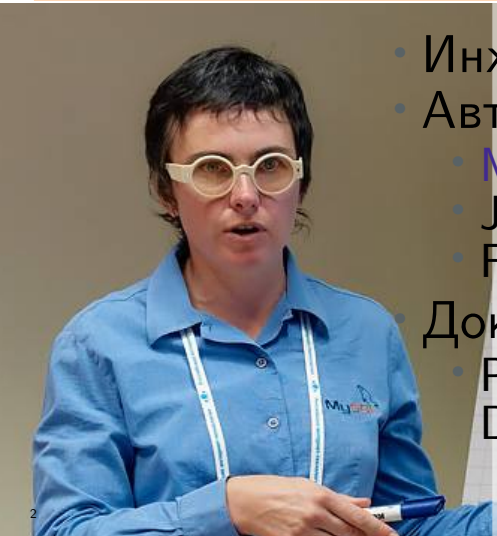
Что нужно знать

о трёх топовых фичах MySQL

August, 22, 2019

Sveta Smirnova

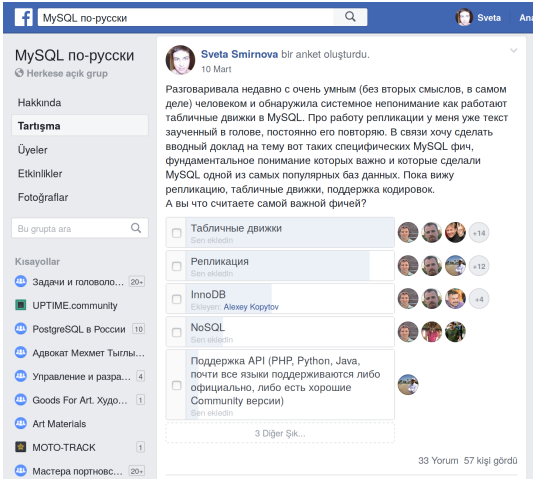
Света Смирнова



- Инженер тех. поддержки MySQL
- Автор
  - MySQL Troubleshooting
  - JSON UDF функции
  - FILTER clause для MySQL
- Докладчик
  - Percona Live, OOW, Fosdem, DevConf, HighLoad...

PERCONA

Идея Доклада



PERCONA

Содержание

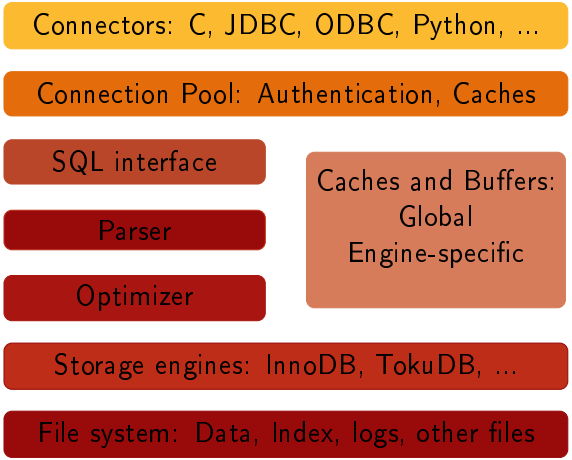
- Табличные Движки
  - InnoDB
- Репликация
- NoSQL

PERCONA

Табличные Движки

PERCONA

Архитектура MySQL



- `mysqld` и его файлы
- Коннекторы
- Оптимизатор
- Кэши
- Табличные движки
- Управление

PERCONA

Табличные Движки

- Владеют данными
- Собственный формат индексов
- Собственные блокировки
- Собственная диагностика
- Собственные журналы
- `CHECK TABLE`

PERCONA

Каждый Хранит Данные По-своему

- InnoDB
  - Сначала пишет в redo log и buffer pool
  - Асинхронно копирует в файлы таблиц
  - Мы не можем скопировать только tablespaces сразу после записи: новые данные будут только в redo logs и buffer pool

PERCONA

Каждый Хранит Данные По-своему

- InnoDB
- MyISAM, CSV, ...
  - Сразу изменяет файлы данных
  - Просто реализовать бинарный бэкап
  - Изменения нельзя откатить

PERCONA

Каждый Хранит Данные По-своему

- InnoDB
- MyISAM, CSV, ...
- Blackhole
  - Игнорирует все записи

PERCONA

Каждый Хранит Данные По-своему

- InnoDB
- MyISAM, CSV, ...
- Blackhole
- Federated
  - Читает и пишет с удалённого сервера
  - Зависит от соединения и табличного движка оригинальной таблицы

PERCONA

Взаимодействие с Сервером: Блокировки

- Сервер
  - Метаданных (MDL) защищают структуру
  - Табличные защищают данные
    - Имплицитно
    - `LOCK TABLES`

PERCONA

Взаимодействие с Сервером: Блокировки

- Сервер
- Табличный движок
  - Строковые
  - Табличные
  - Какие угодно
  - Не коррелируют с серверными

PERCONA

Взаимодействие с Сервером: Блокировки

- Сервер
- Табличный движок
- Возможны неопределяемые "дедлоки"

PERCONA

Как Найти Опции Движка?

