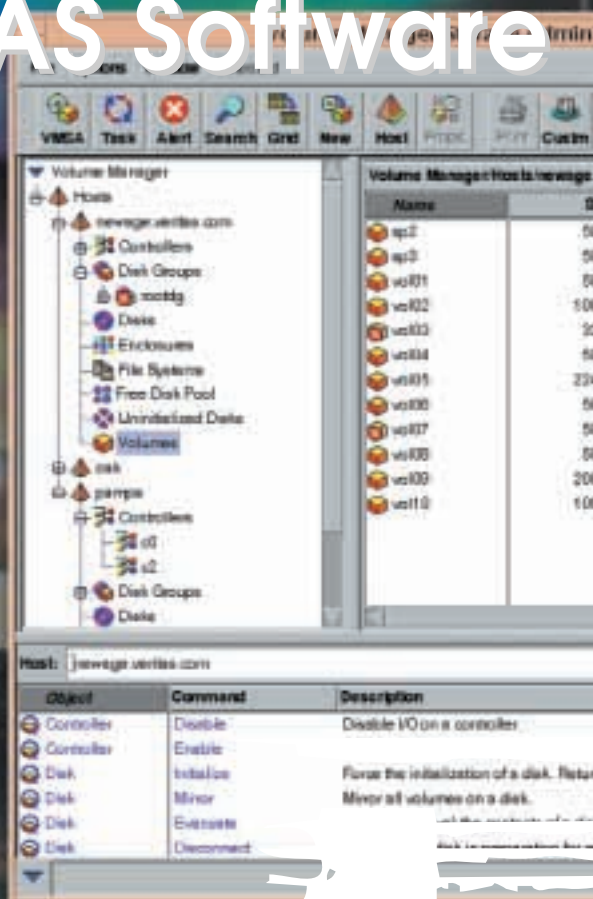


Jet Info

ИНФОРМАЦИОННЫЙ БЮЛЛЕТЕНЬ

№ 2 (93)/2001

Технологии управления хранением данных компании VERITAS Software



КОРПОРАТИВНЫЕ
СИСТЕМЫ

data
access

Технологии управления хранением данных компании VERITAS Software

Пол Массиглиа

Введение: требования к данным в современных информационных системах

Сегодня невозможно себе представить современное предприятие без развитой информационной инфраструктуры. Успешное ведение бизнеса напрямую связано с обработкой больших объемов информации, которая, в основном, представляется в виде электронных данных, хранимых и управляемых компьютерными системами. Информация должна быть:

- **Надежной.** Даже в случае выхода из строя дисков, остановки приложений и реконфигурирования систем информация должна быть доступна, чтобы предприятие могло реагировать на запросы партнеров и клиентов.
- **Быстродоступной.** Зачастую, в конкурентной борьбе все решает щелчок мышью. Когда клиенты или деловые партнеры запрашивают информацию, они ожидают получить незамедлительный ответ; никакие оправдания не принимаются.
- **Управляемой.** Информация должна быть защищена от отказов оборудования, ошибок по вине человека и злонамеренных действий. Необходимо сохранять онлайн-данные, а также перемещать их туда, где они нужны, зачастую во время работы с ними приложений.
- **Масштабируемой.** Количество данных, которое сегодня приходится обрабатывать предприятиям, постоянно растет. Поистине поражают воображение возможности роста, например, электронного бизнеса. По оценкам аналитиков, компания может увеличить свои доходы до 50% только за счет подключения к Интернету. Если растущий электронный бизнес не в состоянии расширить свои возможности доставки информации, то успех может превратиться в неудачу.
- **Невосприимчивой к авариям.** Подключение к Интернету позволяет модернизировать бизнес-операции, сокращает затраты и поднимает прибыльность. Однако, подключившись к Интернету, назад дороги нет. Необходимо обеспечить до-

ступность электронных данных и приложений даже в случае пожара, наводнения или выхода из строя электрической сети. Электронный бизнес должен уметь продолжать работать «как часы» при возникновении аварий.

Настоящая статья посвящена управлению хранением данных в современных информационных системах в плане поддержания доступности электронных данных, предоставления быстрого доступа к ним, управления, обеспечения возможности роста объема данных с расширением бизнеса, а также обеспечения невосприимчивости к авариям. В ней описывается использование технологий компании VERITAS для управления хранением информационных ресурсов, необходимых для работы бизнес-приложений.

Данные и базы данных

Технология реляционных баз данных (БД) лежит, как правило, в основе каждого современного бизнес-приложения. Данная технология надежна и проверена временем. Системы управления базами данных (СУБД) обеспечивают целостность транзакций (обрабатывается либо вся транзакция, либо она отбрасывается и не обрабатывается; при этом данные не повреждаются в результате таких «наполовину обработанных» транзакций) даже в том случае, когда несколько приложений получают одновременный доступ к данным.

Кроме того, БД осуществляют защиту целостности данных. Они хранят одну копию каждого элемента данных независимо от того, сколько приложений его используют. Встроенные фильтры предотвращают внесение недопустимых значений в БД. **Журналы отката (redo logs)** помогают восстановить непротиворечивость БД после отказа системы или приложения.

Все крупные бизнес-приложения, такие, как SAP R3, Baan и PeopleSoft, основаны на технологии баз данных. Компании, разрабатывающие СУБД, подобные Oracle, предлагают полные пакеты бизнес-приложений, которые используют лежащую в их основе технологию управления БД.

Системы управления базами данных

Бизнес-приложения предъявляют самые жесткие требования к информационной технологии в плане вычислений. С одной стороны, требования круглосуточной доступности 7 дней в неделю (схема 24x7) означают, что обработка данных в корпоративном вычислительном центре должна быть высоконадежной. С другой стороны, взрывной рост объемов обрабатываемых данных, наблюдаемый в наши дни, предполагает частое реконфигурирование систем по мере добавления и перераспределения аппаратных средств, перемещения и реструктурирования данных, а также эволюции приложений вместе с потребностями бизнеса и способами его ведения. Вопрос состоит в том, как обеспечить надежный доступ клиентов, поставщиков и сотрудников к необходимой им информации на уровне производительности, который бы их не разочаровал, посреди всего этого хаоса.

Очевидно, что СУБД помогают навести порядок в хаотической информационной среде современного бизнеса. основополагающий принцип управления БД заключается в отделении структуры и организации данных от приложений. СУБД обеспечивают стабильность работы приложений, целостность данных, семантику транзакций, а также возможности восстановления, которые позволяют «сохранить данные в целом виде» по мере роста, изменения и замены приложений и информационной инфраструктуры.

Онлайновые устройства хранения данных для СУБД

Применение СУБД — это очевидный способ удовлетворения растущих требований бизнеса. Однако, СУБД не существуют в вакууме. Чем выше качество устройств хранения данных, тем лучше будут работать СУБД. Устройства хранения данных должны обладать следующими характеристиками:

- **Надежность.** СУБД организуют содержимое огромного количества дисковых блоков во взаимосвязанные таблицы пользовательских данных и метаданных, которые в совокупности представляют состояние бизнеса. Они прекрасно справляются с задачей обеспечения логической целостности данных, но для обеспечения их физической целостности им необходим прочный фундамент.
- **Высокая производительность.** Для обеспечения гарантий целостности СУБД тщательно выстраивают порядок выполнения операций ввода—вывода, включая значительное количество вспомогательных операций, поддерживающих целостность данных и позволяющих восстановить их после отказа систем. Частота обращений к физическому устройству хранения БД зачастую значительно выше, чем предполагаемая частота запросов приложений к БД. В то время как

СУБД пытаются свести количество физических операций ввода—вывода к минимуму за счет широкого использования кэша, в конечном счете, данные приходится считывать с дисков и записывать на них. Более высокопроизводительный доступ к данным означает более быстрое реагирование приложений.

- **Масштабируемость.** В целом СУБД хорошо справляются с ростом объема данных. Большинство систем поддерживают возможность добавления дисковой памяти к БД во время их работы в режиме онлайн. Однако, чтобы воспользоваться этой возможностью, базовая структура устройств хранения должна быть также способна расти динамически. Но решение проблемы роста не означает только добавление дисковой памяти. Зачастую необходимо перераспределить рабочую нагрузку операций ввода—вывода между устройствами хранения, чтобы избежать появления горячих мест, в которых одни диски и шины ввода—вывода используются слишком интенсивно, а другие простаивают.

Варианты хранения данных: диски, тома и файлы

Всякий раз, когда реализуется БД, необходимо выбрать технологию ее хранения. Разработчики должны сделать выбор между:

- неструктурированными («raw») разделами диска,
- логическими томами,
- файлами,

которые будут использоваться в качестве контейнеров для таблиц БД и метаданных.

Неструктурированные («raw») разделы — это области последовательно пронумерованных дисковых блоков, доступ к которым осуществляется посредством прямых запросов ввода—вывода к драйверам ввода—вывода. Для таких разделов непроизводительные затраты на доступ минимальны, но они ограничены емкостью, производительностью операций ввода—вывода и характеристиками доступности дисков, на которых они размещены.

Тома являются логическими комбинациями дисковых блоков, выделяемых для приложений **диспетчером томов**, таким, как **VERITAS Volume Manager**. Диспетчеры томов можно реализовать программно на хосте, в RAID-контроллерах или в специализированных устройствах хранения данных. Роль диспетчера томов состоит в разбиении или объединении физических дисков для достижения желаемой производительности и доступности. Диспетчеры томов позволяют:

- разбивать большой диск на несколько меньших томов,
- объединять части или все пространство нескольких дисков, создавая один большой том,

- создавать зеркальные копии данных на двух или нескольких дисках или наборах дисков,
- использовать технологию RAID с контролем четности для защиты данных от потери при выходе диска из строя.

Диспетчеры томов предоставляют приложениям дисковую память, которая больше по объему, быстрее, надежнее и удобнее, чем самые большие, быстродействующие и самые надежные диски, существующие в настоящее время. Как правило, пользователи БД предпочитают тома неструктурированным разделам диска за их гибкость, несмотря на то, что некоторые реализации усложняют систему:

- **Целостность данных.** Некоторые диспетчеры томов используют кэш с обратной записью для повышения производительности. Без принятия надлежащих защитных мер кэш с обратной записью вносит элемент непредсказуемости, который может усложнить задачу обеспечения целостности данных СУБД в случае отказа системы. Меры защиты от системных сбоев продукта VERITAS Volume Manager рассмотрены ниже.
- **Интеграция с операционной средой.** Некоторые диспетчеры томов, особенно те, которые реализованы в RAID-контроллерах, эмулируют функции диска и не предусматривают тесную интеграцию с остальной частью операционной среды. Без такой интеграции использование возможностей томов, таких, как динамическое расширение памяти, затруднено. Интеграция VERITAS Volume Manager с файловой системой рассмотрена ниже.

Интуитивно понятно, что файлы должны быть идеальными «контейнерами» для хранения таблиц БД и индексов. Файлы удобны и управляемы. Они могут увеличиваться в размере по мере необходимости, а также перемещаться с места на место с помощью простых и понятных операций управления. Файлы являются естественными объектами для резервного копирования. Однако, в противовес этим преимуществам, файловые системы для хранения БД также имеют свои недостатки:

- **Слабые гарантии целостности.** В большинстве файловых систем используется кэш с обратной записью для повышения производительности приложений. Приложениям сообщается, что их запросы на запись обработаны до того, как данные реально записываются на диск. Записи на диск делаются медленно, порою спустя значительное время после того, как приложения предприняли действия, исходя из предположения, что данные сохранены. Хотя это повышает производительность приложений, отказы систем при наличии не записанных на диск данных в кэше могут вызвать безвозвратную потерю изменений, вносимых в БД, что приведет к повреждению данных (сценарий «несоответствия дебета кредиту»).

- **Дополнительная нагрузка на процессор.** Большинство файловых систем перемещают данные из памяти приложения в собственный буферный кэш, прежде чем записать их на диск. Это обеспечивает запись правильных данных даже для тех приложений, которые модифицируют данные в памяти до завершения операций ввода—вывода. Однако, перемещение данных в системной памяти сильно загружает процессор. Кроме того, СУБД не нуждаются в такой защите. Запись данных непосредственно из области памяти СУБД, как правило, уменьшает загруженность процессора.
- **Излишне ограничительная семантика совместного использования.** Большинство файловых систем позволяют любому количеству приложений одновременно читать файл, но только одному записывать в него. Для большинства приложений это вполне разумная политика. Одновременная запись файла двумя приложениями может привести к повреждению данных. Большинство СУБД имеют большое количество параллельных потоков выполнения, поэтому для файловой системы они выглядят как множество отдельных приложений. Однако, в отличие от обычных приложений, потоки СУБД координированы. Два потока получают одновременный доступ к объекту данных только в том случае, когда они могут это сделать безопасно. Файловые системы, которые предоставляют доступ к данным только одному потоку в определенный момент времени, излишне ограничивают производительность БД.

В связи с этим некоторые разработчики не решаются использовать файловые системы в качестве контейнеров для хранения БД. Файловая система VERITAS File System была специально разработана для преодоления указанных ограничений. Кроме того, она имеет и другие преимущества, рассматриваемые ниже, которые делают ее естественным выбором для хранения БД современных бизнес-приложений.

Полное управление хранением данных

Для того чтобы БД оптимально работала при всех непредсказуемых обстоятельствах, являющихся, например, неотъемлемой частью функционирования бизнес-приложений в режиме онлайн, ей необходима инфраструктура, которая:

- Обеспечивает надежное, высокопроизводительное и гибкое хранение данных в режиме онлайн.
- Позволяет осуществлять резервное копирование и управление в масштабе реального времени в процессе работы БД.
- Поддерживает возможность расширения БД.
- Максимально повышает доступность данных, включая защиту от аварий.

