Руководство по аппаратному обеспечению (hardware) FirebirdSQL

Вопрос «Какое аппаратное обеспечение лучше выбрать для СУБД Firebird» довольно часто возникает у пользователей Firebird. Популярность этой темы неизменна, так как требования к hardware для различных задач отличаются, а с течением времени меняется и аппаратное обеспечение.

Мы решили написать это руководство, чтобы дать необходимые знания каждому, кто желает подобрать действительно эффективное аппаратное обеспечение для своей базы данных Firebird. Для этого потребуется разобраться в основных моментах функционирования Firebird, операционной системы (ОС) и, конечно же, аппаратного обеспечения.

Оглавление

Немного теории

Чтобы понять, какое аппаратное обеспечение лучше всего подойдет для вашей БД Firebird, мы должны понять, как Firebird использует его компоненты: CPU, RAM, HDD/SSD, и как эти компоненты взаимодействуют с ОС (например, с файловым кэшем).

Функциональные модули сервера Firebird

Прежде всего, мы рассмотрим функциональные компоненты Firebird, в чем нам поможет следующий рисунок:

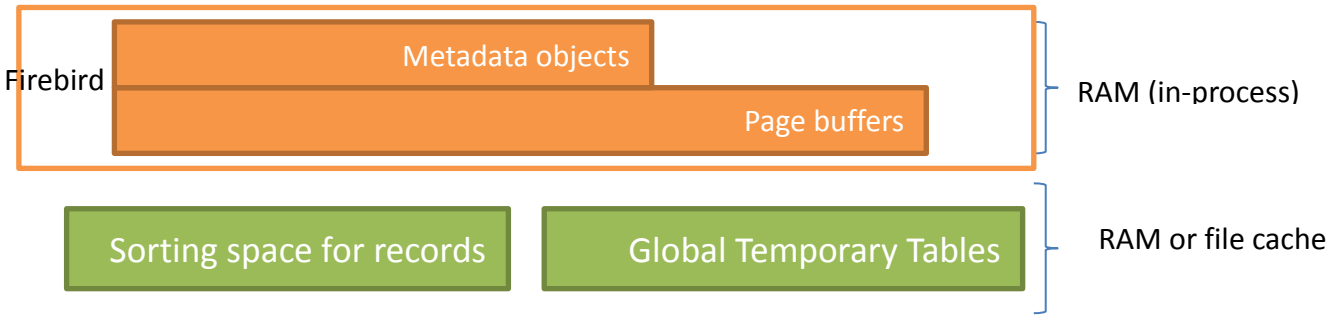


Figure 1. Firebird modules

Firebird включает в себя следующие основные функциональные компоненты:

1) Объекты метаданных - это представления таблиц, индексов, триггеров, хранимых процедур и других объектов БД. Объекты метаданных располагаются в адресном пространстве процесса Firebird (это может быть fbserver, fb_inet_server или firebird.exe).
2) Кэш страничных буферов содержит страницы базы данных, прочитанные с диска, и располагается в адресном пространстве процесса сервера. Механизм кэширования страниц достаточно сложный, поэтому ограничимся утверждением, что Firebird кэширует наиболее часто используемые страницы базы данных.
3) При выполнении сортировок Firebird выполняет ее в памяти (в адресном пространстве процесса сервера), пока размер используемой памяти для всех выполняемых одновременно сортировок не достигнет предела, установленного параметром TempCacheLimit (firebird.conf). При превышении этого лимита создается временный файл (с соответствующим флагом операционной системы) в папке временных файлов, и в нем выполняется сортировка. В случае, если в системе есть свободная память (RAM), то файл сортировки будет кэширован на уровне ОС и сортировка будет производиться в памяти.
4) Глобальные временные таблицы (GTT) создаются как временные файлы ОС. При наличии свободной памяти у ОС операции с GTT выполняются в RAM.

Базовые операции с hardware

Давайте рассмотрим взаимодействие функциональных модулей Firebird с компонентами hardware в контексте операций, которые выполняются при работе с базами данных.

При старте Firebird процесс сервера занимает в RAM минимальный объем (несколько мегабайт), и не производит никаких интенсивных операций с CPU или RAM.

При соединении с БД сервер начинает читать её метаданные и создавать соответствующие объекты в памяти, отчего размер процесса увеличивается пропорционально количеству используемых таблиц, индексов, триггеров и других метаданных. Использование памяти увеличивается, но CPU практически не задействован на этом этапе.

Когда клиент начинает выполнять SQL-запросы (включая хранимые процедуры), сервер выполняет соответствующие операции, обращающиеся к hardware. Среди этих операций мы можем выделить следующие базовые операции, взаимодействующие с компонентами hardware:

- чтение страниц базы данных с диска,

-запись страниц БД на диск,

- чтение страниц базы данных из кэша,

- запись страниц БД в кэш,

- чтение и запись страниц данных в глобальные временные таблицы (Global Temporary Tables),

- Обработка SQL запроса (например, JOINs),

- сортировка записей в результирующем наборе данных (resultset).

Для выполнения каждой из этой операции требуется определенный набор системных ресурсов. В таблице ниже представлено потребление ресурсов, выраженное в условных единицах интенсивности (1 означает небольшую интенсивность, 10 – максимальную):

	Read page from disk	**Write page to disk**	**Read page from page buffers cache**	**Write page to page buffers cache**	**Read from GTT**	**Write to GTT**	**Сортировка записей**	**Обработка SQL**
CPU	1	1	1	1	1	1	5	10
RAM	5	5	5	5	5	5	5	2
Disk IO	10	10	1	1	1	1	1	1

Как видите, что наиболее тяжелыми операциями являются те, которые включают работу с диском, так как дисковая подсистема, несмотря на прогресс последних лет, связанный с SSD, все равно является наиболее медленным компонентом hardware.

> *Отсюда вытекает одно из направлений оптимизации производительности, в полной мере относящееся к hardware – максимально переносить все операции чтения-записи в RAM. Сразу отметим, что подход «давайте сделаем страничный кэш побольше» не работает. Мы подробно рассмотрим этот вопрос в разделе «RAM».*

Параллельность выполнения операций

Обычно требует подобрать hardware для сервера, который будет обслуживать большое количество клиентов, поэтому важно понимать, как реализована параллельность выполнения операций.

С точки зрения компонентов аппаратного обеспечения можно говорить о параллельности использования CPU, диска и RAM. Современные CPU имеют несколько ядер, которые могут параллельно выполнять наборы инструкций, поэтому сервер СУБД распределяет операции между ядрами, то есть, можно сделать простой вывод – чем больше ядер у CPU, тем больше клиентов сможет работать на этом сервере.

