- Даже когда базовые показатели производительности, предлагаемые поставщиком публичного облака, отвечают требованиям, задержки ввода-вывода часто оказываются неприемлемыми для рабочих нагрузок баз данных. Это стало еще более актуальным после миграции баз данных на all-flash СХД (полностью на flash-дисках), в результате чего бизнес-заказчики стали измерять задержки даже не миллисекундами, а микросекундами.
- Хотя публичные облачные хранилища в целом обеспечивают высокую доступность, она все еще не отвечает требованиям большинства критически важных сред.
- В публичных облачных сервисах хранения данных имеются возможности резервного копирования и восстановления, но они обычно не отвечают требованиям нулевого RPO и почти нулевого RTO, налагаемым большинством баз данных. Для защиты данных требуется по-настоящему мгновенное резервное копирование и восстановление на основе моментальных снимков, а не потоковое резервное копирование в другое место (и восстановление из другого места) облака.
- Гибридные облачные среды должны перемещать данные между локальными и облачными системами хранения, что требует общей основы для управления хранением.
- Во многих странах приняты строгие законы о суверенитете данных, которые запрещают перемещать данные за пределы национальных границ.

Системы NPS обеспечивают максимум производительности, контроля и гибкости СХД для поставщиков публичных облаков, включая Amazon AWS, Microsoft Azure и IBM SoftLayer. Эта возможность обеспечивается системами AFF и FAS (включая варианты MetroCluster) в ЦОДах, подключенных прямо к публичным облакам. Всю вычислительную мощь гиперскейлеров можно использовать без ограничений, накладываемых системой хранения. Кроме того, NPS обеспечивает независимость от облака и возможность работы с несколькими облаками, поскольку данные, такие как двоичные файлы приложений, базы данных, резервные копии баз данных и архивы, полностью остаются в системе NPS. Не нужно тратить время, полосу пропускания или деньги на передачу данных между поставщиками облачных сервисов.

Примечательно, что некоторые заказчики NetApp использовали модель NPS по собственной инициативе. Во многих местах ЦОДам заказчиков по запросу предоставляется высокоскоростной доступ к одному из гиперскейлеров. В других случаях заказчики для размещения используют ЦОД, в котором уже есть высокоскоростной доступ к инфраструктуре гиперскейлеров. Это привело к тому, что Amazon AWS, Azure и SoftLayer стали использоваться, по сути, как источники виртуализированных серверов, предоставляемых по запросу на основе фактического потребления ресурсов. В некоторых случаях в повседневной работе заказчиков ничего не изменилось. Они просто используют сервисы гипермасштабируемого ЦОДа как более мощную, гибкую и экономически эффективную замену своей традиционной инфраструктуры виртуализации.

Предлагается также вариант NPS как услуги (NPSaaS). Во многих случаях потребности сред баз данных достаточно высоки, чтобы оправдать покупку системы NPS в ЦОД. Однако в некоторых случаях заказчики предпочитают использовать и облачные серверы, и облачное хранилище по модели операционных, а не капитальных расходов. В этих случаях они хотят использовать ресурсы хранилища исключительно по запросу и в том объеме, в котором нужно. Несколько поставщиков сейчас предлагают таким заказчикам NPS как услугу.

2.3 ONTAP Select

ONTAP Select работает на принадлежащей заказчику инфраструктуре виртуализации и предоставляет сервисы ONTAP и подключение к «фабрике данных» на основе обычных дисков внутри серверного оборудования третьих производителей. ONTAP Select позволяет ONTAP и гостевым ОС совместно использо-

вать одно и то же физическое оборудование как высоко-конвергентную инфраструктуру. Это не меняет наилучшие практики по использованию Oracle на ONTAP. Основной упор делается на производительность, но не следует недооценивать возможности ONTAP Select.

Среда ONTAP Select не обеспечивает такую же пиковую производительность, как у систем AFF высшего класса, но большинству баз данных и не нужно 300 000 IOPS. Типичные базы данных требуют производительность лишь около 5-10 тысяч IOPS, что вполне достижимо с помощью ONTAP Select. Кроме того, производительность большинства баз данных ограничена скорее задержками СХД, чем ее производительностью в IOPS, а эту проблему можно решить, развернув ONTAP Select на твердотельных дисках.

2.4 Cloud Volumes ONTAP

Система Cloud Volumes ONTAP аналогична ONTAP Select за тем исключением, что она работает в облачной среде гиперскейлера, обеспечивая аналитику и подключение фабрики данных к томам облачного провайдера. Это не влияет на передовые практики по запуску Oracle на ONTAP. Основные соображения – производительность и (в меньшей степени) стоимость.

Cloud Volumes ONTAP частично ограничены производительностью базовых (подлежащих) томов, управляемых поставщиком облачных услуг. В результате получается лучше управляемое хранилище, а в некоторых случаях возможность кэширования позволяет повысить производительность.В результате получается более управляемая среда хранения данных, а кэширующие возможности Cloud Volume ONTAP в некоторых случаях позволяют повысить производительность. Однако всегда есть некоторые ограничения по IOPS и задержках из-за зависимости от поставщика публичного облака. Это не означает, что производительность базы данных неприемлема. Это лишь означает, что потолок производительности ниже, чем, скажем, у реальной физической системы AFF. Кроме того, производительность томов хранения, предлагаемых различными поставщиками облачных услуг, которые использует систему Cloud Volumes ONTAP, постоянно улучшается.

Система Cloud Volumes ONTAP в настоящее время в основном используется для разработки и тестирования, но некоторые заказчики использовали ее и для производственной деятельности. В одном довольно известном отчете описано использование функции Oracle In-Memory для снижения ограничений производительности хранилища. Это позволяет хранить больше данных в ОЗУ виртуальной машины, на которой размещается сервер базы данных, снижая требования к производительности хранилища.

3 Конфигурация ONTAP

Полное описание конфигурации ОС ONTAP выходит за рамки настоящего документа. Наилучшие практики для среды с 2000 виртуализированных баз данных может не подходить для конфигурации из трех очень больших баз данных, используемых для систем управления корпоративными ресурсами. Даже небольшие изменения в требованиях к защите данных могут серьезно повлиять на архитектуру хранилища. Ряд основных моментов рассматривается в этом разделе. Более полное объяснение дается в документе TR-4591. Для получения исчерпывающей помощи в дизайне свяжитесь с компанией NetApp или ее партнером.

3.1 Уровни RAID

Иногда возникают вопросы относительно уровней RAID в конфигурации систем хранения NetApp. Во многих старых документах и книгах Oracle по конфигурированию Oracle есть предупреждения об использовании RAID с зеркалированием и/или о необходимости избегать определенных типов RAID. Хотя в этих материалах и поднимаются важные вопросы, они не применимы к RAID 4 и к технологиям NetApp RAID DP® и RAID-TEC™, используемым в ONTAP.

Чтобы не потерять данные из-за отказа диска, в массивах RAID 4, RAID 5, RAID 6, RAID DP и RAID-TEC используется контроль четности. Эти варианты RAID обеспечивают гораздо большую эффективность хранилища по сравнению с зеркалированием, но у большинства реализаций RAID есть недостаток, который влияет на операции записи. Завершение операции записи в других реализациях RAID требует нескольких операций чтения с дисков для повторной генерации данных четности (это обычно называется RAID-пенальти).

Однако в случае ONTAP RAID-пенальти отсутствует, так как NetApp WAFL® (Write Anywhere File Layout - файловая структура с записью повсюду) интегрирована с подсистемой RAID. Операции записи объеди-

няются в ОЗУ и подготавливаются как полный RAID-страйп, включая генерацию данных четности. Для завершения записи выполнять дополнительные операции чтения не нужно, а значит, в ONTAP и WAFL нет RAID-пенальти. Производительность критичных к задержке операций, таких как журналирование транзакций (redo logging), не страдает, и случайные записи данных в файл не влекут за собой RAID-пенальти из-за повторной генерации данных четности.

Что касается статистической надежности, даже RAID DP обеспечивает лучшую защиту, чем RAID с зеркалированием. Основной проблемой является нагрузка на диски во время восстановления RAID. Если используется RAID с зеркалированием, то риск потери данных из-за сбоя диска во время восстановления его диска-партнера по RAID намного выше, чем риск сбоя трех дисков в RAID DP.

3.2 Пределы емкости

Чтобы обеспечить высокую и предсказуемую производительность системы хранения, требуется некоторое свободное пространство для метаданных и организационных задач работы с данными. Свободное пространство определяется как любое пространство, не используемое для фактических данных, и включает в себя нераспределенное пространство на самом агрегате и неиспользуемое пространство на составляющих его томах. Следует учитывать и тонкое выделение пространства. Например, том может содержать LUN емкостью 1 ТБ, из которого фактическими данными может быть занято лишь 50%. В среде с динамическим предоставлением ресурсов это будет выглядеть правильно – как занятые 500 ГБ. Однако в среде с полным предоставлением ресурсов будет казаться, что используется полностью 1 ТБ. 500 ГБ нераспределенного пространства будут скрытыми. Это пространство не используется фактическими данными и поэтому должно быть включено в расчет общего свободного пространства.

Рекомендации NetApp относительно систем хранения, используемых для баз данных, описаны в следующих разделах.

SSD-агрегаты, включая системы AFF

NetApp рекомендует оставлять минимум 10% свободного места. Сюда входит все неиспользуемое пространство, включая свободное пространство на агрегате или томе и любое свободное пространство, выделенное из-за использования полного предоставления, но не занятое фактическими данными.

Рекомендация насчет 10% свободного пространства является консервативной. SSD-агрегаты могут обслуживать базы данных и при более высоком заполнении без ущерба для производительности. Однако по мере роста заполнения агрегата риск нехватки места также увеличивается, если заполнение тщательно не контролировать.

HDD-агрегаты, включая агрегаты Flash Pool

NetApp рекомендует оставлять минимум 15% свободного места. Сюда входит все неиспользуемое пространство, включая свободное пространство на агрегате или томе и любое свободное пространство, выделенное из-за использования полного предоставления, но не занятое фактическими данными.

При заполнении менее 85% никакого измеримого влияния на производительность быть не должно. Когда заполнение достигает 90%, на определенных рабочих нагрузках можно заметить некоторое снижение производительности. Когда заполнение достигает 95%, большинство рабочих нагрузок с БД испытывают снижение производительности.

3.3 Резервное копирование на основе моментальных снимков

Наибольшую важность для структуры файловой системы представляет план использования технологии NetApp Snapshot. Есть два основных подхода:

- Консистентные после сбоя (Crash-consistent) резервные копии
- Оперативные резервные копии, защищенные моментальными снимками

Консистентное резервное копирование базы данных требует снимка всей структуры БД (включая файлы данных, журналы транзакций и контрольные файлы) в один момент времени. Если БД хранится на одном томе NetApp FlexVol®, то моментальный снимок можно легко создать в любое время. Если же БД распределена по нескольким томам, то нужно создать копию в виде моментального снимка группы консистентности (consistency group, CG). Есть несколько вариантов создания копий в виде моментальных снимков групп консистентности, включая ПО NetApp SnapCenter®, фреймворк NetApp Snap Creator®,

NetApp SnapManager® для Oracle (SMO), NetApp SnapDrive® для UNIX и поддерживаемые пользователями скрипты.

Резервные копии в виде консистентных после сбоя моментальных снимков используются в основном тогда, когда достаточно восстановления на момент создания резервной копии. В некоторых случаях можно применять архивы журналов, но, когда требуется более детальное восстановление на определенный момент времени, предпочтительнее будет оперативная резервная копия.

Базовая процедура оперативного резервного копирования на основе моментальных снимков выглядит так:

- Переведите БД в режим `backup`.
- Снимите копии в виде моментального снимка со всех томов, на которых находятся файлы данных.
- Выйдите из режима `backup`.
- Запустите принудительное архивирование журналов командой `alter system archive log current`.
- Создайте копии в виде моментального снимка всех томов, на которых находятся архивные журналы

Данная процедура создает набор мгновенных снимков, содержащих файлы данных в режиме резервного копирования и критичные архивные журналы, созданные во время нахождения в режиме резервного копирования. Эти файлы являются необходимыми для восстановления базы данных. Для удобства можно защитить и управляющие файлы, но единственным абсолютным требованием является защита файлов данных и архивных журналов.

Хотя у разных заказчиков могут быть очень разные стратегии, почти все они в конечном счете основаны на принципах, изложенных в этом разделе.

3.4 Восстановление на основе моментальных снимков

Проектируя структуры томов для БД Oracle, прежде всего нужно решить, использовать ли технологию NetApp SnapRestore® (VBSR, volume-based SnapRestore)для восстановления томов.

SnapRestore позволяет практически мгновенно вернуть том к состоянию более раннего момента времени. Поскольку к состоянию более раннего момента времени возвращаются все данные на томе, VBSR может подходить не во всех случаях. Например, если вся БД (включая файлы данных, журналы транзакций и архивные журналы) хранится на единственном томе и этот том восстанавливается с помощью VBSR, то данные теряются, поскольку более новые архивные журналы и данные транзакций удаляются.

Не обязательно использовать именно технология VBSR для восстановления. Многие БД можно восстановить с помощью технологии файлового SnapRestore (SFSR, single-file SnapRestore) или простым копированием файлов из моментального снимка обратно в активную файловую систему.

VBSR предпочтительнее, когда БД очень велика или когда ее нужно восстановить как можно быстрее, кроме того использование VBSR требует изоляции файлов данных. В среде NFS файлы данных БД должны храниться на выделенных томах, на которых отсутствуют файлы любого другого типа. В среде SAN файлы данных должны храниться на выделенных LUN на выделенных томах FlexVol. Если используется менеджер томов (включая Oracle Automatic Storage Management, или ASM), то для файлов данных тоже должна быть выделена группа дисков.

Такая изоляция файлов данных позволяет вернуть их в более раннее состояние без повреждения других файловых систем.

3.5 Резервирование места под моментальные снимки

Для каждого тома с данными Oracle в среде SAN параметр percent-snapshot-space следует установить в нулевое значение, так как резервирование места для копии в виде моментального снимка в среде LUN бесполезно. Если для частичного резерва (fractional reserve) установить значение 100, то моментальный снимок тома с LUN будет требовать свободное место в томе (исключая резерв под моментальные снимки), достаточное для сохранения 100% объема переменной части данных. Если для частичного резерва установить меньшее значение, то потребуется соответственно меньше свободного пространства, но в него никогда не включается резерв под моментальные снимки. Это означает, что резерв места под моментальные снимки в среде LUN тратится впустую.

В среде NFS есть два варианта:

- Установить параметр percent-snapshot-space, исходя из ожидаемой потребности моментальных снимков в дисковом пространстве.

- Установить для параметра percent-snapshot-space нулевое значение и совместно управлять потреблением активного пространства и пространства под моментальные снимки.

В первом варианте параметр percent-snapshot-space устанавливается в ненулевое значение (обычно около 20%). Это пространство будет скрыто от пользователя. Однако это не задает ограничения на использование. Если БД с 20-процентным резервом имеет оборот 30%, то пространство для моментальных снимков может разрастись сверх 20-процентного резерва и занять незарезервированное пространство.

Главная польза от установки, скажем, 20-процентного резерва состоит в том, что моментальным снимкам всегда будет доступно некоторое пространство. Например, том емкостью 1 ТБ с резервом 20% позволит администратору БД хранить только 800 ГБ данных. Эта конфигурация гарантирует минимум 200 ГБ пространства для моментальных снимков.

Когда для параметра percent-snapshot-space установлено значение 0, все пространство тома доступно конечному пользователю, что обеспечивает большую наглядность. Администратор БД должен понимать, что если он видит том объемом 1 ТБ, на котором используются моментальные снимки, то этот 1 ТБ пространства распределяется между активными данными и оборотом данных, хранящимся в моментальных снимках.

Среди конечных пользователей нет четких предпочтений между первым и вторым вариантами.

3.6 ONTAP и моментальные снимки, сделанные сторонними средствами

В документе Oracle Doc ID 604683.1 разъясняются требования к поддержке моментальных снимков, сделанных сторонними средствами, и различные варианты выполнения операций резервного копирования и восстановления.

Сторонний вендор должен гарантировать, что снимки, создаваемые его средствами, отвечают следующим требованиям:

- Снимки должны интегрироваться с операциями восстановления, рекомендованными компанией Oracle.
- Моментальные снимки должны быть консистентными на момент создания.
- Порядок операций записи сохраняется для каждого файла в моментальном снимке.

ONTAP и продукты NetApp для управления Oracle отвечают этим требованиям.

3.7 Кластерные операции – перехват (Takeover) и переключение (Switchover)

Необходимо понимать такие функции хранилища, как перехват и переключение, чтобы гарантировать, что они не будут нарушать операции БД:

- В нормальных условиях операции записи, поступающие в данный контроллер, синхронно зеркалируются на его партнера. В среде NetApp MetroCluster™ операции записи зеркалируются также на удаленный контроллер. До тех пор, пока операция записи не будет сохранена на энергонезависимом носителе во всех местах, ее выполнение не будет подтверждено приложению.
- Носитель, хранящий данные операции записи, называется энергонезависимой памятью (NVMEM). Иногда его называют энергонезависимой оперативной памятью (NVRAM), и его можно рассматривать как кэш записи, хотя он функционирует как журнал. При нормальной работе данные из NVMEM не читаются – она используется только для защиты данных в случае программного или аппаратного сбоя. Когда данные записываются на диск, данные передаются в систему из ОЗУ, а не из NVMEM.
- При перехвате (takeover) один узел в паре высокой доступности (high availability, HA) перехватывает операции от своего партнера. Переключение (switchover) – по сути то же самое, но применяется к конфигурациям MetroCluster, где удаленный узел перехватывает функции локального узла.

Во время планового обслуживания перехват или переключение хранилища должны выполняться незаметно, кроме, быть может, короткой паузы в операциях с БД, вызванной изменением сетевых путей. Сетевые подключения – дело сложное и чреватое ошибками, поэтому NetApp настоятельно рекомендует тщательно протестировать перехват и переключение на БД, прежде чем запускать систему хранения в эксплуатацию. Это единственный способ убедиться в том, что все сетевые пути настроены правильно. В среде SAN внимательно проверьте вывод команды sanlun lun show -p, чтобы убедиться, что все ожидаемые основные и резервные пути доступны.

Команды на принудительный перехват или переключение нужно выдавать осторожно. Принудительное изменение конфигурации хранилища с помощью этих параметров означает, что состояние контроллера, управ-

ляющего дисками, больше не учитывается и альтернативный узел принудительно перехватывает контроль над дисками. Некорректно выполненный перехват может привести к потере или повреждению данных из-за того, что принудительный перехват или переключение могут сбросить содержимое NVMEM. После завершения перехвата или переключения потеря этих данных означает, что данные, хранящиеся на диске, могут вернуться в немного более старое состояние с точки зрения БД.

В нормальной паре высокой доступности принудительный перехват требуется редко. Почти во всех сценариях отказа узел выключается и информирует своего партнера, что позволяет выполнить автоматическую обработку отказа. Бывают экстремальные ситуации (такие как «волна отказов»), когда теряется соединение между узлами, а затем отказывает один контроллер, в результате чего требуется принудительный перехват. В такой ситуации зеркалирование между узлами теряется до отказа контроллера, а это значит, что у сохранившего работоспособность контроллера больше не будет копии незавершенных операций записи. Тогда необходим принудительный перехват, ведущий к потенциальной потере данных.

Та же логика применима к переключению в кластере MetroCluster. В нормальных условиях переключение почти незаметно. Однако авария может привести к потере подключения между сохранившей работоспособность и аварийной площадками. С точки зрения работоспособной площадки, проблема может быть вызвана просто обрывом соединения между площадками, и основная площадка, возможно, по-прежнему обрабатывает данные. Если узел не может проверить состояние основного контроллера, то возможно только принудительное переключение.

NetApp рекомендует принять следующие меры предосторожности:

- Будьте крайне аккуратны, чтобы случайно не инициировать принудительный перехват или переключение – обычно это не требуется, а любое форсирование изменений может привести к потере данных.
- Если требуется принудительный перехват или переключение, то убедитесь, что БД выключена, размонтируйте все файловые системы, закройте все экземпляры ASM и отключите все группы томов в менеджере логических томов (LVM).
- В случае принудительного переключения в MetroCluster отгородите отказавший узел от всех ресурсов хранения, сохранивших работоспособность. Дополнительные сведения см. в Руководстве по управлению и аварийному восстановлению MetroCluster для соответствующей версии ONTAP.

MetroCluster и несколько агрегатов

MetroCluster – это технология синхронной репликации с переключением в асинхронный режим при обрыве соединения. Именно это чаще всего и нужно заказчикам – ведь гарантированная синхронная репликация означает, что разрыв соединения между площадками полностью останавливает ввод-вывод БД и выводит ее из работы.

При использовании MetroCluster агрегаты быстро восстанавливают синхронизацию после восстановления подключения. В отличие от других технологий хранения, MetroCluster никогда не должен требовать полного повторного зеркалирования после отказа площадки. Зеркалируются только изменения.

В базах данных, распределенных по нескольким агрегатам, существует небольшой риск того, что в случае волны отказов потребуются дополнительные шаги по восстановлению данных. В частности, если (а) связь между площадками прервется, (б) связь восстановится, (в) агрегаты достигнут состояния, в котором какие-то из них синхронизированы, а какие-то нет, затем (г) откажет основная площадка, то результатом будет сохранившая работоспособность площадка с агрегатами, которые не синхронизированы друг с другом. В таком случае части базы данных синхронизируются друг с другом, и запустить БД без восстановления невозможно. Если БД распределена по нескольким агрегатам, то NetApp настоятельно советует использовать резервные копии на основе моментальных снимков и один из множества доступных инструментов, чтобы убедиться в возможности быстро восстановиться в том случае, если этот маловероятный сценарий все же реализуется.

NVFAIL

Из-за больших внутренних кэшей базы данных могут повредиться в случае принудительного перехвата или переключения. Если произойдет принудительная обработка отказа, то ранее подтвержденные изменения будут фактически отменены. Содержимое системы хранения, по сути, отбрасывается назад во времени, и состояние кэша БД перестает отражать состояние данных на диске. Результат – повреждение данных.

Кэширование может происходить на уровне приложения или сервера. Например, сервер БД Oracle кэширует данные в системной глобальной области (system global area, SGA) Oracle. Операция, вызвавшая потерю данных, может поставить БД под угрозу повреждения, поскольку блоки, хранящиеся в SGA, могут не совпадать с блоками на СХД. Менее очевидное использование кэширования – на уровне файловой системы ОС. Блоки из смонтированной файловой системы NFS могут кэшироваться в ОС, либо файловая система на базе LUN может кэшировать данные в буферном кэше ОС. Отказ NVRAM или принудительный перехват в этих ситуациях может привести к повреждению файловой системы.