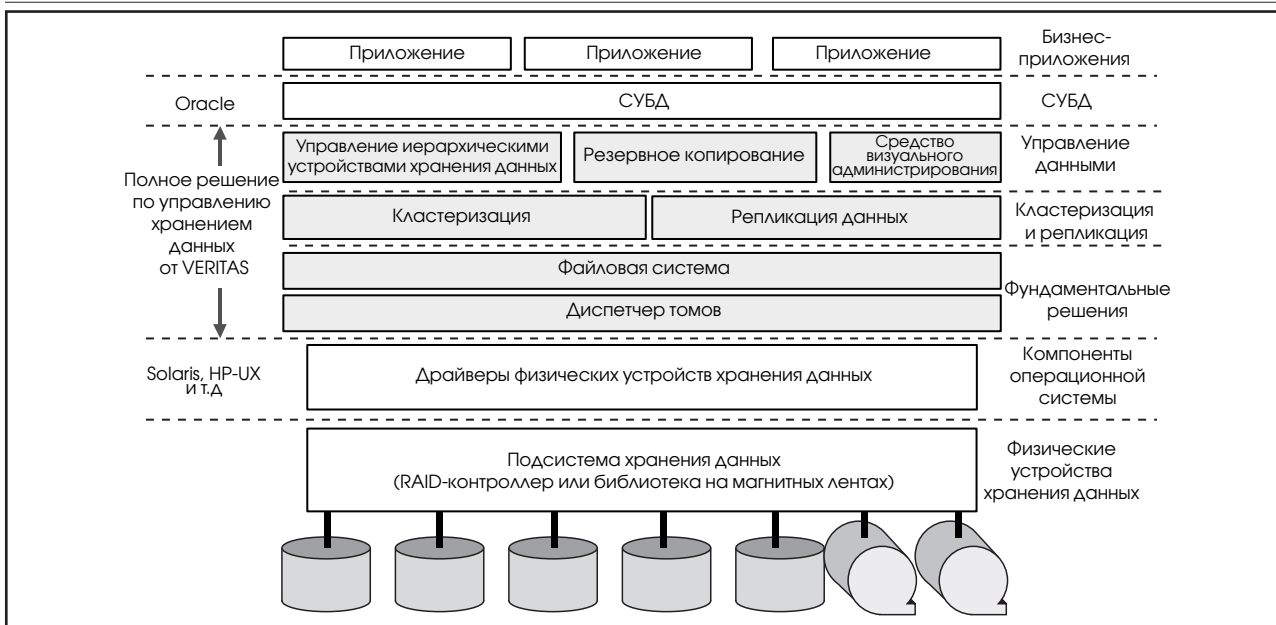


Рис. 1. «Стек» управления данными и обеспечения доступа к нимю.

Компания VERITAS предлагает комплексную инфраструктуру для БД современных бизнес-приложений и называет ее «**полное управление хранением данных**». Полное управление хранением данных VERITAS для БД — это интеграция технологий управления данными и их хранением, охватывающих весь спектр существующих устройств хранения данных. Компания VERITAS предприняла дальнейшие шаги по интеграции некоторых из своих технологий с основными бизнес-приложениями и СУБД, создав специализированные пакеты управления хранением данных, оптимизированные для таких сред. Набор продуктов Database Edition for Oracle представляет собой один из таких пакетов.

Пакет Database Edition for Oracle

Пакет Database Edition for Oracle состоит из следующих компонентов:

- **VERITAS File System.** Высокопроизводительная и быстро восстанавливаемая файловая система, оптимизированная для рабочих нагрузок БД.
- **VERITAS Volume Manager.** Диспетчер томов, поддерживающий рассеивание данных (data striping), зеркалирование (mirroring) и другие средства для повышения доступности данных, производительности и гибкости.
- **VERITAS Visual Administrator.** Простой в использовании графический пользовательский интерфейс для администрирования VERITAS File System и VERITAS Volume Manager.
- **VERITAS Quick I/O for Databases.** Дополнительная опция файловой системы, которая повышает производительность БД Oracle, хранимых в VERITAS File System, до уровня неструктурированных («raw») дисков. Опция Cached Quick I/O позволяет еще больше увеличить производи-

тельность за счет использования больших областей системной памяти для буферизации данных, доступ к которым осуществляется часто.

- **VERITAS NetBackup Block-Level Incremental (BLI) Backup Extension for Oracle.** Данный продукт поддерживает инкрементальное резервное копирование БД, позволяющее минимизировать время простоя БД, время резервного копирования, объем резервируемых данных, а также загрузку процессора и сети.

VERITAS File System, VERITAS Volume Manager, VERITAS Visual Administrator — являются продуктами общего назначения, преимуществами которых может воспользоваться любое приложение и интегрируются с возможностями СУБД Oracle для обеспечения преимуществ, свойственных только ей.

«Стек» управления данными и обеспечения доступа к ним

Чтобы помочь в определении местоположения компонентов инфраструктуры **полного управления хранением данных**, как включенных в **Database Edition for Oracle**, так и других компонентов, полезных в среде Oracle, на рис. 1 представлена схема управления данными и обеспечения доступа к ним для среды выполнения приложения. По аналогии со стеками в сетевой модели, рис. 1 можно рассматривать как «стек» управления данными и обеспечения доступа к ним для бизнес-приложений.

Как видно из рис. 1, использующие БД приложения требуют наличия сложной программной инфраструктуры управления данными и их хранением. Такой инфраструктурой является **полное управление хранением данных** компании VERITAS. В настоящей статье рассматриваются следующие программные технологии этой инфраструктуры:

- **Базовые технологии.** VERITAS File System и VERITAS Volume Manager можно использовать по отдельности или совместно для создания высокопроизводительного, гибкого и масштабируемого фундамента для БД бизнес-приложений. Они позволяют реализовать рассеивание данных по дискам, зеркалирование и технологию RAID, создавать моментальные снимки данных (snapshots), производить инкрементальное резервное копирование на уровне блоков, расширять и реконфигурировать дисковую память в режиме онлайн, а также быстро восстанавливать данные после системных сбоев.
- **Технологии реплицирования данных.** Продукт VERITAS Storage Replicator обеспечивает доступность данных в случае, когда центр обработки данных полностью выходит из строя. Кроме того, репликация позволяет своевременно как распространять информацию в географически распределенные центры обработки данных, так и консолидировать информацию из территориально-распределенных офисов в центральные, а также выполнять различные служебные функции за пределами вычислительного центра, повышающие доступность приложений.
- **Технология кластеризации.** Продукт VERITAS Cluster Server позволяет сохранять работоспособность системы в случае выхода из строя приложений или компьютеров, на которых они запущены. Технология кластеризации поддерживает возможность переноса выполнения приложений на альтернативные компьютеры при возникновении проблем. Для некоторых классов приложений кластеризация также допускает масштабирование за счет добавления новых компьютеров к существующим кластерам. Продукт VERITAS Global Cluster Manager поддерживает возможность объединения кластеров, разбросанных по всему миру, в единый глобальный вычислительный ресурс, которым можно управлять с одной консоли управления.
- **Технология защиты данных.** Информацию необходимо защищать, особенно при функционировании информационной системы в режиме онлайн по схеме 24x7. Продукты NetBackup и Storage Migrator компании VERITAS позволяют решить эту задачу несколькими способами. Резервное копирование для СУБД гарантирует создание непротиворечивых с точки зрения транзакций резервных копий БД, а также поддерживает свежие копии образов программ и других вспомогательных файлов для быстрого восстановления всей среды в случае необходимости. Система управления иерархической памятью выявляет редко используемые файлы и переводит их в оффлайн на более дешевые носители (магнитные ленты и т.п.), освобождая дисковую память и ресурсы ввода-вывода для выполнения бизнес-операций.

В следующих разделах данной статьи вышеперечисленные группы технологий будут рассмотрены подробно, а также будет показано, как они интегрируются для удовлетворения потребностей в управлении хранением БД современных бизнес-приложений.

I. Базовые технологии

Сущность операций ввода—вывода в работе БД

На основе интеграции базовых технологий управления томами и файловой системы компания VERITAS создала высокодоступный, высокопроизводительный, гибкий и масштабируемый фундамент для хранения БД, лежащих в основе бизнес-приложений. Чтобы понять причины, по которым базовые технологии VERITAS являются идеальным решением по хранению БД, рассмотрим основные характеристики операций ввода-вывода БД.

Большинство приложений интенсивно работают с запросами на ввод-вывод. Они делают большое количество запросов на ввод-вывод в секунду, но в каждом запросе пересылается лишь несколько килобайт данных. Порядок, в котором приложения запрашивают данные, непредсказуем, поэтому СУБД обращаются к устройствам хранения в произвольном порядке. Сочетание произвольного доступа с небольшими по объему запросами на ввод-вывод означает, что получение доступа к данным (позиционирование головок чтения/записи диска в нужное место на нужной дисковой пластине) важнее для производительности, чем их передача. Производительность операций обращения к дискам — это важный фактор производительности приложений. Существуют два способа повышения производительности обращения к дискам для приложений, интенсивно работающих с запросами на ввод-вывод:

- Кэширование данных в памяти временного хранения (что исключает обращения к диску).
- Добавление новых дисков и распределение нагрузки операций ввода—вывода между ними.

Кэш и СУБД

Кэш хорошо работает при считывании данных из БД. Данные можно считывать с диска в кэш один раз и пересылать приложениям десятки раз до тех пор, пока они не изменятся. СУБД широко используют кэш для чтения данных. Крупная БД может использовать больший объем кэша, чем способна непосредственно адресовать СУБД. Средство VERITAS Cached Quick I/O for Databases позволяет использовать очень большую кэш-память для чтения БД.

Кэш для записи имеет некоторые отличия. В случае кэша со **сквозной записью** данные записываются на диск до того, как приложению подается сигнал о завершении операции записи. Производитель-

ность операции записи при наличии кэша со сквозной записью примерно совпадает с ее производительностью при отсутствии кэша. Преимущества возникают тогда, когда приложения повторно считывают недавно записанные данные, и их можно непосредственно переслать из кэша с очень малой задержкой.

В случае кэша с **обратной записью** диспетчер кэша объявляет о завершении операций записи данных приложения, как только они перемещены в кэш. Реальные операции записи на диск откладываются до тех пор, пока диспетчер кэша не сможет их выполнить в удобное время. Кэш с обратной записью может существенно повысить производительность приложений потому, что ему не требуется ждать ввода—вывода на физический диск, прежде чем продолжить свою работу. Однако, при хранении данных в кэше с обратной записью возникает опасность исчезновения тех данных, которые приложения «считают» успешно записанными, в случае, когда отказ системы приводит к уничтожению содержимого кэша до того, как они смогут быть записаны на диск.

Репутация СУБД, касающаяся целостности данных, основана на обеспечении гарантий защищенности пользовательских данных. В частности, СУБД требуют, чтобы обновления БД записывались на диск до того, как приложениям сообщается о завершении их транзакций. Причина этого проста. Если обновления данных для завершенной транзакции БД существуют только в энергозависимом кэше, то системный сбой может уничтожить их, что, возможно, приведет к расхождению между реальными бизнес-операциями и электронными записями.

Производители БД обычно не рекомендуют использовать энергозависимый кэш с обратной записью. Большинство из них советуют отключить подсистему RAID и дисковый кэш с обратной записью. Они считают, что если кэш с обратной записью используется прозрачно для СУБД, то она не сможет узнать, когда именно данные были сохранены на диске, а, следовательно, не сможет доверять журналу транзакций, который позволяет ей гарантировать целостность данных. СУБД зачастую не могут воспользоваться одним из наиболее важных средств повышения производительности операций ввода—вывода — кэшем с обратной записью.

Распределение нагрузки операций ввода—вывода

Поскольку обращения к диску играют решающую роль в производительности приложений, интенсивно работающих с запросами на ввод—вывод, существует возможность ее повышения посредством добавления новых дисков (т. е. увеличения объема дисковой памяти для получения одновременного доступа к различным элементам данных) и распределения данных между ними. К сожалению, ситуация далеко не так проста, как кажется. Большинство прило-

жений вызывают появление «горячих мест» в БД — нескольких сегментов таблиц или индексов, обращение к которым происходит гораздо чаще, чем к остальной части БД. Одни «горячие места», такие, как корни деревьев индексов, являются постоянными, другие — временными. Так, например, в начале рабочего дня сегменты таблиц, содержащие клиентские записи восточного побережья США, могут оказаться «горячим местом» клиентской БД. Однако, со временем можно наблюдать смещение активности операций ввода—вывода в сторону сегментов таблицы, содержащей клиентские записи западного побережья.

Если индекс или сегмент таблицы хранится на одном диске, то производительность данного диска будет ограничивать скорость доступа к нему. Нередко встречается ситуация, когда некоторые из дисков, содержащих крупную БД, перегружаются запросами на ввод—вывод, тогда как большинство других практически бездействуют. Один из способов оптимизации производительности приложений состоит в реорганизации индексов и сегментов таблиц таким образом, чтобы те из них, обращение к которым происходит часто, находились на разных дисках. В принципе этот способ работает, однако, он зачастую неэффективен по трем причинам:

- **Он требует много времени и ресурсов.** Для повышения производительности с помощью указанного метода администратор БД должен уделить время на отслеживание скорости доступа, анализ результатов и копирование объектов данных с перегруженных дисков на слабонагруженные. Меньше всего большинству администраторов БД нужна дополнительная повторяющаяся работа, и меньше всего большинству БД, работающих в режиме онлайн, нужны дополнительные непроизводительные фоновые операции ввода—вывода.
- **Не гарантируется его точность.** Любой анализ, приводящий к перемещению объектов данных между дисками, предполагает, что будущий характер доступа к БД будет таким же, как прошлый, на котором основан анализ. Если различные части БД используются во вторник интенсивнее, чем в понедельник, то настройка на основе характера доступа в понедельник не даст никакого выигрыша.
- **Его реализация может оказаться невозможной.** Большинство СУБД поддерживают физическую сегментацию таблицы по одному индексу (например, по диапазону клиентских имен или почтовых кодов). Сегменты можно хранить на разных дисках. Однако, если обращения к дискам коррелируют с такой сегментацией, то «горячие места» все еще могут возникнуть. Например, упорядоченная группа транзакций может последовательно обновлять записи клиентов, чьи имена начинаются с буквы В. Это приведет к обращению к диску, содержащему данный сегмент таблицы, тогда как другие диски будут бездействовать.

Диспетчер томов VERITAS Volume Manager позволяет распределить почти любую нагрузку операций ввода-вывода посредством рассеивания блоков данных по разным дискам. Рассеивание данных позволяет распределить сегменты таблицы между несколькими дисками прозрачным для СУБД образом. В случае рассеивания данных не имеет значения, какие сегменты таблицы являются «горячими»; каждая таблица распределяется между несколькими дисками, поэтому обращения к ним имеют тенденцию распределяться равномерно между множеством аппаратных ресурсов.