С точки зрения дисковой подсистемы все не так однозначно. Традиционные жесткие диски (HDD) при считывании информации физически перемещают головку по магнитным дорожкам с некоторой конечной скоростью. База данных может быть достаточно большой, например, размером в 3 терабайта, и если клиентские SQL запросы будут параллельно обращаться к данным, расположенным в разных областях файла, то головка диска будет метаться между разными областями диска, серьезно затормаживая процессы чтения-вывода. При этом значительно вырастет очередь диска, а остальные ресурсы (CPU, RAM) будут простаивать. Разумеется, кэш дисковой подсистемы (кэш диска или RAID-контроллера) в какой-то мере компенсирует такое замедление, но недостаточно.

SSD диски, в отличие от традиционных HDD, в значительно меньшей мере страдают от деградации производительности при параллельном доступе к данным. Особенно заметно преимущество SSD при параллельной записи данных – наши тесты показывали 7-кратное преимущество SSD над SATA (http://ib-aid.com/en/articles/firebird-performance-degradation-tests-myths-and-truth/). Однако, у SSD есть ряд моментов, которые надо обязательно учитывать при эксплуатации (см. раздел «Выбор дисковой подсистемы»), чтобы избежать падения скорости работы, преждевременной поломки диска и потери данных.

Операции с RAM на современном компьютере выполняются очень быстро, практически ограничены только пропускной способностью шины данных, и поэтому не являются узким местом даже при множестве параллельных SQL-запросов.

Потоки данных

При выполнении SQL-запросов Firebird читает и записывает большое количество данных, перемещает их между функциональными модулями и соответствующими компонентами hardware. Нам необходимо изучить пути, по которым происходит обмен данными, чтобы идентифицировать возможные узкие места, в этом нам поможет рисунок ниже:

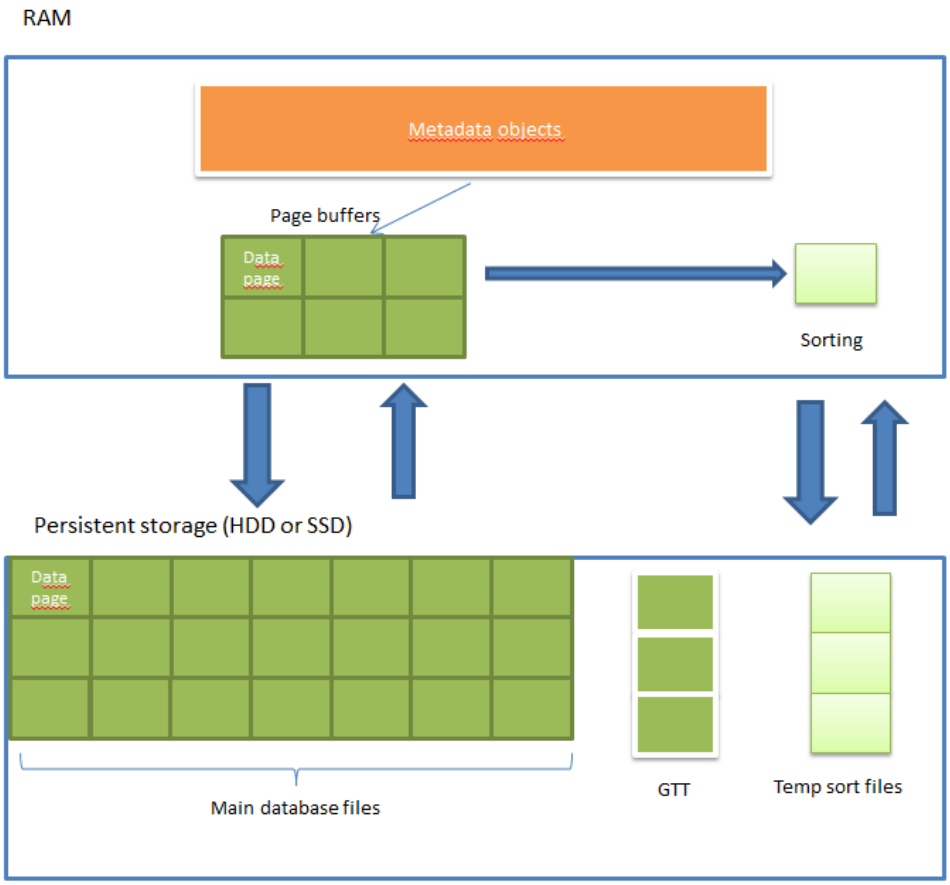


Figure 2. Data flows between RAM and persistent storage

Очевидно, что основные затраты времени возникают при перемещении данных из постоянного хранилища в RAM и назад. При этом возникает два потока данных – чтение/запись страниц данных файла базы данных и чтение/запись файлов сортировок. Так как файлов сортировок может быть несколько и они могут быть достаточно большими, то они могут существенно нагружать дисковую подсистему, поэтому желательно развести эти потоки ввода-вывода по разным дискам.

Резервное копирование

Firebird предлагает 2 способа резервного копирования – верифицированный backup с помощью утилиты gbak, и неверифицированный инкрементальный backup с помощью утилиты nbackup.

Бэкап с помощью gbak является верифицированным, т.к. при его создании происходит чтение каждой записи в каждой таблице БД, и таким образом проверяется целостность информации в БД, а nbackup при создании бэкапа переносит страницы данных целиком, не проверяет целостность данных, и соответственно, является неверифицированным.

> *Мы рекомендуем использовать комбинацию средств резервного копирования – частые запуски nbackup (например, каждый час, день и неделю) и ежедневный ночной верифицируемый backup с помощью gbak.*

При резервном копировании любым способом происходит чтение файла базы данных (всего или части), и запись резервной копии (полной или частичной). Операции записи при создании резервной копии идут последовательно, это означает, что обычные недорогие жесткие диски с

интерфейсом SATA (HDD SATA) хорошо подойдут для хранения резервных копий, так как скорость последовательной записи у них довольно велика.

Выбор подходящего «железа»

Теперь, когда мы представляем, как Firebird взаимодействует с аппаратным обеспечением, необходимо подробно рассмотреть факторы, влияющие на выбор конкретного компонента и его характеристики.

В ряде случаев на выбор компонентов hardware сильно влияют реальные статистические данные конкретной БД, поэтому мы будем использовать инструменты из HQbird (профессионального дистрибутива Firebird для предприятий от IBSurgeon) для получения этих характеристик. Вы можете скачать триальную версию HQbird на сайте http://hqbird.com/en/hqbird/.