Системы ONTAP защищают БД и ОС от этого с помощью NVFAIL и связанных с ним параметров.

4 Виртуальные системы хранения и логические интерфейсы

В этом разделе дается обзор основных принципов управления. Для получения более полной документации см. Руководство по управлению сетью ONTAP для используемой версии ONTAP. Как и в случае с другими аспектами архитектуры БД, лучшие варианты организации SVM (формально также известных как Vserver) и логических интерфейсов (LIF) сильно зависят от требований к масштабированию и потребностей бизнеса.

Создавая стратегию логических интерфейсов, учитывайте следующие основные вопросы:

- Производительность. Достаточна ли пропускная способность сети?
- Отказоустойчивость. Если ли в архитектуре единые точки отказа?
- Управляемость. Можно ли масштабировать сеть без нарушения работы?

Эти вопросы относятся ко всему сквозному решению «хост – коммутаторы – СХД».

4.1 Виртуальные СХД

Виртуальная СХД (SVM) – это основная функциональная единица хранения, поэтому полезно сравнивать ее с гостевой машиной на сервере VMware ESX. При первой установке ESX не имеет преднастроенных возможностей, таких как хостинг гостевой ОС или поддержка приложений конечного пользователя. Пока не определена ВМ, это просто пустой контейнер. То же самое и с ONTAP. При первой установке эта ОС не имеет возможностей по обслуживанию данных – сначала нужно определить SVM. Именно специализация SVM определяет сервисы данных.

Некоторые заказчики используют одну основную виртуальную систему хранения для большинства повседневных потребностей и создают небольшое количество дополнительных виртуальных СХД для особых нужд, в том числе:

- SVM для критически важной БД, управляемой командой специалистов
- SVM для группы разработчиков, которой предоставлен полный административный контроль, чтобы они могли самостоятельно управлять собственным хранилищем
- SVM для конфиденциальных бизнес-данных (таких как кадровая информация или финансовая отчетность), для которых группу администраторов нужно ограничить

В многопользовательской среде виртуальная СХД может выделяться под данные каждого арендатора. Рекомендуемый лимит составляет приблизительно 125 SVM на узел кластера, но обычно максимумы LIF достигаются раньше, чем будет достигнут этот лимит. Существует точка, в которой многопользовательскую среду лучше разделять по сетевым сегментам, а не изолировать их в выделенные SVM.

4.2 Типы LIF

Есть много типов логических интерфейсов (LIF). Официальная документация ONTAP содержит более полную информацию по этой теме, но с функциональной точки зрения LIF можно разделить на следующие группы:

- **LIF управления кластерами и узлами.** LIF, используемые для управления кластером хранения.
- **LIF управления виртуальными СХД.** Интерфейсы, которые разрешают доступ к SVM через API ONTAP (известный как ZAPI) для реализации таких функций, как создание копии в виде моментального снимка или изменение размера тома. Такие продукты, как SnapManager for Oracle и SnapCenter, должны иметь доступ к интерфейсам управления SVM.
- **LIF данных.** Интерфейсы, которые предоставляют данные по FC, iSCSI, NFS или CIFS.

Примечание: LIF данных, используемый для трафика NFS, можно использовать и для управления, изменив политику брандмауэра с data на mgmt или другую политику, разрешающую HTTP, HTTPS или SSH. Это изменение может упростить конфигурирование сети, так как это позволит не настраивать каждый хост для доступа как к LIF данных NFS, так и к отдельному LIF управления. Невозможно настроить интерфейс и для трафика iSCSI, и для трафика управления, хотя оба используют протокол IP. В средах iSCSI для управления требуется отдельный LIF.

4.3 Архитектура логических интерфейсов SAN

Архитектура LIF в среде SAN относительно проста по одной причине – множественность путей. Все современные реализации SAN позволяют клиенту получать доступ к данным по нескольким сетевым путям и выбирать для доступа лучший путь или пути. Это упрощает разработку высокопроизводительной архитектуры LIF, потому что клиенты SAN автоматически распределяют нагрузку ввода-вывода по наилучшим доступным путям.

Если путь становится недоступным, то клиент автоматически выбирает другой путь. Получающаяся в результате простота архитектуры делает логические интерфейсы SAN в целом более управляемыми. Это не означает, что средой SAN всегда легче управлять, потому что есть много других аспектов хранилищ SAN, которые намного сложнее, чем NFS. Это лишь означает, что дизайн логических интерфейсов SAN проще.

Производительность

Важнейшим фактором, влияющим на производительность LIF в среде SAN, является пропускная способность. Например, кластер ONTAP с четырьмя узлами и двумя 16-гигабитными FC-портами на узел обеспечивает пропускную способность до 32 Гбит/с по каждому узлу. Ввод/вывод автоматически балансируется между портами, и все операции ввода/вывода направляются по оптимальному пути.

Устойчивость

Логические интерфейсы SAN не восстанавливаются при отказе. Если логический интерфейс SAN откажет, то это будет обнаружено хостом, а ввод/вывод – перенаправлен на другой логический интерфейс.

Управляемость

Миграция LIF является гораздо более распространенной задачей в среде NFS, поскольку она часто связана с перемещением томов в кластере. В среде SAN при перемещении томов нет нужды в миграции LIF. Это объясняется тем, что после завершения перемещения тома ONTAP отправляет в SAN уведомление об изменении путей, и клиенты SAN автоматически выполняют повторную оптимизацию. Миграция LIF в SAN вызывается главным образом серьезными изменениями физического оборудования. Например, если требуется обновление контроллеров без нарушения работы, то логический интерфейс SAN переносится на новое оборудование. Если будет обнаружена неисправность FC-порта, то LIF можно будет перенести на неиспользуемый порт.

Рекомендации по проектированию

NetApp дает следующие основные рекомендации:

- Не создавайте больше путей, чем нужно. Избыток путей усложняет общее управление и может вызвать на некоторых хостах проблемы с переключением путей в случае отказа. Кроме того, некоторые хосты имеют неожиданные ограничения путей для таких конфигураций, как загрузка по SAN.
- Очень немногим LUN требуется больше четырех путей к хранилищу. Ценность наличия более двух узлов, объявляющих пути к LUN, невелика, поскольку агрегат, на котором размещается LUN, будет

недоступен, если выйдут из строя узел, которому принадлежит LUN, и его HA-партнер. Создание путей на узлах, отличных от первичной HA-пары, в такой ситуации не поможет.

- Хотя количеством видимых путей LUN можно управлять, выбирая порты, включаемые в зоны FC, обычно проще включить все потенциальные целевые точки в зону FC и управлять видимостью LUN на уровне ONTAP.
- В ONTAP 8.3 и более поздних версиях функция избирательного отображения LUN (selective LUN mapping, SLM) задействована по умолчанию. При использовании функции SLM любой новый LUN автоматически объявляется из узла, которому принадлежит соответствующий агрегат, и из HA-партнера узла. Такая организация избавляет от необходимости создавать наборы портов или настраивать зонирование для ограничения доступности портов. Каждый LUN доступен на минимальном количестве узлов, необходимых как для оптимальной производительности, так и для отказоустойчивости.
- Если необходимо вынести LUN за пределы двух контроллеров, то дополнительные узлы можно добавить командой lun mapping add-reporting-nodes, чтобы LUN объявлялись на новых узлах. Это создает в SAN дополнительные пути к LUN для их миграции. Однако, чтобы использовать новые пути, хост должен выполнить операцию обнаружения.
- Особо не беспокойтесь о непрямом трафике. Лучше избегать непрямого трафика в среде с очень интенсивным вводом/выводом, где критична каждая микросекунда задержки, но для типичных рабочих нагрузок видимое влияние на производительность незначительно.
- Следуйте правилам зонирования, описанным в разделе 13.1.

4.4 Архитектура логического интерфейса NFS

В отличие от протоколов SAN, возможности NFS по заданию нескольких путей к данным ограничены. Расширения «параллельной NFS» (pNFS) для NFSv4.1 устраняют это ограничение, но pNFS еще не поддерживается для БД Oracle и не рассматривается в этом документе.

Производительность и отказоустойчивость

В то время как измерение производительности логических интерфейсов SAN в первую очередь зависит от расчета общей пропускной способности для всех основных путей, определение производительности логических интерфейсов NFS требует более тщательного изучения конфигурации конкретной сети. Например, два порта 10 Гбит/с можно использовать как есть, либо объединить их в интерфейсную группу под управлением протокола LACP. Если их настроить как группу интерфейсов, то станет доступен ряд политик балансировки нагрузки, которые работают по-разному в зависимости от того, коммутируется трафик или маршрутизируется. Наконец, Direct NFS (DNFS) предлагает конфигурации с балансировкой нагрузки, которых в настоящее время нет в NFS-клиентах ни для каких ОС.

В отличие от протоколов SAN, файловые системы NFS требуют отказоустойчивости на уровне протоколов. Например, LUN всегда конфигурируется с включенной поддержкой множества путей, поэтому системе хранения доступно несколько резервных каналов, каждый из которых использует протокол FC. С другой стороны, файловая система NFS зависит от доступности одного TCP/IP-канала, который можно защитить только на физическом уровне. Поэтому и существуют такие варианты, как переключение портов при отказе и агрегирование портов по протоколу LACP.

В среде NFS производительность и отказоустойчивость обеспечиваются на уровне сетевого протокола. Так что обе темы переплетаются и должны обсуждаться вместе.

Привязка LIF к группам портов

Чтобы привязать LIF к группе портов, сопоставьте IP-адрес LIF с группой физических портов. Основным методом агрегирования физических портов является LACP. Отказоустойчивость LACP обеспечивается довольно просто: каждый порт в группе LACP контролируется и исключается из группы портов в случае неисправности. Однако существует много заблуждений относительно работы LACP в плане производительности:

- LACP не требует, чтобы конфигурация на коммутаторе соответствовала оконечному устройству. Например, ONTAP можно настроить с балансировкой нагрузки на основе IP-адресов, а коммутатор может использовать балансировку нагрузки на основе MAC-адресов.
- Каждое оконечное устройство, использующее LACP-соединение, может независимо выбирать порт

для передачи пакетов, но не может выбирать порт, используемый для приема. Это означает, что трафик из ONTAP в конкретный пункт назначения привязан к определенному порту, а обратный трафик может поступать через другой интерфейс. Впрочем, проблемы это не создает.

- LACP не распределяет трафик все время равномерно. В крупномасштабной среде со множеством NFS-клиентов результатом обычно является равномерное использование всех портов в LACP-агрегате. Однако любое конкретное подключение NFS в среде ограничено пропускной способностью только одного порта, а не всего агрегата портов.
- Несмотря на то, что в ONTAP доступны политики циклической балансировки с помощью LACP, эти политики не регулируют соединение между коммутатором и хостом. Например, конфигурация с четырехпортовой LACP-магистралью на хосте и четырехпортовой LACP-магистралью на ONTAP по-прежнему способна читать файловую систему только через один порт. Хотя ONTAP может передавать данные через все четыре порта, в настоящее время отсутствуют технологии коммутации, которые отправляют данные с коммутатора на хост через все четыре порта. Используется только один.

В крупномасштабных средах со множеством хостов БД самый распространенный подход состоит в создании LACP-агрегата из соответствующего количества 10-гигабитных интерфейсов и использовании балансировки нагрузки по IP-адресу. При таком подходе ONTAP может обеспечивать равномерное использование всех портов при наличии достаточного количества клиентов. С уменьшением числа клиентов балансировка нагрузки нарушается, поскольку LACP не перераспределяет нагрузку динамически.

Когда соединение установлено, трафик в конкретном направлении проходит только через один порт. Например, БД, выполняющая полное табличное сканирование в файловой системе NFS, подключенной через четырехпортовую LACP-магистраль, считывает данные только через одну сетевую карту (NIC). Если в такой среде находятся только три сервера БД, то возможно, что все три считывают данные с одного и того же порта, а остальные три порта простаивают.

Привязка LIF к физическим портам

Привязка LIF к физическому порту позволяет более тонко управлять конфигурацией сети, поскольку данный IP-адрес в системе ONTAP связан только с одним сетевым портом в каждый конкретный момент времени. Отказоустойчивость достигается настройкой групп обработки отказов и политик обработки отказов.

Политики обработки отказов и группы обработки отказов

Поведение логических интерфейсов при нарушении работы сети контролируется политиками обработки отказов и группами обработки отказов. Параметры конфигурации менялись в разных версиях ONTAP. Подробную информацию о развертываемой версии ONTAP см. в Руководстве по управлению сетью ONTAP.

Для ONTAP 8.2 и более ранних версий следуйте следующим общим правилам:

1. Настройте группу обработки отказов так, чтобы она определялась пользователем.
2. Включите в группу обработки отказов порты контроллера обработки отказов хранилища (storage failover, SFO) партнерского узла, чтобы при обработке отказа хранилища LIF следовали за агрегатами. Эта конфигурация позволяет избежать создания непрямого трафика.
3. Для обработки отказа используйте порты с параметрами производительности, которые соответствуют исходному LIF. Например, LIF на одном 10-гигабитном физическом порту должен включать в себя группу отработки отказа с одним 10-гигабитным портом. Четырехпортовый логический интерфейс LACP должен в случае отказа переключиться на другой четырехпортовый логический интерфейс LACP.
4. Установите политику обработки отказа в значение «Priority».

ONTAP 8.3 позволяет управлять обработкой отказов LIF на основе широковещательных доменов. Поэтому администратор может определить все порты, имеющие доступ к данной подсети, и разрешить ONTAP выбирать подходящий LIF при отказе. Этот подход можно использовать для некоторых заказчиков, но он имеет ограничения в высокоскоростной сетевой среде хранения данных из-за недостаточной предсказуемости. Например, среда может включать как 1-гигабитные порты для обычного доступа к файловой

системе, так и 10-гигабитные порты для ввода/вывода файлов данных (например БД). Если порты обоих типов сосуществуют в одном и том же широковещательном домене, то при обработке отказа LIF ввод/вывод файла данных может быть переключен с 10-гигабитного порта на 1-гигабитный порт.

NetApp рекомендует использовать подход, реализованный в ONTAP 8.2, который определяет, какие порты можно использовать для обработки отказа LIF. Итак, соблюдайте следующие правила:

1. Настройте группу обработки отказов как определяемую пользователем.
2. Включите в группу обработки отказов порты на контроллере партнерского узла, чтобы при обработке отказа хранилища LIF следовали за агрегатами – так вы избежите создания непрямого трафика.
3. Для обработки отказа используйте порты с параметрами производительности, которые соответствуют исходному LIF. Например, LIF на одном 10-гигабитном физическом порту должен включать в себя группу отработки отказа с одним 10-гигабитным портом. Четырехпортовый логический интерфейс LACP должен в случае отказа переключиться на другой четырехпортовый логический интерфейс LACP. Эти порты будут представлять собой подмножество портов, определенных в широковещательном домене.
4. Установите для политики обработки отказов значение «SFO-partner only». Это гарантирует, что LIF будет следовать за агрегатом при обработке отказа.

Параметр «Auto-revert [Автоматический возврат]»

Установите для параметра `auto-revert` нужное вам значение. Большинство заказчиков предпочитают устанавливать для него значение true, чтобы обеспечить возврат LIF на домашний порт. Однако в некоторых случаях заказчики устанавливают для него значение `false`, чтобы иметь возможность исследовать неожиданное аварийное переключение, прежде чем возвращать LIF на домашний порт.

Число интерфейсов на том

Распространенным заблуждением является то, что отношение между томами и логическими интерфейсами NFS должно быть 1:1. Хотя эта конфигурация и нужна для перемещения тома в любое место внутри кластера без создания дополнительного межузлового трафика, это ни в коей мере не является требованием. Учитывать межузловой трафик нужно, но само его присутствие проблем не создает. Многие опубликованные результаты измерения производительности ONTAP включают в себя преимущественно непрямой ввод/вывод.

Например, проект БД с относительно небольшим количеством баз данных, критически важных для производительности, для которого требуется всего 40 томов, может потребовать соотношения «LIF-to-volume», равного 1:1, что означает 40 IP-адресов. Затем любой том можно переместить куда угодно в пределах кластера вместе со связанным интерфейсом LIF, и трафик всегда будет прямым, а все вносимые задержки будут минимизированы, даже когда речь пойдет о микросекундах.

И обратный пример: управлять большой средой, предоставляемой в аренду, будет легче при отношении между числом заказчиков и LIF, равным 1:1. Со временем может потребоваться перенос тома на другой узел, что породит некоторый непрямой трафик. Однако влияние на производительность не должно быть заметно, если только сетевые порты кластерного коммутатора не загружаются полностью. В случае опасений новый LIF можно организовать на дополнительных узлах, а хост можно обновить в следующем окне обслуживания, чтобы исключить непрямой трафик из конфигурации.

5 Качество обслуживания

Более широкое внедрение all-flash СХД также привело к консолидации рабочих нагрузок БД. Массивы хранения на традиционных жестких дисках, как правило, поддерживают лишь небольшое количество БД из-за ограничений старой технологии по параметру IOPS. Одна или две высокоактивные БД полностью загрузят диски, на которых они размещаются, задолго до того, как контроллеры СХД достигнут предела своих возможностей. Сейчас ситуация изменилась. При нынешней производительности SSD-накопителей уже относительно небольшое их число может полностью загрузить даже самые мощные контроллеры СХД. А значит, возможности контроллеров можно использовать полностью, не опасаясь внезапного падения производительности из-за резкого увеличения задержки на вращающихся дисках.

Пример: простая двухузловая система высокой доступности AFF8080 способна обслуживать около

400 000 случайных операций ввода/вывода в секунду при задержке в пределах 1 мс. Предположительно, в такой производительности нуждаются менее 1% БД, а использовать AFF8080 для 10 000 операций ввода/вывода в секунду будет расточительством.

В ONTAP качество обслуживания (QoS) измеряется двумя параметрами: IOPS (числом операций ввода/вывода) и полосой пропускания. Средства управления качеством обслуживания (QoS) можно применять к SVM, томам, LUN и файлам.

5.1 Качество обслуживания – IOPS

Управление качеством обслуживания по параметру IOPS очевидно зависит от общего показателя IOPS конкретного ресурса, но есть ряд аспектов управления качеством по параметру IOPS, которые могут и не быть интуитивно понятными. Некоторые заказчики были сначала озадачены явным ростом задержки при достижении порогового значения IOPS. На самом деле, это единственный реальный способ ограничить IOPS. С точки зрения логики он работает аналогично маркерной системе. Например, если определенный том с файлами данных имеет ограничение в 10 000 IOPS, то каждая поступающая в очередь операция ввода/вывода должна сначала получить маркер (токен) для дальнейшей обработки. Пока в течение одной конкретной секунды не будет использовано более 10 000 маркеров, никаких задержек не будет.

5.2 Качество обслуживания – полоса пропускания

Во-первых, разные операции ввода/вывода имеют разный размер. БД могут выполнять множество совершенно случайных операций чтения блоков (что может привести к достижению порога IOPS), а могут выполнять операцию полного табличного сканирования, состоящую из очень небольшого числа операций чтения больших блоков и потребляющую очень много полосы пропускания, но относительно мало IOPS.

5.3 Минимальное/гарантированное качество обслуживания

Многие клиенты ищут решение, которое гарантирует качество обслуживания. Это труднее обеспечить, чем может показаться, и достижение этого может быть весьма затратным делом. Например, размещение 10 БД с гарантированным значением 10 000 IOPS требует рассчитывать систему, исходя из сценария, когда все 10 БД одновременно работают с производительностью 10 000 IOPS, что в совокупности составляет 100 000 IOPS.

Политика обеспечения минимального (гарантированного) качества обслуживания больше всего подходит для защиты критических рабочих нагрузок. Например, рассмотрим контроллер ONTAP с максимально возможным значением IOPS = 500 000 и комбинацией продуктивных рабочих нагрузок и рабочих нагрузок разработки. К рабочим нагрузкам разработки примените политики максимального качества обслуживания, чтобы предотвратить монополизацию контроллера любой конкретной БД. Затем примените политики минимального качества обслуживания к продуктивным рабочим нагрузкам, чтобы гарантировать, что при необходимости им всегда будет доступно требуемое значение IOPS.

6 Эффективность

Сжатие, уплотнение и дедупликация – это функции повышения эффективности хранения, которые увеличивают количество логических данных, помещаемых в определенный объем физической памяти. В общих чертах, сжатие – это математический процесс, выявляющий повторяющиеся структуры данных, которые затем кодируются так, чтобы уменьшить требования к дисковому пространству. Напротив, дедупликация обнаруживает реальные повторяющиеся блоки данных и удаляет лишние копии. Уплотнение позволяет нескольким логическим блокам данных совместно использовать один и тот же физический блок на носителе.

6.1 Сжатие

До появления all-flash СХД польза от сжатия БД была ограниченной, поскольку большинству БД все равно требовалось очень много жестких дисков для обеспечения приемлемой производительности. СХД неизменно содержали гораздо больше емкости, чем требовалось, что было побочным эффектом большого количества дисков. С развитием хранилищ на твердотельных накопителях ситуация изменилась. Стало не нужно предоставлять чрезмерное количество дисков просто для получения хорошей производительности. Дисковое пространство СХД может соответствовать фактическим потребностям.

Увеличенные параметры IOPS твердотельных накопителей (SSD) почти всегда дают экономию средств по сравнению с жесткими дисками, но сжатие может обеспечить дополнительную экономию путем увеличения эффективной емкости твердотельных накопителей. Хотя сжатие может быть выполнено самой БД, в среде Oracle это делается редко. Встроенная функция сжатия не подходит для быстро меняющихся данных, а лицензия на расширенную функцию сжатия стоит дорого. Кроме того, высока стоимость самой БД Oracle. Нет особого смысла платить за дорогую лицензию по числу ЦП, если ЦП занят сжатием и распаковкой данных, а не реальной работой с БД. Лучший вариант – переложить работу по сжатию на систему хранения.

Адаптивное сжатие

Адаптивное сжатие было тщательно протестировано с рабочими нагрузками Oracle без какого-либо заметного влияния на производительность даже в среде с all-flash СХД, где задержка измеряется микросекундами. При первоначальном тестировании некоторые заказчики сообщали об увеличении производительности при использовании сжатия.

ONTAP управляет физическими блоками размером 4 КБ. Следовательно, максимально возможный коэффициент сжатия при использовании только адаптивного сжатия составляет 2:1 для БД с блоком размером 8 КБ. Предварительное тестирование на данных реальных заказчиков показало, что коэффициенты сжатия приближаются к этому уровню, но результаты варьируются в зависимости от типа хранимых данных.