Томы: целостность данных и производительность операций ввода—вывода

Первое требование, предъявляемое к приложению, работающему по схеме 24x7 (т.е. непрерывно), состоит в том, чтобы обеспечить круглосуточную доступность данных. Диски, каналы ввода—вывода и даже компьютеры могут выйти из строя, однако, для успеха бизнеса необходима постоянная доступность информации. VERITAS Volume Manager позволяет объединять диски в отказоустойчивые виртуальные томы, которые сохранят работоспособность в случае отказов дисков и каналов ввода—вывода. Для СУБД и других приложений тома функционально эквивалентны физическим дискам. Однако, они могут сохранять работоспособность при отказах дисков за счет использования **технологии RAID и технологии зеркалирования**. Будучи примерно функционально эквивалентными аппаратным дисковым подсистемам RAID, тома, созданные с помощью программных RAID, базирующихся на хосте, имеют следующие два основных преимущества:

- **Значительно меньше начальные и дополнительные капиталовложения в оборудование.** Бизнес на начальном этапе развития, как правило, не

имеет больших объемов информации, тем более в режиме онлайн. Поскольку программный RAID, выполняемый на хосте, может работать с обычными дисками, непосредственно подключенными к шинным адаптерам хоста (HBA), он в состоянии защитить небольшое количество данных без крупных капиталовложений в подсистему RAID. По мере роста бизнеса и увеличения объема его данных программный RAID, базирующийся на хосте, может обеспечить увеличение дискового пространства небольшими порциями размером в один диск, в отличие от аппаратных подсистем RAID, которые зачастую требуют более крупных дополнительных капиталовложений для того, чтобы оправдать их установку.

- **Томы, реализуемые на хосте, функционально дополняют аппаратные подсистемы RAID.** По мере роста бизнеса и увеличения количества приложений, требующих доступ к возросшему объему данных в режиме онлайн, аппаратные подсистемы RAID становятся, в конце концов, предпочтительными, хотя бы только по причинам физической конфигурации. Томы, использующие программный RAID на хосте, способны объединять виртуальные диски аппаратных подсистем с целью увеличения их объема, а также повышения доступности и производительности (например, посредством зеркалирования данных на дисковых подсистемах, подключенных посредством аппаратного RAID-контроллера).

Какой тип подсистемы RAID выбрать?

VERITAS Volume Manager поддерживает как зеркалированные отказоустойчивые тома, так и тома на основе **RAID с контролем четности**, а также тома, не являющиеся отказоустойчивыми. В таблице 1 приводятся характеристики доступности томов различных типов, поддерживаемых диспетчером томов VERITAS Volume Manager.

Тип тома	Доступность данных	Приложение, основанное на БД
Зеркалированный (Mirrored) (2 или более) (иногда называется RAID 1)	Очень высокая	Критически важные объекты данных размера не более одного диска (например, БД людских ресурсов)
Рассеянный по дискам и зеркалированный (Striped and Mirrored) (две и более копии)	Очень высокая	Крупные критически важные объекты данных на нескольких дисках (например, данные о продажах)
Рассеянный по дискам с контролем четности (Striped with Parity) ("RAID 5")	Высокая	Важные данные, в основном для чтения
Рассеянный по дискам без контроля четности (Striped without Parity) (иногда называется RAID 0)	Ниже, чем отдельный диск	Легко заменяемые некритичные данные (наименее используемые в бизнесе)
Составной (с различными возможностями)	Ниже, чем отдельный диск	Легко заменяемые некритичные данные (наименее используемые в бизнесе)

Таблица 1. Применения RAID: доступность.

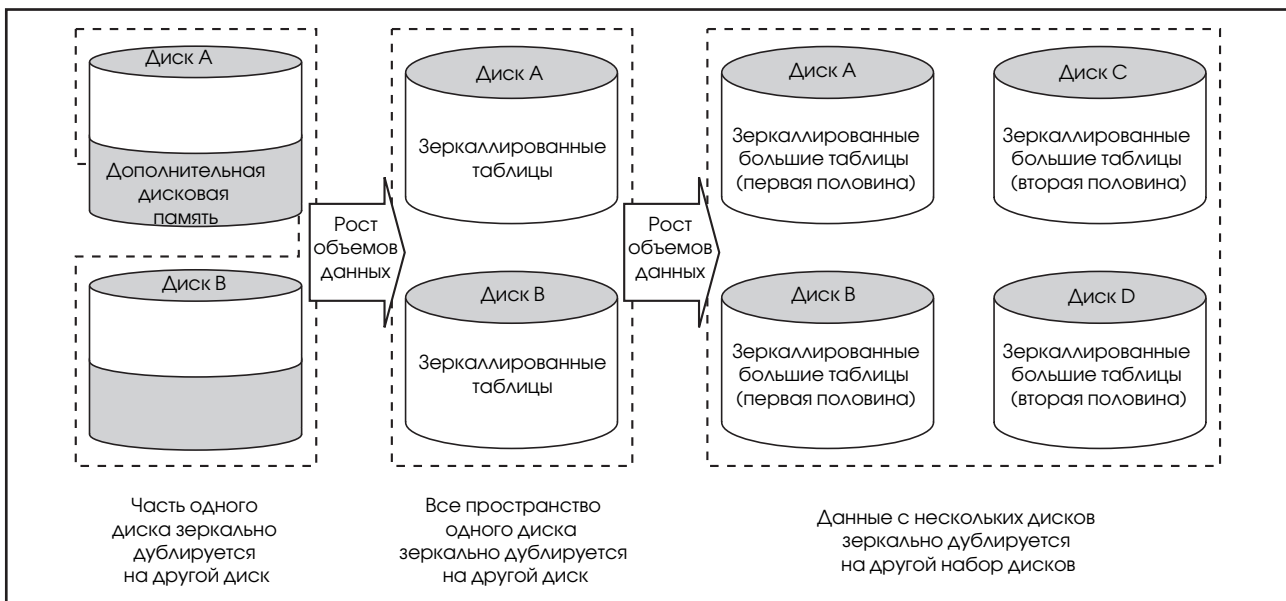


Рис. 2. Различные конфигурации зеркалирования с помощью VERITAS Volume Manager.

Как видно из таблицы 1, зеркалированные тома обеспечивают наилучшую защиту от отказов дисков потому, что при наличии нескольких дисков, содержащих избыточные данные, они могут сохранить работоспособность в случае множественных различных отказов дисков. Кроме того, когда диск выходит из строя, отказ оказывает меньшее воздействие на производительность приложения, чем в случае RAID с контролем четности. VERITAS Volume Manager позволяет зеркалировать данные на участках двух и более дисков, на двух и более целых дисках или на двух и более наборах дисков, как показано на рис. 2.

Такая гибкость позволяет бизнесу начать работу при скромных затратах на аппаратные средства, а также добавлять дисковую память по мере его роста, откладывая крупное инвестирование в оборудование до тех пор, пока это не будет необходимо. В процессе роста бизнеса и добавления дисковой памяти, онлайн-утилиты управления диспетчера томов компании VERITAS поддерживают перемещение данных с меньших на большие тома (как указано стрелками «роста объемов данных» на рис. 2) во время использования данных приложениями.

Еще одно преимущество зеркалированных томов

VERITAS Volume Manager позволяет сохранить три и более зеркальных копий данных на отдельных дисках. Зеркалирование с использованием трех копий полезно в том случае, когда прикладную БД необходимо «заморозить» на некоторое время и использовать ее для другой цели, например, для резервного копирования, извлечения информации или тестирования новых приложений. С помощью технологии трехкратного зеркалирования компании VERITAS одну копию «замороженного» тома можно «от-

делять» от основной БД для других применений, в то время как приложения возобновят свою работу на фоне все еще отказоустойчивой БД.

Рассеивание данных по дискам для распределения нагрузки операций ввода—вывода

VERITAS Volume Manager также поддерживает рассеивание данных по нескольким дискам для того, чтобы распределить нагрузку и, тем самым, повысить производительность операций ввода-вывода. Данные можно распределить по дискам в трех конфигурациях:

- **Без защиты от отказов.** Данная конфигурация, иногда называемая RAID 0, позволяет распределить нагрузку операций ввода—вывода между доступными дисковыми ресурсами, но не предусматривает защиту от отказов дисков. Следует быть осторожным при использовании данной конфигурации для критически важных для бизнеса данных.
- **С защитой посредством контроля четности.** Данную конфигурацию часто называют RAID 5. Она обеспечивает защиту от отказа одного диска при меньших затратах, чем в случае зеркального дублирования, потому что один диск с «контрольными данными» позволяет защитить произвольное количество дисков с пользовательскими данными от отказа одного из них. Однако, RAID 5 теряет в производительности на операциях записи, что делает ее непригодной для часто обновляемых таблиц. В бизнес-приложениях RAID 5 подходит для хранения важных данных, которые обновляются нечасто, таких, как Web-страницы или хранилища данных.

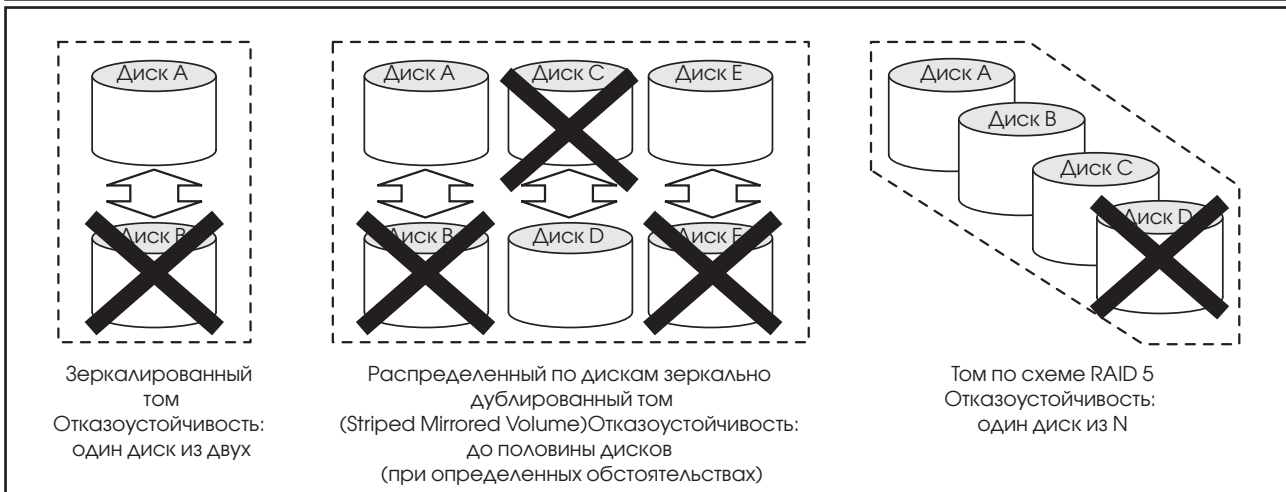


Рис. 3. Отказоустойчивость RAID-томов различных типов.

• **С зеркалированием.** Данную конфигурацию иногда называют RAID 1+0 или RAID 0+1. Она добавляет отказоустойчивость зеркалирования в процесс распределения нагрузки, обеспечиваемый рассеиванием данных. Комбинация рассеивания данных и зеркалирования — это предпочтительное сочетание отказоустойчивости, оптимальной производительности и практичности для критически важных онлайн-данных. Рассеивание данных в сочетании с зеркалированием следует использовать для всей критически важной информации бизнеса, такой, как записи продаж и финансовые записи, а также для любых часто обновляемых данных, таких, как инвентарные и производственные записи.

Обеспечение доступности данных, защищенных с помощью RAID

Пригодность того или иного типа RAID для бизнес-приложений, функционирующих по схеме 24x7, в значительной степени определяется тем, что происходит с данными при отказе диска. Отказы двух важных видов оказывают влияние на доступность данных при использовании программного RAID на хосте:

- Отказ диска, входящего в состав тома RAID.
- Выход из строя системы, в которой работает диспетчер томов RAID.

RAID и отказы дисков

Когда зеркалированный диск выходит из строя, диспетчер томов просто обращается к одному из оставшихся дисков тома. На томе RAID с контролем четности, содержащем отказавший диск, диспетчер томов должен восстановить данные, запрашиваемые приложениями. Таким образом, как зеркалированные тома, так и тома RAID допускают отказы отдельных дисков. Однако, быстрое восстановление данных после отказа диска также имеет большое значение для предотвращения потери данных в случае выхода из строя второго диска. Процесс

восстановления данных после отказа диска состоит из двух частей:

- Замена отказавшего диска.
- Восстановление содержимого отказавшего диска на новом диске.

Функция «горячего» перемещения (**hot relocation**) VERITAS Volume Manager позволяет администратору устройств хранения данных предварительно назначить один и более дисков в качестве **запасных**, которые будут автоматически использоваться при отказе диска. При наличии запасного диска диспетчер томов немедленно его выделяет и начинает восстановление содержимого отказавшего диска на нем. Это позволяет устранить вмешательство человека в процесс восстановления данных, а также сводит к минимуму время, требуемое для полного восстановления избыточности данных.

VERITAS Volume Manager восстанавливает содержимое отказавшего диска прозрачным для приложений образом во время использования поврежденного тома приложениями. Однако, восстановление — это процесс, интенсивно работающий с операциями ввода-вывода, поэтому оно может неблагоприятно сказаться на выполнении приложений. VERITAS Volume Manager позволяет системному администратору контролировать операции ввода-вывода при восстановлении, чтобы свести к минимуму воздействие на приложения.

Меры защиты Volume Manager от отказов системы

Операции записи данных приложений на тома RAID должны быть **неделимыми (atomic)**, даже если они требуют большого количества операций ввода-вывода на нескольких дисках; это значит, что каждая запись на том должна выполняться либо полностью, либо вообще не выполняться. Например, каждая запись данных приложения на зеркалированный том должна отражаться на всех его дисках. Аналогично, всякий раз, когда приложение записывает данные на том RAID 5, необходимо обновлять как пользовательские данные, так и данные контроля четности.

