Производительность для очень больших БД

Дмитрий Кузьменко
Генеральный директор
iBase.ru

FIREBIRD CONF 2024 | РЕДСОФТ

Содержание

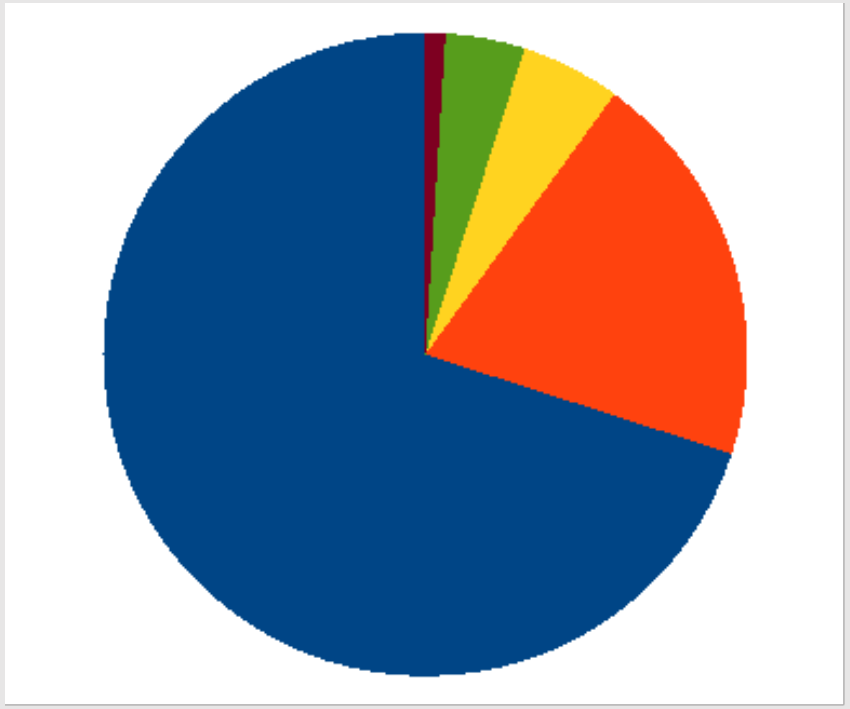
- Что такое «большие базы данных»
- Выбор и настройка железа
- Оптимизация ОС
- Конфигурация Firebird
- Обслуживание БД
- Анализ работы приложений с БД

О каких именно «больших БД» пойдет речь

- Firebird давно перевалил за 1Тб, только среди наших клиентов:
- Медицина МИС Инфоклиника: 2.9Тб x 2000 пользователей, 937Гб x 1100 пользователей
- Логистика DPD, Болгария, 1.5Тб - 3.5Тб, 500 пользователей
- Финансы Bomfim, Бразилия, 1Тб, 700 пользователей
- Размер относителен
- В середине 90х у компьютеров были диски 2 гигабайта. Теперь – в 1000-2000 раз больше
- В 2000х уже люди жаловались на «проблемы» с 200мб базами данных (на Windows XP)
- Цитата – «Firebird настолько нетребователен, что может работать на самом дешевом оборудовании и не обслуживаться годами»
  - Уже не работает. За 20 лет с момента этой цитаты базы данных существенно выросли, и выросло количество пользователей, которые работают с этой БД

Сейчас

Размер БД	Доля, %
Менее 200Гб	70
200-500 Гб	20
Более 500 Гб	5
Более 1Тб	4
Более 5 Тб	1



По наблюдениям - размер баз увеличивается вдвое каждые три года за последние полтора года – на 25-39%.

Основные признаки «большой БД»

- **Когда активная часть БД не помещается в кэш Firebird и кэш операционной системы**
- Вопрос в определении «активной части»
  - большие таблицы - сканирование pointer pages
  - большие индексы – глубина индексов

- Исключения (обслуживание <> регулярная работа)
  - Бэкап – читается вся БД
  - NBACKUP – сканируется вся БД (при –b 0)
  - Restore – все страницы имеют флаг swept = 0
    - первый же sweep перезапишет все страницы БД!

