PALOMINODB

OPERATIONAL EXCELLENCE
FOR DATABASES

Грабли в MySQL.

От детских игрушек до Страшных Облачных Граблей

Vladimir Fedorkov

DevConf, Москва, 2013

www.palominodb.com

Доброе утро!

- Кто все эти люди?
- Зачем мы здесь?

Мои первые грабли

- Конфигурация MySQL
  - 5% настроек обеспечивают 95% производительности
  - InnoDB наше все, MyISAM уже не наше!
- Что тюнить сразу
  - innodb_buffer_pool_size
  - innodb_file_per_table
  - key_buffer
  - server-id

Мои вторые грабли

- Сеть
  - skip_name_resolv
  - max_connect_errors
    - FLUSH HOSTS
  - max_allowed_packet
- IO
  - innodb_flush_log_at_trx_commit
  - innodb_log_file_size

Тест «насколько админу скучно»

- max_join_size – два балла
- sort_buffer_size – пять баллов
- tx_isolation – десять баллов
- Правило 5/95 работает безукоризнено

Путь самурайской базы

- Готовим MySQL к смерти
  - log_bin
    - Для point-in-time recovery
    - Желательно на другой диск
  - expire_logs_days
  - max_binlog_size
  - sync_binlog
    - Если некуда девать производительность B/B
    - Или если данные очень нужны
- Идем на доклад Светы Смирновой в **17:45**

Ура, сконфигурили!

- Правильные настройки
  - не мешают работе MySQL
  - обеспечивают отказоустойчивость
    - только на уровне операционной системы!
  - не обеспечивают его производительности
- Допустим все работает

Запросы

- Обеспечивают 99,99% тормозов
  - И головную боль администраторов
- Лучший MySQL запрос тот который до MySQL не дошел.
  - MySQL не сможет выполнить некоторые запросы быстро
  - Просто потому что так спроектирован

А что с ним не так, турецкий?

- Ограничение дизайна MySQL
  - ACID медленный!
  - B-Tree работает только для спец. случаев
  - Запрос обрабатывается одним ядром
  - Full-text индесы гавн... Где там Аксенов?
  - Про репликацию можно говорить часами
    - Что и делаем на собеседованиях
  - Если вышеперечисленное для вас большая проблема – вы используете неправильный инструмент. Вам нужен не MySQL.
    - Если вы не FaceBook.

Прыгаем на грабли, full table scans

SELECT * FROM table …

- WHERE DAY(FROM_UNIXTIME(`ts`)) = 205
- WHERE deleted != 1
- WHERE id NOT IN (1,2,3,…,10)
- WHERE url LIKE ‘%чтототам%’
  - Не путать с LIKE ‘чтототам%’ !
- ORDER BY RAND()

Где грабли?

- Индексы не помогают!
  - Значение функции считается для всех строк
  - B-Tree не эффективен
- Читать всю таблицу долго
- Пока читаем вымываем память базы ненужными данными
- Как бороться?
  - Пересматривать логику запросов
  - Использовать предварительно аггрегированные данные вместо расчета функций «на лету»
  - Использовать deleted = 0 вместо deleted != 1

Высокоселективные запросы

- SELECT * / COUNT(*)
  FROM users WHERE sex='female'
  - Здравствуй половина таблицы
- Как бороться?
  - Читаем только то, что показываем
    - Используем LIMIT
  - Делаем агрегацию для счетчиков
  - Кешируем все, что можем
    - Не в MySQL query cache!
  - Используем внешние инструменты (Mongo, Sphinx, etc)

Временные таблицы

- Плохо
  - Если попадает на диск – очень плохо!
    - Если есть TEXT или BLOB поля на диск таблица попадет
- Когда создаются
  - GROUP BY
  - Подзапросы
  - DISTINCT + ORDER BY

Что делать?

- Тюнить буфферы временных таблиц
  - tmp_table_size
  - max_heap_table_size
- Запускать «тяжелые» запросы на отдельной реплике
  - Например для генерации отчетов

Если нагрузка большая

- Query cache
- Thread cache
- Table cache
- Slow query log
- wait_timeout
- connection pooling
- Репликация и шардинг

От сервера кластеру

- Один сервер мало, два – плохо.
  - Репликация медленная
    - Работает в один поток (исправили в 5.6)
  - Репликация хрупкая (не исправили в 5.6)
  - Доверия репликации нет
    - Консистентность данных требует проверок
      - Даже если видимых сбоев небыло
      - pt-table-checksum + pt-table-sync
- Есть варианты
  - Tungsten и Galera

Прелести кластера

- Если у тебя упал единственный сервер это трагедия
  - А если один из десяти?
- Можно использовать реплики для бекапа
- Можно балансировать нагрузку
  - Можно сделать реплику для генерации «тяжелых» отчетов

Что еще может кластер

- Встать колом весь от вовремя запущенного ALTER TABLE
- Эффектно среплицировать команду DROP DATABASE
- С разной скоростью выполнять одинаковые запросы на разных нодах

Ежедневная рутина

- Отказы железа
- Баги софта
- Надежные бекапы и восстановление
- Мониторинг
- Апгреды и настройка репликации
- Настройка сети и безопасности

- Можно ли это все автоматизировать?