- Обычно префиксом `engine_name_`
- У MyISAM есть несколько исключений

PERCONA

Как Найти Инструменты Движка?

- Обычно начинаются с `engine_name`
  - `myisamchk`
  - `myisam_ftdump`
  - `myisampack`

PERCONA

Как Найти Инструменты Движка?

- Обычно начинаются с `engine_name`
- Иногда только часть `engine_name`
  - `innochecksum`
  - `tokuftdump`
  - `tokuft_logprint`

PERCONA

Табличные Движки

InnoDB

PERCONA

InnoDB Вкратце

- Транзакционный
  - Каждый запрос - часть транзакции!
  - Даже `SELECT`
    - `READ ONLY` транзакции в 5.6+

PERCONA

InnoDB Вкратце

- Транзакционный
- Поддерживает MVCC
  - Хранит старые версии до тех пор пока все транзакции, которые могут их запросить, завершили работу

PERCONA

InnoDB Вкратце

- Транзакционный
- Поддерживает MVCC
- ACID
  - Atomicity
  - Consistency
  - Isolation
  - Durability

PERCONA

InnoDB Вкратце

- Транзакционный
- Поддерживает MVCC
- ACID
- Физическое расположение
  - Системный (shared) tablespace
  - Redo log
  - Индивидуальные tablespaces

PERCONA

Структуры InnoDB

- Системный tablespace
  - InnoDB data dictionary
    - Метаданные
  - Doublewrite buffer
    - Хранит страницу после удаления из Buffer Pool и до записи в нужное место
    - Для восстановления после крэша во время записи страницы
  - Change buffer
    - Кэш изменений вторичных индексов
    - Если их нет в Buffer Pool
  - Undo log (rollback segment)
    - Копии данных, изменённых активными транзакциями
    - Могут храниться отдельно в 5.6+

Структуры InnoDB

- Системный tablespace
- Buffer pool
  - В памяти
  - Кэш данных и индексов
  - Устаревшие данные удаляются с применением LRU алгоритма

PERCONA

Структуры InnoDB

- Системный tablespace
- Buffer pool
- Redo log
  - Кодированные изменения страниц
  - Записываются в циклической манере
  - Незавершенные записи в файлы таблиц проигрываются из журнала после крэша

PERCONA

Структуры InnoDB

- Системный tablespace
- Buffer pool
- Redo log
- File-Per-Table Tablespaces
  - Данные и индексы таблицы
  - Поддерживают разные форматы
  - Могут храниться на разных дисках

PERCONA

Конкуренция

- Больше потоков - больше работы может быть выполнено одновременно

PERCONA

Конкуренция

- Больше потоков - больше работы может быть выполнено одновременно
- Поток InnoDB $\neq$ соединение
  - Соединения могут бездействовать пока InnoDB выполняет свою работу
  - Если активных соединений (`Running_threads`) больше, чем потоков InnoDB
    - Часть ждёт в очереди

PERCONA

Конкуренция

- Больше потоков - больше работы может быть выполнено одновременно
- Поток InnoDB $\neq$ соединение
- Ограничено количеством ядер CPU

PERCONA

Конкуренция

- Больше потоков – больше работы может быть выполнено одновременно
- Поток InnoDB $\neq$ соединение
- Ограничено количеством ядер CPU
- Слишком большое количество потоков может привести к конфликтам
  - Производительность уменьшится
  - Сервер зависнет!