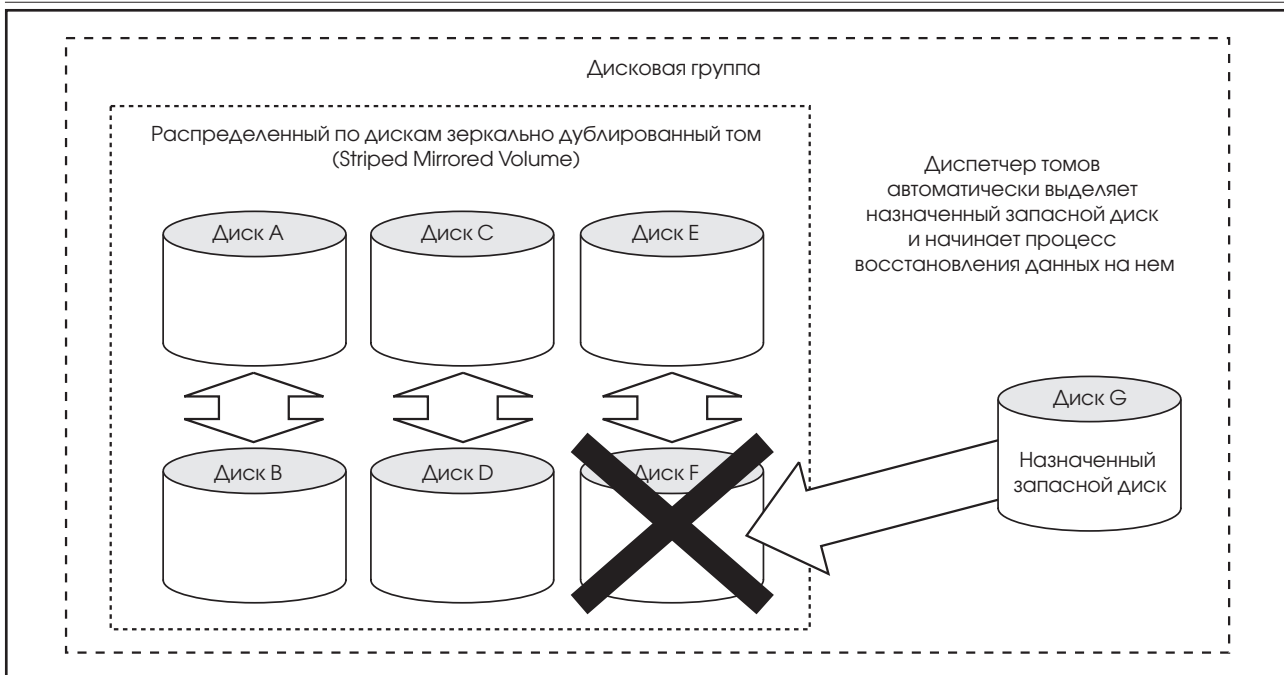


Рис. 4. Функция «горячего» перемещения позволяет уменьшить время восстановления.

Гарантирование неделимости обновлений является одной из самых сложных проблем технологии RAID, поскольку системы могут выйти из строя тогда, когда операции обновления на томе RAID уже частично завершены. Если это произойдет, то возможно повреждение данных, которое может быть обнаружено спустя значительное время после отказа. Приведем примеры:

- Если отказ системы происходит до того, как данные были записаны на все диски зеркалированного тома, то последующие запросы приложений на чтение могут дать различные результаты, в зависимости от того, с какого диска выполняется данная операция чтения.
- Если отказ системы происходит после обновления пользовательских данных на томе RAID 5, но до соответствующего обновления данных контроля четности, то повреждение может оставаться незамеченным в течение нескольких недель или месяцев после перезапуска системы. Только при отказе другого диска и восстановлении его данных с помощью устаревших данных контроля четности повреждение будет замечено.

Гарантирование неделимости обновлений томов RAID требует следующего:

- Выявления возможной противоречивости данных на томе после выхода из строя системы.
- Максимально быстрое восстановление после сбоя внутренней непротиворечивости тома (т.е. обеспечение идентичности содержимого всех дисков зеркалированного тома или соответствия данных контроля четности пользовательским данным для всех блоков тома RAID 5).

В крайнем случае, диспетчер томов может просто предположить, что все его тома являются противоречивыми после отказа системы. Для тома RAID такое предположение потребовало бы обновления всех его блоков данных контроля четности. Для зеркалированного тома возможно несколько решений. Например, содержимое одного диска можно скопировать на другие диски полностью. Какими бы ни были конкретные алгоритмы, предположение о том, что нельзя доверять никаким данным на отказоустойчивом томе после выхода из строя системы, приводит к длительному восстановлению данных, поскольку необходимо повторно скопировать и восстановить все содержимое тома. Более практичной стратегией является ведение журналов регистрации обновляемых областей данных тома. Такой журнал можно прочитать при возобновлении работы системы после отказа, чтобы определить, какие области данных необходимо восстановить.

VERITAS Volume Manager использует два похожих метода регистрации обновлений для восстановления целостности зеркалированных томов и томов RAID с контролем четности после выхода из строя системы:

- Регистрация обновлений рассеянных по дискам областей тома RAID 5 в **журнале обновлений RAID**.
- Регистрация областей (нескольких блоков) зеркалированного тома, в которые приложениями недавно были записаны данные (т.е. которые могут быть «грязными»), в **журнале «грязных» областей**. После отказа необходимо восстановить непротиворечивость только «грязных» областей,

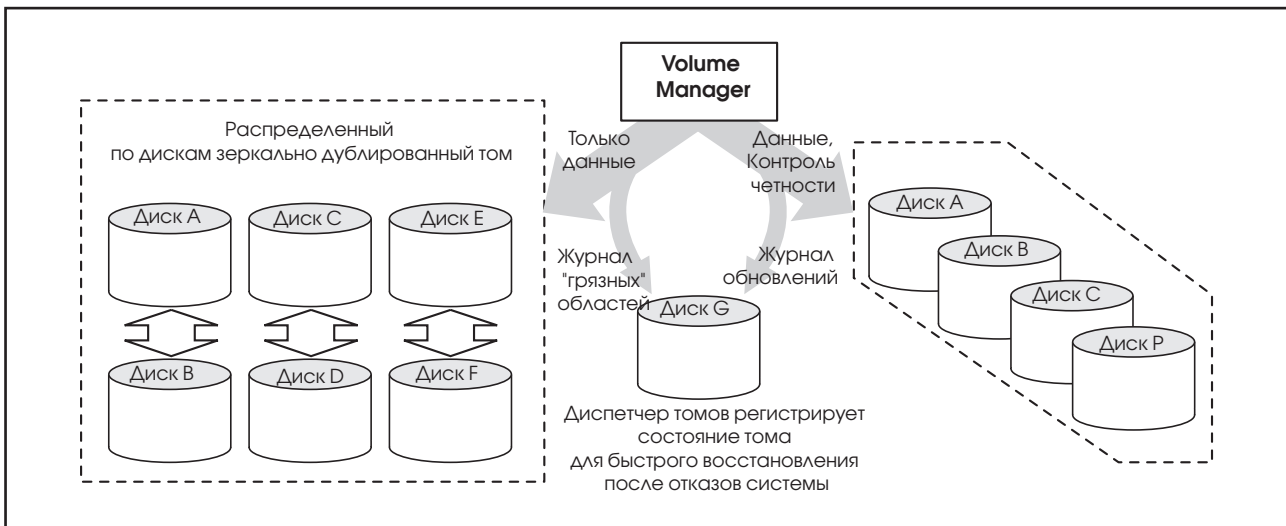


Рис. 5. Ведение диспетчером томов журнала обновлений и журнала «грязных» областей.

поскольку только они подвержены риску стать противоречивыми.

На рис. 5 проиллюстрированы оба метода регистрации обновлений.

Прежде чем начать выполнение последовательности операций чтения и записи на диски, соответствующей операции записи данных приложения на том RAID 5, Volume Manager делает запись в журнале регистрации, указывающую, какие области обновляются. При восстановлении данных после отказа системы Volume Manager читает журнал регистрации и восстанавливает данные контроля четности для каждой области, указанной в нем. Поскольку лишь небольшая часть чередующихся областей тома, вероятно, подвергается обновлению в любой момент времени, это позволяет свести к минимуму время восстановления данных (и, соответственно, время восстановления доступа приложения к данным).

Аналогично, прежде чем начать запись данных на диск зеркалированного тома, Volume Manager обновляет битовую карту (bit map) в журнале «грязных» областей, указывая диапазон обновляемых блоков. Чтобы свести к минимуму потерю производительности, связанную с ведением журнала регистрации, в нем регистрируются только недавно «загрязненные» области блоков. При восстановлении данных после системного сбоя Volume Manager читает битовую карту и восстанавливает непротиворечивость только тех областей, которые подвергались изменениям. Поскольку измененные области обычно составляют лишь небольшую долю объема тома, время восстановления мало по сравнению с повторным копированием всех его блоков.

Интеграция с механизмами восстановления БД

СУБД включают в свой состав механизмы ведения журналов регистрации событий и восстановления данных, предназначенных для восстановления

целостности БД после выхода из строя системы. Даже при восстановлении содержимого тома, на котором располагается БД, отсутствует гарантия **целостности транзакций**, т. е. отражения в БД только завершенных транзакций. Например, СУБД Oracle обрабатывает **журнал отката (redo logs)** после отказа системы для восстановления непротиворечивости БД. Поскольку СУБД имеет более точную информацию о том, какие данные подвержены риску, имеет смысл позволить ей контролировать процесс восстановления данных после системного сбоя.

В базе данных, использующей зеркалированные тома, пакет продуктов **VERITAS Database Edition for Oracle** ускоряет восстановление данных после отказа системы, позволяя СУБД Oracle контролировать большую часть процесса. Volume Manager инициирует процесс восстановления, используя свой журнал «грязных» областей для восстановления (зеркалированного) журнала отката СУБД Oracle. Затем СУБД обрабатывает свой журнал отката и просит Volume Manager восстановить только те данные, которые действительно могли пострадать. Volume Manager не применяет свой журнал «грязных» областей к томам, содержащим таблицы БД. Данный интегрированный метод имеет два преимущества:

- **Быстрый перезапуск.** Поскольку диспетчер томов не восстанавливает тома, содержащие таблицы данных, которые составляют основную часть большинства БД, СУБД (а, значит, приложения) быстрее начинают свою часть процесса перезапуска.
- **Точное восстановление.** Поскольку СУБД способна точно определить, какие именно данные требуют восстановления, возможность восстановления ненужных данных (например, областей, отмеченных в журнале регистрации как «грязные», а на самом деле являющихся «чистыми») практически исключена.

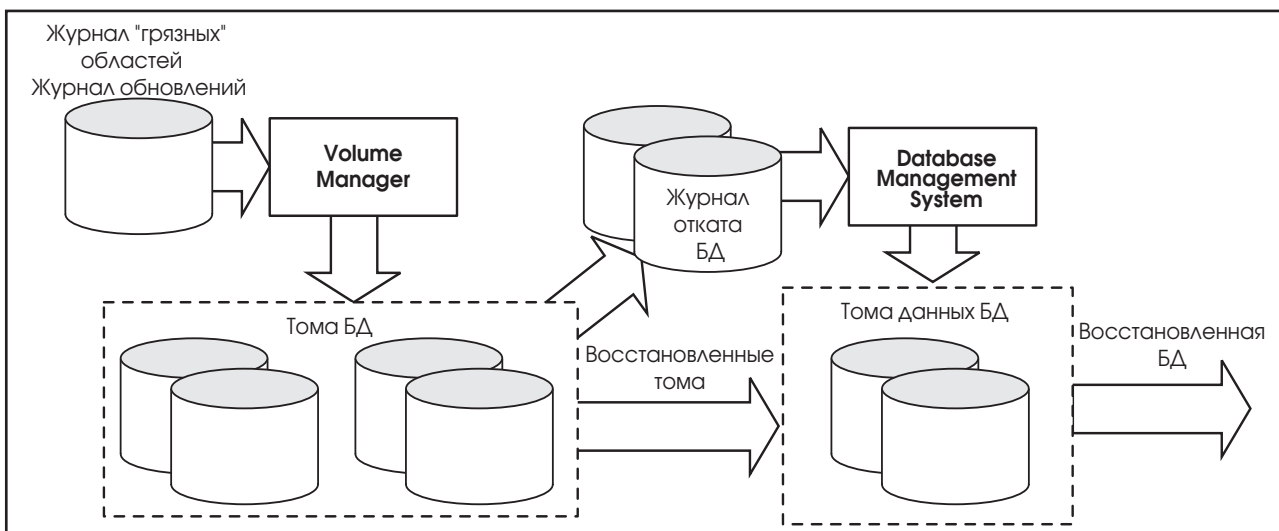


Рис. 6. Восстановление БД после сбоя в системе.

Использование технологии томов для обеспечения высокой производительности

Помимо того, что тома, управляемые VERITAS Volume Manager, обеспечивают высокую доступность данных в режиме онлайн, они, как правило, превосходят неструктурированные диски по производительности. При проектировании БД полезно знать характеристики производительности томов при принятии решений по размещению данных. Такие характеристики представлены в таблице 2.

Как видно из таблицы 2, зеркалированные тома обычно превосходят отдельные диски по производительности при чтении данных. Это объясняется тем, что Volume Manager может свести к минимуму задержку, связанную с вводом—выводом, выбрав наименее загруженный диск зеркалированного тома для выполнения каждого запроса на чтение.

Ожидаемая производительность операции записи для зеркалированных томов несколько ниже (но сравнима), чем для отдельного диска. Volume Manager зеркалирует каждый запрос приложения на запись на каждом из дисков тома, однако, операции записи независимы друг от друга (в отличие от операций записи RAID) и поэтому могут выполняться параллельно. Кроме того, регистрация «грязных» областей в журнале не требует больших затрат ресурсов при записи ранее не измененной области данных.

Операции записи RAID 5, как правило, имеют низкую производительность по сравнению с отдельными дисками, поскольку для каждой записи данных приложения Volume Manager должен выполнить ряд операций чтения и записи на дисках тома RAID, а также внести запись в журнал обновлений. По этой причине тома RAID 5 не рекомендуется использовать для часто обновляемых таблиц БД.