Вторичное сжатие

При вторичном сжатии используется фиксированный блок большего размера – 32 КБ. Это позволяет ONTAP сжимать данные сильнее, чем в соотношении 2:1, но вторичное сжатие в первую очередь предназначено для неактивных данных и данных, которые записываются последовательно и требуют максимального сжатия.

NetApp рекомендует использовать вторичное сжатие для наборов данных с большим объемом неизменяемых данных, таких как архивные журналы или резервные копии Recovery Manager (RMAN). Файлы этих типов пишутся последовательно и не обновляются. Это не значит, что адаптивное сжатие не рекомендуется. Просто если объем хранимых данных велик, то вторичное сжатие обеспечивает лучшую экономию по сравнению с адаптивным.

Рассмотрим вторичное сжатие файлов данных, когда объем данных очень велик, а сами файлы данных доступны только для чтения или редко обновляются. Файлы данных,использующие размер блока 32 КБ, должны сжаться сильнее при вторичном сжатии, которому соответствует блок размером 32 КБ. Однако необходимо убедиться, что на этих томах не размещены данные с размерами блоков, отличными от 32 КБ. Используйте этот метод только тогда, когда данные не обновляются часто.

Предупреждение

Не следует совместно использовать вторичное сжатие и дедупликацию при резервном копировании с помощью RMAN. Причина в том, что даже небольшие изменения в данных резервного копирования влияют на окно сжатия 32 КБ. Если окно сдвинется, то сжатые данные будут отличаться по всей длине файла. Поскольку дедупликация выполняется после сжатия, механизм дедупликации будет видеть каждую сжатую резервную копию по-разному. Если требуется дедупликация резервных копий, созданных с помощью RMAN, то следует избегать использования вторичного сжатия в этом случае. Использование адаптивного сжатия более предпочтительно, т.к. оно работает с меньшим размером блока и не снижает эффективность дедупликации. Сжатие на стороне хоста также влияет на эффективность дедупликации по тем же причинам, что и вторичное сжатие на СХД.

Выравнивание

Адаптивное сжатие в среде БД требует определенного учета выравнивания блоков сжатия. Это касается только данных, в которых некоторые вполне конкретные блоки могут произвольно перезаписываться. Этот подход принципиально аналогичен общему выравниванию файловой системы, как описано в разделе “Зонирование.”

Например, в Oracle запись 8-килобайтного блока в файл данных сжимается, только если он выравнивается

по границе 8-килобайтного блока внутри самой файловой системы. Это означает, что он должен попадать в первые 8 КБ файла, вторые 8 КБ файла и т.д. Такие данные, как резервные копии RMAN или архивные журналы, представляют собой последовательно записываемые операции, которые охватывают несколько блоков, каждый из которых сжимается. Следовательно, нет необходимости учитывать выравнивание. Единственная проблема с вводом/выводом – случайные перезаписи файлов данных.

NFS

При использовании NFS файлы данных выравниваются. Каждый блок файла данных выравнивается относительно начала файла.

SAN

Среды SAN требуют, чтобы данные выравнивались по границам 8-килобайтных блоков для оптимального сжатия. Для SAN есть два аспекта выравнивания: LUN и файловая система. LUN должен быть сконфигурирован либо как устройство на весь диск (без разделов), либо как раздел с границей, выровненной по 8-килобайтному блоку. Посмотрите нижеследующие разделы по конкретным ОС, в которых даются подробные сведения о сжатии и выравнивании для конкретных конфигураций.

Примечание. См. раздел “Частичное резервирование”, в котором разъясняется взаимосвязь между сжатием и частичным резервированием.

6.2 Уплотнение данных на лету

Уплотнение данных на лету (inline data compaction) – это введенная в ONTAP 9 технология, которая повышает эффективность сжатия. Как уже говорилось ранее, одно только адаптивное сжатие может обеспечить в лучшем случае экономию 2:1, поскольку оно ограничено сохранением 8-килобайтного ввода/вывода в 4-килобайтном блоке WAFL. Такие методы сжатия, как вторичное сжатие, используют блок большего размера и обеспечивают лучшую эффективность. Однако они не подходят для данных, которые могут перезаписываться небольшими блоками. Распаковка 32-килобайтных блоков данных, обновление 8-килобайтной части, повторное сжатие и запись на диск создают дополнительные непроизводительные затраты.

Уплотнение данных на лету позволяет хранить несколько логических блоков в одном физическом блоке. Например, БД с хорошо сжимаемыми данными (такими как текст или частично заполненные блоки), может сжиматься от 8 КБ до 1 КБ. Без уплотнения этот 1 килобайт данных все равно будет занимать весь 4-килобайтный блок. Уплотнение данных на лету позволяет хранить один килобайт сжатых данных всего лишь в одном килобайте физического пространства вместе с другими сжатыми данными. Это не технология сжатия. Это просто более эффективный способ распределения пространства на диске, поэтому он не должен сколько-нибудь заметно влиять на производительность.

Величина достигаемой экономии варьируется. Данные, которые уже сжаты или зашифрованы, как правило, не удается сжать еще сильнее, поэтому такие наборы данных не выигрывают от использования технологий сжатия. Вновь инициализированные файлы данных Oracle, которые не содержат почти ничего, кроме метаданных блоков и нулей, сжимаются до 80:1. Это открывает широчайший спектр возможностей. Лучший способ оценить потенциальную экономию – использовать инструмент оценки экономии пространства (NetApp Space Savings Estimation Tool, SSET), доступный на веб-сайте NetApp Field Portal или через представителя компании NetApp.

6.3 Дедупликация

Блок Oracle содержит заголовок, который глобально уникален для БД, и концевую метку, которая почти уникальна. В результате дедупликация БД Oracle редко обеспечивает экономию более 1%.

В некоторых случаях наблюдалась экономия пространства до 15% в БД с размером блока 16 КБ и больше. Начальные 4 КБ каждого блока содержат глобально уникальный заголовок, а последние 4 КБ блока содержат почти уникальный хвостовик. Внутренние блоки являются кандидатами на дедупликацию, хотя на практике это почти полностью относится на дедупликацию нулевых данных.

Во многих конкурирующих массивах заявлена возможность дедупликации БД Oracle в том предположении, что БД копируется несколько раз. В этом случае также можно использовать дедупликацию NetApp, но ONTAP предлагает лучший вариант – технологию NetApp FlexClone®. Конечный результат такой же: создается несколько копий БД Oracle, которые совместно используют большую часть базовых физиче-

ских блоков. Использовать FlexClone намного эффективнее, чем тратить время на копирование файлов данных с последующей дедупликацией. По сути, это не дедупликация – ведь дубликат изначально не создается.

В том редком случае, когда есть несколько копий одних и тех же файлов данных, дедупликация может обеспечить преимущества.

6.4 Функции обеспечения эффективности и тонкое выделение пространства

Функции обеспечения эффективности – это формы тонкого выделения пространства. Например, LUN на 100 ГБ, занимающий том размером 100 ГБ, можно, если повезет, сжать до 50 ГБ. Фактической экономии пока нет, потому что размер тома все равно составляет 100 ГБ. Сначала нужно уменьшить размер тома, чтобы высвобожденное место можно было использовать где-то еще в системе. Если последующие изменения в LUN на 100 ГБ приводят к тому, что данные становятся менее сжимаемыми, то размер LUN увеличивается, а добавленный объем может заполняться.

Настоятельно рекомендуется использовать тонкое выделение, так как оно может упростить управление, существенно увеличив полезную емкость при одновременной экономии средств. Причина проста: среды Oracle часто включают в себя много пустого места, томов и LUN, а также сжимаемые данные. При статическом выделении пространства резервируется пространство для хранения томов и LUN на тот случай, если когда-нибудь они заполнятся на 100% и будут содержать абсолютно несжимаемые данные. Вряд ли это когда-либо произойдет. Динамическое выделение позволяет освободить это пространство и использовать его где-то еще, а также управлять емкостью в зависимости от самой СХД, а не от множества более мелких томов и LUN.

Отдельные заказчики предпочитают использовать статическое выделение ресурсов либо для определенных рабочих нагрузок, либо основываясь на устоявшейся практике эксплуатации.

Внимание! Если том выделен статически, то нужно полностью отключить на нем все функции обеспечения эффективности, включая приведение сжатых и дедуплицированных данных в оригинальное состояние с помощью команды sis undo. Этот том не должен отображаться в выводе команды `volume efficiency show`. Если он отображается, значит, какие-то функции обеспечения эффективности на нем еще не отключены. Как следствие, гарантии перезаписи работают по-другому, и это увеличивает вероятность того, что из-за огрехов конфигурации на томе может внезапно закончиться свободное место, что приведет к ошибкам ввода/вывода БД.

6.5 Передовые приемы обеспечения эффективности

NetApp дает следующие рекомендации для ONTAP 9 и выше. Если у вас ONTAP до версии 9, свяжитесь с вашим представителем NetApp для консультации.

Настройки AFF по умолчанию

Тома, созданные в ONTAP, работающей на all-flash СХД семейства AFF, по умолчанию используют тонкое выделение пространства со всеми включенными технологиями эффективного хранения. Хотя БД Oracle, как правило, не выигрывают от дедупликации и могут содержать несжимаемые данные, тем не менее, настройки по умолчанию, подходят практически для всех рабочих нагрузок. Платформа ONTAP разработана для эффективной обработки всех типов данных и шаблонов ввода/вывода независимо от того, ведут ли они к экономии. Значения по умолчанию следует изменять только в том случае, если вы полностью осознаете причины, и отход от значений по умолчанию может принести пользу.

Общие рекомендации

- Пост-дедупликацию не следует использовать из-за возможного влияния на производительность при сканировании данных на наличие дублирующихся блоков, которых нет в БД Oracle. Дедупликация на лету, напротив, не должна вызывать проблем, поскольку она работает только с ранее выявленными дублирующимися блоками.
- Если тома и/или LUN создаются не по модели тонкого выделения, то следует отключить настройки обеспечения эффективности, поскольку использование этих функций не дает экономии. Кроме того, см. раздел 6.4 и убедитесь, что обеспечение эффективности полностью выключено в настройках, прежде чем использовать стратегию статического выделения ресурсов.
- Если хранится очень много архивных журналов, можно добиться большей эффективности, переме-

стив архивные журналы на том, использующий вторичное сжатие.

- Файл данных может содержать значительный объем несжимаемых данных, например, когда сжатие уже включено на уровне БД. Кроме того, БД может содержать сжатые объекты или быть зашифрована. В любом из этих случаев подумайте об отключении сжатия, чтобы обеспечить более эффективную работу на других томах, содержащих сжимаемые данные.
- Не используйте одновременно сжатие и дедупликацию с резервными копиями Oracle RMAN. Подробные сведения см. в разделе 6.1.

7 Тонкое выделение пространства

Тонкое выделение пространства – это конфигурирование большего пространства в СХД, чем технически доступно. Такое конфигурирование выполняется по-разному и является неотъемлемой частью многих функций ONTAP для среды БД Oracle.

Практически любое использование моментальных снимков требует тонкого выделения. Например, типичная 10-терабайтная БД в СХД NetApp содержит моментальные снимки за 30 дней. При этом в активной файловой системе приблизительно 10 ТБ данных видимы, а 300 ТБ выделены под моментальные снимки. Для хранения 310 ТБ информации обычно нужно 12-15 ТБ места. Активная база данных занимает 10 ТБ, а оставшимся 300 ТБ данных требуется только 2-5 ТБ пространства, поскольку сохраняются только изменения исходных данных.

Еще один пример динамического предоставления ресурсов – клонирование. Так, один из крупных заказчиков NetApp создал 40 клонов 80-терабайтной БД для использования разработчиками. Если бы все 40 разработчиков, использующих эти клоны, перезаписали каждый блок в каждом файле данных, то потребовалось бы более 3,2 ПБ пространства. На практике оборот невелик, а потребность в коллективном пространстве близка к 40 ТБ, так как на диске хранятся только изменения.

7.1 Управление пространством

При использовании тонкого выделения в среде Oracle нужно соблюдать осторожность, поскольку скорость изменения данных может неожиданно возрасти. Например, потребление пространства моментальными снимками может резко возрасти при повторном индексировании таблиц, либо неправильно расположенная резервная копия RMAN может записать большой объем данных за очень короткое время. Наконец, может оказаться трудно восстановить БД Oracle, если во время расширения файла данных в файловой системе закончится свободное пространство.

К счастью, эти риски можно снизить точной настройкой политик `volume-autogrow` и `snapshot-autodelete`. Как следует из названий, эти параметры позволяют пользователю создавать политики, которые автоматически освобождают пространство, занимаемое моментальными снимками, или увеличивают размер тома для размещения дополнительных данных. Потребности у всех заказчиков разные, так что есть много вариантов.

Полное описание этих функций см. в «Руководстве по управлению логическим хранилищем ONTAP».

7.2 Тонкое выделение LUN

В среде Oracle тонкое выделение активных LUN применяется ограниченно, поскольку Oracle сразу создает и инициализирует файлы данных максимального размера. Эффективность тонкого выделения активных LUN в среде файловой системы может со временем снижаться, так как удаленные и стертые данные занимают все больше и больше свободного пространства в файловой системе.

Есть одно исключение, когда используется LVM. При использовании LVM, таких как Veritas VxVM или Oracle ASM, базовые LUN делятся на экстенты, которые используются только при необходимости. Например, если размер БД начинается с 2 ТБ, но со временем может вырасти до 10 ТБ, то эту БД можно разместить на динамически предоставляемых LUN размером 10 ТБ, организованных в группу дисков LVM. Она будет занимать лишь 2 ТБ дискового пространства в момент создания и требовать дополнительного пространства только при выделении экстентов для компенсации разрастания БД. Этот процесс безопасен, пока ведется мониторинг пространства.

7.3 Частичное резервирование (fractional reservation)

Частичный резерв описывает поведение LUN в томе применительно к эффективности использования

пространства. Если для параметра fractional-reserve установлено значение 100%, то все данные в томе могут обновиться на 100% независимо от шаблона ввода-вывода без исчерпания пространства в томе.

В качестве примера использования мгновенного снимка рассмотрим БД на одном LUN размером 250 ГБ в томе размером 1 ТБ. Создание мгновенного снимка сразу же приведет к резервированию дополнительных 250 ГБ пространства в томе, чтобы гарантировать, что на томе не исчерпается все свободное место по какой-либо причине. Использование частичных резервов, вообще говоря, расточительно, поскольку крайне маловероятно, что потребуется перезаписать каждый байт на томе БД. Нет причин резервировать место для события, которое никогда не происходит. Тем не менее, если заказчик не может отслеживать расходование пространства в СХД и хочет быть уверен, что пространство никогда не будет исчерпано, то для использования моментальных снимков потребуется 100% частичное резервирование.

7.4 Сжатие и дедупликация

И сжатие, и дедупликация – это формы динамического предоставления ресурсов. Например, пространство, занимаемое БД размером 50 ТБ, может сжаться до 30 ТБ, давая экономию в 20 ТБ. Чтобы сжатие давало какие-либо выгоды, нужно либо использовать часть этих 20 ТБ для других данных, либо купить СХД с емкостью менее 50 ТБ. Результат – возможность хранить больше данных, чем технически доступно в СХД. С точки зрения БД, у вас все равно имеется 50 ТБ данных, пусть даже они и занимают на диске только 30 ТБ.

Всегда есть вероятность, что сжимаемость БД изменится, а реально занимаемое пространство увеличится. Это означает, что сжатием нужно управлять так же, как и при других формах динамического предоставления ресурсов – вести мониторинг и использовать `volume-autogrow` и `snapshot-autodelete`.

Сжатие и дедупликация рассматриваются далее более подробно в разделах “Сжатие” и “Дедупликация.”

7.5 Утилита ASM Reclamation и обнаружение нулевых блоков

ONTAP эффективно удаляет обнуленные блоки, записываемые в файл или LUN, когда включено сжатие на лету. Такие утилиты, как Oracle ASM Reclamation Utility (ASRU), работают путем записи нулей в неиспользуемые экстенты ASM. Это позволяет администраторам БД высвобождать пространство на системе хранения после удаления файлов. ONTAP перехватывает запись нулей и забирает выделенное пространство у LUN. Процесс высвобождения пространства выполняется очень быстро, потому что в СХД не записываются никакие данные.

С точки зрения БД группа дисков ASM содержит нули. Чтение этих областей LUN привело бы к потоку нулей, но ONTAP не хранит нули на дисках. Вместо этого в метаданные вносятся простые изменения – обнуленные области LUN внутренне помечаются как не содержащие данных.

По тем же причинам тестирование производительности на обнуленных данных не отражает реальной картины, поскольку блоки нулей на самом деле не обрабатываются как записи в массиве хранения.

Примечание. При использовании ASRU убедитесь, что установлены все программные исправления (патчи), рекомендованные компанией Oracle.

7.6 Сжатие и частичное резервирование

Сжатие – это одна из форм тонкого выделения ресурсов. Частичное резервирование влияет на использование сжатия с одним важным комментарием: пространство резервируется до создания моментального снимка. Обычно частичный резерв важен, только если существует моментальный снимок. Если его нет, то частичный резерв не важен. Но это не относится к случаю со сжатием. Если LUN создается на томе со сжатием, то ONTAP сохраняет пространство для размещения моментального снимка. Такое поведение может сбивать с толку во время конфигурирования, но это вполне ожидаемо.

В качестве примера рассмотрим том размером 10 ГБ с LUN размером 5 ГБ, сжатым до 2,5 ГБ без моментальных снимков. Рассмотрим два сценария:

- Если частичный резерв равен 100, то использовано будет 7,5 ГБ
- Если частичный резерв равен 0, то использовано будет 2,5 ГБ

В первом сценарии используется 2,5 ГБ дискового пространства для текущих данных и 5 ГБ дискового пространства для 100% перезаписи исходных данных в предположении использования моментального снимка. Во втором сценарии дополнительное пространство не резервируется.

Хотя эта ситуация может сбить с толку, на практике она вряд ли произойдет. Сжатие подразумевает

тонкое выделение пространства, а тонкое выделение в среде LUN требует частичного резервирования. Сжатые данные всегда могут быть перезаписаны чем-то несжимаемым, поэтому том должен быть создан с использованием тонкого выделения пространства, чтобы допускать сжатие, которое обеспечит хоть какую-то экономию.

NetApp рекомендует следующие конфигурации резерва:

- Установите для параметра `fractional-reserve` значение 0, когда есть возможность мониторинга пространства, а также включите параметры `volume-autogrow` и `snapshot-autodelete`.
- Установите для параметра `fractional-reserve` значение 100, если возможности мониторинга нет или если пространство невозможно исчерпать ни при каких обстоятельствах.

8 Оптимизация и измерения производительности

Точное тестирование производительности системы хранения, на которой размещена БД, является чрезвычайно сложной задачей и требует понимания таких аспектов, как:

- IOPS и пропускная способность
- Разница между приоритетными и фоновыми операциями ввода/вывода
- Влияние задержек на БД
- Многочисленные настройки ОС и сети, которые также влияют на производительность системы хранения

Кроме того, необходимо учитывать процессы БД не связанные с хранением данных. Начиная с какого-то момента, оптимизация производительности СХД не дает преимуществ, так как она перестает быть узким местом.

Большинство заказчиков, использующих БД, теперь выбирают all-flash массивы, что создает ряд дополнительных вопросов для рассмотрения. Например, рассмотрим тестирование производительности в системе с двумя узлами AFF8080:

- При соотношении чтение/запись = 75/25 два узла AFF8080 могут обеспечить более 300 000 операций случайного ввода/вывода в БД, прежде чем задержка превысит 1 мс. Это настолько превышает текущие требования большинства БД к производительности, что трудно спрогнозировать ожидаемое улучшение. СХД в значительной степени перестанет быть узким местом.
- Пропускная способность сети становится все более частым ограничителем производительности. Например, решения с традиционными жесткими дисками часто являются узким местом для производительности БД из-за очень высокой задержки ввода/вывода. Когда вызванные задержками ограничения снимаются all-flash массивом, узкое место часто перемещается в сеть. Это особенно заметно в виртуализированных средах и блейд-системах, в которых трудно визуализировать реальные связи в сети. Это может усложнить тестирование производительности, если саму СХД невозможно полностью использовать из-за ограничений пропускной способности.
- Сравнить производительность all-flash массива и массива вращающихся дисков, как правило, невозможно из-за радикально меньшей задержки all-flash массивов. Результаты тестов обычно не имеют смысла.
- Сравнивать производительность по пиковому значению IOPS с all-flash массивом часто бесполезно, поскольку БД не ограничены вводом/выводом СХД. Пусть, например, один массив может выдавать 500 000 операций случайного ввода/вывода в секунду, а другой – 300 000. В реальном мире эта разница не имеет значения, если БД тратит 99% своего времени на обработку данных процессорами. Рабочие нагрузки никогда не используют все возможности системы хранения. Напротив, пиковые возможности IOPS могут быть критически важными для платформы консолидации, в которой, как ожидается, система хранения будет загружена до предела возможностей.
- При любом тестировании СХД всегда учитывайте и задержку, и IOPS. Для многих предлагаемых на рынке систем хранения заявляются крайне высокие значения IOPS, но задержка делает эти значения IOPS бесполезными. Типичное целевое значение для all-flash массива – 1 мс. Лучше измерять не максимально возможное число операций ввода/вывода в секунду, а число операций ввода/вывода в секунду, которое может обеспечить система хранения, прежде чем средняя задержка превысит 1 мс.

8.1 Oracle Automatic Workload Repository и измерение производительности

«Золотой стандарт» для сравнения производительности Oracle – это отчет Oracle Automatic Workload

Repository (Автоматический репозиторий рабочих нагрузок, AWR).

Есть несколько типов отчетов AWR. С точки зрения хранилища отчет, сгенерированный командой `awrrpt.sql`, является наиболее полным и ценным, поскольку он нацелен на конкретный экземпляр БД и содержит ряд подробных гистограмм, на которых события ввода/вывода хранилища показаны с разбивкой по задержке.

Сравнение двух СХД по производительности в идеале предполагает выполнение одной и той же рабочей нагрузки на каждом массиве и создание отчета AWR, точно ориентированного на рабочую нагрузку. В случае очень длительной рабочей нагрузки можно использовать один отчет AWR за все время ее выполнения, включающий в себя время начала и окончания, но предпочтительно разбивать данные AWR на несколько отчетов. Например, если пакетное задание выполнялось с полуночи до 6 часов утра, создайте серию одночасовых отчетов AWR с полуночи до 1 часа ночи, с 1 часа ночи до 2 часов ночи и т.д.

В других случаях, наоборот, очень короткий запрос следует оптимизировать. Наилучшим вариантом является отчет AWR, основанный на моментальном снимке AWR, созданном в момент начала запроса, и втором моментальном снимке AWR, созданном в момент окончания запроса. В противном случае сервер БД должен быть неактивным, чтобы минимизировать фоновую активность, которая может скрыть активность анализируемого запроса.