Выбор и настройка железа

- Процессор, ядра
  - частота ЦПУ пропорциональна скорости исполнения фетчей
  - запрос не распараллеливается, выполняется на одном ядре
  - Выбор модели ЦПУ «количество ядер против частоты» зависит от профиля нагрузки
- Память — также зависит от профиля нагрузки. Общие соображения:
  - Firebird-у для больших БД отдавать более 50% RAM нельзя. Остальные 50% ОС использует как файловый кэш и прочее.
  - В 50% для Firebird входит
    - кэш БД
    - память метаданных
    - память для сортировок
    - битовые маски индексов
- Диски
  - SSD, RAID 10, кэширование 75% reads, 25% writes, SSD

Процессор

- Откуда может быть высокая загрузка:

- выборка больших объемов данных
- частые коннекты-дисконнекты
- частые короткие запросы
- очередь хэш-таблицы

- Общая современная рекомендация: одно ядро >=3гГц
- Если все ядра загружены – нужен процессор с большим количеством ядер

```
LOCK_HEADER BLOCK
        Version: 146, Creation timestamp: 2023-10-18 16:50:57
        Active owner:      0, Length: 20971520, Used: 348744
        Enqs:     98, Converts:      3, Rejects:      0, Blocks:      0
        Deadlock scans:      0, Deadlocks:      0, Scan interval:  10
        Acquires:    116, Acquire blocks:      0, Spin count:   0
        Mutex wait: 0.0%
        Hash slots: 40099, Hash lengths (min/avg/max):    0/   0/   6
        Hash lengths distribution:
                0  :    40013   (99%)
                1  :       82   (0%)
                2  :        2   (0%)
                3  :        1   (0%)
                4  :        0   (0%)
                5  :        0   (0%)
                6  :        1   (0%)
```

Память

- X RAM
- 50% оставляем для ОС, файлового кэша и прочего
- 50% можем оставить для Firebird
- 30-35% на кэш Firebird
  - Не забывайте, что общий кэш firebird.conf действует для всех баз (включая security)
    - поэтому в databases.conf для security указано DefaultDbCachePages = 256
  - Указывайте конкретный кэш для БД в databases.conf
- 20-15% на память для сортировок
  - в Firebird 4 память для сортировок раздельная для разных БД. Указывайте TempCacheLimit для каждой БД отдельно в databases.conf
- Windows – RAMMAP, Linux – free –h
- Калькулятор Конфигураций поможет избежать ошибок

Диск

- RAID 10
- RAID 5 – не нужно (запись плохая, восстановление долгое)
- RAID 6 – можно
- Настройки RAID – 75% reads, 25% writes (для файловых серверов умолчание 50/50)

- Работа с БД – это всегда random IO. Поэтому оценивать нужно параллельные IOPS
  - Тесты – CrystalDiskMark (4я строка!), IOMeter, FIO
  - Тест Simple Insert Update Delete – однопоточный тест на оценку снизу.
    Если меньше 7000 вставок в секунду – остальные тесты можно не смотреть,
    так как дело плохо.

- Выключенный кэш записи – плохо (primary domain controller, и т.д.)
- BBU – обязательно
- Эффективность SSD NVME – зависит от модели материнской платы и поколения PCI

Виртуальные машины

- Гарантированная потеря на процессоре и памяти - ~3-5%
- Потеря на дисках может быть существенной!
  - зависит от драйверов ОС, драйверов ВМ и настроек ВМ

- Бэкап виртуальной машины не является заменой обычных бэкапов

Оптимизация ОС

- Минимизируйте посторонний софт, для исключения «внезапного» перерасхода памяти
- **Windows или Linux ?**
- Windows – настраивать нечего
  - (кроме performance plan – high, +20% производительности)
- Linux – *нужна версия яда 5+*
  - Max Open Files - LimitNOFILE= 100000
  - vm.max_maps – 65536 -> 512000
  - Для ВМ с большим количеством RAM возможно vm.swapiness = 1
  - ext3/ext4, не используйте xfs (и другие ФС с журналированием)
  - nobarrier (barrier = 0) – отключение принудительной записи (ускоряет до 10 раз).
  - ! на ZFS не работает nbackup –d ON
  - Swap не отключать! Размер 4-8 Гб

- Если больше 500 коннектов и >1тб база – можно использовать Linux (с учетом тюнинга)

