



PERCONA  
Performance Consulting Experts

---

# Innodb – Масштабируемость И НОВЫЕ ВОЗМОЖНОСТИ

Peter Zaitsev

Percona Inc

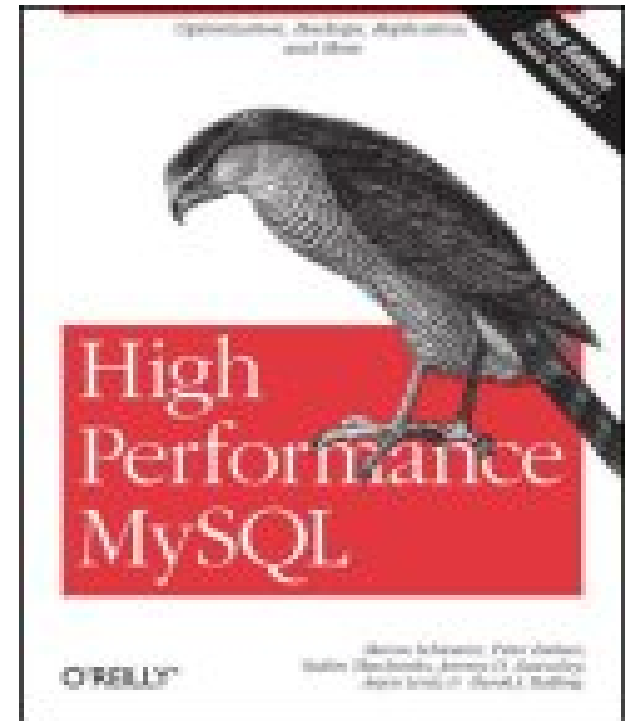
HighLoad.RU 2008

Moscow, Russia

Oct 6-7, 2008

# Немного о Докладчике

- Основатель Persona Inc
  - Оптимизация производительности, масштабируемость, надежность систем на MySQL
- Основатель <http://www.mysqlperformanceblog>
- Co-Автор “High Performance MySQL Second Edition



# О чем эта презентация ?

- InnoDB – наиболее популярная система хранения в MySQL
- Которая к сожалению не слишком хорошо масштабируется
  - Рассмотрим проблемы и их решения
- Посмотрим на другие аспекты производительности InnoDB
- Фокус – существующий код MySQL 5.0 и 5.1
- А так же сторонние данные.

# Масштабируемость

- В широком смысле – масштабируемость приложения
- Задача – обеспечение нужной производительности за разумную стоимость
  - Есть так же требования по надежности безопасности доступности итд
- “Рост” приложения – размер базы данных чисто запросов, часто их сложность.

# Вопрос выживания или денег ?

- Многие системы спроектированы используя один сервер
  - Часто не за какие деньги невозможно купить сервер с характеристиками нужными для роста
  - Максимальное использование ресурсов сервера вопрос выживания
- Другие системы могут использовать произвольное число серверов с произвольной производительностью
  - Ее максимизация – вопрос эффективности

# Масштабируемость роста

- Как ведет себя производительность при росте объема данных ?
  - Зачастую вопрос архитектуры/схемы баз данных
  - Поведение координально меняется когда данные перестают влезать в память
- Как ведет себя производительность при росте сложности запросов ?
  - Понимать и контролировать сложность
  - Параллельное выполнение
  - Регенерация данных итд

# Масштабируемость железа

- Вопрос насколько большое приложение должно стать до того как потребуются использование многих серверов
- Вопрос эффективного использования
  - Всегда есть конфигурация с оптимальной ценой производительностью
  - Важна возможность ее эффективного использования
  - Последние годы число ядер процессора в доступных системах резко увеличилось и проблемы затрагивают все более широкие слои

# Тестирование Микро-Тестами

- Микро-Тесты Производительности – простые операции
- Тестирование какого-то определенного аспекта поведения
- Проблемы найденные при них проявляются и в реальных приложениях
  - Однако они не раскрывают все проблемы разумеется
- Часто могут приувеличивать проблему
  - Может проявляться в меньшей степени в реальных приложениях

# Не забудьте об обслуживании

- Часто забывают о требовании производительности обслуживания
- Очень важно для реально используемых приложений
- Обычно связаны с размером данных и потреблением ресурсов

# Работа с большими объемами

- Сложны в обращении !
- Резервное Копирование обязательно физическое
- В Innodb таблицы нельзя перемешать между серверами
- Нет возможности REPAIR TABLE
  - При повреждениях часто приходится загружать таблицу из дампа, часто быстрее восстановить все из бакапа
- **ALTER TABLE** очень медленная
  - Master-Master репликация может помочь
- Обслуживание таблиц **OPTIMIZE TABLE**