CPU

Для выбора CPU необходимо учитывать 3 вещи:

1) Какие запросы преобладают в приложении,
2) Количество активных соединений с БД в среднем и в моменты пиковой нагрузки,
3) Версию и архитектуру Firebird.

Какие запросы преобладают в приложении?

Firebird всегда исполняет один запрос на одном ядре, поэтому сложные или плохо оптимизированные запросы могут занимать до 100% одного ядра, заставляя остальные запросы переместиться на менее загруженные ядра, и чем больше ядер, тем меньше шанс, что все процессорные мощности будут заняты, а пользователи увидят замедление работы приложения.

Если приложение в основном исполняет простые короткие SQL запросы, все запросы хорошо отлажены, и не используется генерация запросов на лету (ad hoc) (например, для отчетов), то CPU не будет являться узким местом производительности, и можно выбрать младшую модель с меньшим количеством ядер.

Если приложение содержит генератор отчетов или большое количество медленных запросов, возвращающих большое количество данных, то необходим процессор с большим количеством ядер.

Количество активных соединений с БД в среднем и в моменты пиковой нагрузки

Количество соединений (активных пользователей) также влияет на выбор CPU. К сожалению, часто даже разработчики приложений не представляют, сколько в точности соединений, запросов и транзакций выполняется в данный конкретный момент. Для уточнения этой информации мы рекомендуем воспользоваться инструментом MON$ Logger из HQbird, и снять несколько снэпшотов в моменты рабочей нагрузки, на которых будет четко видно, сколько в действительности установлено соединений.

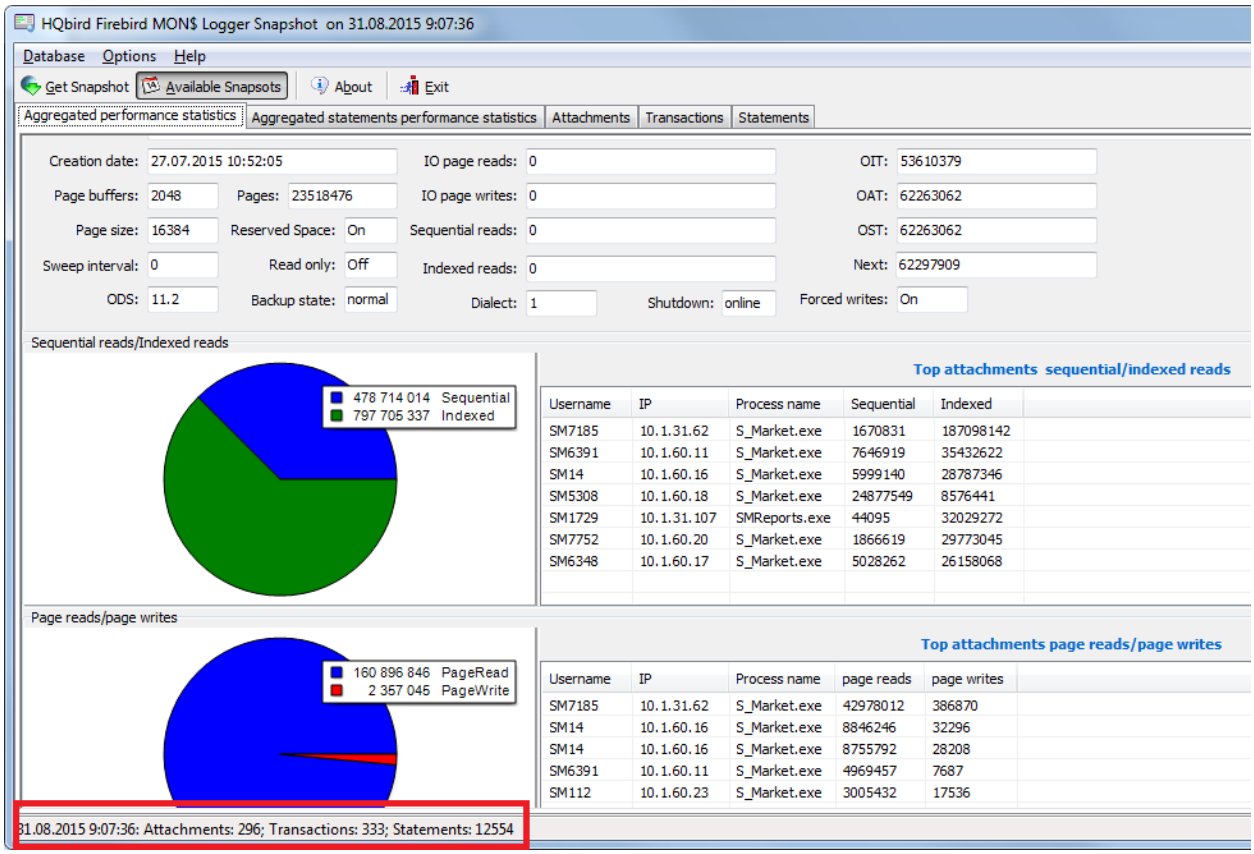


Figure 3. MON$ Logger: number of attachments

Например, в данном примере видно, что число соединений составляет 296. Очевидно, что использование 4-х ядерного процессора в данном случае будет слишком оптимистичным, а вот 24-ядерный вариант вполне подойдет. Также желательно подсчитать число одновременно активных запросов, так как соединения могут простаивать, не выполняя никаких SQL запросов.

> *Для грубой оценки необходимого количества ядер в CPU можно пользоваться правилом от 10 до 30 соединений на 1 ядро. 10 пользователей/ядро – приложение с преобладанием сложных и медленных запросов, 30 соединений/ядро – приложение с преобладанием простых, хорошо отлаженных запросов.*
>
> *Для более точной оценки можно посмотреть на закладке Statements число активных запросов (состояние Active) и всех остальных (IDLE И др.)*

Версия и архитектура Firebird

Если Вы используете версию Firebird 2.5, то следует иметь в виду, что для распараллеливания обработки на несколько ядер следует использовать архитектуры Classic или SuperClassic. Архитектура SuperServer в версии 2.5 может использовать только одно ядро на одну БД, поэтому ее не следует применять в высоконагруженных системах.

В версии Firebird 3.0 и SuperServer, и Classic, и SuperClassic используют возможности многоядерных CPU. Наибольшую производительность показывает Firebird 3.0 SuperServer.

RAM

При выборе RAM следует уделить внимание 2 моментам:

1) модули памяти должны быть с коррекцией ошибок (ECC RAM)

2) Правильно рассчитать объем RAM

ECC RAM

ECC RAM значительно снижает количество ошибок при работе с памятью и настоятельно рекомендуется для использования в промышленных системах.