Примечание. Если отчеты AWR недоступны, то хорошей альтернативой будут отчеты Oracle statspack. Они содержат большую часть той же статистики ввода/вывода, что и отчет AWR.

8.2 Oracle AWR и устранение проблем производительности

Отчет AWR также является наиболее важным инструментом анализа проблем производительности.

Как и в случае измерения производительности, устранение проблем требует точного измерения конкретной рабочей нагрузки. По возможности предоставляйте данные AWR, когда сообщаете о проблеме производительности в центр поддержки NetApp или когда работаете с командой по взаимодействию с заказчиками NetApp или партнера в связи с новым решением.

Предоставляя данные AWR, учитывайте следующие требования:

- Запустите команду `awrrpt.sql`, чтобы сгенерировать отчет. Отчет можно вывести либо в текстовом формате, либо в формате HTML.
- Если используете Oracle Real Application Clusters (RACs), то сгенерируйте отчеты AWR для каждого экземпляра в кластере.
- Задайте конкретное время, в которое проблема существовала. Максимально допустимое время, затраченное на создание отчета AWR, обычно составляет один час. Если проблема сохраняется в течение нескольких часов или связана с многочасовой операцией, такой как пакетное задание, то предоставьте несколько одночасовых отчетов AWR, которые охватывают весь анализируемый период.
- Если возможно, установите интервал создания моментальных снимков AWR в 15 минут. Это позволит выполнить более подробный анализ. Это также требует дополнительного выполнения скриптов `awrrpt.sql`, чтобы предоставлять отчет за каждые 15 минут.
- Если проблема вызвана очень быстро выполняющимся запросом, то предоставьте отчет AWR, основанный на моментальном снимке AWR, созданном в момент начала операции, и втором моментальном снимке AWR, созданном в момент окончания операции. В противном случае сервер БД должен быть неактивным, чтобы минимизировать фоновую активность, которая может скрыть активность анализируемой операции.
- Если проблема производительности возникает в одни моменты, но не наблюдается в другие, то предоставьте для сравнения дополнительные данные AWR, которые демонстрируют хорошую производительность.

8.3 calibrate_io

Команду `calibrate_io` никогда не следует использовать для тестирования, сравнения или измерения производительности СХД. Согласно документации Oracle, эта процедура калибрует возможности ввода/вывода хранилища.

Калибровка – не то же самое, что измерение производительности. Эта команда используется для ввода/вывода с целью помочь откалибровать операции БД и повысить их эффективность путем оптимизации

ввода/вывода на хосте. Поскольку ввод/вывод, выполняемый командой calibrate_io, не является реальным вводом/выводом, который возникает при работе пользователей с БД, результаты непредсказуемы и часто даже невоспроизводимы.

8.4 SLOB2

Предпочтительным инструментом оценки производительности БД стало ПО Silly Little Oracle Benchmark (SLOB2), которое разработал Кевин Клоссон (Kevin Closson) и которое доступно по этой ссылке. Для установки и настройки требуется несколько минут, а для создания паттернов ввода/вывода в определяемом пользователем табличном пространстве используется реальная БД Oracle. Это один из немногих доступных вариантов тестирования, который может насытить all-flash массив операциями ввода/вывода. Этот инструмент также полезен для создания гораздо менее интенсивных потоков ввода/вывода, имитирующих рабочие нагрузки хранилища с небольшим числом операций ввода/вывода в секунду, но с высокой чувствительностью к задержке.

8.5 Swingbench

Инструмент Swingbench может быть полезен для тестирования производительности БД, но перегрузить им хранилище чрезвычайно трудно. Компания NetApp не встречала тестов Swingbench, которые обеспечили бы ввод/вывод с интенсивностью, достаточной для того, чтобы серьезно нагрузить любой массив AFF. В редких случаях для оценки задержки хранилища можно использовать Order Entry Test (OET). Это может быть полезно, когда зависимость задержки от типа запроса к БД известна. Нужно убедиться, что хост и сеть правильно сконфигурированы, чтобы по максимуму использовать скоростные возможности all-flash массива.

8.6 HammerDB

HammerDB – это инструмент тестирования БД, который, помимо прочего, имитирует стандартные тесты производительности TPC-C и TPC-H. Создание достаточно большого набора данных для правильного выполнения теста может занять много времени, но он может быть эффективным инструментом для оценки производительности приложений OLTP и хранилищ данных.

8.7 Orion

Инструмент Oracle Orion обычно использовался с Oracle 9, но в нем не обеспечена совместимость с изменениями в различных ОС хоста. Он редко используется с Oracle 10 или Oracle 11 из-за несовместимости с ОС и конфигурацией хранилища.

Компания Oracle доработала инструмент, и он устанавливается по умолчанию вместе с Oracle 12c. Хотя этот продукт был доработан и использует многие из тех же вызовов, что и реальная БД Oracle, он не использует точно такой же путь выполнения кода или поведение операций ввода/вывода, как БД Oracle. Так, большая часть операций ввода/вывода Oracle выполняется синхронно, т.е. БД останавливается до тех пор, пока ввод/вывод не будет завершен, поскольку операция ввода/вывода выполняется в фоновом режиме. Простое заполнение СХД операциями случайного ввода/вывода не является воспроизведением реальных операций ввода/вывода Oracle и не позволяет напрямую сравнить системы хранения или измерить влияние изменений конфигурации.

Тем не менее, Orion можно использовать, например, для общего измерения максимальной возможной производительности конкретной конфигурации «хост-сеть-хранилище» или для оценки работоспособности СХД. При тщательном тестировании можно было бы разработать пригодные для использования тесты на базе Orion, чтобы сравнивать ссистемы хранения или оценивать влияние изменений конфигурации, при условии, что в параметрах учитываются IOPS, пропускная способность и задержка, а также делается попытка точно воспроизвести реалистичную рабочую нагрузку.

9 Общая конфигурация Oracle

Следующие параметры обычно применимы ко всем конфигурациям.

9.1 filesystemio_options

Параметр инициализации Oracle filesystemio_options контролирует использование асинхронного и прямого ввода/вывода. Вопреки распространенному мнению, асинхронный и прямой ввод/вывод не исключают друг друга. По наблюдениям NetApp, в средах заказчиков этот параметр часто настроен неправильно, что

напрямую создает многие проблемы производительности.

Асинхронный ввод/вывод означает, что операции ввода/вывода Oracle можно выполнять параллельно. До появления асинхронного ввода/вывода в различных ОС пользователи настраивали многочисленные процессы dbwriter и изменяли конфигурацию процессов сервера. При асинхронном вводе/выводе ОС сама выполняет ввод/вывод от имени ПО БД высокоэффективно и параллельно. При этом данные не подвергаются риску, а критически важные операции, такие как журналирование транзакций Oracle, по-прежнему выполняются синхронно.

Прямой ввод/вывод обходит буферный кэш ОС. Поток ввода/вывода в системе UNIX обычно проходит через буферный кэш ОС. Это полезно для приложений, которые не поддерживают внутренний кэш, но Oracle имеет собственный буферный кэш в SGA. Почти во всех случаях лучше включить прямой ввод/вывод и выделить оперативную память сервера для SGA, а не полагаться на буферный кэш ОС. Oracle SGA использует память более эффективно. Кроме того, когда поток ввода/вывода проходит через буфер ОС, он подвергается дополнительной обработке, которая увеличивает задержки. Увеличенные задержки особенно заметны при интенсивной записи, когда малое время задержки является ключевым требованием.

Возможные значения параметра f`ilesystemio_options`:

- **async.** Oracle отправляет запросы ввода/вывода в ОС для обработки. Этот процесс позволяет Oracle выполнять другую работу, а не ждать завершения ввода/вывода, что улучшает распараллеливание ввода/вывода.
- **directio.** Oracle выполняет ввод/вывод физических файлов напрямую, а не маршрутизирует ввод/вывод через кэш ОС хоста.
- **none.** Oracle использует синхронный и буферизованный ввод/вывод. В этой конфигурации более важны выбор между общими и выделенными серверными процессами и количество процессов dbwriter.
- **setall.** Oracle использует как асинхронный, так и буферизованный ввод/вывод.

Почти во всех случаях использование значения `setall` является оптимальным, но примите во внимание следующее:

- Некоторые заказчики в прошлом сталкивались с проблемами асинхронного ввода/вывода, особенно в предыдущих релизах Red Hat Enterprise Linux 4 (RHEL4). Однако об этих проблемах больше не сообщается, и асинхронный ввод/вывод стабилен во всех текущих ОС.
- Если БД использует буферизованный ввод/вывод, то переключение на прямой ввод/вывод также может потребовать изменения размера SGA. Отключение буферизованного ввода/вывода нивелирует выигрыш в производительности, обеспечиваемый кэшем ОС хоста для БД. Эта проблема устраняется увеличением размера оперативной памяти в SGA. Конечным результатом должно стать повышение производительности ввода/вывода.
- Хотя оперативную память почти всегда лучше использовать для Oracle SGA, чем для буферного кэширования ОС, определить наилучшее значение может оказаться невозможно. Например, может быть предпочтительно использовать буферизованный ввод/вывод с очень маленькими размерами SGA на сервере БД с множеством периодически активизирующихся экземпляров Oracle. Такая организация позволяет всем работающим экземплярам БД гибко использовать оставшуюся свободную оперативную память в ОС. Это очень маловероятная ситуация, но она наблюдалась на некоторых площадках заказчиков.

Примечание. Параметр `filesystemio_options` не оказывает влияния в средах DNFS и ASM. Использование DNFS или ASM автоматически ведет к использованию как асинхронного, так и прямого ввода/вывода.

Рекомендация компании NetApp:

- Установите для параметра `filesystemio_options` значение `setall`, но помните, что в некоторых случаях отсутствие буферного кэша на хосте может потребовать увеличения размера Oracle SGA.

9.2 db_file_multiblock_read_count

Параметр `db_file_multiblock_read_count` контролирует максимальное число блоков БД Oracle, которые Oracle читает за одну операцию во время последовательного ввода/вывода. Однако этот параметр не влияет на количество блоков, которые Oracle читает во время всех без исключения операций чтения, и не влияет на случайный ввод/вывод. Он влияет только на последовательный ввод/вывод.

Oracle рекомендует пользователям не устанавливать значение для этого параметра – в этом случае ПО БД будет автоматически выбирать оптимальное значение. Это в общем означает, что для этого параметра установлено значение, при котором размер ввода/вывода равен 1 МБ. Например, чтение 1 МБ блоками по 8 КБ потребует чтения 128 блоков, и поэтому значение по умолчанию для этого параметра будет 128.

Большинство проблем с производительностью БД, которые NetApp наблюдает на площадках клиентов, связаны с неправильной настройкой этого параметра. Для изменения этого значения в Oracle версий 8 и 9 были веские причины. В результате этот параметр может присутствовать в файлах `init.ora` (и пользователи могут не знать об этом), так как имевшаяся БД была обновлена до Oracle 10 или более поздних версий. Устаревшее значение 8 или 16 по сравнению со значением по умолчанию 128 значительно снижает производительность последовательного ввода/вывода.

Рекомендация компании NetApp:

- Параметра `db_file_multiblock_read_count` не должно быть в файле `init.ora`. Компания NetApp никогда не сталкивалась с ситуацией, когда изменение этого параметра повышало производительность, но во многих случаях это приводило к явному снижению пропускной способности последовательного ввода/вывода.

9.3 Размер блока redo

Oracle поддерживает блоки redo размером 512 Б или 4 КБ. Значение по умолчанию – 512 байт. Ожидается, что лучшим вариантом будет 512 байт, потому что при таком размере минимизируется объем данных, записываемых во время операций redo. Тем не менее, возможно, при очень высоких скоростях записи размер 4 КБ обеспечит выигрыш в производительности. Например, одна БД, в которой журналирование транзакций ведется на скорости 50 МБ/с, может работать эффективнее, если увеличить размер блока redo. СХД, поддерживающая множество БД с большим общим объемом журналирования транзакций, может выиграть от установки размера блока redo в значение 4 КБ. Это объясняется тем, что этот параметр устраняет неэффективную частичную обработку ввода/вывода в случае, когда нужно обновить только часть 4-килобайтного блока.

Неверно думать, что все операции ввода/вывода укладываются в один блок журнала транзакций. При очень высоких скоростях журналирования БД обычно выполняет очень большие операции ввода/вывода, состоящие из множества блоков redo. Фактический размер этих блоков redo в общем не влияет на эффективность журналирования.

Рекомендация компании NetApp:

- Изменяйте размер блока по умолчанию только при наличии веской причины (например, задокументированного требования для конкретного приложения) или по рекомендации службы поддержки заказчиков NetApp или Oracle.

9.4 Контрольные суммы и целостность данных

Компанию NetApp часто спрашивают, как обеспечить целостность данных в БД. Этот вопрос возникает особенно часто, когда заказчик, привыкший использовать потоковое резервное копирование Oracle RMAN, переходит к резервным копиям на основе моментальных снимков. Одна из особенностей RMAN состоит в выполнении проверки целостности во время операций резервного копирования. Хотя эта особенность и имеет некоторую ценность, основное ее преимущество достигается на БД, которые не используют функционал современных систем хранения. Когда для БД Oracle используются физические диски, повреждение почти наверняка произойдет по мере старения дисков. В настоящих современных системах хранения эта проблема решается с помощью использования контрольных сумм.

В системах хранения корпоративного уровня целостность данных защищается контрольными суммами на нескольких уровнях. Если данные повреждаются в IP-сети, то уровень TCP отклоняет пакетные данные и запрашивает повтор передачи. Протокол FC (как и инкапсулированные данные SCSI) включает в себя контрольные суммы. После запуска на СХД операционная система ONTAP обеспечивает защиту посредством RAID и контрольных сумм. Повреждение может возникнуть, но, как и в большинстве массивов корпоративного класса, оно обнаруживается и исправляется. Обычно отказывает диск целиком, вызывая перестроение RAID, так что целостность БД не нарушается. Реже ONTAP обнаруживает ошибку контрольной суммы, означающую, что данные на диске повреждены. После этого диск объявляется сбойным и также начинается перестроение RAID-группы. При этом целостность данных все равно не нарушается.

Архитектура файла данных и журнала транзакций Oracle также призвана обеспечить максимально возможный уровень целостности данных даже в экстремальных условиях. На самом базовом уровне блоки

Oracle включают в себя контрольную сумму и базовые логические проверки почти каждой операции ввода/вывода. Если не происходит аварийный отказ Oracle и табличное пространство не переводится в автономный режим, то данные не повреждаются. Степень проверки целостности данных регулируется, и Oracle также можно настроить для подтверждения записи. В результате почти все сценарии аварийных отказов и сбоев допускают восстановление, а в крайне редких случаях, когда это невозможно, повреждение быстро обнаруживается.

Большинство заказчиков NetApp, использующих БД Oracle, прекращают использовать RMAN и другие продукты резервного копирования после перехода к резервным копиям на основе моментальных снимков. Тем не менее, по-прежнему существуют ситуации, где RMAN можно использовать для восстановления на уровне блоков посредством SMO. Однако в повседневной деятельности RMAN, NetBackup и другие продукты используются только изредка для создания ежемесячных или ежеквартальных архивных копий.

Некоторые заказчики предпочитают периодически запускать dbv для проверки целостности своих БД. NetApp не рекомендует так делать, поскольку это создает избыточную нагрузку ввода/вывода. Как обсуждалось выше, если в БД ранее не было проблем, то вероятность найти проблему с помощью dbv близка к нулю, но зато эта утилита очень сильно нагружает сеть и СХД операциями последовательного ввода/вывода. Если нет причины подозревать повреждение данных (например, из-за известной ошибки Oracle), то не стоит запускать `dbv`.

10 Флэш-технологии

Подробное описание использования технологий флэш-памяти и SSD для БД Oracle выходит за рамки этого документа, но некоторые общие вопросы и заблуждения обсудить нужно. Все принципы, описанные в этом разделе, в равной степени применимы ко всем протоколам и файловым системам, включая Oracle ASM.

10.1 SSD-агрегаты

По поводу использования SSD и флэш-накопителей для журналов транзакций есть много путаницы. Высокопроизводительное журналирование транзакций требует записи данных на SSD. SSD-накопитель может быть полезен для повышения производительности журналирования, если используется в виде устройства, подключенного напрямую (внутренние диски или DAS), но массивы хранения NetApp уже содержат энергонезависимое зеркалируемое твердотельное хранилище на основе NVRAM или NVMEM. Когда БД Oracle выполняет операцию записи, запись подтверждается, как только она журналируется в NVRAM или NVMEM. Тип дисков, на которые в конечном счете производится запись, не влияет на производительность записи напрямую.

В лучшем случае использование SSD-агрегата или платформы AFF для размещения последовательных записей (таких как журналирование транзакций) или для ввода/вывода временных файлов данных не окажет никакого эффекта. Однако в некоторых случаях выбор AFF косвенно повышает производительность записи. Например, система с интенсивным случайным вводом/выводом, который перегружает обычные жесткие диски, может достичь точки, когда диски больше не смогут принимать входящие операции записи со скоростью, достаточной для того, чтобы предотвратить заполнение NVMEM/NVRAM. В этих случаях переход на SSD-агрегат или платформу AFF может повысить производительность redo, но это непрямая выгода. Проблема производительности записи будет решена, потому что эта система лучше сможет обрабатывать случайный ввод/вывод. Затем поведение операций записи вернется к норме, при этом все входящие операции записи будут записываться в NVMEM/NVRAM без задержки.

Иногда заказчики совершали ошибки планирования, из-за которых снижалась производительность SSD-агрегата. Хотя SSD-накопители имеют гораздо более высокую производительность, чем обычные жесткие диски, SSD-агрегаты иногда содержат гораздо меньше устройств, чем SAS- или SATA-агрегаты в системе. Например, компания NetApp обнаружила серьезные проблемы с производительностью в средах заказчиков, вызванные перемещением рабочих нагрузок с интенсивной последовательной записью, включая журналы транзакций, из большого SAS-агрегата, который может содержать 100 дисков, в небольшой SSD-агрегат, состоящий всего лишь из 4 или 5 устройств. SSD-диски могут быть быстрее, чем SAS, но не бесконечно быстрее.

Основной областью применения SSD-агрегатов является обслуживание рабочих нагрузок со случайным вводом/выводом. Особенно хорошо размещать на SSD-дисках индексы. Другие типы операций ввода/вывода не должны пострадать, если только агрегат не состоит из слишком малого количества дисков, но не стоит ждать улучшения производительности, если предыдущая система не была сильно перегружена.

10.2 Гибридные агрегаты: Flash Pool

Технология NetApp Flash Pool™ уменьшает время задержки случайного чтения, что обычно является основным узким местом производительности БД Oracle. Кроме того, Flash Pool помогает экономить деньги. Многие СХД для Oracle имеют значительное количество жестких дисков, обслуживающих пики запросов на произвольное чтение с минимальной задержкой. Небольшое пространство, выделенное из пула флэш-накопителей, может заменить большое количество жестких дисков.

Использование технологии Flash Pool для кэширования записи напрямую не влияет на производительность записи, так как запись производится в первую очередь в NVRAM или NVMEM. Если говорить о задержке, то ввод/вывод завершается, когда данные журналируются в NVRAM или NVMEM. Тип носителя, на котором впоследствии сохраняется входящая запись, сам по себе на производительность не влияет. Однако, если кэширование записи с помощью Flash Pool снижает нагрузку на жесткие диски, то это косвенно может повысить производительность записи. Это может повысить общую производительность ввода/вывода всего массива.

Кэширование записи с помощью Flash Pool также уменьшает время задержки чтения для случайно перезаписанных блоков, которые быстро читаются снова. Этот процесс не применим ко всем БД, потому что БД обычно сохраняет копию записанного блока. По мере увеличения буферного кэша Oracle и кэширования все большего количества операций записи необходимость в повторном чтении блока с диска становится все менее вероятной. В таких случаях может быть предпочтительнее отключить кэширование записи и зарезервировать ценное флэш-пространство для операций случайного чтения.

С другой стороны, преимущество есть и в том, чтобы не пропускать повторные перезаписи одних и тех же блоков дальше слоя SSD, чтобы снизить нагрузку на жесткие диски. Эта проблема может возникнуть, когда буферный кэш Oracle сильно нагружен, а быстро устаревающие блоки удаляются из кэша, вскоре после чего возникает необходимость прочитать их снова.

Общие рекомендации компании NetApp:

- Не меняйте установленную по умолчанию политику Flash Pool, которая включает в себя кэширование как случайного чтения, так и случайной записи.
- Хотя кэширование записи может и не дать особых выгод, общая интенсивность случайной записи, наблюдаемая в большинстве БД Oracle, не настолько высока, чтобы вызвать чрезмерное использование пространства SSD-накопителей. По умолчанию кэширование записи доступно при необходимости.

Основным исключением для использования этих рекомендаций является рабочая нагрузка БД со следующими характеристиками:

- Рабочая нагрузка является основной нагрузкой на агрегат и, следовательно, на нее приходится большая часть операций кэширования Flash Pool.
- Известно, что рабочая нагрузка ограничена задержкой случайного чтения.
- Активность записи относительно низкая.
- Буферный кэш Oracle относительно велик.

В таких случаях изменение политики кэширования записи Flash Pool на none может быть целесообразным. Тогда для кэширования чтения на SSD-накопителях будет доступно максимальное пространство.

Технология Flash Pool часто полезна для резервных БД Oracle, включая использование с Oracle DataGuard, так как у резервной БД обычно нет настоящего буферного кэша. Эта ситуация порождает ресурсоемкий шаблон ввода/вывода, когда одни и те же блоки читаются, обновляются и записываются многократно. Flash Pool перехватывает эту концентрированную перезапись в слое SSD, снижая нагрузку на жесткие диски. До появления таких технологий, как Flash Pool, резервная БД нередко требовала больше жестких дисков, чем основная БД, которая являлась источником репликации.

10.3 Платформы AFF

NetApp AFF делает SSD-агрегаты еще более полезными благодаря повышению производительности и оптимизации поведения специально для all-flash платформы. Полная документация доступна на веб-сайте поддержки NetApp.

Следует учитывать, что флэш-память – это не только IOPS. У нее есть и другие преимущества, такие как непрерывность и предсказуемость производительности, меньшее энергопотребление, меньшее тепловыделение и общая перспективность решения.

Во многих случаях all-flash платформа может снизить затраты, так как позволяет не устанавливать жесткие диски один за другим только для того, чтобы обеспечить малое время отклика. Затраты продолжают радикально снижаться, в результате чего все больше и больше заказчиков выбирают AFF.