Том RAID	Производительность	Приложение, основанное на БД
Зеркалированный (Mirrored) (2 или более) (иногда называется RAID 1)	Высокая для чтения средняя для записи	Критически важные объекты данных размера не более одного диска (например, БД людских ресурсов)
Рассеянный по дискам и Зеркалированный (Striped and Mirrored) (две и более копии)	Очень высокая для чтения высокая для записи	Крупные критически важные объекты данных на нескольких дисках (например, данные о продажах)
Рассеянный по дискам с контролем четности (Striped with Parity) ("RAID 5")	Высокая для чтения Низкая для записи	В основном данные для чтения (например, Web-страницы, хранилища данных)
Рассеянный по дискам без контроля четности (Striped without Parity) (иногда называемый RAID 0)	Высокая для чтения и записи	Легко заменяемые некритичные данные (наименее используемые в бизнесе)
Составной (с различными возможностями)	Средняя для чтения и записи	Легко заменяемые некритичные данные (наименее используемые в бизнесе)

Таблица 2. Применения RAID: производительность.

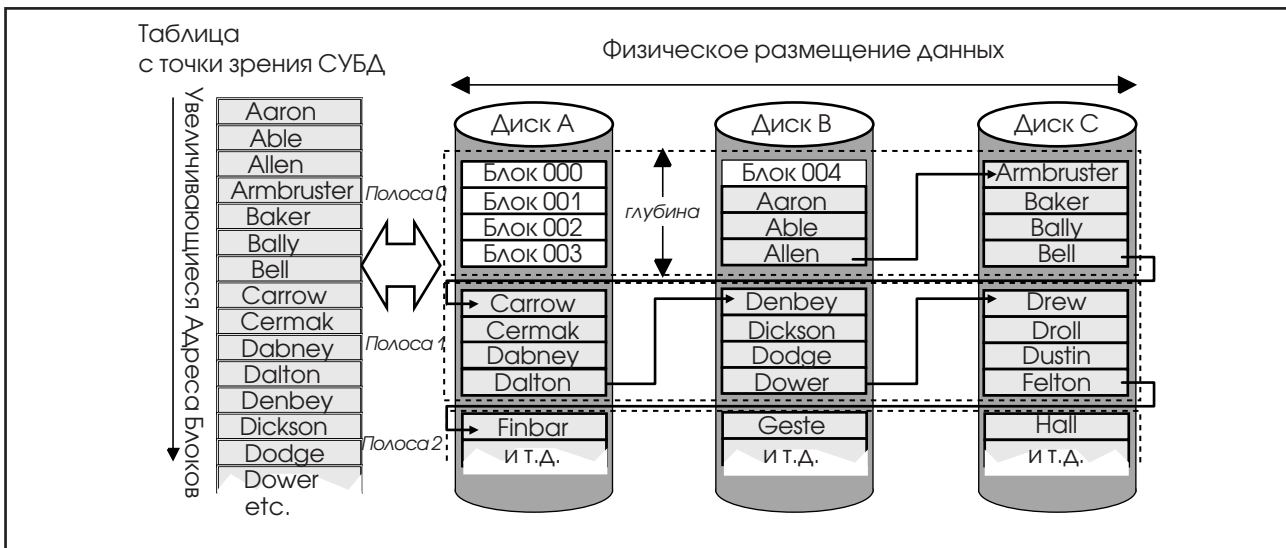


Рис. 7. Влияние рассеивания данных (striping) на структуру таблицы БД.

Рассеянные по дискам тома

Как зеркалированные тома, так и тома RAID используют **рассеивание данных** для прозрачного распределения файлов и табличных областей по нескольким дискам. На рис. 7 показано, какое влияние оказывает рассеивание данных на структуру таблицы БД.

Как видно из левой части рис. 7, СУБД рассматривает таблицу как совокупность записей, хранимых в последовательно пронумерованных блоках (для упрощения рисунка делается предположение, что каждая запись БД занимает один блок на томе). Однако, Volume Manager устанавливает соответствие между блоками тома и дисков в виде логических горизонтальных «полос» (**stripes**). На рис. 7 блоки 0—3 тома соответствуют блокам 0—3 диска А, блоки 4—7 тома соответствуют блокам 0—3 диска В и так далее.

В организации данных Volume Manager адресное пространство блоков каждого диска иногда называется **столбцом** по причинам, понятным из рис. 7. Количество последовательно пронумерованных блоков тома, соответствующих последовательным блокам на отдельном диске, называется **глубиной полосы** (четыре блока на рис. 7). Совокупность таких блоков на всех дисках тома называется **глубиной полосы** (например, полоса 0, полоса 1 и полоса 2 на рис. 7).

Как показано на рис. 7, рассеивание данных позволяет распределить по дискам тома упорядоченную последовательность элементов данных, таких, как таблица БД, записи которой сортируются по основному ключу. Это, как правило, повышает производительность операций ввода—вывода для приложений, интенсивно работающих с вводом—выводом:

- Если запросы приложения на ввод—вывод равномерно распределены по блокам тома, то запросы на ввод—вывод на диск, как правило, будут распределены равномерно по дискам тома. Это

позволяет довести до максимума количество одновременно используемых дисков.

- Даже если запросов приложения на ввод—вывод будет слишком много, например, в результате последовательности транзакций делаются записи, соответствующие именам, начинающимся с буквы D, как на рис. 7, то рассеивание данных все равно позволяет распределить запросы на ввод—вывод по дисковым ресурсам.

Так как карта соответствий между блоками тома и дисков при рассеивании данных не зависит от того, как СУБД представляет себе таблицу, нагрузка операций ввода вывода, как правило, распределена для любого характера запросов на управление БД.

Рассеивание данных используется как с зеркалированными томами, так и с томами RAID 5. Производительность операций чтения будет высокой в обоих случаях. При записи на зеркалированные тома рассеивание данных позволяет сократить время записи посредством распределения нагрузки операций ввода-вывода по дискам. Для томов RAID 5 временные затраты на операции ввода-вывода, вызванные каждой операцией записи БД, снижают производительность даже в случае использования рассеивания данных по дискам. По этой причине компания VERITAS рекомендует использовать тома RAID 5 только для таблиц БД, которые в основном предназначены для чтения.

Файловая система для БД электронного бизнеса

Томы, созданные с помощью VERITAS Volume Manager, — это надежный и высокопроизводительный фундамент для СУБД. Использование файловой системы VERITAS для хранения таблиц и индексов БД позволяет повысить производительность, гибкость и управляемость.

Как было отмечено во Введении, основными ограничениями файлов с точки зрения их использования в качестве контейнеров для БД являются:

- **Слабые гарантии сохранности или низкая производительность.** Большинство современных файловых систем подают сигнал приложениям о завершении обработки запросов на запись до того, как данные реально записываются на диск. Если система выйдет из строя до того, как такие данные будут записаны, то они могут «исчезнуть». Это усложняет задачу обеспечения сохранности, а, следовательно, целостности данных СУБД, использующих файлы для хранения БД. В качестве альтернативы СУБД может заставить файловую систему сбросить содержимое своего кэша на диск после всех критичных операций записи. Это приведет к последовательному выполнению операций ввода—вывода, что понизит производительность многопоточных операций.
- **Нежелательные непроизводительные затраты.** Большинство файловых систем перемещают данные в буферный кэш ядра, прежде чем записать их на диск. Это обеспечивает защиту от ошибок программирования, существующих в приложениях, но приводит к большим затратам на обработку, что нежелательно для БД.
- **Излишне ограниченная семантика совместного использования.** Большинство файловых систем требуют, чтобы приложение, собирающееся записать файл, установило блокировку ядра для каждой операции записи. Для СУБД это излишняя функция, уменьшающая возможность параллельного выполнения операций.

Файловая система компании VERITAS не имеет указанных ограничений. Кроме того, она обладает дополнительными преимуществами для файлов, ис-

пользуемых в качестве контейнеров для таблиц БД. Многие из преимуществ, описываемых в последующих разделах, обусловлены архитектурой файловой системы, построенной на основе экстенгов.

Традиционная и основанная на экстентах файловые системы

Как в традиционной, так и в построенной на экстентах файловой системе индексные дескрипторы файлов (inodes) (структуры данных на диске, описывающие файлы) содержат **дескрипторы файловых блоков**, которые указывают, какие файловые блоки (последовательности дисковых блоков) на диске или томе заняты областью данных файла. Файловая система на основе блоков показана на рис. 8.

Как видно из рис. 8, каждый дескриптор файлового блока содержит указатель (номер дискового блока) на фиксированное число блоков. В большинстве файловых систем ОС UNIX фиксированное число блоков заключено между 4 и 16. Когда приложение делает запрос на ввод—вывод файла, файловая система использует дескрипторы файловых блоков для преобразования адреса данных, указанного в запросе приложения, в адрес блока, которому адресован запрос на ввод—вывод.

Поскольку индексные дескрипторы файлов имеют ограниченный объем пространства для дескрипторов, файлы, требующие большое количество дескрипторов, должны использовать дополнительные индексные дескрипторы для отображения своих блоков, как показано на рис. 8. Чтение и запись таких файлов требуют дополнительных операций ввода—вывода файловой системы для получения таких вспомогательных индексных дескрипторов, прежде чем будет получен доступ к данным.

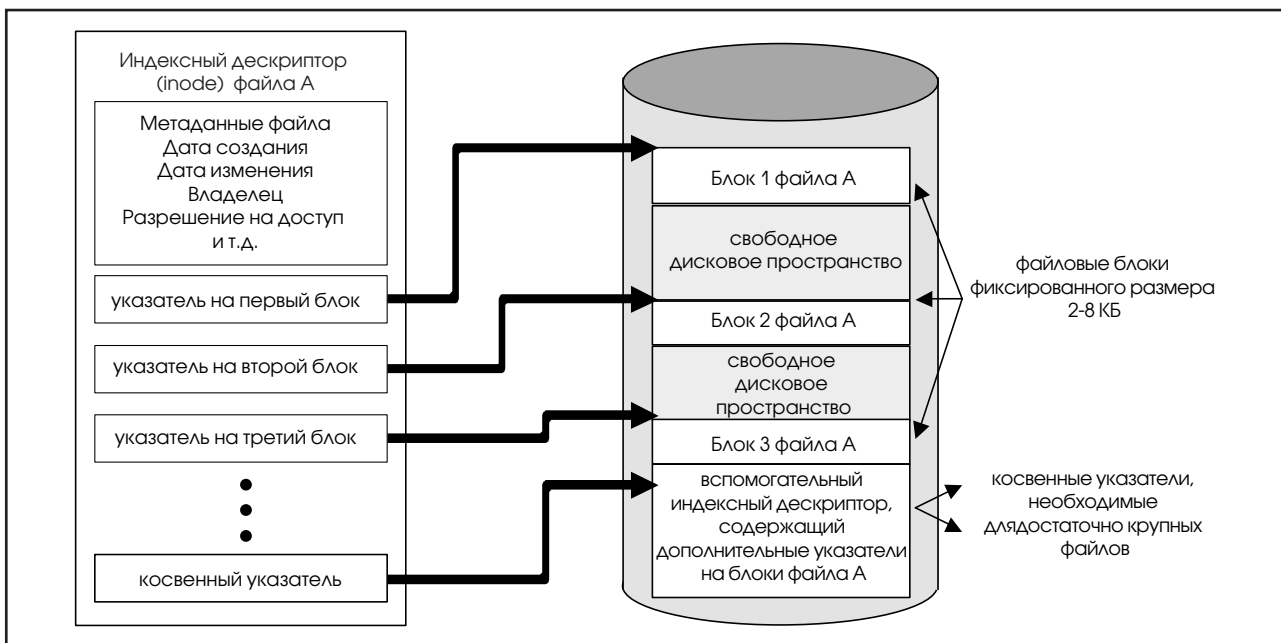


Рис. 8. Файловая система на основе блоков.

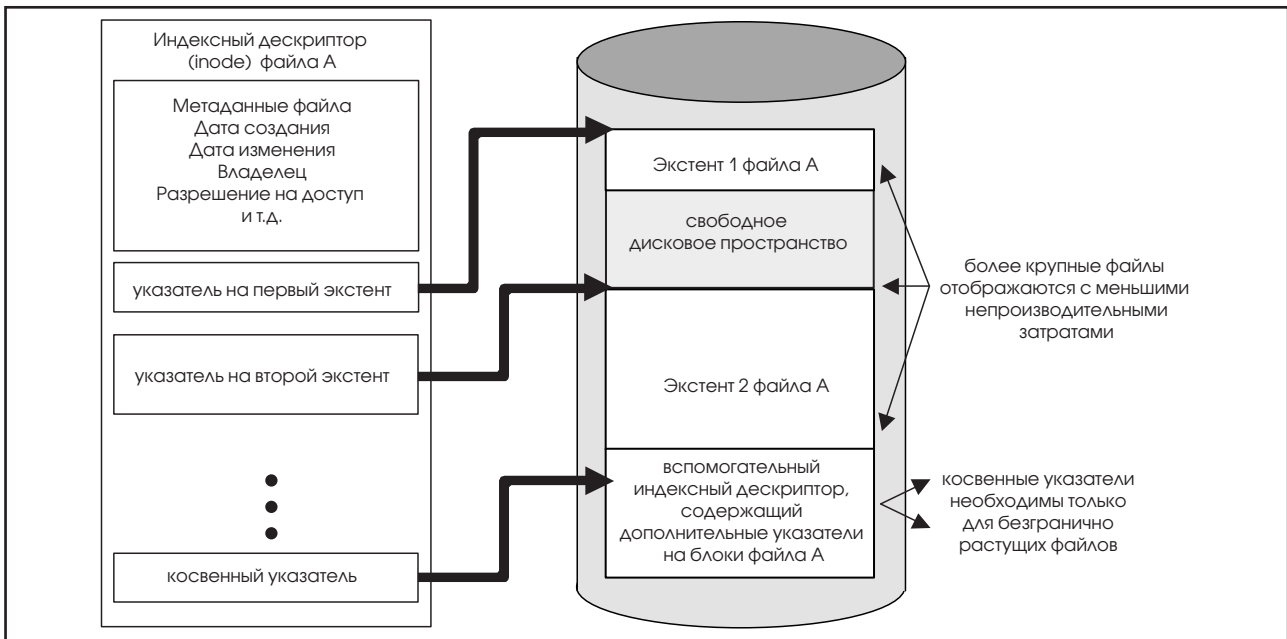


Рис. 9. Файловая система на основе экстенгов.