# Быстрое создание индексов

- MySQL 5.1 Innodb плагин от Oracle
  - Так же включен в MySQL 5.1 от Percona
- Innodb может создавать индексы без перестройки всей таблицы
  - И что важно с помощью сортировки
- Загрузка в 10 раз быстрее и более компактные индексы (специальный метод загрузки)

Метод загрузки	Время	Размер данных	Размер индекса
SQL Dump	11m	1333788672	1867013806
LOAD INFILE	90m	1333788672	1867013806
ALTER from MYISAM	90m	1333788672	1867013806
LOAD + ADD INDEX	7m+0m	1333788672	1124.73472
SQL Dump MyISAM	7m	1000000000	312079.72

# Как сортировка влияет на индекс

- Индекс построенный с помощью сортировки физически сортирован и имеет большее заполнение страниц
- Полное Сканирование индекса (с диска)
  - **31 sec** стандартный и **22 sec** сортированный
    - Процессор становится ограничивающим фактором
- Обновление ндекса:
  - **update sample set c=md5(i) where i%1000=1;**
    - **3 min 20 sec** стандартаны и **8 min 16 sec** сортированный
  - Рост размера индекса
    - **0%** (стандартный) и **30%** (сортированный)

# Сжатие Данных

- Innodb расширение также имеет ф-ю сжатия данных
  - Постраничное сжатие без ограничение на обновление
  - Дополнительно большие BLOB/TEXT поля сжимаются келиком
  - Много продвинутых технологий позволяющих делать уделение и ряд обновлений без повторного сжатия
  - Балансировка сжатых и расжатых страниц в кэше
  - Несколько сложно в использовании
    - “Угадай какое сжатие будет наиболее эффективно”

# Много Таблиц

- Много маленьких таблиц часто может использоваться вместо одной большой
  - InnoDB держит мета данные о всех таблиц к которым было обращение в памяти что может занять много памяти. Не такая большая проблема для 64bit платформ
  - **innodb\_file\_per\_table=1**
    - Увеличивает потребление место маленькими таблицами
    - Если таблиц очень много то восстановление после сбоя может быть очень долгим
  - “Разогрев” существенная таблица
    - Открытие строго сериализовано (MySQL 5.0)
    - И медленное так как происходит обновление статистик

# Работа с большим buffer pool

- Обычно чем больше buffer pool тем лучше
- Расчитывайте на “Разогрев”
  - Чем больше buffer pool тем он дольше
  - 32GB будет заполняться 1 час
    - При скорости чтения 600 страниц в сек
- Checkpoint активность может приводить к большим провалом производительности
- Иногда восстановление после сбоя происходит существенно медленнее с большим объемом кэша

# Работа с большим Buffer pool

- Корректная остановка сервера занимает дольше времени
  - Все модифицированные страницы Buffer Pool должны быть сохранены на диск
  - Установите **innodb\_max\_dirty\_pages\_pct=0** заранее чтобы почти все страницы были агрессивно сброшены.

# Репликация

- Репликация отстает куда быстрее чем мастер или слейв полностью загружены
  - Все большая проблема с многоядерными проц-ами.
  - А так же когда нет VBU кэша в RAID на слейве
- Часто ограничено скоростью транзакций
  - установите **innodb\_flush\_log\_at\_trx\_commit=2**
    - Репликация то все равно асинхронно
- Ограничена скоростью собственно обновлений
  - Диск - префетчинг может помочь (кэш)
  - Процессор – смотрите row level replication в 5.1
  - Оптимизация траффика репликации