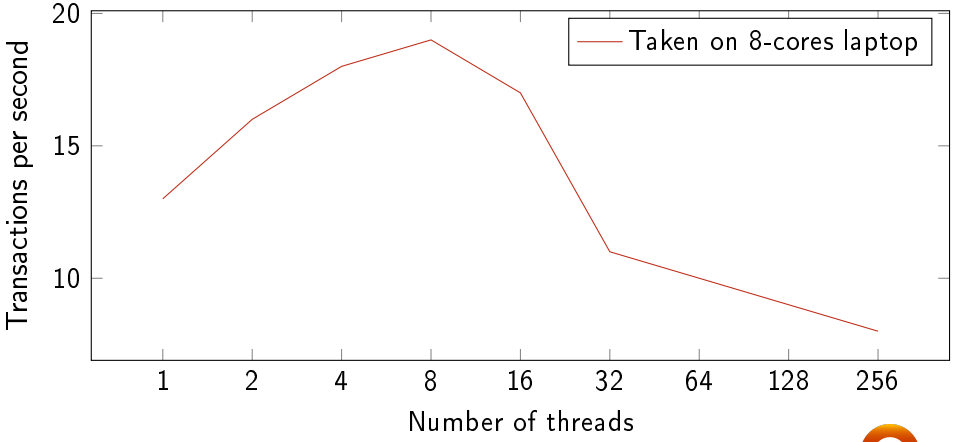
PERCONA

Конкуренция

- Больше потоков - больше работы может быть выполнено одновременно
- Поток InnoDB $\neq$ соединение
- Ограничено количеством ядер CPU
- Слишком большое количество потоков может привести к конфликтам
- `innodb_thread_concurrency`

PERCONA

Неоправданно большое количество потоков



PERCONA

Ввод-Вывод: Производительность

- Меньше записи-чтения
  - Больше производительность
  - Выше риск потери данных

PERCONA

Ввод-Вывод: Производительность

- Меньше записи-чтения
  - Больше производительность
  - Выше риск потери данных
- Не всегда нужно снижать количество операций ввода-вывода
  - Быстрые диски
  - Приложение много пишет

PERCONA

Настройка ввода-вывода для InnoDB

- Обязательные
  - `innodb_flush_method`
  - `innodb_log_file_size`
  - `innodb_io_capacity`

PERCONA

Настройка ввода-вывода для InnoDB

- Обязательные
- С риском потери данных
  - `innodb_doublewrite`
  - `innodb_flush_log_at_trx_commit`
  - `innodb_flush_log_at_timeout`
  - Изменяйте только на репликах!

PERCONA

Топ настроек InnoDB

- innodb_buffer_pool_size
- innodb_log_file_size
- innodb_thread_concurrency
- innodb_io_capacity
- innodb_flush_method

PERCONA

Репликация

PERCONA

Репликация в MySQL

- Начиная с очень ранних версий

PERCONA

Репликация в MySQL

- Начиная с очень ранних версий
- Легко использовать

PERCONA

Репликация в MySQL

- Начиная с очень ранних версий
- Легко использовать
- Минимальные настройки

PERCONA

Как включить

- Мастер
  - `-log-bin`
  - `-server-id`
  - `GRANT REPLICATION SLAVE ON *.* ...`

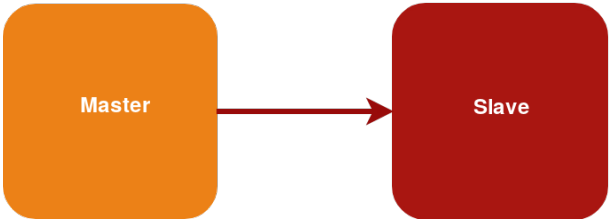
PERCONA

Как включить

- Мастер
  - `-log-bin`
  - `-server-id`
  - `GRANT REPLICATION SLAVE ON *.* ...`
- Слейв
  - `-server-id`
  - `CHANGE MASTER ...`
  - `START SLAVE`