Файловая система VERITAS использует архитектуру на основе экстенгов для устранения этого ограничения. Она выделяет дисковую память в виде экстенгов переменного размера, а не в виде блоков фиксированного размера, используемых традиционными файловыми системами ОС UNIX. Каждый экстенг в файловой системе VERITAS представляет собой набор последовательно пронумерованных блоков диска или тома произвольного размера, поэтому его можно описать начальным номером блока и количеством блоков. Поскольку каждый экстенг может быть каким угодно большим, один индексный дескриптор файлов часто содержит полную карту распределения файловых блоков даже для очень больших файлов. Это позволяет уменьшить непроизводительные затраты на определение места нахождения блоков файла, что приводит к повышению производительности приложений.

Для СУБД главное преимущество файловой системы на основе экстенгов состоит в том, что она позволяет осуществлять доступ к крупным непрерывным областям дисковой памяти с минимальными затратами. Большие непрерывные области дисковой памяти выгодны по следующим причинам:

- **Более компактные структуры дескрипторов дисковой памяти.** Поскольку каждый дескриптор экстенга может ссылаться на большее количество блоков, для карты распределения файла требуется меньше дескрипторов. Преобразование запросов приложений в запросы на ввод—вывод к диску или тому осуществляется быстрее.
- **Более эффективный просмотр таблицы.** Более крупные непрерывные области дисковой памяти означают, что запросы на ввод—вывод большего количества байт можно использовать для последовательного просмотра таблицы. Это уменьшает

как количество запросов на ввод—вывод, требуемых для просмотра таблицы, так и задержку при вращении диска (время, теряемое на ожидание момента, когда при вращении диска головка чтения—записи окажется над точкой начала передачи данных).

- **Уменьшение фрагментации диска.** Более крупные непрерывные области дисковой памяти позволяют уменьшить фрагментацию диска, при которой логически непрерывные данные хранятся в файловых блоках, не являющихся смежными. Фрагментация диска понижает эффективность операций ввода—вывода, поскольку она требует разбиения некоторых запросов приложения на множество запросов к диску для считывания блоков данных, не являющихся смежными.
- **Более агрессивная политика последовательного ввода—вывода.** Данные, размещенные в смежных областях дисковой памяти, можно считывать с упреждением запросов приложений во время последовательного просмотра таблицы. Это уменьшает время ожидания доступа и ускоряет просмотр таблицы.
- **Более крупные файлы.** Выделение дисковой памяти на основе экстенгов делает практичным использование крупных по размеру файлов. Файловая система VERITAS способна работать с отдельными файлами размером до двух терабайт в средах операционных систем, которые поддерживают файлы такого размера.

Предварительное выделение и выравнивание дисковой памяти

Во многих приложениях создание, расширение и удаление файлов происходит достаточно часто. Однако, БД имеют сравнительно статичные табличные

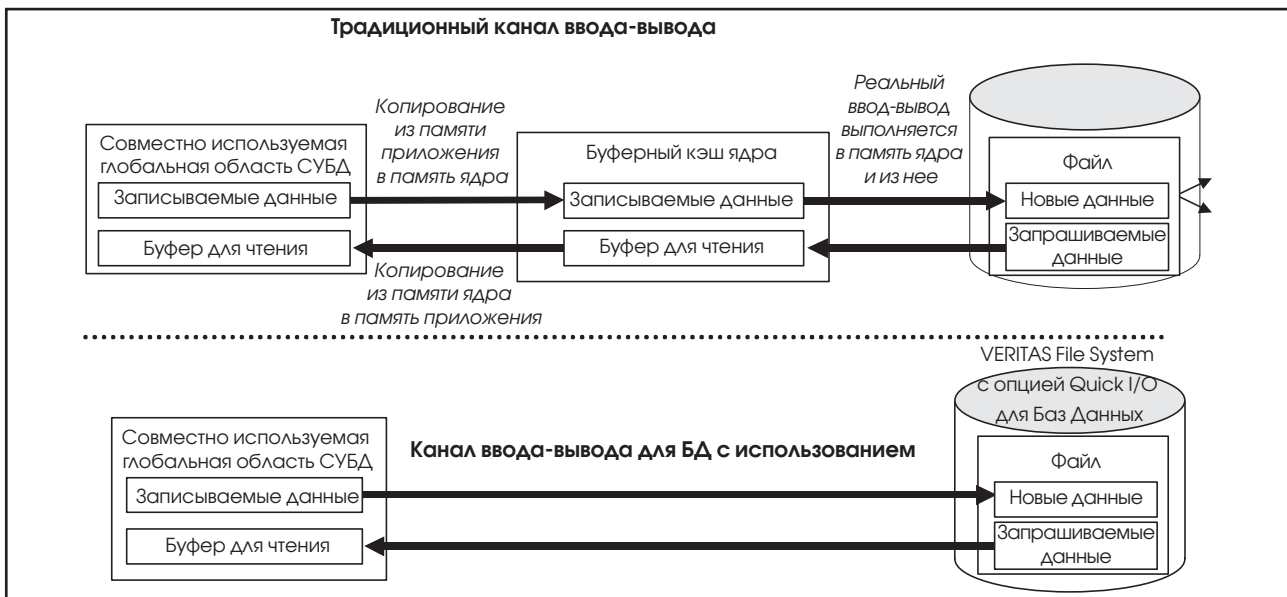


Рис. 10. Опция файловой системы компании VERITAS Quick I/O for Databases поддерживает прямой ввод-вывод данных из приложений.

структуры. Файлы—контейнеры БД создаются, расширяются и удаляются довольно редко, хотя записи могут добавляться к ним или удаляться из них часто.

Файловая система VERITAS позволяет резервировать дисковую память при создании файлов (до записи данных). Такой подход имеет два преимущества для БД:

- **Оптимальная схема размещения данных.** Администратор БД, проектирующий новую БД, может зарезервировать дисковое пространство с оптимальной схемой размещения файлов—контейнеров при известных ему требованиях к системе.
- **Минимальные сбои, связанные с выделением дискового пространства для БД.** В случае предварительно выделенных файлов—контейнеров БД, транзакции и пакетные задания, создающие временные файлы, не могут вызвать сбои, связанные с выделением дискового пространства для БД.

Пакет программ **VERITAS Database Edition for Oracle** позволяет выравнивать предварительно выделенные файлы—контейнеры БД по границам экстенгов файловой системы. Это сводит к минимуму возможность появления разделенных страниц, выходящих за пределы экстенгов файловой системы, а также позволяет СУБД группировать смежные запросы на ввод—вывод для достижения максимальной эффективности.

Сочетание архитектуры на основе экстенгов с предварительным выделением дисковой памяти лежит в основе еще одной технологии компании VERITAS, которая увеличивает производительность файловой системы до уровня производительности неструктурированных дисковых разделов — **Quick I/O for Databases**.

Quick I/O for Databases: то же быстродействие при использовании файлов в качестве контейнеров таблиц БД, что и при применении для этих целей неструктурированных устройств

Опция **Quick I/O for Databases** используется в сочетании с файловой системой VERITAS для достижения такой же производительности операций ввода-вывода с файлами-контейнерами, как при работе с неформатированными устройствами хранения, позволяя СУБД Oracle читать и записывать предварительно выделенные файлы так, как будто они являются неструктурированными устройствами ОС UNIX. Символическая ссылка ОС UNIX на каждый файл-контейнер БД предоставляет Oracle доступ к неструктурированным устройствам. Обычные пути доступа к файловой системе также доступны для служебных операций. Четыре фактора повышают производительность **Quick I/O for Databases** до уровня производительности операций ввода-вывода с неструктурированными дисками:

- **Прямой ввод—вывод.** С помощью **Quick I/O for Databases** данные перемещаются непосредственно между **совместно используемой глобальной областью (SGA) СУБД Oracle** и диском без создания промежуточной копии в пространстве ядра. Это исключает существенные непроизводительные затраты на обработку.
- **Усовершенствованное кэширование.** Поскольку SGA СУБД Oracle сама является кэшем, то отпадает необходимость кэширования данных БД в кэше операционной системы. **Quick I/O for Databases** производит чтение и запись данных непосредственно из SGA, устраняя такую избыточность.

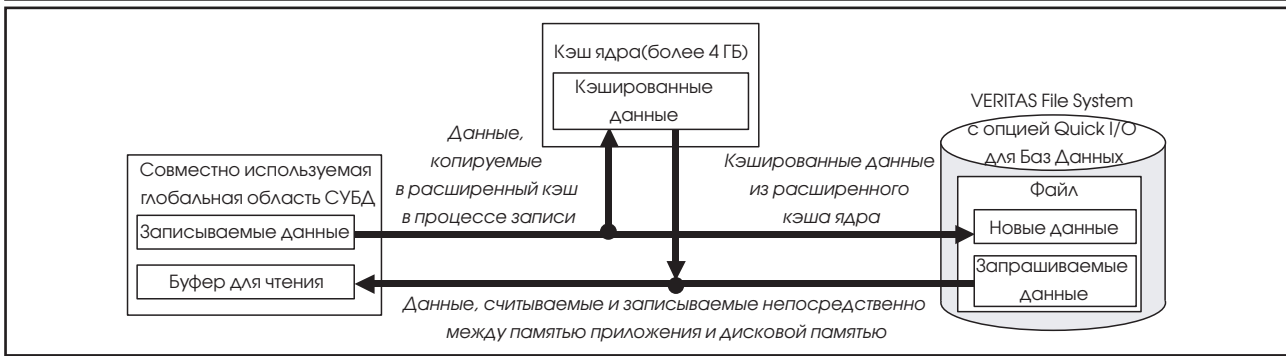


Рис. 11. Работа утилиты Cached Quick I/O для Баз Данных.

- **Большее количество параллельных операций ввода—вывода.** Используя операции ввода—вывода без буферизации, предоставляемые опцией **Quick I/O for Databases**, СУБД Oracle способна делать множество одновременных запросов. Это позволяет параллельно выполнять больше транзакций, а также предоставляет более широкие списки запросов на ввод—вывод драйверам, контроллерам RAID и дискам для использования в алгоритмах оптимизации производительности.
- **Минимальная блокировка файлов ядром.** Операции ввода—вывода без буферизации, предоставляемые опцией **Quick I/O for Databases**, дают возможность обойти блокировку файлов ядром. Поскольку СУБД осуществляют свою собственную внутреннюю блокировку, блокировка файлов ядром является не только излишней, но и приводит к последовательному выполнению операций БД. Устранение блокировки ядром позволяет добиться параллельного выполнения максимального количества операций СУБД Oracle без нарушения целостности данных.

Использование СУБД функций Quick I/O for Databases для прямого ввода—вывода не ограничивает функциональные возможности файловой системы. Доступ к файлам, используемым Quick I/O for Databases, можно также получить по обычным путям файловой системы, например, для резервного копирования, тиражирования, перемещения и других целей.

Cached Quick I/O for Databases: преодоление четырехгигабайтного барьера

32-разрядное приложение или СУБД способно адресовать не более четырех гигабайт памяти. Крупные БД или приложения БД с повышенными требованиями к производительности могли бы легко использовать память большего объема для кэширования данных, если бы они только умели ее адресовать. Опция **Cached Quick I/O for Databases** в сочетании с файловой системой компании VERITAS позволяет 32-разрядным СУБД использовать большой физический кэш и увеличить гибкость распределения кэша среди нескольких копий БД.

При использовании **Cached Quick I/O for Databases** данные записываются непосредственно из кэша СУБД на диск, как и в случае с **Quick I/O for Databases**. Однако, в процессе записи данные также копируются в кэш операционной системы. Последующие операции чтения можно выполнить из этого кэша со скоростью, намного превышающей скорость доступа к диску. Аналогично, данные, считанные с диска, копируются в системный буферный кэш по мере их поступления, так что если они будут повторно запрошены, то их можно переслать незамедлительно.

Поскольку необходимо копировать данные между памятью БД и системным буферным кэшем, **Cached Quick I/O for Databases** является решением с более высокими непроизводительными затратами по сравнению с 64-разрядной СУБД с прямым доступом к очень большому объему памяти. Однако, до тех пор, пока приложения не будут преобразованы в 64-разрядную СУБД, **Cached Quick I/O for Databases** позволяет проводить масштабирование СУБД посредством использования физической памяти очень большого объема для кэша.

Для систем, в которых параллельно работают несколько копий БД, **Cached Quick I/O for Databases** обладает еще одним важным преимуществом. Системная память, выделенная для копии БД (например, SGA СУБД Oracle), доступна только для этой копии. Одна копия БД может практически простаивать, но ее кэш недоступен для других копий, даже если они используются гораздо интенсивнее. С помощью **Cached Quick I/O for Databases** каждой копии БД можно выделить отдельный кэш меньшего объема, поскольку системный кэш, доступ к которому осуществляется посредством **Cached Quick I/O for Databases**, доступен для всех копий в случае необходимости.

Быстрое восстановление данных

Даже в наиболее подготовленных центрах обработки данных дела могут пойти неважно. Приложения, аппаратные компоненты и ошибки в эксплуатации могут привести к отказу системы, поэтому необходимо иметь план быстрого восстановления системы после любых возможных отказов. Максимально быстрый перезапуск приложений после устранения сбоя - это важная составная часть процесса восстановления данных.

Специальные методы восстановления БД Oracle были рассмотрены выше. Однако не менее важной задачей является восстановление файлов, не входящих в состав БД, — десятки и сотни файлов конфигурации и регистрации, а также других вспомогательных файлов, необходимых для работы приложений.

Когда система выходит из строя, целостность структур ее файловой системы ставится под сомнение. Возможно, что приложения создали, расширили или удалили файлы, но не исключено, что постоянные (на диске) метаданные файловой системы не были полностью обновлены. Отказ в неподходящий момент может привести к потере файлов или пространства дисковой памяти, или, что еще хуже, к выделению дисковых блоков двум и более файлам.

Обычным средством проверки целостности структуры файловой системы является запуск программы, проверяющей ее метаданные и осуществляющей их восстановление в случае необходимости, прежде чем монтировать и использовать файловую систему. В UNIX-системах такая программа называется **fsck** (проверка файловой системы). ОС Windows имеют аналогичное средство — программу **CHKDSK**.