# А теперь бенчмаки

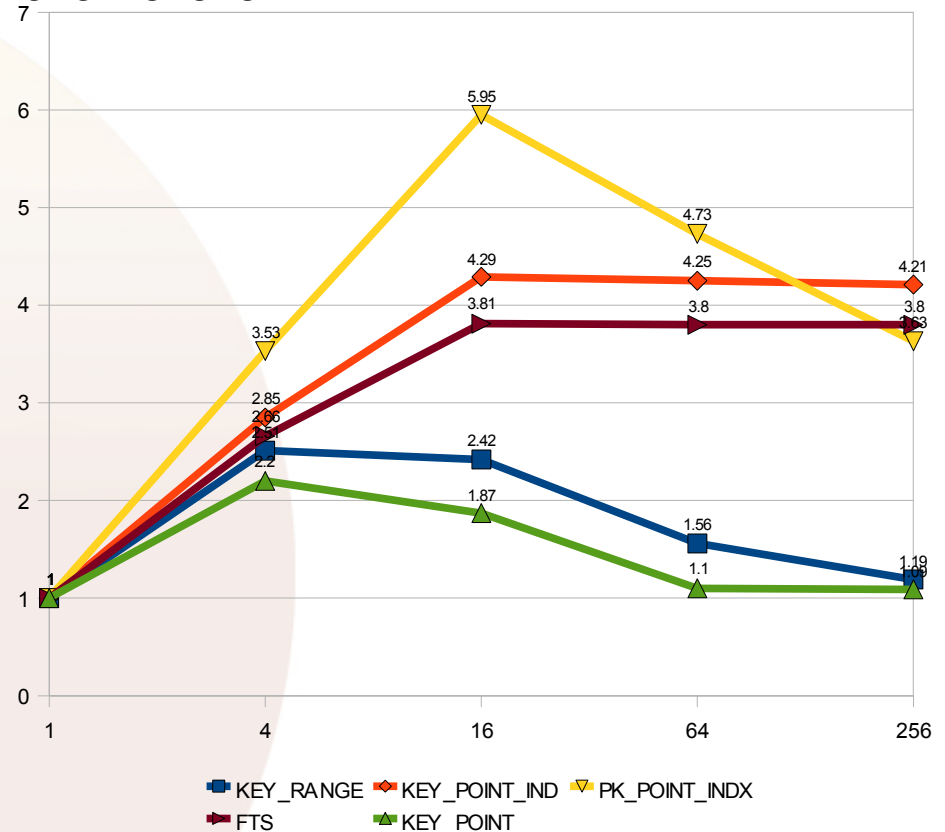
---

Относитесь к результатам критично

# Масштабируемость 5.0

- MySQL 5.0.51a
- Dell PE 2950
- 2\* Quad Core CPUs
  - Intel Xeon L5335
- Нет Дискового ввода вывода
- Масштабируемость сильно зависит от нагрузки

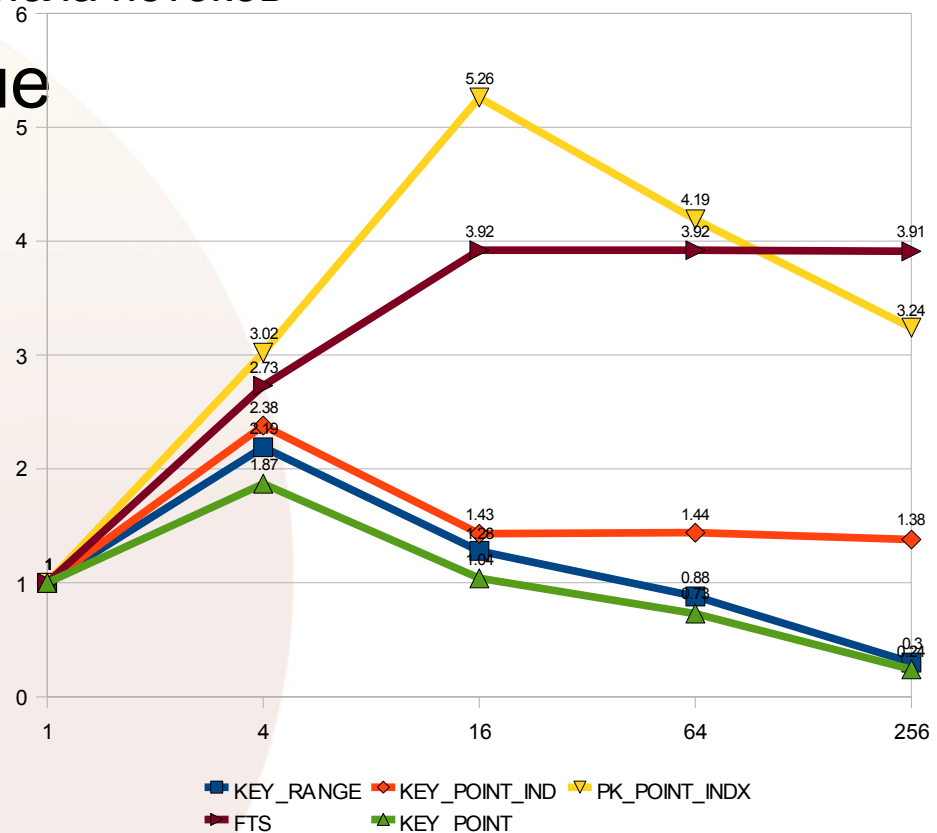
- Фактор масштабирования относительно числа потоков