Расчет необходимого объема RAM

Для расчета памяти нам придется немного углубиться в особенности различных архитектур Firebird.

Firebird 2.5 архитектуры Classic и Firebird 3.0 Classic запускают отдельный процесс для обслуживания каждого соединение, SuperClassic запускает отдельный поток для каждого соединения, но практически с той же структурой потребления памяти – каждое соединение имеет свой независимый страничный кэш.

Firebird SuperServer запускает один процесс с единым страничным кэшем для всех соединений.

Таким образом, потребление памяти складывается из следующих основных параметров:

1) Количество соединений
2) Размер страницы базы данных
3) Размер объектов метаданных (пропорционален количеству таблиц, триггеров, хранимых процедур и др., не регулируется, определяется по фактическому использованию)
    a. Для Classic и SuperClassic – на соединение
    b. Для SuperServer – на экземпляр открытой базы данных
4) Размер страничного кэша (определяется параметрами в заголовке БД или в firebird.conf или в свойствах конкретного соединения)
    a. Для Classic и SuperClassic – на соединение
    b. Для SuperServer – на экземпляр открытой базы данных

    Размер кэша для сортировок (определяется параметром в firebird.conf*)* Обратите внимание, что память для сортировок выделяется по мере необходимости, а не сразу же.

5)
    a. Для Classic – на соединение
    b. Для Super Server и SuperClassic – на процесс (т.е., кэш сортировок единый)
6) Для Classic/SuperClassic – размер таблицы блокировок (так как обычно он небольшой, то его выведем из расчетов).

Компанией IBSurgeon в ряде экспериментов получен ряд оптимальных значений количества страниц в страничном кэше Firebird:

- Classic/SuperClassic – от 256 до 2000 страниц
- SuperServer 2.5 – 10000 страниц
- SuperServer 3.0 – 100000 страниц

На основании этих экспериментов были созданы оптимизированные конфигурационные файлы Firebird для серверов с 4-6Гб памяти, вы можете скачать их отсюда: http://ib-aid.com/ru/optimized-firebird-configuration/

Формулы расчета необходимого объема RAM

Ниже представлены формулы приблизительного расчета необходимого объема памяти для Firebird. Реальное значение потребления памяти может отличаться, так как в этом расчете не учитывается объем памяти под метаданные, под битовые маски индексов, и т.д., что может увеличить расход памяти, но одновременно предполагается, что память под сортировки будет использована полностью во всех соединениях, чего обычно не происходит.

Когда база данных уже находится в эксплуатации, можно просто посмотреть средний размер памяти, используемый процессом Firebird (с помощью TaskManager или ProcessExplorer).

Расчет для Classic:

Количество соединений * ((Кол-во страниц в кэше * Размер страницы) + Размер кэша для сортировок)

Пример для Classic: пусть мы ожидаем 100 активных пользователей, размер страницы БД мы установили в 8Кб, а количество страниц в страничном кэше установили в 256, размер кэша для сортировок увеличили с 8Мб (значение по умолчанию для Classic и SuperClassic) до 64Мб:

100* ((256*8кб)+64) = 6600 Мб

Расчет для SuperClassic:

Количество соединений X (Кол-во страниц в кэше X Размер страницы) + Размер кэша для сортировок

Пример для SuperClassic: 100 пользователей, размер страницы БД 8Кб, количество страниц в страничном кэше 256, размер кэша для сортировок 1024 Мб

100*(256*8Кб) + 1024 Мб = 2024 Мб

Расчет для SuperServer

(Кол-во страниц в кэше X Размер страницы) + Размер кэша для сортировок

Пример для SuperServer (Firebird 2.5): 1БД, 100 пользователей, размер страницы БД 8кб, количество страниц в страничном кэше 10000, размер кэша для сортировок 1024 Мб:

(10000*8Кб) + 1024 = 1102 Мб

Пример для SuperServer (Firebird 3.0): 1 БД, 100 пользователей, размер страницы БД 8кб, количество страниц в страничном кэше 100000, размер кэша для сортировок 1024 Мб:

(100000*8Кб) + 1024 = 1805 Мб

«Излишняя память»

Часто Firebird упрекают в неэффективном использовании памяти, когда работающий процесс сервера потребляет небольшое количество RAM, а остающаяся память якобы не используется.

На самом деле это неверное суждение, происходящее, в основном, из-за непонимания работы механизма кэширования Firebird и несовершенства инструментов мониторинга операционных систем.

Прежде всего, необходимо четко представлять, что Firebird активно использует файловый кэш операционной системы. Когда страница загружается в страничный кэш Firebird, она проходит через файловый кэш ОС. Когда Firebird выгружает страницу из своего страничного кэша, операционная система при наличии свободной памяти продолжает держать этот кусок базы данных в RAM.

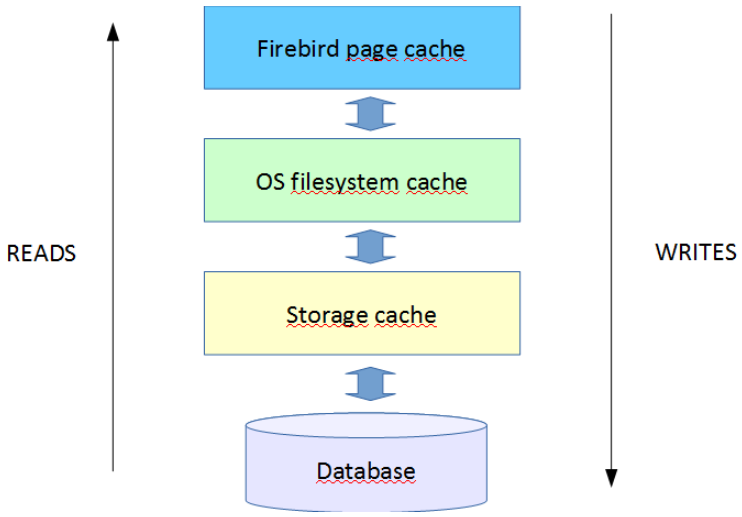


Figure 4. Cache levels: Firebird, OS and storage

Однако, при поверхностном взгляде, операционная система не показывает занятую под файловую кэш память как используемую. Например, вот типичная ситуация распределения памяти при работающем сервер Firebird, как ее показывает TaskManager:

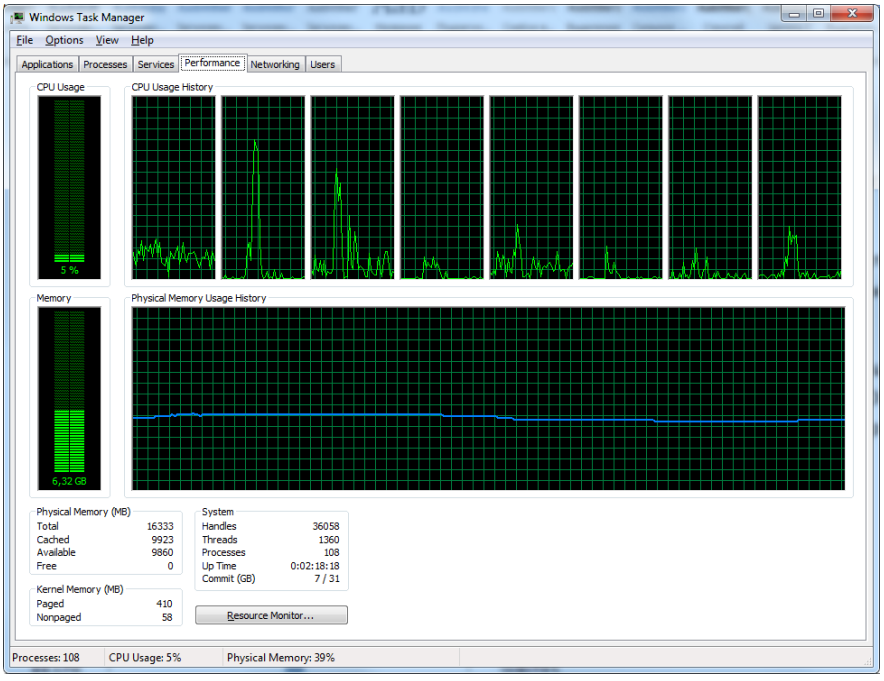


Figure 5. TaskManager does not show file cache usage

Выглядит так, как будто используется только 6.3Гб из 16Гб.

Однако, если использовать инструмент RAMMap (из набора утилит SysInternals от Microsoft), то картина выглядит более логичной:

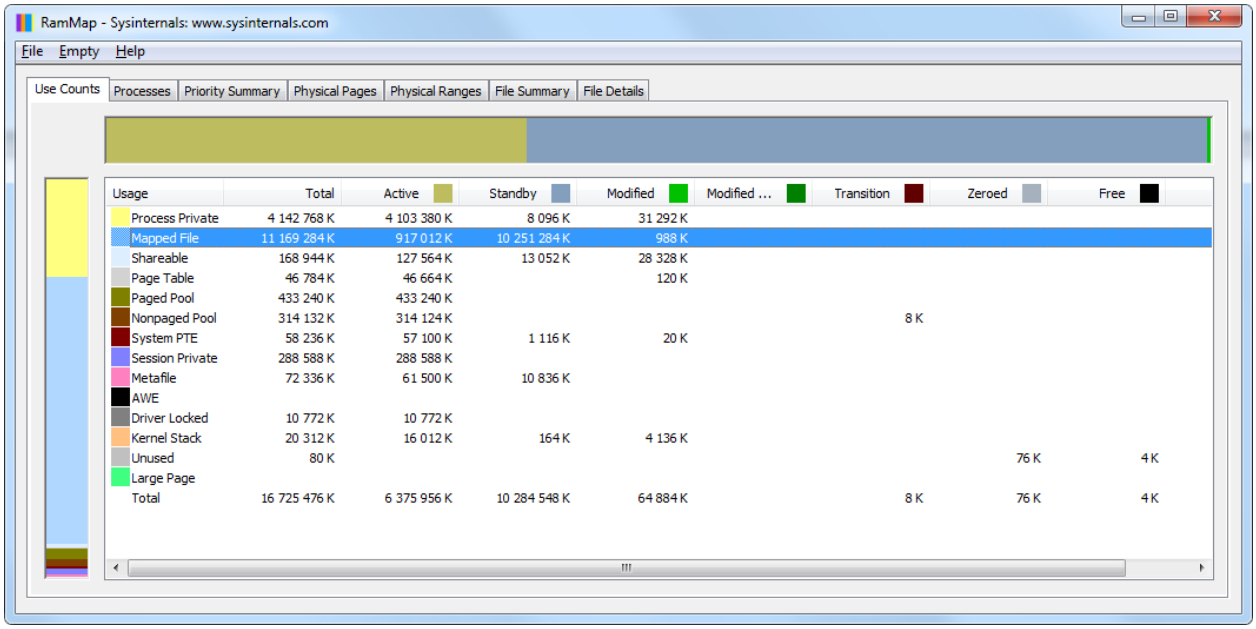


Figure 6. RAMMap shows details about memory usage: mapped files are cached databases

Файлы баз данных (dbw350_fb252x64.fdb и dbw250_fb252x64.fdb) закэшированы ОС, и занимают всю память, декларированную TaskManager как свободную :

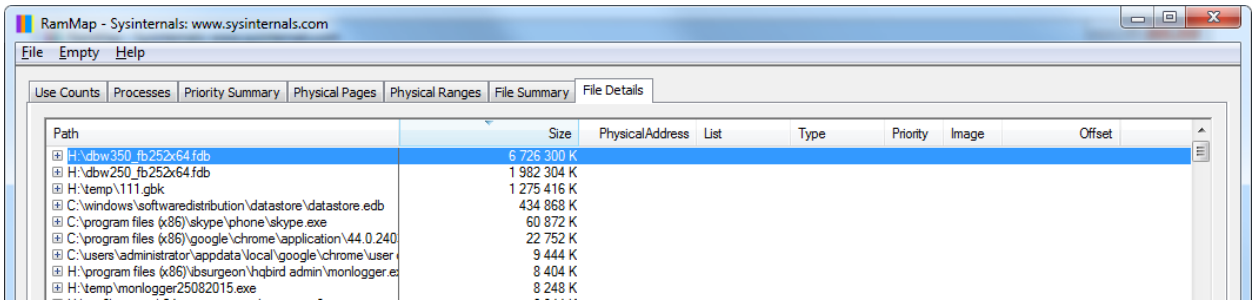


Figure 7. RAMMap: details about file cache usage

Отсюда вывод – операционная система эффективно использует всю доступную память для кэширования базы данных, вплоть до полной загрузки БД в память.

Дисковая подистема

Правильная конфигурация дисковой подсистемы является очень важной частью выбора и конфигурирования hardware для Firebird, так как ошибки на этом этапе дорого обходятся и тяжело исправляются.