Программы **fsck** и **CHKDSK** проводят проверку правильной структуры файловой системы, ищут «потерянные» или выделенные нескольким файлам дисковые блоки. Они могут отменить незавершенные обновления метаданных, что приведет к потере недавно внесенных изменений, но оставят нетронутой структуру соответствующих файловых систем. Проблема, связанная с использованием **fsck** для восстановления данных после сбоя, состоит в том, что в случае больших файловых систем она может потребовать много времени для своего выполнения (по одной из оценок, от 2 до 5 минут на гигабайт данных). Поскольку диски нельзя монтировать до завершения выполнения **fsck**, ни одно приложение, включая СУБД, использующие файлы-контейнеры для хранения данных, не смогут начать выполнение. Время восстановления может оказаться очень большим, что уменьшает доступность системы.

Файловая система VERITAS включает в свой состав альтернативный метод восстановления, основанный на ведении **журнала намерений (invent log)**, аналогичного журналу регистрации Volume Manager. Всякий раз, когда планируется изменение метаданных файловой системы VERITAS (например, при создании, расширении или удалении файла или каталога), все обновления метаданных, необходимые для внесения изменений, регистрируются в собственном журнале намерений файловой системы до их выполнения. Журнал намерений файловой системы представляет собой небольшой циклический буфер, предусматривающий самоочистку (т.е. более свежие обновления записываются поверх старых).

При восстановлении данных после системного сбоя файловая система читает **журнал намерений** и проверяет, чтобы все обновления метаданных в нем

были полностью внесены в метаданные файловой системы. Поскольку количество незавершенных обновлений метаданных в любой момент времени, как правило, составляет весьма малую долю общего объема метаданных в большой файловой системе, обработка **журнала намерений** обычно осуществляется гораздо быстрее (на несколько порядков величины), чем выполнение программы **fsck** для анализа всех метаданных файловой системы. Использование **журнала намерений** позволяет быстрее монтировать файловые системы после отказа системы, что в свою очередь позволяет быстрее начать восстановление БД, в результате чего общая доступность приложений повышается.

Администрирование в режиме онлайн

Можно смело утверждать, что информационная система будет со временем расти и изменяться. Рост и изменение неизбежно потребуют реорганизации данных. Например:

- БД превышают по объему свои устройства хранения, так что последние придется заменить, а БД должны быть перемещены на новые устройства хранения.
- Определенные таблицы БД следует переместить на разные устройства хранения для распределения нагрузки операций ввода—вывода.
- БД необходимо переместить с одного компьютера или географического места на другой по организационным или другим причинам и т.д.

Значительная часть реорганизации данных предполагает перемещение крупных объектов данных с одного физического или виртуального устройства хранения на другое. К сожалению, необходимость перемещения данных вступает в противоречие с основным требованием обеспечения максимальной доступности приложений и данных. Однако, возможность администрирования файловой системы VERITAS в режиме онлайн позволяет выполнять самые важные административные функции в процессе использования файлов. В последующих разделах будет показано, как файловая система VERITAS позволяет выполнять четыре самых важных операции управления файлами — дефрагментацию, перемещение, расширение файловой системы и создание моментальных снимков.

Дефрагментация в режиме онлайн

В процессе выполнения приложений создаются, изменяются и удаляются файлы, при этом, дисковая память постоянно выделяется и освобождается. Со временем невыделенная область дисковой памяти на томе становится фрагментированной. На фрагментированном томе, как показано справа на рис. 12, содержимое файлов и свободное пространство перемешиваются. Свободное пространство, доступное для новых файлов, можно распределить в виде множества небольших областей блоков. Запрос на создание большого непрерывного файла может окончить-

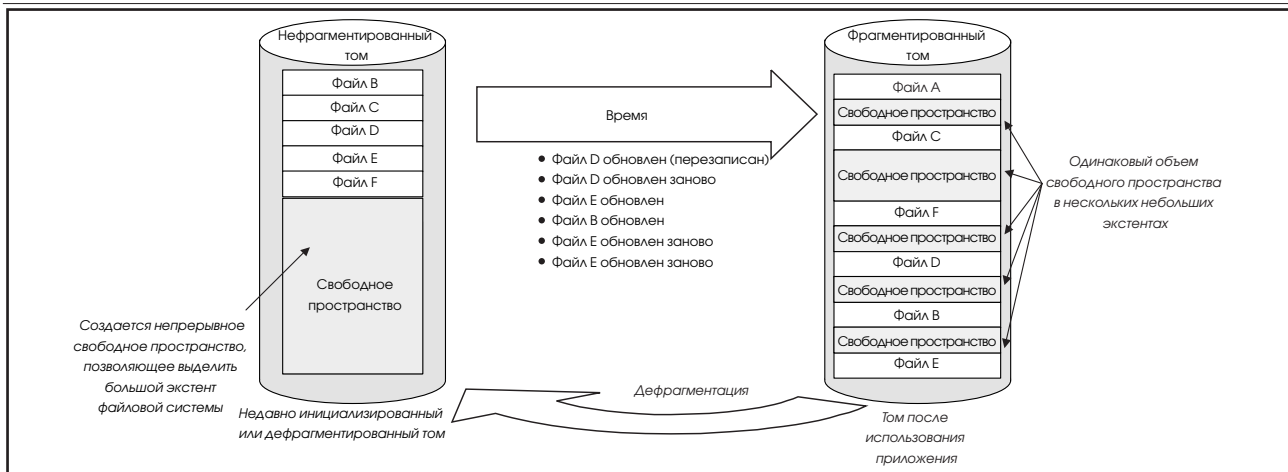


Рис. 12. Фрагментация и дефрагментация томов.

ся неудачей, даже если на томе имеется достаточное пространство. Запросы на выделение файлов, допускающие множество экстентов, не являющихся смежными, могут оказаться успешными в случае выделения большого количества экстентов, однако высокие непроизводительные затраты на обращение к файлу, фрагментированному на множество экстентов, может привести к понижению производительности операций ввода-вывода приложений.

Файловая система VERITAS способна осуществлять дефрагментацию устройств хранения данных посредством перемещения файлов из одной области блоков в другую для создания непрерывной области свободного пространства, а также обновления метаданных для отражения нового местоположения каждого перемещенного файла. Перемещение файлов — это прозрачная для приложений операция, которая осуществляет доступ к файлам одинаковым образом до, во время и после их перемещения. Если все файлы на томе расположить рядом, как это показано слева на рис. 12, то говорят, что том дефрагментирован. В данном состоянии область свободного пространства на томе является непрерывной, и можно выделять большие непрерывные экстенты.

Перемещение файла, который не используется, является простой задачей — по существу, это операция копирования. Перемещение файла, с которым работают приложения, представляет более серьезную проблему. На рис. 13 показано, как файловая система VERITAS перемещает файлы, используемые приложениями.

Для перемещения файла файловая система сначала выделяет пространство, предназначенное для целого файла. Затем осуществляется блокировка областей блоков и их копирование. Метаданные файловой системы обновляются, чтобы отразить новые адреса блоков для каждой области блоков в процессе ее перемещения, а блокировка области блоков снимается для предоставления доступа приложениям. Если приложение пытается получить доступ к блокам в перемещаемой области, выполнение его запроса на

ввод—вывод приостанавливается до тех пор, пока эта область не будет перемещена, а блоки не будут разблокированы. Если файловая система пытается заблокировать область, содержащую блоки, которые уже заблокированы приложением, то выполнение ее операций ввода—вывода приостанавливается до тех пор, пока приложение не снимет блокировку.

Способность перемещать файл в процессе его использования — это один из критически важных элементов дефрагментации файловой системы. Другим критически важным элементом является способность проводить текущий обзор пространства дисковой памяти файловой системы и принимать оптимальные решения по перемещению файлов при дефрагментации.

Перемещение файлов—контейнеров БД

Структура и размер табличных областей, таблиц и индексов, составляющих БД, обычно определяются при реализации БД. Только в редких случаях структуры добавляются к существующей БД. Размеры существующих структур также изменяются сравнительно редко. Поэтому случаи фрагментации файловой системы (как показано на рис. 12), содержащей файлы—контейнеры БД, сравнительно редки. Таким образом, хотя дефрагментация устройства хранения БД иногда необходима, в контексте указанного приложения это не является основным использованием возможности перемещения файлов в режиме онлайн.

Администраторы БД все же находят полезной возможность перемещения табличных областей во время их использования для удовлетворения потребностей в расширении, перемещении или реконфигурировании таблиц и индексов. Если для хранения БД используются файлы—контейнеры файловой системы VERITAS, то возможно перемещение файлов в режиме онлайн для всех указанных целей. Приведем примеры:

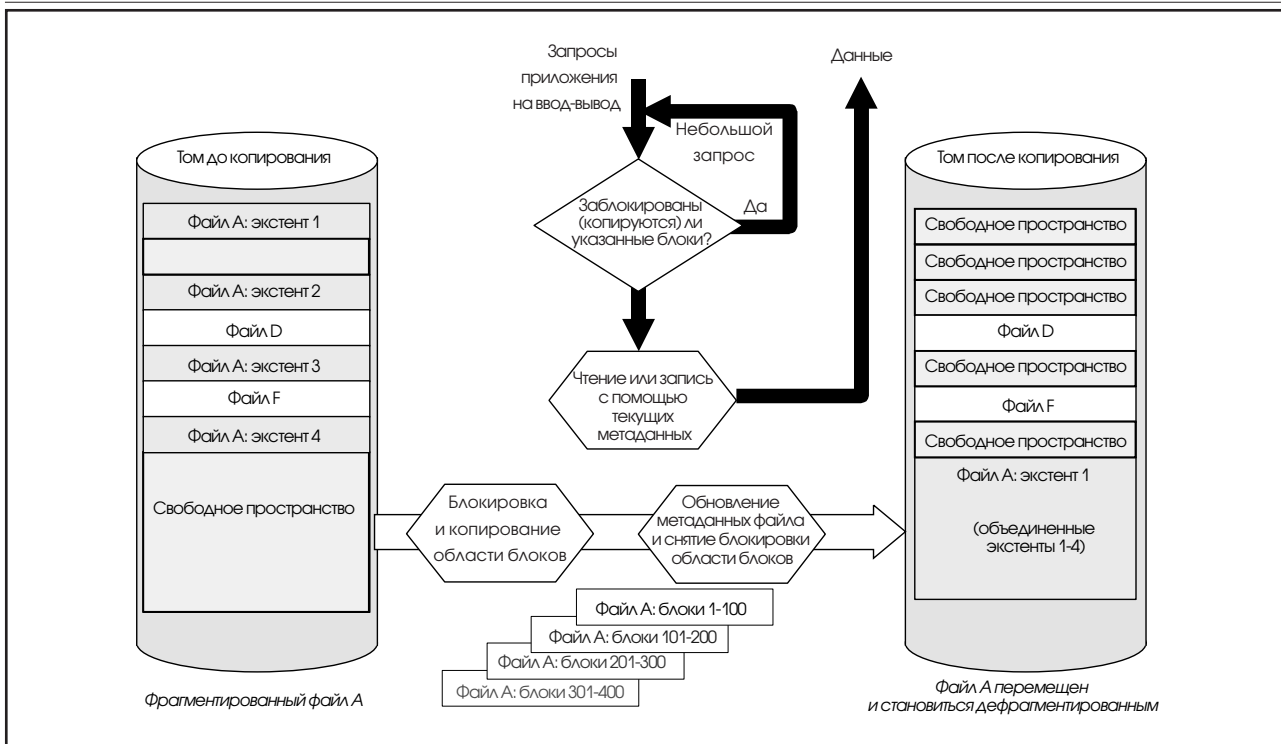


Рис. 13. Перемещение открытого файла.

- Изменения политики хранения данных, например, перемещение таблицы, критичность которой возросла, с простого тома или тома RAID 5 на зеркалированный том, можно реализовать во время использования соответствующих таблиц.
- Таблицы, которые занимают практически все пространство устройств их хранения, можно переместить на более крупные устройства или устройства с большим объемом доступной памяти, предотвращая тем самым отказы выполнения заданий вследствие недостатка свободного пространства.
- Файлы—контейнеры БД можно переместить из отказывающих устройств хранения, выявленных аппаратными или программными методами предупреждающего анализа отказов, на другие устройства.

Расширение в режиме онлайн

Едва ли не наиболее часто встречающейся причиной перемещения файлов во время их использования является расширение онлайн-ой дисковой памяти. Например, в процессе развития электронного бизнеса:

- перечень его продуктов и услуг расширяется,
- он приобретает новых клиентов,
- история его транзакций растет и
- он начинает использовать дополнительные онлайн-ые приложения.

Все эти факторы приводят к увеличению объемов обрабатываемой информации в режиме онлайн, что в свою очередь требует расширения дисковой памяти. Вопрос состоит в том, как реагировать

на, зачастую, непредсказуемые потребности в расширении дисковой памяти, не вызвав чрезмерно длительный простой. Для приложений на основе реляционных БД ответ на этот вопрос обычно сравнительно прост. Например, табличные области БД Oracle могут быть расширены в режиме онлайн либо автоматически, либо администратором БД. Файлы-контейнеры, используемые для хранения баз данных, можно легко расширить в режиме онлайн до предела, установленного файловой системой. Базовые технологии компании VERITAS сделали еще один шаг в этом направлении: они поддерживают расширение томов в режиме онлайн с последующим расширением онлайн-ых файловых систем, чтобы воспользоваться заново выделенным пространством. Поэтому БД, для хранения которых используется файловая система VERITAS, обладают гибкостью расширения в режиме онлайн в сочетании с другими средствами управления файлами.

Моментальные снимки

Требование постоянной доступности данных противоречит общей практике регулярного резервного копирования критически важных данных. Резервная копия БД представляет наибольшую ценность, когда она дает непротиворечивое представление о бизнес-операциях в определенный момент времени. Поэтому резервное копирование следует проводить тогда, когда транзакции БД не выполняются. Резервная копия БД, снятая в процессе обработки транзакций, может содержать неполные результаты их обработки, в результате чего, она, возможно, не представляет непротиворечивый одномоментный образ данных.

Один из способов обеспечения одномоментного резервного копирования состоит в приостановлении использования БД на время его выполнения. Созданные таким образом резервные копии часто называются «холодными». Поскольку БД не используется во время ее «холодного» резервного копирования, это наиболее быстрый метод полного резервного копирования БД. Для проведения «холодного» резервного копирования БД системный администратор должен найти так называемое окно для этой операции, т.е. промежуток времени, в течение которого допускается недоступность БД для приложений. Обычно окна резервного копирования приходятся на ночные часы или выходные дни. Однако, например, в случае электронного бизнеса данные должны быть доступны как в эти промежутки времени, так и в течение всего рабочего дня. Поэтому «холодное» резервное копирование БД не удовлетворяет требованиям доступности данных, предъявляемым электронным бизнесом. Кроме того, по мере роста электронного бизнеса увеличивается объем его онлайн-данных. Время резервного копирования увеличивается пропорционально, но размер его окна остается неизменным — «холодное» резервное копирование становится все менее и менее приемлемым решением.