От кластера к облаку

- На примере Amazon RDS
  - Relational database service
- MySQL, Oracle & MSSQL
  - Был MySQL 5.5 и уже есть 5.6
- Что позволяет?
  - Создавать/удалять ноды и реплики
  - Выделять реплики из кластера
  - Автоматизировать развертывание БД и фронтов
    - Вплодь до полного скриптования

Что это значит для нас?

- Полностью автоматизированное развертывание приложение с использованием RDS и EC2
- Гибкий контроль производительности и стоимости с помощью добавления и удаления машин
  - В зависимости от времени дня
  - В зависимости от текущей нагрузки
- Изменение параметров кластера на лету

Что сделать не получиться?

- Зайти на RDS инстанс по SSH
- Прочитать binary log
- Починить репликацию
  - Сделать кросс-региональну реплику
    - Команда RDS работает над этим
- Сделать внешний бекап с помощью xtrabackup
  - Только mysqldump, только хардкор

Что мы можем контролировать?

- Выбирать регион для инстанса
- Конфигурировать MySQL
- Выбирать тип инстанса
- Открывать/закрывать доступ по сети
- Выбирать параметры дисковой подсистемы

Регионы и availability zones

- US
  - East 1 (Northern Virginia)
  - West 1 (Northern California)
  - West 2 (Oregon)
- EU: Ireland
- Asia Pacific: Singapore, Tokyo, Sydney
- South America: São Paulo
- More to come across the world

Availability zone

- В каждом регионе нескольк AZ
- Можно сделать AZ мастер
  - Поможет в случае креша основного мастера
    - Репликация может быть сломана

Storage

- PIOPS & non-PIOPS диски
  - Полностью разные архитектурно и физически
- Вы в облаке

RDS Sizes

- **Micro DB Instance**: 630 MB memory, Up to 2 ECU (for short periodic bursts), 64-bit platform, Low I/O Capacity, Provisioned IOPS Optimized: No
- **Small DB Instance:** 1.7 GB memory, 1 ECU (1 virtual core with 1 ECU), 64-bit platform, Moderate I/O Capacity, Provisioned IOPS Optimized: No
- **Medium DB Instance**: 3.75 GB memory, 2 ECU (1 virtual core with 2 ECU), 64-bit platform, Moderate I/O Capacity, Provisioned IOPS Optimized: No
- **Large DB Instance**: 7.5 GB memory, 4 ECUs (2 virtual cores with 2 ECUs each), 64-bit platform, High I/O Capacity, Provisioned IOPS Optimized: 500Mbps
- **Extra Large DB Instance**: 15 GB of memory, 8 ECUs (4 virtual cores with 2 ECUs each), 64-bit platform, High I/O Capacity, Provisioned IOPS Optimized: 1000Mbps
- **High-Memory Extra Large DB Instance** 17.1 GB memory, 6.5 ECU (2 virtual cores with 3.25 ECUs each), 64-bit platform, High I/O Capacity, Provisioned IOPS Optimized: No
- **High-Memory Double Extra Large DB Instance**: 34 GB of memory, 13 ECUs (4 virtual cores with 3,25 ECUs each), 64-bit platform, High I/O Capacity, Provisioned IOPS Optimized: No
- **High-Memory Quadruple Extra Large DB Instance**: 68 GB of memory, 26 ECUs (8 virtual cores with 3.25 ECUs each), 64-bit platform, High I/O Capacity, Provisioned IOPS Optimized: 1000Mbps

Parameter groups

- Содержит все возможные настройки MySQL
- Дефолтные настройки не оптимальны
  - Нужно создать свою группу и поменять как надо
  - Число PG не бесконечно
- Некоторые настройки поменять нельзя
- Статические и динамические настройки почти как в «родном» MySQL

Добро пожаловать в облако!

- Железо можеть быть разное
  - Даже на каждом запуске бенчмарков
- IO зависит от сети
- Стоят ограничители CPU & IO
- Ты не знаешь своих соседей

Как жить?

- Всегда ориентируемся на худший случай
- Максимально уменьшить нагрузку на IO
  - Перейти на PIOPS (SSD) где возможно
- Агрессивный шардинг наш друг и верный помошник
- Внимательно следить за
  - нагрузкой на CPU/IO
  - Использованием памяти
  - Вовремя обнаруживать битые реплики

Недавние улучшения

- Можно переименовывать сервера
- Добавили точность до микросекунд
- Теперь можно смотреть логи
- Можно апгредится до MySQL 5.6

Когда использовать облако?

- Когда не очень много данных
  - Или когда вы можете их зашардить
- Когда сложно предугадывать нагрузку
- Когда быстро нужно много ресурсов
  - Или когда ресурсы нужны не на долго
- Когда не хватает рук для администрирования

Что дальше?

- Зафолловить @vfedorkov
- Посмотреть другие доклады на astellar.com
- Посмотреть блог на http://palominodb.com
- Сходить:
  - «Новые возможности репликации в MySQL 5.6» в 16:35 зал 5
  - «Как делать backup MySQL» в 17:45 зал 7
  - «Новые фичи в MariaDB и в MySQL - что общего и в чем разница» в 18:45 зал 9
  - На мастерклассы Badoo и Светы Смирновой завтра
  - Шодан (Аксенов А.А.) тоже будет сегодня жечь
- Вопросы!

Спасибо!