11 Конфигурация Ethernet

Параметры TCP/IP, необходимые для установки ПО БД Oracle, обычно достаточны для обеспечения хорошей производительности всех ресурсов хранения NFS или iSCSI. В некоторых случаях компания NetApp наблюдала рост производительности в 10-гигабитных средах после выполнения конкретных рекомендаций производителя сетевого адаптера.

11.1 Управление потоком в сетях Ethernet

Эта технология позволяет клиенту попросить отправителя остановить на время передачу данных. Обычно это делается потому, что получатель не может обработать входящие данные достаточно быстро. Когда-то запрос к отправителю прекратить передачу причинял меньший вред, чем необходимость для получателя отбрасывать пакеты из-за заполнения буферов. Сегодня это уже не так, учитывая TCP-стеки, используемые в ОС. На самом деле управление потоком создает проблем больше, чем решает.

В последние годы проблемы производительности, вызванные управлением потоком в сети Ethernet, усиливаются. Это происходит потому, что в сети Ethernet управление потоком работает на физическом уровне. Если конфигурация сети позволяет любому серверу БД отправлять в СХД запрос на управление потоком Ethernet, то результатом является пауза ввода/вывода для всех подключенных клиентов. Поскольку один контроллер СХД обслуживает все больше и больше клиентов, возрастает вероятность того, что один или несколько из них отправят запросы на управление потоком. Проблема часто наблюдается на площадках заказчиков с экстенсивной виртуализацией ОС.

Сетевая карта в системе NetApp не должна получать запросы на управление потоком. Метод, используемый для достижения этого результата, зависит от производителя сетевого коммутатора. В большинстве случаев для управления потоком на коммутаторе Ethernet можно установить значение `receive desired` или `receive on,` чтобы запрос на управления потоком не пересылать контроллеру системы хранения. В других случаях сетевое соединение на контроллере СХД может не позволить отключить управление потоком. В этих случаях клиенты должны быть сконфигурированы так, чтобы никогда не отправлять запросы на управление потоком, путем изменения конфигурации сетевой карты либо на самом сервере БД, либо на портах коммутатора, к которым подключен сервер БД.

Рекомендация компании NetApp:

- Убедитесь, что контроллеры СХД NetApp не принимают пакеты управления потоком Ethernet. Обычно это можно сделать, настроив порты коммутатора, к которым подключен контроллер, но некоторые аппаратные коммутаторы имеют ограничения, которые могут потребовать изменений на стороне клиента.

11.2 Jumbo-кадры

Было показано, что использование jumbo-кадров несколько повышает производительность в 1-гигабитных сетях, уменьшая нагрузку на ЦП и сеть, но это повышение обычно невелико. Тем не менее, NetApp рекомендует по возможности использовать jumbo-кадры, чтобы реализовать любые пути повышения производительности и обеспечить перспективу решения на будущее.

Использование jumbo-кадров в 10-гигабитной сети почти обязательно. Это объясняется тем, что без jumbo-кадров большинство 10-гигабитных сетей достигают предела по числу пакетов в секунду раньше, чем скорости 10 Гбит/с. Jumbo-кадры повышают эффективность обработки TCP/IP, поскольку с ними сервер БД, сетевые карты и СХД обрабатывают меньшее количество больших пакетов. Повышение производительности зависит от сетевой карты, но в любом случае оно существенно.

Применительно к реализации jumbo-кадров существует распространенное, но неверное убеждение, что все подключенные устройства должны поддерживать jumbo-кадры и что размер MTU должен соответствовать сквозному. На самом деле нет – два оконечных сетевых устройства согласуют максимальный взаимоприемлемый размер кадра при установлении соединения. В типичной среде для сетевого коммутатора значение MTU равно 9216, для контроллера NetApp – 9000, а для клиентов – 9000 и 1514. Клиенты, поддерживающие MTU = 9000, могут использовать jumbo-кадры, а клиенты, поддерживающие только 1514, могут договориться о меньшем значении.

Такие проблемы редки в полностью коммутируемой среде. Однако в маршрутизируемой среде позаботьтесь о том, чтобы ни один промежуточный маршрутизатор не был настроен на принудительную фрагментацию jumbo-кадров.

Рекомендация компании NetApp:

- Jumbo-кадры желательны, но не обязательны для сетей Ethernet 1 Гбит/с (GbE).
- Jumbo-кадры необходимы для обеспечения максимальной производительности в сети Ethernet 10 Гбит/с (10GbE).

11.3 Параметры TCP

Следующие три параметра часто настраивают неправильно: временны́е метки TCP (TCP timestamps), селективное подтверждение (selective acknowledgment, SACK) и масштабирование TCP-окна (TCP window scaling). Во многих устаревших документах, которые можно найти в интернете, для повышения производительности рекомендуется отключать один или несколько из этих параметров. Эта рекомендация имела под собой определенные основания много лет назад, когда ЦП были гораздо слабее и уменьшать накладные расходы на обработку TCP при каждой возможности имело смысл.

Однако в современных ОС отключение любой из этих функций TCP обычно не дает ощутимых преимуществ и даже может снизить производительность. Снижение производительности наиболее вероятно в виртуализированных сетевых средах, поскольку эти функции необходимы для эффективной обработки потерь пакетов и изменений качества сети.

Рекомендация компании NetApp:

- Разрешить временны́е метки TCP, SACK и масштабирование окна TCP на хосте.

12 Общая конфигурация NFS

12.1 Версии NFS

Ранее документация Oracle требовала использования NFSv3, но последние обновления и изменения политики поддержки позволяют использовать NFSv4 с Oracle 12.1.0.2 и выше, в том числе при использовании как одного экземпляра, так и Oracle RAC. В настоящее время NetApp не поддерживает NFSv4.1.

12.2 Таблицы TCP-слотов

В NFSv3 таблицы TCP-слотов эквивалентны глубине очереди адаптера (HBA). Эти таблицы контролируют количество операций NFS, ожидающих выполнения в каждый момент времени. Значение по умолчанию обычно равно 16, что слишком мало для оптимальной производительности. Противоположная проблема возникает в более новых ядрах Linux, которые могут автоматически увеличивать лимит таблицы TCP-слотов настолько, что это перегрузит NFS-сервер запросами.

Чтобы получить оптимальную производительность и избежать проблем, настройте параметры ядра, управляющие таблицами TCP-слотов.

Выполните команду `sysctl -a | grep tcp.*.slot_table` и проверьте следующие параметры:

```
# sysctl -a | grep tcp.*.slot_table
sunrpc.tcp_max_slot_table_entries = 128
sunrpc.tcp_slot_table_entries = 128
```

Все системы Linux должны содержать параметр `sunrpc.tcp_slot_table_entries`, но лишь немногие содержат параметр

`sunrpc.tcp_max_slot_table_entries`. Для обоих нужно установить значение 128.

12.3 Установка ПО и патчей

Наличие следующих параметров монтирования в `ORACLE_HOME` приводит к отключению кэширования на хосте:

```
cio, actimeo=0, noac, forcedirectio
```

Это может серьезно замедлить установку ПО Oracle и исправлений. Многие заказчики временно удаляют эти параметры монтирования в процессе установки или исправления двоичных файлов Oracle. Это удале-

ние можно выполнить безопасно, если пользователь убедится, что никакие другие процессы не используют активно целевой `ORACLE_HOME` во время установки ПО или исправлений.

12.4 ONTAP и NFS Flow Control

В некоторых случаях использование ONTAP требует изменения параметра ядра Oracle или Linux. Причина связана с управлением потоком в NFS, поэтому не путайте эти изменения с управлением потоком в Ethernet. Управление потоком в NFS позволяет NFS-серверу, например, ONTAP, ограничивать сетевое взаимодействие с NFS-клиентом, который не подтверждает получение данных. Это защищает NFS-сервер в тех случаях, когда неисправный NFS-клиент запрашивает данные со скоростью, превышающей его способность обрабатывать ответы. Без защиты сетевые буферы NFS-сервера заполняются пакетами, получение которых не подтверждено.

В редких случаях пиковая нагрузка по вводу/выводу, создаваемая как DNFS-клиентами Oracle, так и более новыми NFS-клиентами Linux, может превышать возможности самозащиты NFS-сервера ONTAP. NFS-клиент отстает с обработкой входящих данных, при этом продолжая отправлять запросы на все новые и новые данные. Это отставание может привести к проблемам с производительностью и стабильностью при подключении NFS.

Хотя проблемы встречаются редко, NetApp рекомендует принимать следующие меры защиты, признанные наилучшими. Эти меры применяются только к ONTAP, и изменения не должны снижать производительность.

- Если используется Oracle DNFS, то для параметра `DNFS_BATCH_SIZE` установите значение `128`. Этот параметр есть в Oracle 11.2.0.4 и более новых версиях. Если это невозможно, то не используйте DNFS.
- Убедитесь, что для обоих параметров таблицы TCP-слотов, которые обсуждались ранее, установлено значение 128.
- Команда Oracle `calibrate_io` не работает, если для параметра `DNFS_BATCH_SIZE` установлено не значение по умолчанию. Если нужно выполнить калибровку ввода/вывода, то удалите параметр `DNFS_BATCH_SIZE` на время калибровки.

12.5 Direct NFS

DNFS-клиент Oracle разработан для обхода стека NFS операционной системы хоста и выполнения файловых операций NFS непосредственно на NFS-сервере. Чтобы его включить, нужно лишь изменить библиотеку Oracle Disk Manager. Инструкции для этого процесса приведены в документации Oracle.

Использование DNFS приводит к общему повышению производительности ввода/вывода и снижает нагрузку на хост и СХД, так как ввод/вывод выполняется наиболее эффективным способом. Кроме того, Oracle DNFS позволяет использовать множество путей и обеспечивает отказоустойчивость. Например, можно объединить два интерфейса 10 Гбит/с, чтобы обеспечить пропускную способность 20 Гбит/с. Отказ одного интерфейса приводит к повторной попытке ввода/вывода через другой интерфейс. В общем, работа очень похожа на использование множественных путей в FC.

Когда используется DNFS, крайне важно, чтобы были установлены все патчи, описанные документе Oracle Doc 1495104.1. Если патч установить нельзя, то нужно оценить среду и убедиться, что ошибки, описанные в этом документе, не вызывают проблем. В некоторых случаях невозможность установки необходимых патчей препятствует использованию DNFS.

Предупреждение

- Перед использованием DNFS проверьте, что установлены все патчи, описанные в документе Oracle 1495104.1
- Начиная с Oracle 12c, DNFS включает поддержку NFSv3, NFSv4 и NFSv4.1. NetApp поддерживает использование v3 и v4 для всех клиентов, однако NFSv4.1 не поддерживается для использования совместно с Oracle DNFS.
- Не используйте DNFS с любыми вариантами циклического (round-robin) разрешения имен, включая DNS, DDNS, NIS или иными. Это включает и систему балансировки с помощью DNS доступную в ONTAP. Когда БД Oracle разрешает имя хоста в IP адрес с помощью DNFS, адрес не должен меняться во всех последующих запросах, иначе это может привести к «падению» сервера БД и даже к порче данных.

12.6 Direct NFS и доступ к файловой системе хоста

Использование DNFS может иногда вызывать проблемы для приложений или действий пользователей, которые зависят от видимых файловых систем, смонтированных на хосте, поскольку DNFS-клиент обращается к файловой системе, минуя ОС хоста. DNFS-клиент может создавать, удалять и изменять файлы без ведома ОС.

Если используются параметры монтирования для БД с одним экземпляром, то они разрешают кэширование атрибутов файла и каталога, что также означает, что и содержимое каталога кэшируется. Поэтому DNFS может создать файл, и есть небольшая задержка, прежде чем ОС перечитает содержимое каталога и файл станет видимым для пользователя. Обычно это не составляет проблемы, но в редких случаях у таких утилит, как SAP BR*Tools, могут возникать сложности. В этом случае устраните проблему, изменив параметры монтирования, чтобы использовать рекомендации для Oracle RAC. Это изменение приводит к отключению всего кэширования на стороне хоста.

Изменяйте параметры монтирования только тогда, когда (a) используется DNFS и (b) проблема возникает из-за того, что файл виден не сразу. Если DNFS не используется, то использование параметров монтирования Oracle RAC в БД с одним экземпляром приводит к снижению производительности.

Примечание. См. примечание о `nosharecache` в разделе “Параметры монтирования Linux NFS” применительно к конкретной проблеме с DNFS, характерной для Linux, которая может привести к необычным результатам.

12.7 ADR и NFS

Некоторые заказчики сообщали о проблемах производительности, вызванных чрезмерно интенсивным вводом/выводом данных в ADR. Эта проблема обычно не возникает до тех пор, пока не будет накоплено много данных о производительности. Причина чрезмерно интенсивного ввода/вывода неизвестна, но эта проблема, по-видимому, вызывается процессами Oracle, многократно сканирующими целевой каталог на наличие изменений.

Удаление параметров монтирования `noac` и/или `actimeo=0` разрешает ОС хоста выполнять кэширование и снижает нагрузку по вводу/выводу на хранилище.

Рекомендация компании NetApp:

- Не размещайте данные `ADR` в файловой системе с параметрами `noac` или `actimeo=0`, поскольку это может вызвать проблемы с производительностью. При необходимости выделяйте для данных `ADR` отдельную точку монтирования.

13 Общая конфигурация SAN

13.1 Зонирование

Зона FC никогда не должна содержать больше одного инициатора. Вначале может показаться, что конфигурация с несколькими инициаторами работоспособна, но взаимовлияние инициаторов в конечном итоге будет ухудшать производительность и стабильность.

Зоны с несколькими таргетами (multitarget zones) обычно считаются безопасными, хотя в редких случаях поведение целевых FC-портов на продуктах разных изготовителей вызывало проблемы. Например, не следует включать в одну зону целевые порты из NetApp и массива хранения EMC. Кроме того, размещение системы хранения NetApp и ленточного устройства в одной зоне может вызвать проблемы.

13.2 Выравнивание LUN

Выравнивание LUN – это оптимизация ввода/вывода относительно базовой структуры файловой системы. В системе NetApp хранилище организовано блоками по 4 КБ. 8-килобайтный блок файла данных Oracle нужно выровнять так, чтобы он занимал ровно два 4-килобайтных блока. Если из-за ошибки конфигурирования LUN граница между блоками сместится на 1 КБ в любую сторону, то каждый 8-килобайтный блок Oracle будет существовать в трех разных 4-килобайтных блоках СХД, а не в двух. Это может увеличить задержку и породить дополнительные операции ввода/вывода в СХД.

Как правило, о выравнивании LUN нужно заботиться только в том случае, если не используется менеджер логических томов. На практике это означает, что внимания требуют в первую очередь Linux и Solaris.

Если физический том в группе логических томов определен на всем дисковом устройстве (без создания разделов), то первый 4-килобайтный блок LUN выравнивается по первому 4-килобайтному блоку в СХД. Это – правильное выравнивание. Проблемы возникают при использовании разделов, потому что из-за них точка начала смещается туда, где ОС использует LUN. Пока смещение кратно 4 КБ, LUN остается выровненным.

В средах Linux следует организовывать группы логических томов на всем дисковом устройстве. Если же нужно создать раздел, то проверьте выравнивание командой `fdisk –u` и убедитесь, что начало каждого раздела кратно восьми. Это означает, что раздел начинается в точке, кратной восьми секторам по 512 байт, т.е. 4 КБ.

Кроме того, выравнивание блоков сжатия рассмотрено в разделе “Сжатие.” Любая компоновка, которая выровнена по границам 8-килобайтных блоков сжатия, также выровнена по границам 4-килобайтных блоков.

Выравнивание в средах Solaris более сложное. Дополнительные сведения см. в соответствующей Документации по утилитам хоста.

Предупреждение
В среде Solaris x86 требуется особенное внимание к правильному выравниванию, т.к. большинство конфигураций имеет несколько уровней разделов. Секции разделов Solaris x86 обычно существуют поверх стандартной таблицы разделов (MBR)

13.3 Предупреждения о невыровненном LUN

При журналировании транзакций в Oracle обычно генерируются невыровненные операции ввода/вывода, которые могут вызывать сбивающие с толку предупреждения о невыровненных LUN в ONTAP. При журналировании транзакций в Oracle выполняется последовательная перезапись файла журнала транзакций, причем операции записи имеют разный размер. Операция записи в журнал, которая не выравнивается по границам 4 КБ, обычно не вызывает проблем с производительностью, поскольку следующая операция записи в журнал дополняет блок. В результате ONTAP может обрабатывать почти все записи как полные 4-килобайтные блоки, даже если данные в некоторые из них были записаны двумя отдельными операциями.

Проверьте выравнивание, используя утилиты `sio` или `dd`, которые могут генерировать ввод/вывод с определенным размером блока. Статистику выравнивания ввода/вывода в СХД можно просмотреть, выполнив команду `stats`. Дополнительную информацию см. в «Приложении B. Проверка выравнивания в WAFL».

13.4 Определение размера LUN

LUN – это виртуализованный объект в ONTAP, который распределен по всем жестким дискам агрегата, на котором он размещен. Производительность LUN не зависит от его размера, поскольку LUN использует весь потенциал агрегата.

Из соображений удобства заказчик может использовать LUN определенного размера. Например, если БД построена на группе дисков ASM, состоящей из двух LUN по 1 ТБ каждый, то размер этой группы дисков ASM должен увеличиваться с шагом 1 ТБ. Возможно, лучше создать группу дисков ASM из восьми LUN по 500 ГБ каждый, чтобы увеличивать размер группы дисков с меньшим шагом.

Задавать какой-то универсальный стандартный размер LUN не рекомендуется, потому что это может усложнить управление. Например, стандартный LUN размером 100 ГБ может хорошо работать, когда БД занимает от 1 ТБ до 2 ТБ, но для БД объемом 20 ТБ потребуется 200 LUN. Из-за этого сервер будет перезагружаться дольше, пользовательские интерфейсы будут содержать больше объектов управления, а такие продукты, как SMO, должны будут выполнять обнаружение множества объектов. Этих проблем можно избежать, используя меньшее количество LUN большего размера.

Примечание.

- Количество LUN важнее их размера.
- Размер LUN в основном определяется требованиями к количеству LUN.
- Старайтесь не увеличивать количество LUN сверх необходимого.

13.5 Изменение размера LUN и изменение размера на основе LVM

Когда файловая система на основе SAN достигает предела емкости, увеличить доступное пространство можно двумя способами:

- Увеличить размер набора LUN.
- Добавить LUN к существующей группе томов, тем самым увеличив общий размер логических томов, которые в ней содержатся.

Поддерживаются оба варианта, но увеличить размер LUN обычно сложно и иногда рискованно. Вот некоторые соображения на этот счет:

- LUN, созданные в ONTAP, можно увеличить примерно в 10 раз от первоначального размера. Это ограничение основано на характерной структуре геометрии диска. Иногда можно увеличить размер более, чем в 10 раз, но это может потребовать изменений в таблицах разделов, что требует глубокого понимания конфигурации диска на уровне хоста.
- В качестве резервной меры NetApp рекомендует, прежде чем пытаться изменить размер LUN, завершить работу БД и создать копии в виде мгновенных снимков. Хотя это не является требованием во всех случаях, существует некоторый риск сбоя и ошибки пользователя при повторном обнаружении недавно увеличенных LUN на уровне ОС хоста.
- Если говорить о сложностях, связанных с изменением размера LUN, то единственным исключением является Microsoft Windows, которая предлагает безопасный и неразрушающий способ увеличения размеров LUN с помощью NetApp SnapDrive для Windows[1].

Хотя емкость можно увеличить изменением размера LUN, лучше все же использовать LVM, включая Oracle ASM. LVM позволяет не менять размер LUN – и это одна из основных причин использования LVM. LVM объединяет несколько LUN в виртуальный пул хранения. Логические тома, выделяемые из этого пула дисков, находятся под управлением LVM, а их размер можно легко изменить. Дополнительным преимуществом является предотвращение перегрузки конкретного диска – ведь логический том распределен по всем доступным LUN. Прозрачную миграцию обычно можно выполнить с помощью менеджера томов, чтобы переместить базовые экстенты логического тома в новые LUN.

По вышеуказанным причинам не рекомендуется стратегия, предусматривающая изменение размера LUN, а рекомендуется использовать LVM.

13.6 Количество LUN

В отличие от размера LUN, количество LUN влияет на производительность. На производительность БД Oracle влияет способность выполнять параллельный ввод/вывод через уровень SCSI. В результате два LUN имеют более высокую производительность, чем один. Самый простой способ повысить степень параллелизма – использовать LVM, такой как Veritas VxVM, Linux LVM2 или Oracle ASM.

Обычно заказчики NetApp получали минимум выгоды от увеличения числа LUN сверх 8, хотя тестирование all-flash сред с очень интенсивными операциями случайного ввода/вывода показало дальнейшее улучшение при увеличении числа LUN до 64. NetApp рекомендует строить группу томов с размером экстента, который обеспечивает равномерное распределение операций ввода/вывода. Например, группа томов общим размером 1 ТБ, состоящая из 10 LUN по 100 ГБ, и экстент размером 100 МБ дадут в целом 10 000 экстентов (1 000 экстентов на LUN). В результате ввод/вывод в БД, размещенной на этой группе томов размером 1 ТБ, должен быть равномерно распределен по всем 10 LUN.

Распределение логического тома по экстентам – это не то же самое, что чередование, хотя концепция похожа.

Менеджеры логических томов разбивают LUN на относительно большие экстенты, чтобы упростить управление данными. Большие экстенты предпочтительны, поскольку они обеспечивают лучшую эффективность операций упреждающего чтения. Например, экстент Oracle ASM (его также называют единицей выделения, Allocation Unit) размером 64 МБ позволяет осуществлять упреждающее чтение массива данных в хранилище, чтобы помочь перенести полные 64 МБ данных, прежде чем ASM перейдет в следующий экстент. Меньшие единицы выделения означают, что упреждающее чтение должно сбрасываться чаще, так как операции чтения перемещаются из экстента в экстент.

Более важный случайный ввод/вывод все равно должен быть равномерно распределен по LUN даже при большом размере экстента, если только БД не имеет чрезвычайно концентрированного ввода/вывода.

1 Поддержка данного ПО заканчивается 31 мая 2021 г.

В отличие от распределенных экстентов, следует избегать настоящего чередования (striping). Чередование в основном предназначено для относительно медленных жестких дисков. Например, если было известно, что приложение считывает фрагменты по 1 МБ, то можно создать набор с чередованием из восьми LUN с шириной чередования 128 КБ. В результате операция ввода/вывода 1 МБ может выполняться как восемь одновременных операций ввода/вывода по 128 КБ на каждом LUN. На современных БД и СХД это почти никогда не приносит выгоды. Более того, неправильная настройка группы томов с чередованием может привести к снижению производительности.