Отдельные диски для всего

Чтобы уменьшить конкуренцию за дисковый ввод-вывод между операциями с файлом БД, сортировками и резервными копиями, а также уменьшить шанс одновременной потери и БД и резервных копий, рекомендуется иметь 3 разных диска (или raid-массива): для БД, для временных файлов и для создания и хранения бэкапов.

Когда мы говорим «отдельные диски», это означает, что физически потоки данных должны идти через разные каналы ввода-вывода. Если создать 3 логических диска на одном физическом диске, никакого улучшения производительности не произойдет. Однако, если 3 логических диска будут организованы на устройстве хранения данных (СХД), оснащенном многоканальными контроллерами, то производительность, скорее всего, возрастет, так как устройство может распределять потоки данных между контроллерами.

Иногда сообщают о том, что выделение отдельного диска для хранения файлов ОС и swap-файла ОС приводит к увеличению производительности.

SSD для базы данных

Для работы с базой данных лучше всего использовать SSD-диски, так как они обеспечивают отличное масштабирование при параллельном вводе-выводе. Обязательно следует использовать диски промышленного класса, с увеличенным числом циклов перезаписи, иначе велик риск потери данных из-за поломки SSD.

Некоторое время назад SSD-диски страдали от повышенного износа в случае, если на диске оставалось небольшое количество (менее 30%) свободного места. Упрощенно излагая, каждое изменение на SSD диске пишется в новую свободную ячейку, поэтому недостаток свободного места приводил к повышенному износу ячеек, которые остались свободными, и сокращению срока жизни диска.

Производители современных контроллеров SSD декларируют, что эта проблема была решена с помощью превентивного перемещения статичных данных, и теперь износ ячеек идет более или менее равномерно. Однако, точные спецификации и алгоритмы работы SSD закрыты производителями, поэтому мы все равно рекомендуем оставлять до 30% места на SSD-дисках свободным, а ожидаемый срок жизни занижать и планировать замену дисков не реже чем 1 раз в 3 года.

> *Например, если размер Вашей БД в данный момент составляет 100Гб, рост БД идет по 1Гб в месяц, то нельзя приобретать SSD минимального размера (120Гб), а лучше выбрать следующий в линейке - 250Гб. В тоже время, покупка 512Gb SSD будет пустой тратой денег, т.к. через 3 года диск будет желательно заменить.*

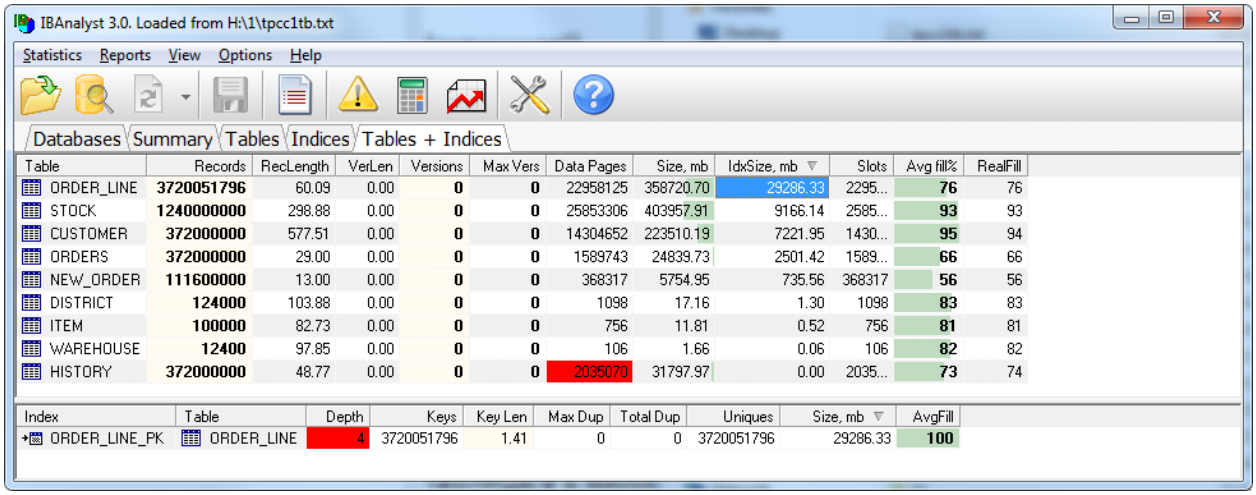
Наилучшим выбором является использование SSD эксклюзивно для работы с базой данных, так как любые операции ввода-вывода сокращают срок службы дисков.

Диск для временных файлов

Так как временные файлы на диске возникают только при отсутствии достаточного количества RAM, то конечно, лучше всего вообще избегать их появления на диске. Оценку количества и размера временных файлов в production системе можно провести лишь путем мониторинга папки с временными файлами. Такой мониторинг осуществляет FBDataGuard из дистрибутива HQbird. Узнав, сколько временных файлов сортировок и когда создается на диске, можно соответствующим образом нарастить RAM и изменить параметры firebird.conf.

В любом случае, Firebird требует указания папки, где будут хранится временные файлы. Обычно его оставляют по умолчанию, т.е. используется папка ОС для временных файлов. В случае, если свободной RAM достаточно, это достаточно хороший выбор.

Однако, есть еще один важный момент в размещении временных файлов на диске – это создание индексов при восстановлении верифицированного бэкапа (созданного утилитой gbak). При создании индекса создается временный файл, который содержит все ключи этого индекса. Если база данных достаточно велика, то размер индекса для какой-нибудь большой таблицы может быть также значительным. Например, в БД размером 1 Тб индекс для самой большой таблицы в 3.2 миллиардов записей в базе данных имеет размер 29 Гб, а при создании такого индекса потребовалось 180 гигабайт свободного места:



Чтобы предупредить исчерпание свободного места на системном диске, в Firebird.conf указывают второй диск в качестве дополнительного резервного места:

TempDirectories =C:\temp; H:\Temp

Если на первом диске пространство кончится, Firebird продолжит использовать для временных файлов второй диск, и так далее.

HDD for backups

Для создания и хранения бэкапов подойдут обычные HDD диски с интерфейсом SATA или nSAS. Они обеспечивают быструю линейную запись и чтение файлов бэкапа, и достаточно дешевы, чтобы не экономить на их размере, и хранить несколько резервных копий.

На диске для резервных копий всегда должно оставаться свободное место в размере последнего бэкапа+10%. В этом случае есть возможность создать свежий бэкап, убедиться в корректном

завершении процесса копирования (для БД размером в несколько терабайт такой процесс может идти несколько часов), и только потом удалить предыдущий бэкап.