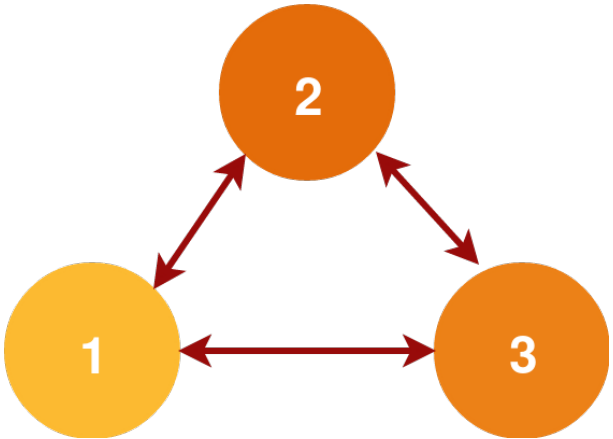
PERCONA

Простая



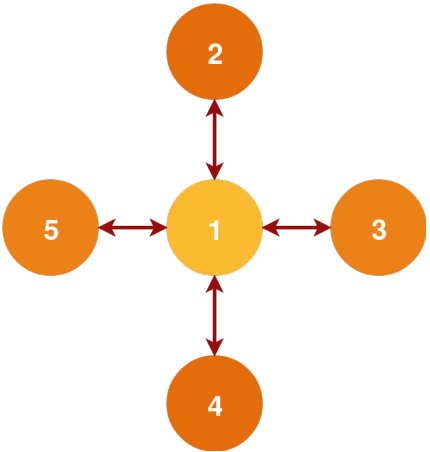
PERCONA

Круговая



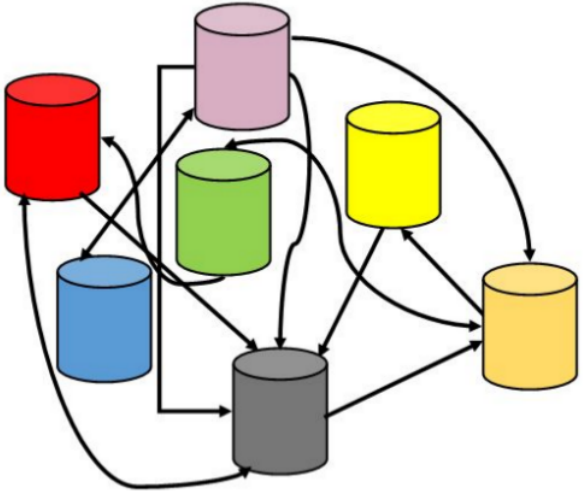
PERCONA

Звезда



PERCONA

Креативная



PERCONA

Асинхронная

Мастер

Слейв

← Инициирует

PERCONA

Асинхронная

Мастер

Слейв

← Инициирует

← Запрашивает пакет

PERCONA

Асинхронная

Мастер	Слейв
	← Инициирует
	← Запрашивает пакет
Пересылает пакет →	

PERCONA

Асинхронная

Мастер	Слейв
	← Инициирует
	← Запрашивает пакет
Пересылает пакет →	
	... ?