Большинство БД ограничены производительностью случайного ввода/вывода, а не последовательного. Файл данных, распределенный по множеству экстентов, позволяет рандомизировать множество случайных операций ввода/вывода по многим экстентам. Это означает, что все LUN в группе томов используются равномерно и что ни один конкретный LUN не ограничивает производительность.

Рекомендация компании NetApp:

- Как правило, от 4 до 8 LUN достаточно для обеспечения хорошей производительности ввода/вывода файлов данных. Менее четырех LUN могут ограничивать производительность из-за ограничений в реализациях SCSI на хосте.
- Не используйте чередование с небольшими страйпами. Вместо этого включите политику LVM, которая распределяет данные по большим экстентам на каждом LUN, чтобы гарантировать, что каждый файл данных распределен по всем доступным LUN.

13.7 Размер блока файла данных

В некоторых ОС размер блока файловой системы можно выбирать. Для файловых систем, поддерживающих файлы данных, блок должен иметь размер 8 КБ при использовании сжатия. Если сжатие не требуется, то можно использовать блок размером 8 КБ или 4 КБ.

В некоторых ОС размер блока файловой системы можно выбирать. Для файловых систем, поддерживающих файлы данных, блок должен иметь размер 4 КБ. Если файл данных помещается в файловую систему с 512-байтными блоками, то возможны нарушения выравнивания файлов. LUN и файловая система могут быть правильно выровнены согласно рекомендациям NetApp, но ввод/вывод файлов может оказаться не выровненным. Это может вызвать серьезные проблемы производительности.

13.8 Размер блока redo

Файловые системы, поддерживающие журналы транзакций, должны использовать блок с размером, кратным размеру блока redo. Обычно для этого требуется, чтобы и файловая система журнала транзакций, и сам журнал транзакций использовали блок размером 512 байт. При очень высоких скоростях журналирования транзакций 4-килобайтный блок может работать лучше, потому что высокие скорости журналирования транзакций позволяют выполнять ввод/вывод за меньшее число операций, а сами операции более эффективны. Если скорость журналирования транзакций превышает 50 МБ/с, то подумайте, не стоит ли попробовать блок размером 4 КБ.

Был выявлен ряд проблем у заказчиков с БД, использующими журналы транзакций с 512-байтным блоком в файловой системе с 4-килобайтным блоком и множеством очень маленьких транзакций. Накладные расходы, связанные с применением нескольких 512-байтовых изменений к одному 4-килобайтному блоку файловой системы, порождали проблемы с производительностью, которые решались изменением файловой системы для использования 512-байтного блока.

Рекомендация компании NetApp:

- Не изменяйте размер блока redo, если это не рекомендовано соответствующей службой поддержки заказчиков или поставщиком профессиональных услуг, либо если изменение не рекомендовано в официальной документации по продукту.

14 Виртуализация

14.1 Обзор

Виртуализация БД с помощью VMware ESX, Oracle OVM или KVM получает все большее распространение среди заказчиков NetApp, которые используют ее даже для критически важных БД.

В отношении политик поддержки виртуализации, особенно для продуктов VMware, есть множество заблуждений. Действительно, можно нередко слышать, что Oracle никак не поддерживает виртуализацию. Думать так – неверно, и из-за этого можно упустить возможности виртуализации. В документе Oracle Doc ID 249212.1 рассматриваются распространенные проблемы в среде Oracle и подробно рассматривается RAC.

Если у заказчика возникла проблема, не известная компании Oracle, то его могут попросить воспроизвести эту проблему на физическом оборудовании. Заказчик Oracle, использующий самую современную версию продукта, может не захотеть использовать виртуализацию из-за опасений столкнуться с новыми багами. Однако это не вызывало проблем у тех заказчиков, которые для виртуализации использовали общедоступные версии продукта.

14.2 Представление пространства для хранения

Заказчики, которые задумываются о виртуализации своих БД, свои решения, касающиеся организации хранения, должны основывать на бизнес-потребностях. И хотя это в целом верно для всех ИТ-решений, это особенно важно для виртуализации, так как размер и содержание проектов значительно различаются.

Существуют 4 основных варианта представления пространства для хранения:

- LUN iSCSI, управляемые инициатором iSCSI на ВМ, а не гипервизором
- Файловые системы NFS, монтируемые самой ВМ, а не диск виртуальной машины (VMDK)
- Диски, «проброшенные» к ВМ с использованием технологии raw device mapping (RDM)
- Хранилище гипервизора (датастор)

В общем, лучше не использовать датасторы для файлов Oracle, и тому есть множество причин:

- Прозрачность. Когда ВМ владеет своими файловыми системами, администратору БД или системному администратору легче определить источник файловых систем для своих данных.
- Производительность. Как показало тестирование, прохождение всех операций ввода/вывода через датастор данных влияет на производительность.
- Управляемость. Когда ВМ владеет своими файловыми системами, использование или неиспользование слоя гипервизора влияет на управляемость. Одни и те же процедуры предоставления, мониторинга, защиты данных и т.д. могут использоваться во всей инфраструктуре, включая как виртуализированные, так и невиртуализированные среды.
- Стабильность и устранение неисправностей. Когда ВМ владеет своими файловыми системами, обеспечить высокую и стабильную производительность плюс возможности устранения неисправностей гораздо проще, так как весь стек хранения находится в ВМ. Единственная роль гипервизора в этом случае – транспортировка FC- или IP-кадров. Если датастор включен в конфигурацию, то это усложняет конфигурирование из-за появления другого набора таймаутов, параметров, файлов журналов и потенциальных багов.
- Переносимость. Когда ВМ владеет своими файловыми системами, процесс переноса среды Oracle существенно упрощается. Файловые системы можно легко перемещать между виртуализированными и невиртуализированными гостевыми сущностями.
- Привязка к вендору. После помещения данных в датастор использование другого гипервизора или полное удаление данных из виртуализированной среды становится очень трудным делом.
- Возможность использования моментальных снимков. В некоторых случаях резервное копирование в виртуализированной среде может стать проблемой из-за относительно ограниченной пропускной способности. Например, четырехпортового агрегированного канала из портов 10 GbE может быть достаточно для удовлетворения повседневных нужд многих виртуализированных БД в производительности. Однако такой магистрали будет недостаточно для резервного копирования с использованием RMAN или других инструментов резервного копирования, для которых требуется потоковая передача полноразмерной копии данных.
- Использование файловых систем, принадлежащих ВМ, упрощает создание резервных копий на основе моментальных снимков и восстановление с них. Когда ВМ владеет своими файловыми системами, нагрузка по созданию резервных копий переносится на СХД. Не нужно закладывать избыточность в конфигурацию гипервизора только лишь для того, чтобы выполнить требования к полосе пропускания и ЦП в окне резервного копирования.

Рекомендация компании NetApp:

- Для получения оптимальной производительности и управляемости старайтесь не размещать данные

Oracle в датасторе гипервизора. Используйте принадлежащие гостевой системе файловые системы, такие как NFS или iSCSI, которые управляются гостевой системой или подключаются с помощью RDM.

14.3 Паравиртуализированные драйверы

Для достижения оптимальной производительности очень важно использовать паравиртуализированные сетевые драйверы. Когда используется датастор гипервизора, требуется паравиртуализированный драйвер SCSI (например, PVSCSI). Паравиртуализированный драйвер устройства позволяет гостевой ОС глубже интегрироваться в гипервизор, в отличие от эмулируемого драйвера, в случае с которым гипервизор тратит больше процессорного времени, имитируя поведение физического оборудования.

Производительность большинства БД ограничена системой хранения. Поэтому дополнительные задержки, вносимые сетевым драйвером или драйвером SCSI, особенно заметны. Служба поддержки заказчиков NetApp неоднократно сталкивалась с жалобами на производительность, которые удавалось урегулировать установкой паравиртуализированных драйверов. В ходе пилотного проекта у одного из заказчиков базы данных продемонстрировали лучшую производительность под ESX, чем на том же оборудовании, работающем как «голое железо». В тестах использовался очень интенсивный ввод/вывод, а разница в производительности была объяснена использованием паравиртуализированных сетевых драйверов ESX.

Рекомендация компании NetApp:

- Всегда используйте паравиртуализированные сетевые драйверы и драйверы SCSI.

14.4 Избыточное выделение оперативной памяти

Избыточное выделение оперативной памяти означает задание в конфигурациях различных хостов большего объема виртуализированной оперативной памяти, чем имеется на физическом оборудовании. Это может привести к неожиданным проблемам с производительностью. При виртуализации БД гипервизор не должен выгружать на диск базовые блоки Oracle SGA, иначе результаты измерения производительности будут крайне нестабильными.

Рекомендация компании NetApp:

- Не конфигурируйте гипервизор так, чтобы была возможна выгрузка блоков Oracle SGA.

15 Кластеризация

15.1 Oracle Real Application Clusters

Этот раздел применим к Oracle 10.2.0.2 и более поздним версиям. Определите оптимальные настройки для более ранних версий Oracle, руководствуясь документом Oracle Doc ID 294430.1 и настоящим документом.

disktimeout

Основной, относящийся к хранению, параметр RAC – это `disktimeout`. Этот параметр управляет пороговым значением, в пределах которого должен завершаться ввод/вывод voting-файла. Если параметр `disktimeout` будет превышен, то узел RAC исключается из кластера. По умолчанию для этого параметра установлено значение 200. Этого должно быть достаточно для стандартных процедур переключения системы хранения (takeover и giveback).

NetApp настоятельно рекомендует тщательно тестировать конфигурации RAC перед их вводом в эксплуатацию, поскольку есть много факторов, влияющих на прямое или обратное переключение. Помимо времени, необходимого для аварийного переключения хранилища, требуется также дополнительное время для распространения изменений протокола управления агрегацией каналов (LACP). Кроме того, ПО поддержки множественных путей передачи в SAN должно обнаружить таймаут ввода/вывода и повторить попытку на альтернативном пути. Если БД чрезвычайно активна, то придется поставить в очередь большой объем операций ввода/вывода и затем попытаться выполнить их повторно, прежде чем обрабатывать ввод/вывод voting-диска.

Если фактический перехват или возврат управления в СХД невозможно выполнить, то эффект можно смоделировать, отключив кабель от сервера БД.

Рекомендация компании NetApp:

- Для параметра `disktimeout` не меняйте значение по умолчанию (200).

- Всегда тщательно тестируйте конфигурацию RAC.

misscount

Параметр `misscount` обычно влияет только на контрольные сигналы «я в порядке» (heartbeat), передаваемые по сети между узлами RAC. Значение по умолчанию – 30 секунд. Этот параметр может стать важным, если двоичные кластерной системы Oracle находятся на системе хранения или если загрузочный диск ОС не является локальным. Сюда входят хосты с загрузочными дисками, расположенными в FC SAN, ОС с загрузкой из NFS и загрузочные диски, расположенные в датасторах виртуализации, например, файл VMDK.

Если доступ к загрузочному диску прерывается из-за процедуры прямого или обратного переключения, то может так случиться, что устройство хранения двоичных файлов кластера или вся ОС временно зависнет. Время, необходимое ONTAP для завершения операции, и время, необходимое ОС для изменения пути и возобновления ввода/вывода, может превысить пороговое значение `misscount`. В результате узел будет немедленно исключен после восстановления связи с загрузочным LUN или двоичными файлами. В большинстве случаев исключение и последующая перезагрузка происходят без журналирования сообщений, которые указывали бы на причину перезагрузки. Это влияет не на все конфигурации, поэтому тестируйте все варианты загрузки в среде RAC (из SAN, из NFS или с хоста на основе датастора), чтобы сохранить стабильность RAC в случае обрыва связи с загрузочным диском.

Если используются не локальные загрузочные диски или если в файловой системе размещаются двоичные файлы, то, возможно, параметр `misscount` придется изменить, чтобы он соответствовал параметру `disktimeout`. Если этот параметр изменится, то проведите дальнейшее тестирование, чтобы также выявить другие факторы, влияющие на поведение RAC, например, время отработки отказа узла.

Рекомендация компании NetApp:

- Для параметра `misscount` оставьте значение по умолчанию (30), если не будет выполнено одно из следующих условий:
  - двоичные файлы кластера находятся на сетевом диске, включая NFS, iSCSI, FC и датасторы гипервизора.
  - ОС загружается по сети хранения (SAN-boot).
- В таких случаях оцените влияние перерывов в работе сети, которые влияют на доступ к файловым системам ОС или `GRID_HOME`. В некоторых случаях такие перерывы вызывают остановку демонов Oracle RAC, что может привести к таймауту по параметру `misscount` и к исключению узла из RAC. По умолчанию для таймаута установлено значение 27 секунд, т.е. `misscount` минус `reboottime`. В таких случаях для параметра `misscount` увеличьте значение до 200 для соответствия параметру disktimeout.

15.2 Solaris Clusters

Solaris Clusters - технология кластеризации типа «активный-пассивный» - гораздо более интегрирована, чем другое кластерное ПО. Эта технология обеспечивает почти полноценные возможности «plug-and-play», позволяя легко развертывать БД и приложения в виде кластеризованных ресурсов. Эта технология также позволяет легко перемещать их по кластеру, включая связанные IP-адреса, файлы конфигурации и ресурсы системы хранения. Как следствие такой тесной интеграции, Oracle предусматривает жесткую процедуру квалификации для кластеров Solaris, чтобы убедиться, что все компоненты работают вместе правильно.

ONTAP предоставляет обширную поддержку кластеров Solaris в среде SAN. Дополнительную информацию см. здесь:Interoperability Matrix Tools (IMT).

В среде NFS поддержка ограничена. В общем, препятствий для поддержки NFS не существует (например, использование автоматически монтируемых домашних каталогов NFS), но контролировать БД средствами кластеров Solaris нельзя. Ранее был доступен агент NFS, но поддержка этого продукта была прекращена в октябре 2012 г. Хотя можно использовать присущие кластерам Solaris возможности создания заказного кластеризуемого сервиса, это, вероятно, неосуществимо для большинства развертываний. Причина в затратах времени и усилий, необходимых для написания скриптов, управляющих ресурсами, включая хранилище.

15.3 Veritas Cluster Server

Veritas Cluster Server (VCS) аналогичен кластерам Solaris тем, что он позволяет пользователям предлагать БД или приложение в виде сервиса, развернутого в кластере типа «активный-пассивный».

VCS и SAN

ONTAP предоставляет обширную поддержку кластеризации VCS в среде SAN. Дополнительную информацию см. здесь: Interoperability Matrix Tool (IMT).

VCS и NFS

Одно время для создания кворума, мониторинга, управления, ограждения и разблокировки в NFS был доступен клиент от компании NetApp. Однако его поддержка была прекращена в октябре 2012 г. главным образом из-за того, что эти возможности перестали быть востребованными. Сейчас VCS может собственными средствами обеспечить управление и мониторинг файловых систем NFS. Для управления кворумом в среде NAS есть несколько вариантов, не требующих агента.

VCS и ограждение в NFS (fencing)

Одним из факторов, которые необходимо иметь в виду, относительно любой кластеризации типа «активный-пассивный» является ограждение (fencing), означающее, что ресурсы системы хранения доступны только одному узлу в кластере. В среде SAN ограждение обычно означает использование постоянно зарезервированных дисков SCSI, что позволяет узлу требовать исключительного контроля над LUN. В контексте NFS это означает изменение параметров экспорта для файловой системы, чтобы сделать невозможным доступ к ресурсу более чем на одном узле. Разница в том, что в среде SAN ограждение выполняется узлом кластера, заявляющим право на ресурс системы хранения. В среде NAS ограждение должно выполняться на СХД.

В случае NFS ограждение не является строго обязательным. Гораздо важнее иметь ограждение в среде SAN, так как простой процесс монтирования файловой системы SAN более чем на одном сервере обычно сразу же повреждает данные. NFS – кластерная файловая система, а значит, несколько серверов могут монтировать файловую систему без проблем.

Многие заказчики используют кластеризацию «активный-пассивный» с VCS и аналогичными продуктами, такими как HP ServiceGuard и IBM PowerHA, без какого-либо ограждения. Они доверяют ПО самого кластера, которое обеспечивает работу ресурса только на одном узле. Если необходимо ограждение, его можно развернуть как часть ресурса кластера, затратив немного усилий на создание скриптов.

Когда сервис запускается, он выдает СХД команду (а) отключить доступ для целевых файловых систем ко всем узлам, а затем (б) предоставить доступ одному узлу, на котором запускается сервис. Таким образом, только один узел может выполнять ввод/вывод в целевых файловых системах. Когда сервис останавливается, он выдает СХД команду закрыть доступ для него. Существуют и другие варианты, но этот – наиболее всеобъемлющий.

Помощь по этим системам предоставляет подразделение NetApp Professional Services. За дополнительной информацией обращайтесь к вашему представителю NetApp.

VCS и снятие блокировки в NFS

Блокировки NFS в среде Oracle – это одна из форм ограждения. БД Oracle не запустится, если обнаружит блокировку NFS на целевых файлах. В среде VCS блокировка NFS обычно мешает нормальной работе кластера VCS. Единственная ситуация, когда блокировку нужно снимать, это когда один узел перехватывает сервисы другого узла, который не завершил работу корректно. Во время корректного отключения БД Oracle блокировки снимаются. В случае аварийного отказа узла блокировки сохраняются и должны быть сняты перед перезапуском БД.

Большинство заказчиков предпочитают отключать блокировку NFS, для чего активируют соответствующую опцию монтирования NFS, которая в первую очередь предотвращает создание блокировок. Если это нежелательно, то снятие блокировки можно описать скриптом. Как и в случае с ограждением, в написании скриптов взлома блокировки вам поможет подразделение NetApp Professional Services, а в некоторых случаях полностью поддерживаемые параметры можно получить через службу Rapid Response Engineering. За дополнительной информацией обращайтесь к вашему представителю NetApp.

16 IBM AIX

В этом разделе рассматриваются вопросы конфигурации, относящиеся к операционной системе IBM AIX.

16.1 Параллельный ввод/вывод

Для обеспечения оптимальной производительности на IBM AIX нужно использовать параллельный ввод/вывод. Без параллельного ввода/вывода есть вероятность ограничения производительности, поскольку AIX выполняет сериализованный атомарный ввод/вывод, который сопровождается значительными накладными расходами.

Изначально компания NetApp рекомендовала использовать параметр монтирования `cio`, чтобы принудительно задействовать параллельный ввод/вывод в файловой системе, но этот процесс имел недостатки и больше не требуется. С появлением AIX 5.2 и Oracle 10gR1 Oracle в AIX может открывать для параллельного ввода/вывода отдельные файлы, что отличается от принудительного параллельного ввода/вывода во всей файловой системе.

Лучший способ включить параллельный ввод/вывод – установить в файле `init.ora` для параметра `filesystemio_options` значение `setall`. В этом случае Oracle сможет открывать отдельные файлы для параллельного ввода/вывода.

Использование cio как параметра монтирования принудительно активирует параллельный ввод/вывод, что может иметь негативные последствия. Например, принудительный параллельный ввод/вывод отключает упреждающее чтение в файловых системах, что может снизить производительность операций ввода/вывода вне ПО БД Oracle, таких как копирование файлов и резервное копирование на ленточные накопители. Кроме того, такие продукты, как Oracle GoldenGate и SAP BR*Tools, не совместимы с использованием параметра монтирования `cio` в определенных версиях Oracle.

Рекомендация компании NetApp:

- Не используйте параметр монтирования `cio` на уровне файловой системы. Вместо этого разрешите параллельный ввод/вывод командой `filesystemio_options=setall`.
- Параметр монтирования cio используйте, только если невозможно установить `filesystemio_options=setall`.

16.2 Параметры монтирования AIX NFS

В Таблице 1 и Таблице 2 перечислены параметры монтирования AIX NFS.

Таблица 1. Параметры монтирования AIX NFS – один экземпляр.

Тип файла	Параметры монтирования
`ADR_HOME`	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536
Файлы управления Файлы данных Журналы транзакций	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536
`ORACLE_HOME`	rw,bg,hard,[[vers=3,vers=4],vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536,intr

Таблица 2. Параметры монтирования AIX NFS – RAC.

Тип файла	Параметры монтирования
`ADR_HOME`	rw,bg,hard,[[vers=3,vers=4],vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536
Файлы управления Файлы данных Журналы транзакций	rw,bg,hard,[[vers=3,vers=4],vers=4],proto=cp,timeo=600,rsize=65536,wsize=65536,nointr, noac
CRS/Voting	rw,bg,hard,[[vers=3,vers=4],vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr,noac
Выделенный `ORACLE_HOME`	rw,bg,hard,[[vers=3,vers=4],vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536
Общий `ORACLE_HOME`	rw,bg,hard,[[vers=3,vers=4],vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr

Основная разница между параметрами монтирования для случаев одного экземпляра и RAC – добавление параметра `noac` в параметры монтирования. Это добавление отключает кэширование ОС хоста, что позволяет всем экземплярам в кластере RAC иметь согласованное представление о состоянии данных.

Хотя использование параметра монтирования `cio` и параметра `filesystemio_options=setall` в файле `init.ora` точно так же запрещает кэширование на хосте, использовать `noac` все равно необходимо. Параметр `noac` необходим в случае развертывания совместно используемых продуктов `(ORACLE_HOME)`, чтобы было легче обеспечить согласованность таких файлов, как файлы паролей Oracle и файлы параметров `spfile`. Если каждый экземпляр в кластере RAC имеет выделенный `ORACLE_HOME`, то этот параметр не требуется.

16.3 Параметры монтирования AIX jfs/jfs2

В Таблице 3 перечислены параметры монтирования AIX jfs/jfs2.

Таблица 3. Параметры монтирования AIX jfs/jfs2 – один экземпляр.

Тип файла	Параметры монтирования
`ADR_HOME`	Значения по умолчанию
Файлы управления Файлы данных Журналы транзакций	Значения по умолчанию
`ORACLE_HOME`	Значения по умолчанию

Прежде чем использовать устройства AIX hdisk в любой среде, включая БД, проверьте параметр `queue_depth`. Этот параметр не является глубиной очереди HBA; он скорее относится к глубине очереди SCSI отдельного устройства hdisk. В зависимости от того, как сконфигурированы LUN, значение параметра `queue_depth` может быть слишком малым, чтобы обеспечить хорошую производительность. Как показало тестирование, оптимальное значение равно 64.

17 HP-UX

В этом разделе рассматриваются вопросы конфигурации, характерные для ОС HP-UX.

17.1 Параметры монтирования HP-UX NFS

В Таблице 4 перечислены параметры монтирования HP-UX NFS.

Таблица 4. Параметры монтирования HP-UX NFS – один экземпляр.