Если же удалять предыдущий бэкап до окончания процесса создания нового, возможна ситуация, что в результате сбоя копирования новый бэкап не будет создан, старый уже удален, а база данных также повреждена, например, в результате сбоя диска.

Если Вы используете рекомендованный выше способ организации резервного копирования в виде комбинации инкрементального бэкапа в 3 уровнях и верифицированного бэкапа один раз в сутки, хранением только 1 последней копии, то формула расчета минимального необходимого места для бэкапа следующая:

Размер_БД*3+0.2*Размер_БД

Пример расчета места, необходимого для бэкапа

Пусть у нас есть БД в 100Гб, для которой мы храним 3 уровня инкрементального бэкапа неделя-день-час (по 1 копии) и 1 копию ежедневного верифицировованного бэкапа. Тогда резервные копии будут занимать следующее место:

- Nbackup_level_0_weekly - 100Гб
- Nbackup_level_1_daily – 5Гб (приблизительная оценка)
- Nbackup_level_2_hourly – 200 Мб (приблизительная оценка)
- Daily verified backup – 100Гб (приблизительная оценка)
- Плюс нужен резерв в 110Гб для создания следующей копии бэкапа.

Итого – 316Гб.

! размер инкрементного файла уровня 1 и выше зависит от количества страниц, которые были изменены с момента предыдущего выполнения nbackup. Определить размер этих файлов можно только экспериментально, т.к. объем изменений в БД зависит от приложений.

Разумеется, в оценку места для бэкапа следует закладывать и возможный аномальный рост БД, и соответственно увеличивать количество свободного места, иначе бэкап может аварийно прерваться из-за недостатка места.

> *Разумеется, интеллектуальные средства резервного копирования (FBDataGuard из HQbird), отследят недостаток места для бэкапов, предотвратят старт неудачного бэкапа и вышлют соответствующее сообщение администратору.*

HDD для базы данных

Может случиться так, что SSD окажется слишком дорогим решением, или размер БД слишком велик, и необходимо обойтись более дешевыми альтернативами.

В этом случае следует использовать HDD с интерфейсом SAS, при невозможности – диски SATA с интерфейсом nSAS, и совсем бюджетный вариант – обычные диски SATA.

Для ускорения работы жестких дисков (а также, надежности – см. ниже) следует объединять их в RAID10. RAID10 – это комбинация зеркалирования (RAID1) и stripe (RAID0). Хороший и правильно настроенный RAID контроллер с большим кэшем является неплохой альтернативой SSD.

Надежность и RAID

Разумеется, во всех упомянутых выше вариантах (за исключением диска, выделенного эксклюзивно для временных файлов) следует увеличивать надежность дисковой подсистемы путем объединения дисков в RAID.

- Для SSD дисков следует обязательно использовать RAID1 – т.е. 2 «зеркальных» диска, на которые одновременно пишутся изменения, что значительно уменьшит шанс полной потери данных. RAID 10 из SSD скорее всего будет избыточным, т.к. шина RAID будет ограничивать пропускную способность. Например, интерфейс 6 Гбит/с имеет пропускную способность в 600 мегабайт в секунду, а современные одиночные SSD уже достигли такой скорости. Таким образом, для RAID 10 мы получим тот же самый лимит в 600 мб/с. Разве что с PCI Express 3.0 можно организовывать RAID 10 из SSD, т.к. пропускная способность этой шины уже 16 гигабит в секунду и выше.
- Для HDD дисков, используемых для бэкапов, достаточно использовать RAID1, который обеспечит надежное хранение бэкапов и приемлемую скорость записи и чтения.
- HDD диски, используемые для БД, необходимо объединять в RAID10 (минимум 4 диска), который обеспечит оптимальное сочетание стоимости, надежности и производительности. Некоторые пользователи используют также RAID5, жертвуя производительностью взамен увеличенного пространства.

Настройка RAID

Прежде всего, необходимо проверить наличие и заряд батареи резервного питания (Backup Battery Unit, BBU) в RAID. При отсутствии данной батареи большинство RAID переходят в режим безопасной записи (полностью отключено кэширование записи), который обеспечивает меньшую скорость IO, чем обычный SATA диск!

> *Именно с этим фактом связано большинство горестных писем в техподдержку от пользователей, которые приобрели дорогой сервер и обнаружили, что он работает медленнее, чем десктопный компьютер. К сожалению, часть вендоров по умолчанию не снабжает свои RAID батареями, поэтому это первое, что необходимо проверить и, при необходимости, исправить.*

Затем необходимо провести настройку кэширования чтения и записи. Часто по умолчанию кэш полностью выключен, а если мы хотим добиться приличной скорости от RAID, то кэш необходимо включить.

Кроме включения кэша, необходимо проверить режим его работы, это может быть write through и write back. Быстрый способ работы с кэшем – write back, в этом случае изменения пишутся в кэш контроллера, и через некоторое время – непосредственно на диск.

Проверка наличия батареи, кэша и режима работы RAID может быть осуществлена в фирменных утилитах, которые производители поставляют вместе с RAID.

Современные RAID-контроллеры также имеют возможности тонкой настройки кэша – его можно ориентировать на чтение или на запись. Обычно по умолчанию стоит 50%/50% на чтение-запись.

Чтобы определить, в каком отношении стоит настраивать кэш, можно вновь воспользоваться инструментом MON$ Logger из продвинутого дистрибутива HQbird. Она показывает отношение

числа операций чтения к операциям записи (в агрегированном виде с момента первого соединения к серверу):