PERCONA

Два Типа Потоков Слейва

IO thread

Читает с мастера

SQL thread

PERCONA

Два Типа Потоков Слейва

IO thread

Читает с мастера

Записывает в relay log

SQL thread

PERCONA

Два Типа Потоков Слейва

IO thread

Читает с мастера

Записывает в relay log

SQL thread

← Читает из relay log

PERCONA

Два Типа Потоков Слейва

IO thread

Читает с мастера

Записывает в relay log

SQL thread

← Читает из relay log

Выполняет

PERCONA

Единственный SQL Thread

- Прост в отладке

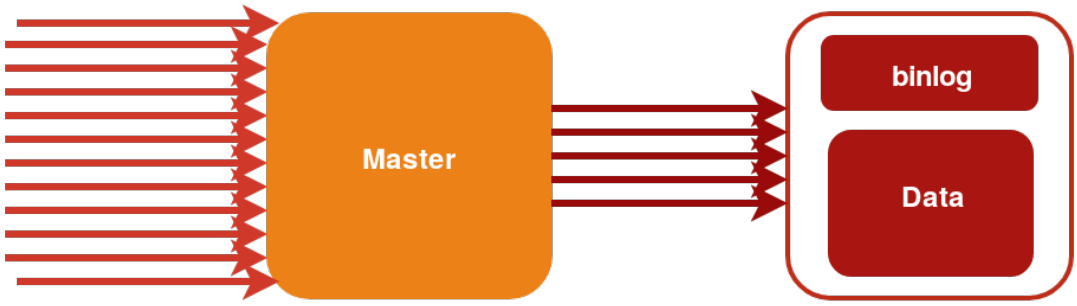
PERCONA

Единственный SQL Thread

- Прост в отладке
- Обычно медленнее мастера
  - Высоконкурентная нагрузка

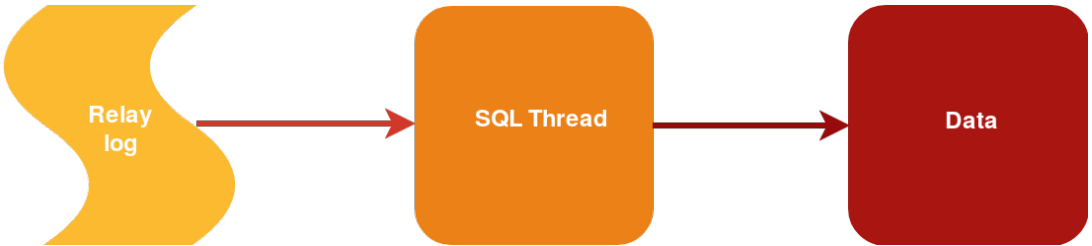
PERCONA

Записи на Мастере



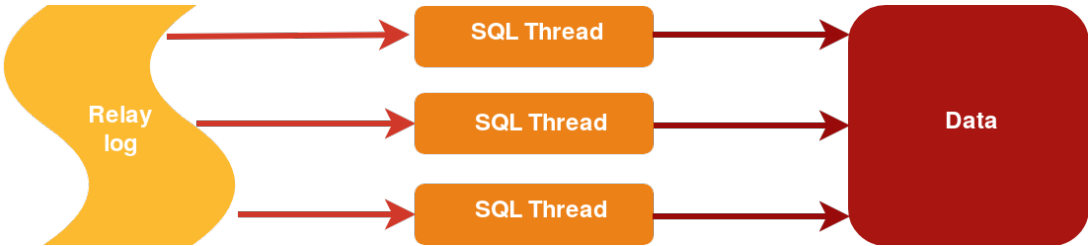
PERCONA

Записи на Слейве: Один SQL Thread



PERCONA

Несколько SQL Threads: 5.6+



PERCONA

Логическая

Master

Получает изменение

Табличный Движок

PERCONA

Логическая

Master

Получает изменение

Передаёт движку $\rightarrow$

Табличный Движок

PERCONA

Логическая

Master

Получает изменение

Передаёт движку $\rightarrow$

Табличный Движок

Пишет в таблицу

PERCONA

Логическая

Master

Получает изменение

Передаёт движку $\rightarrow$

Табличный Движок

Пишет в таблицу

$\leftarrow$ Возвращает контроль

PERCONA

Логическая

Master	Табличный Движок
Получает изменение	
Передаёт движку →	
	Пишет в таблицу
	← Возвращает контроль
Пишет в binary log	

PERCONA

Логическая

Master	Табличный Движок
Получает изменение	
Передаёт движку →	
	Пишет в таблицу
	← Возвращает контроль
Пишет в binary log	
Синхронизируется →	← Синхронизируется

PERCONA

Statement-Based Binary Log Format

Клиент

Binary log

PERCONA

Statement-Based Binary Log Format

Клиент

INSERT INTO ... $\rightarrow$

Binary log

PERCONA

Statement-Based Binary Log Format

Клиент

INSERT INTO ... $\rightarrow$

Binary log

SET TIMESTAMP...

PERCONA

Statement-Based Binary Log Format

Клиент

INSERT INTO ... →

Binary log

SET TIMESTAMP...
SET sql_mode...

PERCONA

Statement-Based Binary Log Format

Клиент

INSERT INTO ... →

Binary log

SET TIMESTAMP...
SET sql_mode...
INSERT INTO ...

PERCONA

Row-Based Binary Log Format

Клиент

Binary log

PERCONA

Row-Based Binary Log Format

Клиент

UPDATE ... $\rightarrow$

Binary log

PERCONA

Row-Based Binary Log Format

Клиент

UPDATE ... $\rightarrow$

Binary log

SET TIMESTAMP...

PERCONA

Row-Based Binary Log Format

Клиент

UPDATE ... $\rightarrow$

Binary log

SET TIMESTAMP...
SET sql_mode...

PERCONA

Row-Based Binary Log Format

Клиент

UPDATE ... $\rightarrow$

Binary log

SET TIMESTAMP...
SET sql_mode...
Строка до изменений

PERCONA

Row-Based Binary Log Format

Клиент

UPDATE ... $\rightarrow$

Binary log

SET TIMESTAMP...
SET sql_mode...
Строка до изменений
Строка с изменениями

PERCONA

Позиционная

- Нужно указать
  - Название binary log
  - Позицию

PERCONA

Позиционная

- Нужно указать
  - Название binary log
  - Позицию
- С точки зрения отладки
  - Выполнение с указанной позиции

PERCONA

Позиционная

- Нужно указать
  - Название binary log
  - Позицию
- С точки зрения отладки
  - Выполнение с указанной позиции
  - Легко пропустить

PERCONA

Позиционная

- Нужно указать
  - Название binary log
  - Позицию
- С точки зрения отладки
  - Выполнение с указанной позиции
  - Легко пропустить
  - Легко переместить назад