Тип файла	Параметры монтирования
`ADR_HOME`	rw,bg,hard,[[vers=3,vers=4],vers=4],proto=tcp,timeo=600,rsize=65536,wsize=6 5536,suid
Файлы управления Файлы данных Журналы транзакций	rw,bg,hard,[[vers=3,vers=4],vers=4],proto=tcp,timeo=600,rsize=65536,wsize=6 5536,forcedirectio,nointr,suid
`ORACLE_HOME`	rw,bg,hard,[[vers=3,vers=4],vers=4],proto=tcp,timeo=600,rsize=65536,wsize=6 5536,suid

Таблица 5. Параметры монтирования HP-UX NFS – RAC.

Тип файла	Параметры монтирования
`ADR_HOME`	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536,no ac,suid
Файлы управления Файлы данных Журналы транзакций	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536,noi ntr,noac forcedirectio, suid
CRS/Voting	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536,noi ntr,noac, forcedirectio,suid
Выделенный `ORACLE_HOME`	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536,sui d
Общий `ORACLE_HOME`	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536,noi ntr,noac,suid

Основная разница между параметрами монтирования для случаев одного экземпляра и RAC – добавле-

ние параметра `noac` и `forcedirectio` в параметры монтирования. Это добавление отключает кэширование ОС хоста, что позволяет всем экземплярам в кластере RAC иметь согласованное представление о состоянии данных. Хотя использование параметра `filesystemio_options=setall` в файле `init.ora` точно так же запрещает кэширование на хосте, использовать `noac` и `forcedirectio` все равно необходимо.

Параметр `noac` необходим в случае развертывания совместно используемых продуктов `ORACLE_HOME`, чтобы было легче обеспечить согласованность таких файлов, как файлы паролей Oracle и файлы spfile. Если каждый экземпляр в кластере RAC имеет выделенный `ORACLE_HOME`, то этот параметр не требуется.

17.2 Параметры монтирования HP-UX VxFS

Для файловых систем, в которых размещаются двоичные файлы Oracle, используйте следующие параметры монтирования:

```
delaylog,nodatainlog
```

Используйте следующие параметры монтирования для файловых систем, содержащих файлы данных, журналы транзакций, архивные журналы и файлы управления, в которых версия HP-UX не поддерживает параллельный ввод/вывод:

```
nodatainlog,mincache=direct,convosync=direct
```

Когда параллельный ввод/вывод поддерживается (VxFS 5.0.1 или более поздние версии, либо при использовании ServiceGuard Storage Management Suite), используйте следующие параметры для файловых систем, содержащих файлы данных, журналы транзакций, архивные журналы и файлы управления:

```
delaylog,cio
```

Примечание. Параметр `db_file_multiblock_read_count` имеет особую важность в средах VxFS. Oracle рекомендует не устанавливать значение для этого параметра в Oracle 10g R1 и более поздних версиях, если только это прямо не требуется. По умолчанию он равен 128 при размере блока Oracle 8 КБ. Если для этого параметра принудительно установить значение 16 или меньше, то удалите параметр `convosync=direct`, так как это может снизить производительность последовательного ввода-вывода. Этот шаг снижает производительность в других отношениях и необходим, только если нужно изменить значение по умолчанию db_file_multiblock_read_count.

18 Linux

В этом разделе рассматриваются вопросы конфигурации, характерные для ОС Linux.

18.1 Linux NFS

Таблицы слотов

Производительность NFS на Linux зависит от параметра `tcp_slot_table_entries`. Этот параметр регулирует количество ожидающих выполнения операций NFS, которые разрешены в ОС Linux.

В большинстве систем на основе ядра 2.6, включая RH5 и OL5, значение по умолчанию равно 16, и это часто вызывает проблемы производительности. Противоположная проблема возникает в ядрах более новых версий, в которых значение `tcp_slot_table_entries` не ограничено и может вызвать проблемы с хранением из-за переполнения системы чрезмерным количеством запросов.

Решение – задать это значение статически. Задайте значение 128 для любой ОС Linux, использующей хранилище NetApp NFS с БД Oracle.

Чтобы установить это значение в RHEL 6.2 и более ранних версиях, вставьте следующую строку в файл `/etc/sysctl.conf`:

```
sunrpc.tcp_slot_table_entries = 128
```

Кроме того, в большинстве дистрибутивов Linux, использующих ядра 2.6, есть ошибка. Процесс запуска читает содержимое `/etc/sysctl.conf` до того, как будет загружен NFS-клиент. В результате, когда NFS-клиент наконец загрузится, он примет значение по умолчанию 16. Для предотвращения этой проблемы отредактируйте `/etc/init.d/netfs` так, чтобы вызывать `/sbin/sysctl -p` в первой строке скрипта, чтобы для параметра `tcp_slot_table_entries` установить значение 128 до того, как NFS смонтирует какую-либо файловую систему.

Чтобы установить это значение в RHEL 6.3 и более поздних версиях, измените файл конфигурации RPC следующим образом:

```
echo “options sunrpc udp_slot_table_entries=64 tcp_slot_table_entries=128
tcp_max_slot_table_entries=128” >> /etc/modprobe.d/sunrpc.conf
```

18.2 Параметры монтирования Linux NFS

В Таблице 6 и Таблице 7 перечислены параметры монтирования Linux NFS.

Таблица 6. Параметры монтирования Linux NFS – один экземпляр.

Тип файла	Параметры монтирования
`ADR_HOME`	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536
Файлы управления Файлы данных Журналы транзакций	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr

Тип файла	Параметры монтирования
`ORACLE_HOME`	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr

Таблица 7. Параметры монтирования Linux NFS – RAC.

Тип файла	Параметры монтирования
`ADR_HOME`	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536,acti meo=0
Файлы управления Файлы данных Журналы транзакций	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536,noint r,actimeo=0
CRS/voting	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536,noint r,noac,actimeo=0
Выделенный `ORACLE_HOME`	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536
Общий `ORACLE_HOME`	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536,noint r,actimeo=0

Основная разница между параметрами монтирования для случаев одного экземпляра и RAC – добавление параметра `actimeo=0` в параметры монтирования. Это добавление отключает кэширование ОС хоста, что позволяет всем экземплярам в кластере RAC иметь согласованное представление о состоянии данных. Хотя использование параметра `filesystemio_options=setall` в файле `init.ora` точно так же запрещает кэширование на хосте, использовать `actimeo=0` все равно необходимо.

Параметр `actimeo=0` необходим в случае многопользовательского развертывания `ORACLE_HOME`, чтобы облегчить достижение согласованности таких файлов, как файлы паролей Oracle и файлы spfile. Если каждый экземпляр в кластере RAC имеет выделенный `ORACLE_HOME`, то этот параметр не требуется.

Как правило, файлы, не относящиеся к БД, нужно монтировать с такими же параметрами, что и файлы данных в случае одного экземпляра, хотя разные приложения могут предъявлять разные требования. По возможности избегайте параметров монтирования `noac` и `actimeo=0`, так как они отключают упреждающее чтение и буферизацию на уровне файловой системы. Это может вызвать серьезные проблемы производительности в таких процессах, как извлечение, трансляция и загрузка.

ACCESS и GETATTR

Некоторые заказчики отмечали, что в их рабочих нагрузках могут доминировать другие процессы с чрезвычайно интенсивным вводом/выводом (такие как ACCESS и GETATTR). В экстремальных случаях операции чтения и записи могут занимать лишь 10% от общего объема. Это нормальное поведение для любой БД при использовании параметров actimeo=0 и/или noac в Linux, потому что эти параметры заставляют ОС Linux все время перезагружать метаданные из СХД. Такие операции, как ACCESS и GETATTR, оказывают малое влияние и обслуживаются из кэша ONTAP в среде БД. Их не следует рассматривать как истинные операции ввода/вывода, такие как операции чтения и записи, которые действительно нагружают СХД. Тем не менее, эти «не истинные» операции ввода/вывода все же создают какую-то нагрузку, особенно в средах RAC. Чтобы справиться с этой ситуацией, включите DNFS, которая обходит кэш буфера ОС и избегает этих лишних операций с метаданными.

Linux Direct NFS

Когда (а) DNFS включена, (б) исходный том монтируется более одного раза на одном сервере и (в) применяется монтирование вложенных NFS, требуется один дополнительный параметр монтирования, называемый `nosharecache`. Эта конфигурация встречается главным образом в средах, поддерживающих приложения SAP. Например, один и тот же том в системе NetApp может иметь каталог, расположенный в `/vol/oracle/base`, а второй – в `/vol/oracle/home`. Если `/vol/oracle/base` монтируется в `/oracle` и `/vol/oracle/home` монтируется в `/oracle/home`, то результатом будут смонтированные вложенные NFS, берущие начало на одном и том же источнике.

ОС может определить тот факт, что `/oracle` и `/oracle/home` находятся на одном томе, который представляет собой одну и ту же исходную файловую систему. Затем ОС использует один и тот же маркер устройства для доступа к данным. Это улучшает использование кэширования ОС и ряд других операций, но мешает работе DNFS. Если DNFS должна обратиться к такому файлу как `spfile`, расположенному в `/oracle/home`, то она может попытаться ошибочно использовать неверный путь к данным. Результатом будет неудачная операция ввода/вывода. В таких конфигурациях добавьте параметр монтирования `nosharecache` в любую файловую систему NFS, которая использует исходный том FlexVol совместно с другой файловой системой NFS на том же хосте. Это заставит ОС Linux выделить независимый маркер устройства для этой файловой системы.

Linux Direct NFS и Oracle RAC

Использование DNFS особенно полезно для повышения производительности Oracle RAC в ОС Linux, поскольку в Linux нет метода принудительного прямого ввода/вывода, который требуется при использовании RAC для обеспечения согласованности между узлами. В качестве обходного решения Linux требует использования параметра монтирования `actimeo=0`, при котором файл данных немедленно удаляется из кэша ОС. Этот параметр заставляет NFS-клиент Linux постоянно перечитывать данные атрибутов, что увеличивает время задержки и нагрузку на контроллер хранилища.

Включение DNFS позволяет обойти NFS-клиент на хосте и избежать этого. Многие заказчики сообщали о значительном повышении производительности кластеров RAC и значительном снижении нагрузки на ONTAP (особенно в части «не истинных» операций ввода/вывода) при включении DNFS.

Linux Direct NFS и файл oranfstab

При использовании DNFS в Linux с параметром, разрешающим передачу по нескольким путям, необходимо использовать несколько подсетей. В других ОС множественные каналы DNFS можно организовать, используя параметры `LOCAL` и `DONTROUTE` для конфигурирования множественных каналов в одной подсети. Однако на Linux это не работает как следует, что может приводить к неожиданным проблемам с производительностью. В случае Linux каждая сетевая карта, используемая для DNFS-трафика, должна быть в отдельной подсети.

18.3 Общие сведения о конфигурировании SAN в Linux

Выравнивание при сжатии – разделы

Для получения оптимальных результатов при сжатии требуется выравнивание по 8-килобайтным границам на диске. Проверьте выравнивание утилитой fdisk с параметром `-u`, отображающим информацию о секторах диска. Рассмотрим следующий пример:

```
[root@jfs0 etc]# fdisk -l -u /dev/sdb
Disk /dev/sdb: 10.7 GB, 10737418240 bytes
64 heads, 32 sectors/track, 10240 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 65536 bytes
Disk identifier: 0xb97f94c1

Device Boot  Start  End          Blocks       Id     System
/dev/sdb1    36     20971519     10485742     83     Linux
Partition 1 does not start on physical sector boundary.
```

Этот раздел не выровнен по 8-килобайтной границе, а смещен на 36 секторов. Это смещение соответствует выравниванию по 4-килобайтной границе (что обычно требуется для хорошей производительности), но не по 8-килобайтной. Чтобы раздел был выровнен, его начало должно быть кратно 16 секторам (512 байт * 16 = 8192).

В этом примере показан корректно выровненный раздел:

```
[root@jfs0 etc]# fdisk -l -u /dev/sdb

Disk /dev/sdb: 10.7 GB, 10737418240 bytes
64 heads, 32 sectors/track, 10240 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 65536 bytes
Disk identifier: 0xb97f94c1

Device Boot  Start  End          Blocks       Id     System
/dev/sdb1    64     20971519     10485728     83     Linux
```

Выравнивание при сжатии – файловые системы

В дополнение к разбиению на разделы, файловую систему нужно также выровнять по 8-килобайтным границам. Это значит, что размер блока файловой системы должен быть 8 КБ. При использовании Oracle ASM выравнивание по 8-килобайтной границе обеспечивается механизмом выделения экстентов и чередования в Oracle ASM.

При использовании других файловых систем размер блока должен быть задан равным 8 КБ. На некоторых файловых системах это может оказаться невозможно.

Планировщик ввода/вывода

Ядро Linux делает возможным низкоуровневый контроль над способом планирования ввода/вывода на блочные устройства. Настройки по умолчанию сильно разнятся от одного дистрибутива Linux к другому.

Тестирование показывает, что обычно лучшие результаты выдает Deadline, хотя иногда немного лучше был NOOP. Разница в производительности минимальна, но протестируйте оба варианта, чтобы добиться максимально возможной производительности на вашей конфигурации БД. CFQ – это значение по умолчанию во многих конфигурациях, и, как показала практика, вызывает серьезные проблемы с производительностью БД.

Инструкции по настройке планировщика ввода-вывода см. в документации соответствующего вендора Linux.

Множественность путей

Некоторые заказчики сталкивались с аварийными отказами во время нарушения работы сети, потому что демон, обеспечивающий работу по нескольким путям, в их системе не работал. В последних версиях Linux процесс установки ОС и демон, обеспечивающий работу по нескольким путям, могут делать эти ОС уязвимыми к этой проблеме. Пакеты установлены правильно, но не настроены на автоматический запуск после перезагрузки.

Например, значение по умолчанию для демона, обеспечивающего работу по нескольким путям, на RHEL5.5 может выглядеть так:

```
[root@jfs0 iscsi]# chkconfig --list | grep multipath
multipathd   0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

Это можно скорректировать следующими командами:

```
[root@jfs0 iscsi]# chkconfig multipathd on
[root@jfs0 iscsi]# chkconfig --list | grep multipath
multipathd   0:off   1:off   2:on   3:on   4:on   5:on   6:off
```

18.4 Зеркалирование в ASM

Зеркалирование в ASM может потребовать изменения настроек Linux, отвечающих за работу по нескольким путям, чтобы ASM распознавал проблему и переключался на альтернативную группу отказа. Большинство конфигураций ASM в ONTAP используют внешнее резервирование, т.е. данные защищаются внешним массивом, а ASM не зеркалирует данные. На некоторых площадках используется ASM с нормальным резервированием, чтобы обеспечить двустороннее зеркалирование (обычно на разных площадках).

Параметры Linux, приведенные в документации по NetApp Host Utilities, включают в себя и те параметры, отвечающие за работу по нескольким путям, которые приводят к неопределенности работы с очередью ввода/вывода. Это означает, что ввод/вывод на устройстве LUN без активных путей ожидает столько времени, сколько требуется для завершения ввода/вывода. Это обычно желательно, потому что хосты Linux ждут столько времени, сколько необходимо для изменения пути SAN, для перезагрузки коммутаторов FC или для завершения обработки отказа СХД.

Такое отсутствие ограничений при обработке очереди вызывает проблему с зеркалированием ASM, поскольку ASM должен получить ошибку ввода/вывода, чтобы повторить ввод/вывод на альтернативном LUN.

Установите следующие параметры в файле Linux `multipath.conf` для LUN ASM, используемых с зеркалированием ASM:

```
polling_interval 5
no_path_retry 24
```

Эти настройки создают 120-секундный таймаут для устройств ASM. Таймаут в секундах рассчитывается как `polling_interval * no_path_retry`. Иногда может потребоваться корректировка точного значения, но в большинстве случаев 120-секундного таймаута должно быть достаточно. В частности, 120 секунд должно хватить, чтобы контролер выполнил перехват или возврат, не выдавая ошибку ввода/вывода, которая приведет к отключению группы отказа.

Меньшее значение параметра `no_path_retry` может сократить время, необходимое для переключения ASM на альтернативную группу отказа, но также увеличивает риск нежелательного переключения во время технического обслуживания, например, переключение контроллера. Этот риск можно уменьшить

путем тщательного мониторинга за состоянием зеркалирования ASM. Если произойдет нежелательное переключение, зеркала можно быстро пересинхронизировать, если пересинхронизация выполняется относительно быстро. Дополнительную информации см. в документации Oracle по ASM Fast Mirror Resync для используемой версии ПО Oracle.

18.5 Размер блока ASMlib

ASMlib – это дополнительная библиотека управления ASM и сопутствующие утилиты. Главная ее ценность состоит в возможности помечать LUN или файл в NFS как ресурс ASM удобочитаемой меткой.

Последние версии ASMlib обнаруживают параметр LUN, называемый «показателем числа логических блоков на физический блок» (Logical Blocks Per Physical Block Exponent, LBPPBE). До недавнего времени подсистема ONTAP SCSI таргет не сообщала этого значения. Теперь она возвращает значение, которое указывает, что предпочтительным является размер блока 4 КБ. Это не определение размера блока, а подсказка любому приложению, использующему LBPPBE, о том, что операции ввода/вывода определенного размера могут обрабатываться более эффективно. А вот ASMlib интерпретирует LBPPBE как размер блока и целенаправленно маркирует заголовок ASM при создании устройства ASM.

Этот процесс может вызвать разные проблемы с обновлениями и миграциями, и все из-за невозможности комбинировать устройства ASMlib с разными размерами блока в одной группе дисков ASM.

Например, старые массивы обычно сообщали, что значение LBPPBE = 0, или вообще не сообщали его. ASMlib трактует это как блок размером 512 байт. Более новые массивы будут считаться массивами с блоками размером 4 КБ. Невозможно смешивать устройства с блоками 512 байт и 4 КБ в одной группе дисков ASM. Это может помешать пользователю увеличить размер группы дисков ASM, используя LUN из двух массивов, или использовать ASM в качестве инструмента миграции. В других случаях RMAN может не разрешать копирование файлов между группой дисков ASM с 512-байтными блоками и группой дисков ASM с 4-килобайтными блоками.

Предпочтительным решением является установка патча на ASMlib. ID ошибки Oracle – 13999609, и патч для нее включен в oracleasm-support-2.1.8-1 и выше. Этот патч позволяет пользователю установить для параметра `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` значение `true` в файле конфигурации `/etc/sysconfig/oracleasm`. Это запрещает ASMlib использовать параметр LBPPBE, что означает, что LUN в новом массиве теперь распознаются как устройства в 512-байтными блоками.

Примечание. Этот параметр не меняет размер блока LUN, которые ранее были помечены ASMlib. Например, если группу дисков ASM с 512-байтными блоками нужно перенести в новую СХД, которая сообщает о 4-килобайтном блоке, то параметр `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` должен быть установлен **до того, как** новые LUN будут помечены ASMlib.

Если устройства уже были помечены с помощью команды oracleasm, то их нужно переформатировать, прежде чем помечать их как устройства с новым размером блока. Во-первых, отмените конфигурирование устройства с помощью команды `oracleasm deletedisk`, а затем очистите первый 1 ГБ устройства с помощью команды `dd if=/dev/zero of=/dev/mapper/device bs=1048576 count=1024`. Наконец, если устройство было ранее разбито на разделы, то используйте команду `kpartx`, чтобы удалить ненужные разделы, или просто перезагрузите ОС.

Если не удается установить патч на ASMlib, то ASMlib можно удалить из конфигурации. Это изменение является разрушительным и требует снятия меток с дисков ASM и проверки правильности установки параметра `asm_diskstring`. Однако это изменение не требует миграции данных.

18.6 Параметры монтирования Linux ext3 и ext4

NetApp рекомендует использовать параметры монтирования, установленные по умолчанию.

19 Microsoft Windows

В этом разделе рассматриваются вопросы конфигурации, характерные для ОС Microsoft Windows.

19.1 NFS

Oracle поддерживает использование Microsoft Windows с клиентом DNFS. Это делает доступными преимущества управления NFS, включая возможность просмотра файлов в разных средах, динамическое изменение размера томов и использование менее дорогого протокола IP. Информацию об установке и конфигурировании БД в Microsoft Windows с использованием DNFS см. в официальной документации Oracle. Никаких особых передовых практик не существует.

19.2 SAN

Для оптимизации сжатия убедитесь, что файловые системы NTFS используют единицу размещения (allocation unit) размером 8192 байт или больше. Использование 4096-байтной единицы размещения, которая обычно задана по умолчанию, снижает эффективность.

20 Solaris

В этом разделе рассматриваются вопросы конфигурации, характерные для ОС Solaris.

20.1 Параметры монтирования Solaris NFS

В Таблице 8 перечислены параметры монтирования Solaris NFS для одного экземпляра.

Таблица 8. Параметры монтирования Solaris NFS – один экземпляр.

Тип файла	Параметры монтирования
`ADR_HOME`	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536
Файлы управления Файлы данных Журналы транзакций	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr,llock,suid
`ORACLE_HOME`	rw,bg,hard,[vers=3,vers=4],proto=tcp,timeo=600,rsize=65536,wsize=65536,suid

Доказано, что использование `llock` существенно повышает производительность в средах заказчиков, устраняя задержку, связанную с установкой и снятием блокировок в СХД. Используйте этот параметр с осторожностью в средах, где множество серверов настроено на монтирование одних и тех же файловых систем, а ПО Oracle настроено на монтирование этих БД. Хотя это и весьма необычная конфигурация, некоторые заказчики ее используют. Если экземпляр случайно запустится повторно, то данные могут повредиться, поскольку Oracle не может обнаружить файлы блокировки на стороннем сервере. В остальных отношениях блокировки NFS не обеспечивают защиту; как и в NFS версии 3, они носят исключительно рекомендательный характер.

Поскольку параметры `llock` и `forcedirectio` взаимно исключают друг друга, важно, чтобы выражение `filesystemio_options=setall` присутствовало в файле `init.ora` для использования `directio`. Без этого параметра будет использоваться буферное кэширование в ОС хоста, что может негативно повлиять на производительность.

В Таблице 9 перечислены параметры монтирования Solaris NFS в RAC.

Таблица 9. Параметры монтирования Solaris NFS – RAC.

Тип файла	Параметры монтирования
`ADR_HOME`	rw, bg, hard, [vers=3,vers=4], proto=tcp, timeo=600, rsize=65536, wsize=65536, noac
Файлы управления Файлы данных Журналы транзакций	rw, bg, hard, [vers=3,vers=4], proto=tcp, timeo=600, rsize=65536, wsize=65536, nointr, noac, forcedirectio
CRS/Voting	rw, bg, hard, [vers=3,vers=4], proto=tcp, timeo=600, rsize=65536, wsize=65536, nointr, noac, forcedirectio
Выделенный `ORACLE_HOME`	rw, bg, hard, [vers=3,vers=4], proto=tcp, timeo=600, rsize=65536, wsize=65536, suid
Общий `ORACLE_HOME`	rw, bg, hard, [vers=3,vers=4], proto=tcp, timeo=600, rsize=65536, wsize=65536, nointr, noac, suid

Основная разница между параметрами монтирования для случаев одного экземпляра и RAC – добавление параметров `noac` и `forcedirectio` в параметры монтирования. Это добавление отключает кэширование ОС хоста, что позволяет всем экземплярам в кластере RAC иметь согласованное представление о состоянии данных. Хотя использование параметра `filesystemio_options=setall` в файле `init.ora` точно так же запрещает кэширование на хосте, использовать `noac` и `forcedirectio` все равно необходимо.

Параметр `actimeo=0` необходим в случае многопользовательского развертывания `ORACLE_HOME`, чтобы облегчить достижение согласованности таких файлов, как файлы паролей Oracle и файлы spfile. Если каждый экземпляр в кластере RAC имеет выделенный `ORACLE_HOME`, то этот параметр не требуется.

20.2 Параметры монтирования Solaris UFS

NetApp настоятельно рекомендует использовать параметр монтирования `logging`, чтобы сохранить целостность данных в случае сбоя хоста Solaris или прерывания FC-подключения. Параметр монтирования `logging` также сохраняет возможность использования моментальных снимков.

20.3 Solaris ZFS

Для достижения оптимальной производительности нужно тщательно устанавливать и конфигурировать Solaris ZFS.

mvector

В Solaris 11 в обработку больших операций ввода/вывода внесены изменения, которые могут привести к серьезным проблемам с производительностью массивов хранения SAN. Эта проблема подробно описана в отчете об ошибке NetApp 630173, «Падение производительности ZFS в Solaris 11». Решение состоит в изменении параметра ОС, называемого `zfs_mvector_max_size.`

От имени root выполните следующую команду:

```
echo "zfs_mvector_max_size/W 0t131072" |mdb -kw
```

Если из-за этого изменения возникнут непредвиденные проблемы, то его можно легко откатить, выполнив от имени root следующую команду:

```
echo "zfs_mvector_max_size/W 0t1048576" |mdb -kw
```

Ядро

Для надежной работы ZFS нужно установить патч на ядро Solaris, чтобы исправить проблемы с выравниванием LUN. Это исправление включено в патч 147440-19 в Solaris 10 и в SRU 10.5 для Solaris 11. С ZFS используйте только Solaris 10 или более позднюю версию.

Конфигурация LUN

Чтобы сконфигурировать LUN, выполните следующие шаги:

1. Создайте LUN типа `solaris`.
2. Установите соответствующий Host Utility Kit (HUK), указанный инструментом IMT.
3. Строго следуйте инструкциям, содержащимся в HUK. Основные шаги приведены в этом разделе, но надлежащая процедура описана в новейшей документации.
   a. Запустите утилиту `host_config`, чтобы обновить файл `sd.conf/sdd.conf`. Это позволит драйверам SCSI корректно обнаруживать LUN ONTAP.
   b. Следуйте инструкциям, выданным утилитой `host_config`, чтобы включить ввод/вывод по нескольким путям (MPIO).
   c. Перезагрузитесь. Этот шаг нужен для того, чтобы все изменения в системе вступили в силу.
4. Разбейте LUN на разделы и убедитесь, что они должным образом выровнены. См. «Приложение B. Проверка выравнивания WAFL ", где приведены инструкции о том, как проверять и подтверждать выравнивание.

Пулы zpool

Создавайте zpool только после выполнения шагов в разделе “Конфигурирование LUN”. Если процедуру выполнить некорректно, то производительность может серьезно упасть из-за выравнивания ввода/вывода. Для достижения оптимальной производительности в ONTAP нужно, чтобы ввод/вывод был выровнен по 4-килобайтным границам на диске. Файловые системы, созданные в zpool, используют эффективный размер блока, определяемый параметром `ashift`, значение которого можно просмотреть, выполнив команду `zdb -C`.

По умолчанию значение параметра `ashift` равно 9, что означает 2^9, или 512 байт. Для получения оптимальной производительности значение параметра `ashift` должно быть равно 12 (2^12 = 4К). Это значение устанавливается при создании zpool и не может быть изменено, поэтому данные из пулов zpool с параметром `ashift`, отличным от 12, нужно скопировать во вновь созданный zpool.

После создания zpool перед продолжением проверьте значение `ashift`. Если значение не равно 12, значит, LUN не были обнаружены корректно. Удалите zpool, убедитесь, что все шаги, приведенные в соответствующей документации на Host Utilities, были выполнены правильно, и заново создайте zpool.

Пулы zpool и логические домены (LDOM) в Solaris

LDOM в Solaris налагают дополнительное требование, призванное обеспечить надлежащее выравнивание ввода/вывода. Хотя LUN может правильно обнаруживаться как устройство с размером блока 4 КБ, виртуальное устройство vdsk на LDOM не наследует конфигурацию от домена ввода/вывода. Виртуальное устройство vdsk на базе этого LUN восстанавливает в настройках по умолчанию 512-байтные блоки.

Требуется дополнительный файл конфигурации. Во-первых, нужно установить для отдельных LDOM патч, исправляющий ошибку Oracle 15824910, чтобы включить дополнительные параметры конфигурации. Этот патч входит во все используемые в настоящее время версии Solaris. После установки патча LDOM готов к конфигурированию новых правильно выровненных LUN, как показано ниже:

1. Укажите один или несколько LUN, которые будут использоваться в новом пуле zpool. В данном примере это – устройство c2d1.

```
root@LDOM1 # echo | format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c2d0 <Unknown-Unknown-0001-100.00GB>
    /virtual-devices@100/channel-devices@200/disk@0
  1. c2d1 <SUN-ZFS Storage 7330-1.0 cyl 1623 alt 2 hd 254 sec 254>
    /virtual-devices@100/channel-devices@200/disk@1
```

2. Найдите vdc-экземпляр устройств, которые будут использоваться в пуле ZFS:

```
root@LDOM1 # cat /etc/path_to_inst
#
# Caution! This file contains critical kernel state
#
“/fcoe” 0 “fcoe”
“/iscsi” 0 “iscsi”
“/pseudo” 0 “pseudo”
“/scsi_vhci” 0 “scsi_vhci”
“/options” 0 “options”
“/virtual-devices@100” 0 “vnex”
“/virtual-devices@100/channel-devices@200” 0 “cnex”
“/virtual-devices@100/channel-devices@200/disk@0” 0 “vdc”
“/virtual-devices@100/channel-devices@200/pciv-communication@0” 0 “vpci”
“/virtual-devices@100/channel-devices@200/network@0” 0 “vnet”
“/virtual-devices@100/channel-devices@200/network@1” 1 “vnet”
“/virtual-devices@100/channel-devices@200/network@2” 2 “vnet”
“/virtual-devices@100/channel-devices@200/network@3” 3 “vnet”
“/virtual-devices@100/channel-devices@200/disk@1” 1 “vdc” << We want this one
```

3. Отредактируйте `/platform/sun4v/kernel/drv/vdc.conf`:

```
block-size-list="1:4096";
```

Это значит, что экземпляру устройства 1 назначен размер блока 4096 байт.

Дополнительный пример: предположим, что экземпляры vdsk с 1 по 6 нужно сконфигурировать с 4-килобайтными блоками, тогда `/etc/path_to_inst` выглядит следующим образом:

```
"/virtual-devices@100/channel-devices@200/disk@1" 1 "vdc"
"/virtual-devices@100/channel-devices@200/disk@2" 2 "vdc"
"/virtual-devices@100/channel-devices@200/disk@3" 3 "vdc"
"/virtual-devices@100/channel-devices@200/disk@4" 4 "vdc"
"/virtual-devices@100/channel-devices@200/disk@5" 5 "vdc"
"/virtual-devices@100/channel-devices@200/disk@6" 6 "vdc"
```

4. Окончательный файл `vdc.conf` должен содержать следующее:

```
block-size-list=»1:8192»,»2:8192»,»3:8192»,»4:8192»,»5:8192»,»6:8192»;
```

Предупреждение

Логический домен (LDOM) должен быть перезагружен после конфигурации `vdc.conf` и создания vdisk. Этот шаг не может быть пропущен. Продолжите настройку zpool и убедитесь, что для параметра `ashift` правильно установлено значение 12, как было описано ранее.

ZIL

Как правило, нет причин помещать журнал ZFS Intent Log (ZIL) на другое устройство. Журнал может использовать то же пространство, что и основной пул. Отдельный ZIL применяется, в основном, когда в современном массиве хранения используются физические диски без кэширования записи.

logbias

Установите параметр `logbias` в файловых системах ZFS, в которых размещаются данные Oracle.

```
zfs set logbias=throughput <filesystem>
```

Использование этого параметра уменьшает общую интенсивность записи. По умолчанию записанные данные сначала фиксируются в ZIL, а затем – в пуле хранения. Этот подход подходит для конфигурации простых дисков (plain disk), в которую входит SSD-устройство для ZIL и жесткие диски для основного пула хранения. Это объясняется тем, что он позволяет выполнять фиксацию за одну транзакцию ввода/вывода на доступном носителе с наименьшей задержкой.

При использовании современного массива хранения, который имеет собственные возможности кэширования, этого обычно не требуется. В редких случаях может быть желательно зафиксировать запись в журнал за одну транзакцию (например, в случае рабочей нагрузки, состоящей из особо концентрированных и чувствительных к задержке случайных операций записи). Это влечет такие последствия, как увеличение объема записи, поскольку внесенные в журнал данные в конечном счете пишутся в основной пул хранения, что приводит к удвоению активности записи.

Прямой ввод/вывод (Direct I/O)

Многие приложения, в том числе продукты Oracle, могут обходить буферный кэш хоста, обеспечивая прямой ввод/вывод. С файловыми системами ZFS эта стратегия не работает как следует. Хотя буферный кэш хоста и обходится, сама система ZFS продолжает кэшировать данные. Это может привести к сбивающим с толку результатам при тестировании производительности такими инструментами, как fio или sio, поскольку трудно предсказать, достигнет ли ввод/вывод системы хранения или будет локально кэширован в ОС. Это также сильно затрудняет использование синтетических тестов для сравнения производительности ZFS и других файловых систем. На практике при реальных рабочих нагрузках пользователей разницы в производительности файловых систем или нет, или она мала.

Множественные пулы zpool

Резервное копирование на основе моментальных снимков, восстановление с них, клонирование и архивирование данных в ZFS должно выполняться на уровне пула zpool и обычно требует нескольких пулов zpool. Пул zpool аналогичен пулу LVM-дисков и должен конфигурироваться по тем же правилам. Например, для БД, возможно, лучший вариант, когда файлы данных находятся в пуле `zpool1`, а архивные журналы, файлы управления и журналы транзакций – в пуле `zpool2`. Такой подход позволяет выполнять стандартное горячее резервное копирование, в котором БД переводится в режим горячего резервного копирования, за которым следует создание моментального снимка `zpool1`. Затем БД выводится из режима горячего резервного копирования, запускается принудительное архивирование журнала и создается моментальный снимок `zpool2`. Операция восстановления требует размонтирования файловых систем zfs и полного отключения zpool от системы, после чего выполняется операция восстановления SnapRestore. Затем zpool можно снова подключить к системе и восстановить БД.

filesystemio_options

В ZFS параметр Oracle `filesystemio_options` работает иначе. Если используется `setall` или `directio`, то операции записи синхронны и обходят буферный кэш ОС, а операции чтения буферизуются системой ZFS. Это затрудняет анализ производительности, поскольку иногда ввод/вывод перехватывается и обслуживается кэшем ZFS, что делает задержку хранения и общий объем ввода/вывода меньше, чем она могла бы быть.

21 Заключение

Как говорилось в начале этого документа, существует немного по-настоящему передовых приемов конфигурирования хранилища Oracle, потому что между реализациями очень много различий. Проект БД может содержать как одну критически важную БД, так и 5 000 унаследованных БД или БД разных размеров от нескольких гигабайт до сотен терабайт. Кластерное ПО и виртуализация только усугубляют эти различия.

Лучше говорить о конструктивных соображениях или сложностях, которые необходимо учитывать при планировании реализации хранилища. Правильное решение зависит как от технических деталей реализации, так и от бизнес-требований, определяющих проект. Эксперты NetApp и партнеров по оказанию профессиональных услуг готовы помочь в сложных проектах. Даже если помощь не требуется на протяжении всего проекта, NetApp настоятельно рекомендует новым заказчикам пользоваться профессиональными услугами при разработке высокоуровневого подхода.

Приложение A. Устаревшие блокировки NFS

В случае аварийного отказа сервера БД Oracle при перезапуске возможны проблемы с устаревшими блокировками NFS. Избежать этого можно, уделяя пристальное внимание настройке разрешения имен на сервере.

Эта проблема возникает из-за того, что для создания и снятия блокировки используются два метода разрешения имен, имеющих незначительные отличия. Участвуют два процесса: диспетчер сетевых блокировок (Network Lock Manager, NLM) и NFS-клиент. NLM использует `uname -n`, чтобы определить имя хоста, а процесс `rpc.statd` использует `gethostbyname()`. Чтобы ОС правильно снимала устаревшие блокировки, эти имена хостов должны совпадать. Например, хост может искать блокировки, принадлежащие `dbserver5`, хотя блокировки были зарегистрированы хостом как `dbserver5.mydomain.org`. Если `gethostbyname()` не возвращает то же самое значение, что и `uname -a`, значит, процесс снятия блокировки не увенчался успехом.

Следующий пример скрипта проверяет, является ли разрешение имен полностью согласованным:

```
#! /usr/bin/perl
$uname=`uname -n`;
chomp($uname);
($name, $aliases, $addrtype, $length, @addrs) = gethostbyname $uname;
print "uname -n yields: $uname\n";
print "gethostbyname yields: $name\n";
```

Если `gethostbyname` не соответствует `uname`, то вероятны устаревшие блокировки. Например, этот результат вскрывает потенциальную проблему:

```
uname -n yields: dbserver5
gethostbyname yields: dbserver5.mydomain.org
```

Проблема обычно решается изменением порядка появления хостов в `/etc/hosts`. Например, предположим, что файл hosts содержит следующую запись:

```
10.156.110.201 dbserver5.mydomain.org dbserver5 loghost
```

Чтобы решить эту проблему, измените порядок отображения полностью квалифицированного имени домена и короткого имени хоста:

```
10.156.110.201 dbserver5 dbserver5.mydomain.org loghost
```

Теперь `gethostbyname()` возвращает короткое имя хоста `dbserver5`, которое соответствует выводу `uname`. Поэтому блокировки снимаются автоматически после аварийного отказа сервера.

Приложение B. Проверка выравнивания в WAFL

Правильное выравнивание в WAFL крайне важно для достижения высокой производительности. Хотя ONTAP управляет 4-килобайтными блоками, этот факт не означает, что ONTAP выполняет все операции в 4-килобайтных блоках. На самом деле ONTAP поддерживает операции с блоками разных размеров, но на базовом уровне WAFL ведет учет 4-килобайтными блоками.

Термин «выравнивание» описывает, как ввод/вывод Oracle соотносится с этими 4-килобайтными блоками. Для оптимизации производительности нужно, чтобы 8-килобайтный блок Oracle располагался на двух 4-килобайтных физических блоках WAFL на диске. Если блок смещен на 2 КБ, значит, он занимает половину одного 4-килобайтного блока, отдельный полный 4-килобайтный блок и еще половину третьего 4-килобайтного блока. Это снижает производительность.

В файловых системах NAS выравнивание не является камнем преткновения. Файлы данных Oracle выравниваются по началу файла в зависимости от размера блока Oracle. Поэтому блоки размерами 8 КБ, 16 КБ и 32 КБ всегда выровнены. Все блочные операции смещены от начала файла кратно 4 КБ.

Напротив, LUN, как правило, содержат в начале некий заголовок диска или метаданные файловой системы, которые создают смещение. Выравнивание редко является проблемой в современных ОС, потому что эти ОС спроектированы для физических дисков, которые могут использовать собственный сектор размером 4 КБ, что также требует выравнивания ввода/вывода по 4-килобайтным границам для оптимизации производительности.

Тем не менее, есть ряд исключений. Возможно, БД была перенесена из более старой ОС, не оптимизированной под 4-килобайтный ввод/вывод, или из-за ошибки пользователя при создании раздела возникло смещение, отличное от 4 КБ.

Следующие примеры относятся к Linux, но процедуру можно адаптировать для любой ОС.

Выравнивание есть

В следующем примере показана проверка выравнивания на одном LUN с одним разделом. Сначала создайте раздел, который использует все доступные на диске разделы.

```
[root@jfs0 iscsi]# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0xb97f94c1.
Changes will remain in memory only, until you decide to write them. After that, of
course, the previous content won't be recoverable.

The device presents a logical sector size that is smaller than
the physical sector size. Aligning to a physical sector (or optimal I/O) size bound-
ary is recommended, or performance may be impacted.

Command (m for help): n Command action
e     extended
p     primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-10240, default 1): Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-10240, default 10240): Using default
value 10240

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table. Syncing disks.
[root@jfs0 iscsi]#
```

Выравнивание можно проверить математически следующей командой:

```
[root@jfs0 iscsi]# fdisk -u -l /dev/sdb

Disk /dev/sdb: 10.7 GB, 10737418240 bytes
64 heads, 32 sectors/track, 10240 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 65536 bytes
Disk identifier: 0xb97f94c1

Device Boot  Start  End          Blocks       Id     System
/dev/sdb1    32     20971519     10485744     83     Linux
```

Вывод команды показывает, что блоки имеют размер 512 байт, а начало раздела смещено на 32 блока. Это составляет 32 x 512 = 16 384 байт, что кратно 4-килобайтному блоку WAFL. Этот раздел выровнен правильно.

Чтобы проверить правильность выравнивания, выполните следующие шаги:

1. Определите универсальный уникальный идентификатор (UUID) LUN.

```
FAS8040SAP::> lun show -v /vol/jfs_luns/lun0 Vserver Name: jfs
LUN UUID: ed95d953-1560-4f74-9006-85b352f58fcd
Mapped: mapped
```

2. Войдите в оболочку узла (node shell) на контроллере ONTAP.

```
FAS8040SAP::> node run -node FAS8040SAP-02
Type 'exit' or 'Ctrl-D' to return to the CLI
FAS8040SAP-02> set advanced
set not found. Type '?' for a list of commands
FAS8040SAP-02> priv set advanced
Warning: These advanced commands are potentially dangerous; use
         them only when directed to do so by NetApp
         personnel.
```

3. Запустите сбор статистики на целевом UUID, определенном на первом шаге.

```
FAS8040SAP-02*> stats start lun:ed95d953-1560-4f74-9006-85b352f58fcd
Stats identifier name is 'Ind0xffffff08b9536188'
FAS8040SAP-02*>
```

4. Выполните какие-нибудь операции ввода/вывода. Важно использовать аргумент `iflag`, чтобы быть уверенным в том, что ввод/вывод синхронный и не буферизуется.

Примечание. Будьте крайне осторожны с этой командой. Если вы перепутаете аргументы `if` и `of`, то вы уничтожите данные.

```
[root@jfs0 iscsi]# dd if=/dev/sdb1 of=/dev/null iflag=dsync count=1000 bs=4096
1000+0 records in
1000+0 records out
4096000 bytes (4.1 MB) copied, 0.0186706 s, 219 MB/s
```

5. Остановите сбор статистики и изучите гистограмму выравнивания. Весь ввод/вывод должен находиться в интервале гистограммы .0, что означает, что ввод/вывод выровнен по границам 4-килобайтных блоков.

```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffff08b9536188
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-4f74-9006-
85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.0:186%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.1:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.2:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.3:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.4:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.5:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.6:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.7:0%
```

Выравнивания нет

В следующем примере показан не выровненный ввод/вывод:

1. Создайте раздел, который не выровнен по 4-килобайтной границе. Это не стандартная ситуация для современных ОС.

```
[root@jfs0 iscsi]# fdisk -u /dev/sdb Command (m for help): n
Command action e   extended
p     primary partition (1-4)
p
Partition number (1-4): 1
First sector (32-20971519, default 32): 33
Last sector, +sectors or +size{K,M,G} (33-20971519, default 20971519): Using default
value 20971519

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table. Syncing disks.
```

2. Созданный раздел смещен на 33 сектора вместо 32, как было бы по умолчанию. Повторите процедуру, описанную в разделе “Выравнивание есть.” Гистограмма выглядит следующим образом:

```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffff0468242e78
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-4f74-9006-
85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.0:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.1:136%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.2:4%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.3:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.4:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.5:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.6:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.7:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_partial_blocks:31%
```

Ясно видно отсутствие выравнивания. Ввод/вывод попадает в основном в интервал гистограммы `.1,` что соответствует ожидаемому смещению. Когда раздел создавался, он был сдвинут «вглубь» устройства на 512 байт дальше, чем оптимизированный раздел по умолчанию, а значит, и гистограмма смещена на 512 байт.

Кроме того, статистика `read_partial_blocks` не равна нулю, а значит, выполнялся ввод/вывод, который не заполнял весь 4-килобайтный блок полностью.

Журналирование транзакций

Описанные здесь процедуры применимы к файлам данных. Журналы транзакций Oracle и архивные журналы имеют разные схемы ввода/вывода. Например, журналирование транзакций – это циклическая перезапись одного файла. Если используется блок размером 512 байт (размер по умолчанию), то статистика записи выглядит примерно так:

```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffff0468242e78
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-4f74-9006-
85b352f58fcd lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.0:12%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.1:8%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.2:4%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.3:10%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.4:13%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.5:6%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.6:8%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.7:10%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_partial_blocks:85%
```

Ввод/вывод будет распределен по всем интервалам гистограммы, но с точки зрения производительности это не повод для беспокойства. Однако при чрезвычайно высокой скорости журналирования транзакций 4-килобайтный блок может быть полезен. В этом случае желательно убедиться в правильном выравнивании LUN журналирования транзакций. Однако это не так критично для хорошей производительности, как выравнивание файлов данных.

Используйте Interoperability Matrix Tool (IMT) на веб-сайте поддержки NetApp, чтобы убедиться, что версии продукта и функциональные версии, описанные в настоящем документе, поддерживаются для вашей конкретной среды. NetApp IMT определяет компоненты и версии продукта, которые можно использовать для построения конфигураций, поддерживаемых компанией NetApp. Конкретные результаты зависят от особенностей установленной у заказчика системы в соответствии с опубликованными спецификациями.

TR-3633-0717