Технология **моментальных снимков (snapshots)** файловой системы VERITAS позволяет разрешить указанную проблему. Моментальные снимки — это одномоментные виртуальные образы файлов в файловой системе, которые создаются и сопровождаются самой файловой системой. Моментальные

снимки прозрачны для СУБД и приложений, за исключением момента их создания. Файловая система VERITAS, с которой создан снимок, называется **«снятой» (snapped)**. Приложения могут обновлять данные в ней как до, так и после создания моментального снимка. Образ моментального снимка представляется приложениям в виде файловой системы, доступной только для чтения.

Создание моментального снимка занимает несколько секунд. Файловая система использует невыделенное пространство для создания карты модифицированных блоков, в которой записываются изменения, вносимые в «снятую» файловую систему. Она также выделяет пространство для области измененных блоков, в которой записывается предыдущее содержание любого блока, измененного после инициирования процесса создания моментального снимка. После создания этих структур как «снятая» файловая система, так и файловая система моментального снимка становятся доступными для приложений.

Как работают моментальные снимки?

Файловая система VERITAS использует метод **копирования при записи** для сопровождения моментальных снимков во время внесения изменений в «снятую» файловую систему. При первой записи данных приложения в блок после создания снимка файловая система копирует его содержимое в область измененных блоков моментального снимка. За-

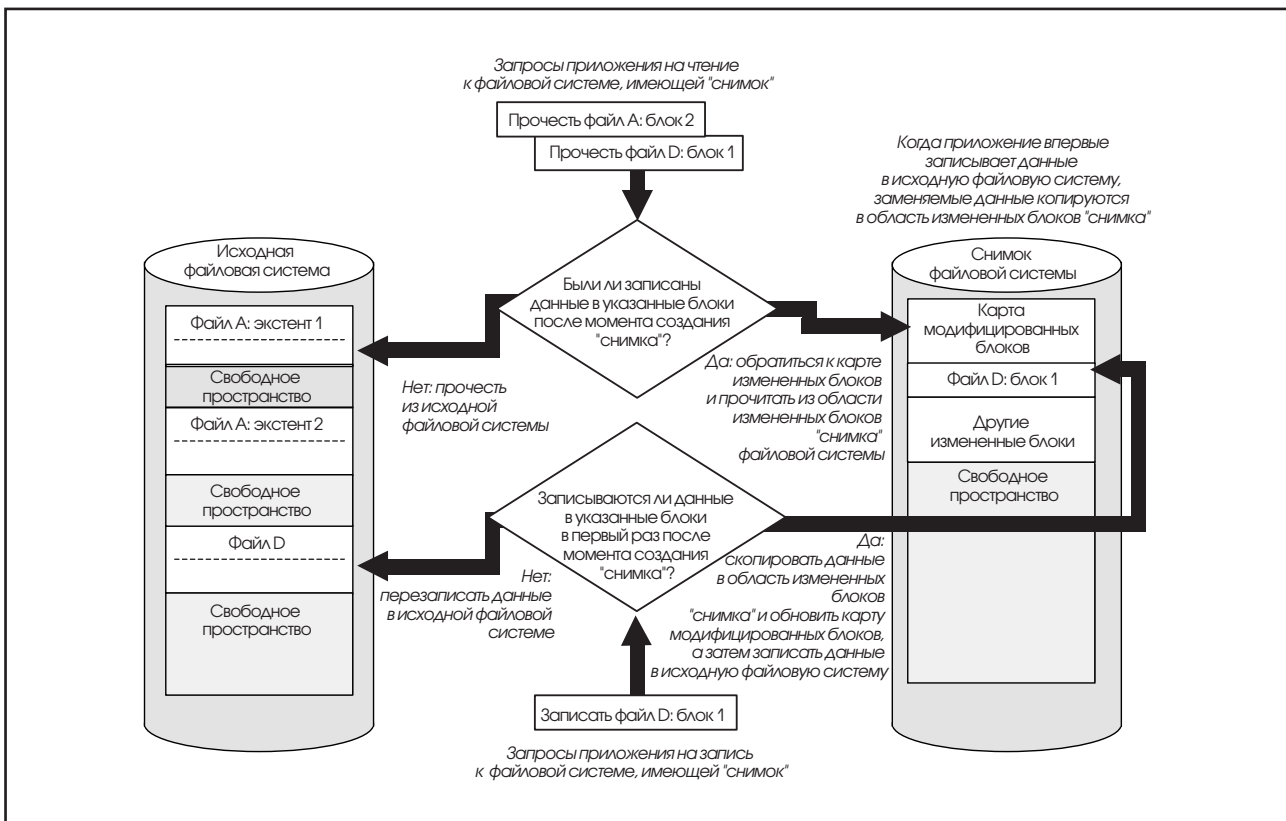


Рис. 14. Принцип работы моментального снимка (snapshot) файловой системы.

тем она обновляет карту модифицированных блоков снимка, чтобы указать, что предварительно измененное содержимое блока хранится в области моментального снимка. В заключение файловая система перезаписывает блок в его исходном положении в «снятой» файловой системе.

За исключением небольшого пространства, занимаемого картой модифицированных блоков, объем дисковой памяти, занимаемой снимком, пропорционален объему данных «снятой» файловой системы, которые обновляются во время активного состояния моментального снимка.

Когда приложение, такое, как программа резервного копирования, считывает блок из моментального снимка, файловая система сначала проверяет его карту модифицированных блоков, чтобы определить, изменился ли блок после момента создания снимка. Если блок не был изменен, то он считывается из своего исходного положения в «снятой» файловой системе. Если блок был перезаписан в «снятой» файловой системе после момента создания моментального снимка, то карта модифицированных блоков указывает положение его исходного содержимого в области измененных блоков снимка, и его считывание происходит из этой области. На рис. 14 показано, как выполняются операции ввода-вывода в файловой системе моментального снимка.

Непротиворечивость моментальных снимков

Поскольку моментальные снимки требуют, чтобы приложения были приостановлены всего лишь на несколько секунд во время их создания, они могут

оказаться полезными для создания замороженных образов БД или других прикладных данных, резервное копирование которых можно провести при возобновлении выполнения приложения. Однако чтобы моментальный снимок был полезным источником для резервного копирования, он должен представлять не только одномоментный образ файловой системы, но и неактивное состояние БД, в котором:

- не происходят никакие незавершенные транзакции,
- все завершенные транзакции записаны на диск (а не содержатся в кэше с обратной записью).

Говорят, что БД в указанном состоянии находится в **покое (quiescent)**.

Когда администратор создает моментальный снимок, файловая система может приостановить выполнение запросов на ввод—вывод и сбросить (записать) на диск кэшированные пользовательские данные и метаданные, чтобы обеспечить непротиворечивость ее образа на диске. Однако, файловая система не может установить, обрабатывает ли СУБД, использующая ее для хранения данных, какие-либо транзакции. БД может иметь транзакции, которые обрабатываются, в то время как не выполняются никакие запросы на ввод—вывод (например, транзакции, в обработке которых участвуют люди, часто требуют время на обдумывание в процессе их выполнения. Обычно никакие операции ввода—вывода не выполняются в это время, поскольку человек обдумывает следующее действие).

VERITAS Database Edition for Oracle решает эту проблему с помощью встроенного средства администрирования, которое предписывает СУБД закон-

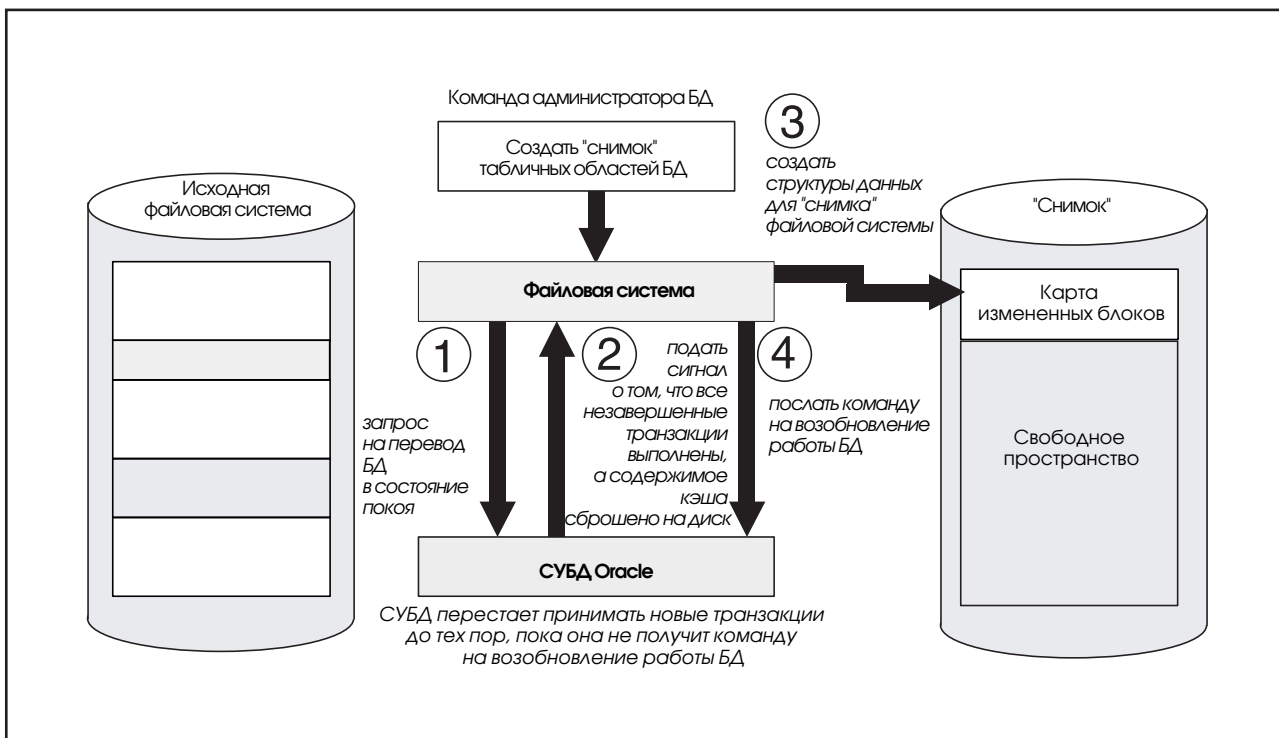


Рис. 15. Создание моментального снимка БД.

чить все незавершенные транзакции, сбросить содержимое ее кэша на диск и приостановить выполнение новых транзакций для того, чтобы можно было создать моментальный снимок непротиворечивой БД. Когда БД находится в покое, файловая система создает моментальный снимок, для чего обычно требуется не более двух секунд. После создания моментального снимка, БД дается указание возобновить выполнение транзакций. Это позволяет свести к минимуму промежутки времени, в течение которого БД недоступна для приложений, при этом гарантируя, что моментальный снимок будет представлять непротиворечивое с точки зрения транзакций состояние БД. На рис. 15 показано, как создается моментальный снимок БД.

После того, как моментальный снимок создан, он ведет себя как файловая система, доступная только для чтения. Программы БД и резервного копирования файлов могут его использовать в качестве источника для проведения структурированного резервного копирования. Данные, прочитанные из моментального снимка, идентичны тем, которые были бы прочитаны из «снятой» файловой системы в момент создания моментального снимка. Таким образом, содержимое резервной копии, снятой с моментального снимка, точно совпадает с содержимым «холодной» резервной копии БД. Различие состоит в том, что БД, основанная на файлах-контейнерах, может использоваться приложениями во время резервного копирования ее моментального снимка (замороженного образа).

Влияние моментальных снимков на производительность

Хотя моментальные снимки позволяют уменьшить время простоя приложений, требуемое для создания резервной копии, с нескольких часов до нескольких секунд, следует рассмотреть их влияние на производительность приложений. Запись данных приложения в «снятой» файловой системе приводит к тому, что активность операций ввода-вывода повышается, по крайней мере, в три раза по сравнению с файловой системой без моментального снимка. Это связано с тем, что записываемые блоки должны быть прочитаны и скопированы в область измененных блоков моментального снимка до того, как обновленные блоки будут записаны в файловую систему. Опыт показывает, что производительность приложений обработки транзакций в режиме онлайн, использующих СУБД Oracle, может понизиться на 15% в процессе использования моментального снимка. Од-

нако, это представляется сравнительно малой платой для возможности создания непротиворечивых с точки зрения транзакций резервных копий при практически постоянной доступности БД.

Заключение

В начале данной статьи были сформулированы требования, предъявляемые бизнес-приложениями к базам данных:

- **Надежность.**
- **Высокая производительность.**
- **Способность масштабируемого роста.**

В данном разделе было показано, как базовые технологии, включающие файловую систему и диспетчер томов компании VERITAS, помогают удовлетворить этим требованиям.

- **Надежность** достигается использованием технологий RAID и зеркалирования, реализуемых в VERITAS Volume Manager, что обеспечивает доступность данных в режиме онлайн в случае отказов дисков и других аппаратных компонент в канале ввода—вывода.
- **Высокая производительность** обусловлена как технологией рассеивания данных по дискам, реализуемой VERITAS Volume Manager, так и средством Quick I/O for Databases, которое устраняет избыточное кэширование, выполняя операции ввода-вывода непосредственно из кэша БД, а также позволяет создать кэш очень большого объема, выходящего за четырехгигабайтный предел 32-разрядных операционных систем.
- **Масштабируемость** проистекает из возможности добавления устройств хранения данных, а также из способности масштабирования томов, файловых систем и самой БД, которая позволяет использовать добавленную дисковую память, причем без остановки приложений.

Базовые технологии компании VERITAS позволяют создать прочный фундамент для управления данными и их хранением, что необходимо для обеспечения постоянной доступности современных бизнес-приложений, основанных на БД.

В следующих разделах данной статьи речь пойдет о других технологиях компании VERITAS, которые строятся на основе базовых технологий, и являются составными частями инфраструктуры «**полного управления хранением данных**».