PERCONA

Позиционная

- Нужно указать
  - Название binary log
  - Позицию
- С точки зрения отладки
  - Выполнение с указанной позиции
  - Легко пропустить
  - Легко переместить назад
  - Все конфликты - ваши

PERCONA

Global Transaction Identifiers (GTID)

- У каждой транзакции свой номер: GTID

PERCONA

Global Transaction Identifiers (GTID)

- У каждой транзакции свой номер: GTID
- Название binary log и позиция не нужны
  - `AUTO_POSITION=1`

PERCONA

Global Transaction Identifiers (GTID)

- У каждой транзакции свой номер: GTID
- Название binary log и позиция не нужны
- Сложно пропустить ошибочные записи

PERCONA

Global Transaction Identifiers (GTID)

- У каждой транзакции свой номер: GTID
- Название binary log и позиция не нужны
- Сложно пропустить ошибочные записи
- Не все команды разрешены
  - Транзакционные и нетранзакционные движки
  - `CREATE TABLE ... SELECT`
  - Временные таблицы
    - 8.0.13+: разрешены при RBR

PERCONA

Табличные Движки и Репликация

- Синхронизация binary log и таблицы

PERCONA

Табличные Движки и Репликация

- Синхронизация binary log и таблицы
- SBR и transaction isolation levels

```
mysql> set transaction isolation level read committed;
Query OK, 0 rows affected (0,00 sec)

mysql> update t1 set f1=f1*2;
ERROR 1665 (HY000): Cannot execute statement: impossible to write to binary log since
BINLOG_FORMAT = STATEMENT and at least one table uses a storage engine limited to
row-based logging. InnoDB is limited to row-logging when transaction isolation level
is READ COMMITTED or READ UNCOMMITTED.
```

PERCONA

Табличные Движки и Репликация

- Синхронизация binary log и таблицы
- SBR и transaction isolation levels
- RBR - независимый от движка формат

PERCONA

Табличные Движки и Репликация

- Синхронизация binary log и таблицы
- SBR и transaction isolation levels
- RBR - независимый от движка формат
- GTIDs независимы от движка

PERCONA

Табличные Движки и Репликация

- Синхронизация binary log и таблицы
- SBR и transaction isolation levels
- RBR - независимый от движка формат
- GTIDs независимы от движка
- У движка могут быть свои специфические фичи для репликации
  - Read free репликация в TokuDB

PERCONA

NoSQL

Что Поддерживает MySQL

- Функции JSON
- MySQL Shell
- Document Store
- Memcached API

PERCONA

Функции JSON

- Манипулируют документами JSON
  - Создают
    - `JSON_OBJECT`
    - `JSON_ARRAY`
  - Ищут
    - `JSON_EXTRACT`
    - `column->path`
  - Модифицируют
    - `JSON_INSERT`
    - `JSON_SET`

PERCONA

Функции JSON

- Манипулируют документами JSON
- Тип данных JSON
  - Оптимизированное хранилище
  - Автоматическая валидация

PERCONA

Функции JSON

- Манипулируют документами JSON
- Тип данных JSON
- Может работать со строками
  - Медленно
  - Автоматическая валидация не поддерживается

PERCONA

MySQL Shell

- Командный клиент
- Поддерживает Python и JavaScript

PERCONA

MySQL как Document Store

- Доступ из MySQL Shell
- Поддерживает
  - Python, JavaScript, C#, C++, Java
  - Your driver
  - X Plugin: асинхронный клиент
    - X Protocol
    - X DevAPi
- Данные хранятся в реляционном формате

PERCONA

Memcached API

- Интерфейс к memcached
- Доступ ключ-значение
- Данные хранятся в InnoDB
- Могут быть кэшированы
  - Кэш может быть настроен для высокой производительности
  - Умолчания: только прямой доступ, высокая надёжности, фактически никакого кэша

Дополнительная Информация

- Вебинары

Introduction to MySQL Troubleshooting

Storage engines troubleshooting

InnoDB Troubleshooting

MySQL Replication Troubleshooting

PERCONA

Дополнительная Информация

- Мануалы

The InnoDB Storage Engine

Replication

JSON Functions

MySQL as a Document Store

InnoDB memcached Plugin

PERCONA

Спасибо!


www.slideshare.net/SvetaSmirnova


twitter.com/svetsmirnova


github.com/svetasmirnova

PERCONA

DATABASE PERFORMANCE
MATTERS