Конфигурация Firebird

- В firebird.conf мало параметров, влияющих на производительность
- Важно “не испортить” firebird.conf/databases.conf
  - Самая тяжелая ошибка это забыть Page Buffers в заголовке БД
- Калькулятор конфигурации
  https://cc.ib-aid.com/democalc.html
  - для 2.5, 3.0, 4.0 и 5.0, разные архитектуры

- Подкручивать параметры поштучно один за другом, с замером результатов в процессе работы

Важные параметры (по умолчанию)

- ServerMode = Super
- DefaultDbCachePages = 2048

- TempBlockSize = 1M
- TempCacheLimit = 64M
- TempDirectories =

- LockMemSize = 1M
- LockHashSlots = 8191

- CpuAffinityMask = 0

- MaxUnflushedWrites = 100
- MaxUnflushedWriteTime = 5

- GCPolicy = combined
- FileSystemCacheThreshold = 64K
- FileSystemCacheSize = 0
- DeadlockTimeout = 10

- WireCrypt = Enabled (для клиента) / Required (для сервера)
- WireCompression = false

- InlineSortThreshold = 1000

- MaxParallelWorkers = 1 (можно 64)
- ParallelWorkers = 1

Изменение параметров

- Всегда добавляйте комментарий # – кто, когда, зачем
- Измененные параметры добавляйте в начало конфигурации
- **DefaultDBCachePages** = 30% от 50% RAM
  - 64гб RAM – 50%=32гб, 20гб можно отдать на кэш Firebird
- **TempCacheLimit** – из остатка в 10%
  - оставшиеся 10гб из примера выше
- **LockHashSlots** – можно сразу установить 64999
- **LockMemSize** = 30M (старая формула не работает, см. fb_lock_print)
- **InlineSortThreshold** = 8192 (см. explain plan запросов в SORT)

- Не забывайте корректировать параметры при изменении железа или архитектуры

Параллельные операции

- Ускоряют «непараллельные» операции в 3-5 раз
- Пример – **MaxParallelWorkers = 8, ParallelWorkers = 4**
- Создание индекса будет идти в 4х тредах
- Автосвип будет идти в 4х тредах
  - Не надо так делать. Выключите автосвип, запускайте gfix –sweep –par 4 в нужное время
- Бэкап – укажите явно gbak –b –g –par 4 (до 5.0 он игнорирует ParallelWorkers)
- Рестор – как минимум индексы будут создаваться параллельно
- С параметрами выше можно gbak –b –g –par 4 + gfix –sweep –par 4 (но не надо)
- Если ParallelWorkers = 1 – многопоточность будет работать только при явном указании
  –par n для бэкапа, рестора и sweep
- Бэкап базы 12 часов или 6 часов – большая разница! Используйте параллельность
- gfix –icu –par n – для Linux, тоже распараллеливается

Статистика – куда смотреть

- gstat –r database.fdb >statyyyymmdd.txt

- Размер таблиц – просто информация
- Глубина индексов – повод для размышления
  - Root page: 42984793,**depth: 3**, leaf buckets: 33034, nodes:9899575
  - Глубина 4 и выше – проблема с производительностью

- Обычно проблемы начинаются на базах данных больше 100 гигабайт

Другая статистика

- gstat –h database.fdb
- **Next Attachment ID** – сравните за интервал времени. Вероятно, слишком много новых коннектов
- **Next Transaction** – сравните за интервал времени. Вероятно, слишком много новых транзакций
- **Oldest Active transaction**
  - Препятствует превращению версий в «мусор»
  - Возможно, кто-то «заснул в IBExpert» ☺
    - или длительные активные транзакции в приложениях
    - ! в 4.0 транзакции RC RO удерживают версии

Обслуживание БД

- Резервное копирование
  - Штатный бэкап-рестор «на грани» по длительности
  - Репликация! (nbackup, асинхронная и синхронная репликация)
- Мониторинг состояния
  - Свободное место!
  - Количество пользователей
  - Потребляемая память
  - Скорость дискового ввода-вывода
- Регулярный sweep
  - с отключением коннектов с длинными пишущими транзакциями перед запуском sweep
- Пересчет селективности индексов