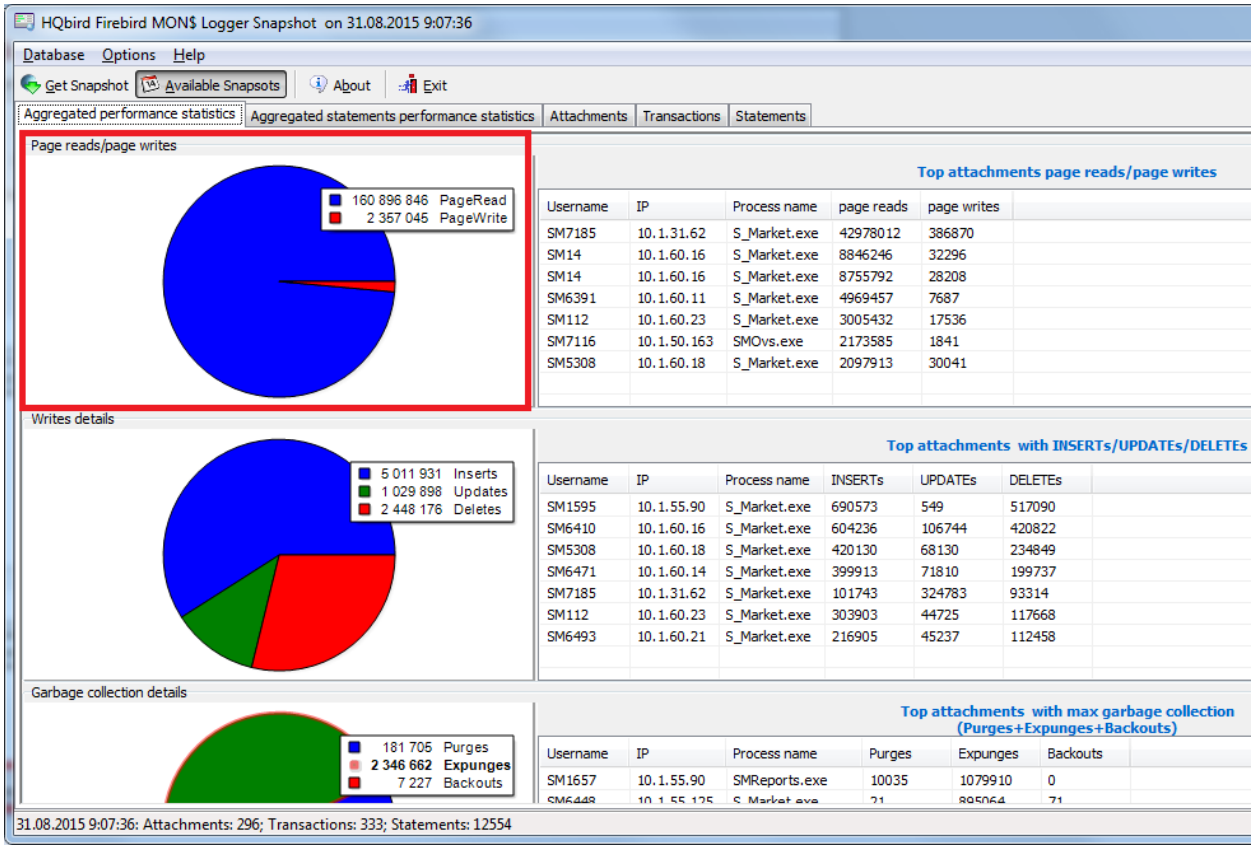


Figure 8. HQbird MON$logger: reads/write ratio

Как видите, в данном примере операций чтения гораздо больше, чем операций записи, поэтому имеет смысл настроить RAID контроллер на 80% чтений и 20% записи.

СХД

В последние годы получили распространение интегрированные системы хранения данных (СХД), которые включают в себя массив дисков с гибкими возможностями конфигурации (все типы RAID) и продвинутым кэшированием. Обычно СХД имеют несколько контроллеров ввода-вывода, что позволяет обслуживать несколько серверов одновременно, а также работать достаточно быстро.

Многие организации приобретают СХД и используют их для работы с базами данных Firebird. При условии правильной настройки от СХД можно добиться очень хорошей производительности. Основные моменты, которые нужно учитывать при использовании СХД:

1) Наличие нескольких высокопроизводительных контроллеров дисков, обеспечивающих многоканальный обмен данными
2) Наличие батарей резервного питания (BBU), если они предусмотрены конструкцией.
3) Диски для баз данных должны быть организованы в RAID10.
4) Кэш должен быть включен, режим записи установлен в write back.
5) Если к СХД подключены несколько компьютеров, то каждый из них обслуживается своим контроллером.
6) Установлены свежие драйвера для СХД. В нашей практике были случаи, когда более новые драйвера обеспечивали 30% прирост скорости.

7) Если на СХД созданы несколько логических дисков (для БД, бэкапов, ОС), то они разделены по каналам ввода вывода. Попытка использовать один канал для всех дисков сразу приведет к снижению производительности.
8) Аналогично, если СХД будут использовать сразу несколько серверов и БД, производительность может уменьшиться из-за превышения полосы пропускания (bandwidth) контроллеров ввода-вывода.
9) Часто используются комбинированные схемы, когда ОС и временные файлы находятся на локальных дисках, а база данных и бэкапы на СХД.

Часто СХД используются в схеме «2 сервера – 1 СХД», с целью построения отказоустойчивого кластера. Надо отметить, что такой кластер поможет победить только проблемы, связанные непосредственно с поломкой hardware одного из серверов, чтобы быстро переключиться на второй сервер. Если проблема возникнет с СХД или с самой базой данных, то такое решение будет бесполезно.

Для построения реального отказоустойчивого решения необходимо использовать решения с репликацией данных между экземплярами баз данных. Такие решения доступны для Firebird 2.5 b 3.0.Вы можете обратиться на support@ib-aid.com, чтобы узнать больше о решениях высокой доступности для Firebird.

Краткие выводы и рекомендации

Давайте коротко суммируем выводы и рекомендации по hardware для Firebird.

1) Необходимо использовать многоядерные CPU для обслуживания большого количества пользователей
2) Оперативная память RAM – обязательное минимальное количество рассчитывается из количества пользователей и параметров БД, количество сверх того будет эффективно использоваться операционной системой для кэширования файла БД.
3) Использовать отдельные диски для БД, temp-файлов и бэкапов
4) Для размещения БД лучше использовать SSD диски.
    a. Резервировать от 30% свободного места на SSD дисках.
    b. Желательно эксклюзивно использовать диск под БД.
    c. Использовать SSD промышленного класса (с большим количеством циклов перезаписи).
5) Обязательно использовать RAID.
    a. Для SSD – RAID 1, для HDD – RAID10, для HDD под бэкапы – RAID1.
    b. Обязательно проверять наличие и заряд батареи RAID
    c. Обязательно проверять настройку write back/through (нужно ставить write back).
    d. Ряд контроллеров имеет настройку размера кэша RAID, например, 75% для чтения, 25% для записи, или 50/50, и т.д. Необходимо ставить Mon$logger, ПО для управления параметрами RAID, смотреть соотношение чтения и записи, и менять настройки RAID.
    e. В использовании СХД есть свои плюсы и минусы, чтобы получить эффект от ее использования, необходимо правильно настраивать СХД.

6) Для построения отказоустойчивого решения необходимо использовать решения с репликацией, работающие на нескольких серверах.

Контакты

Компания IBSurgeon/IBase.ru разрабатывает HQbird - продвинутый дистрибутив Firebird для предприятий, комплексную техническую поддержку для Firebird, разработку кастомных спецсборок и разрешение других сложных вопросов.

Обращайтесь: support@ib-aid.com, +7 495 953 13 34