Анализ работы приложений с БД

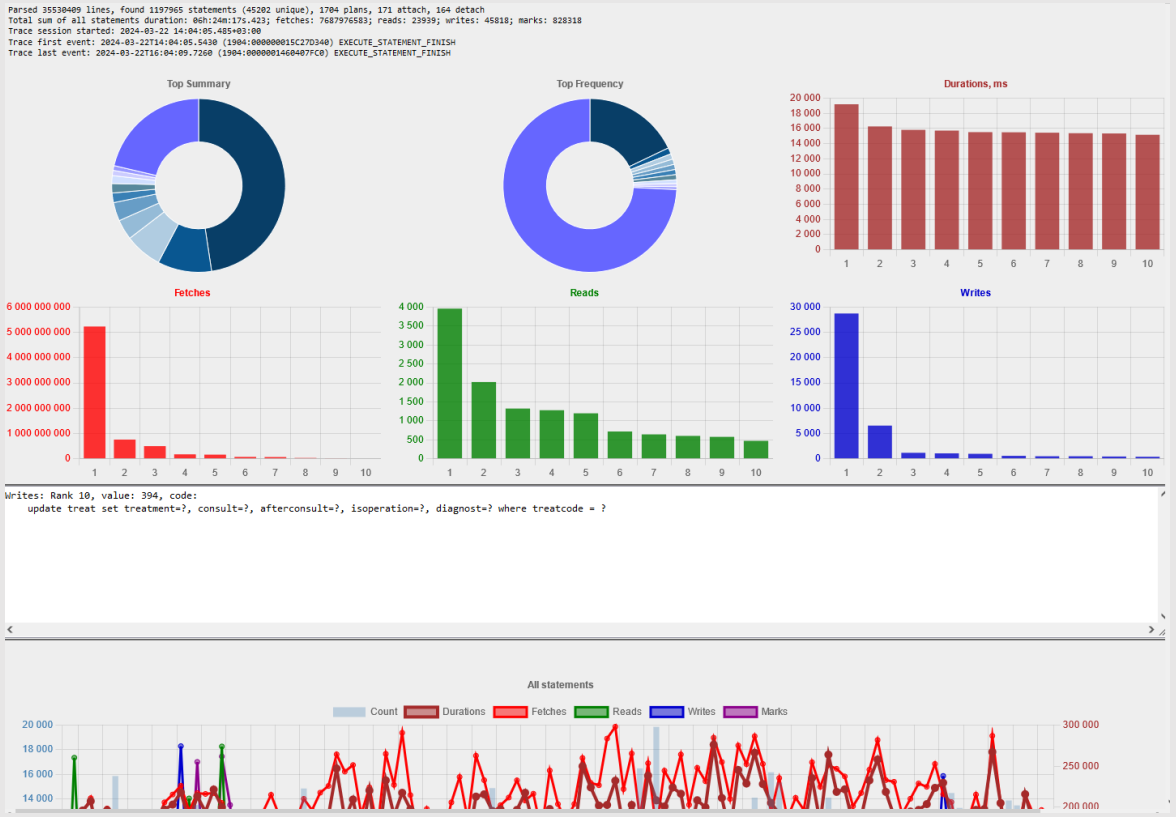
- Трассировка

- С отсечкой более 500мс – длительные тяжелые запросы
- С отсечкой 0мс – частотные и все остальные запросы

- Частотные запросы могут съедать более 50% от производительности сервера
- Натуральные чтения в большом объеме создают проблемы
- Масса индексных чтений может создавать проблемы
- Непараметризированные запросы могут создавать проблемы при анализе трассировки

Анализ трассировки

- Отчет по
  - длительности
  - частотности
  - суммарно-времени
  - обращений к памяти (fetches)
  - чтений с диска
  - записи на диск
  - планы



SORT BY DURATION [VIEW PROCESS SUMMARY OR SORT BY DURATION, FREQUENCY, TIME-SUMMARY, FETCHES, WRITES, READS, PLAN-SUMMARY, PLAN-FREQUENCY]

Резюме

- Оптимальный выбор аппаратного обеспечения (эмпирический)
- Оптимальные настройки firebird.conf и databases.conf под аппаратное обеспечение
- Слежение за загрузкой процессора, потреблением памяти и вводом-выводом
- Анализ статистики базы данных
- Анализ запросов (трассировки)
  - длительные запросы
  - частотные запросы

Спасибо за внимание!

Вопросы?