# Масштабируемость 5.1

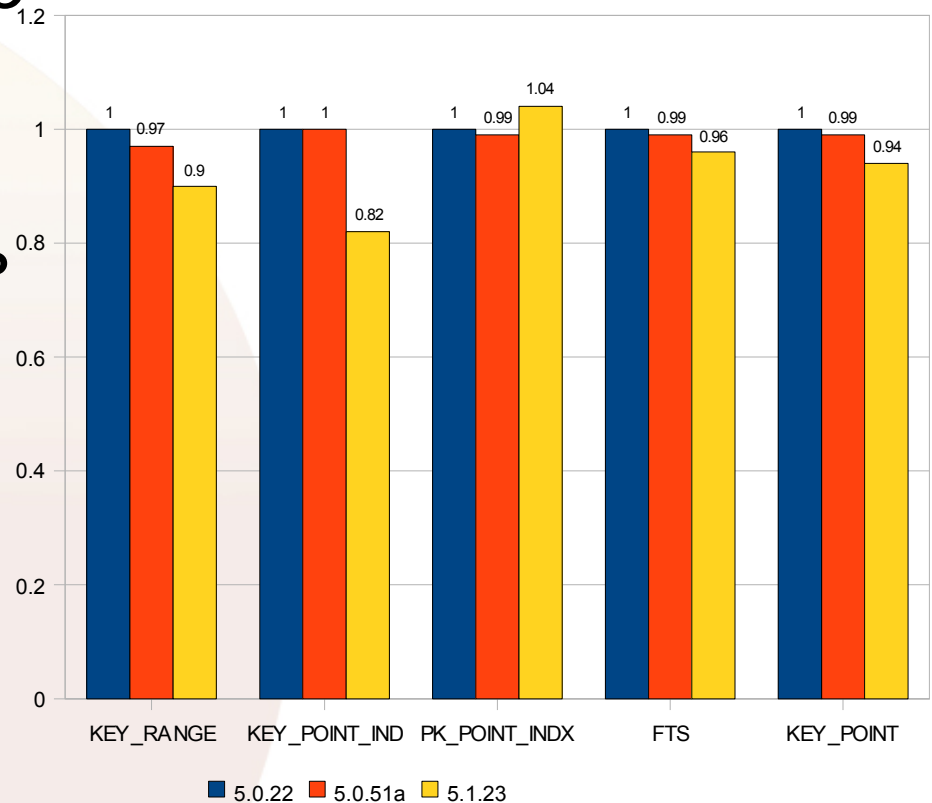
- MySQL 5.1.23-rc
- Все то же самое кроме версии
- Видно большое падение производительности – Вроде исправлено в 5.1.28

- Фактор масштабируемости относительно числа потоков



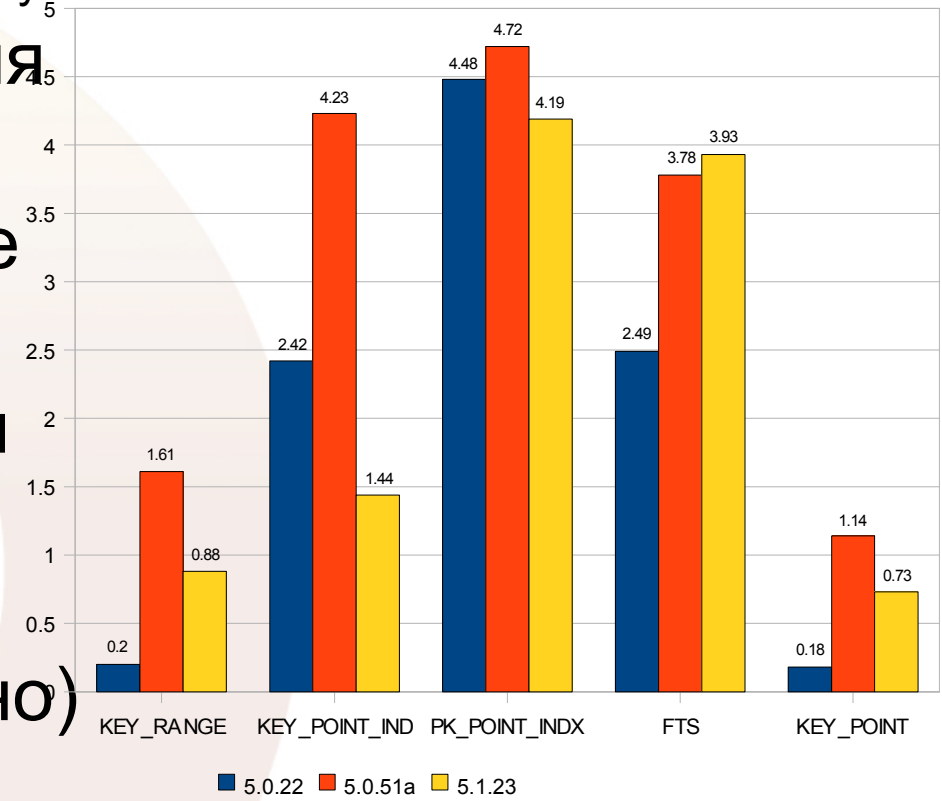
# Пр-сть одного потока

- Относительная производительность разных версий
- Сложно судить только по фактору масштабирования
- Показываем скорость относительно 5.0.22
- 5.0.22 и 5.0.51 очень близки
- 5.1 показывает некоторую потерю производительности



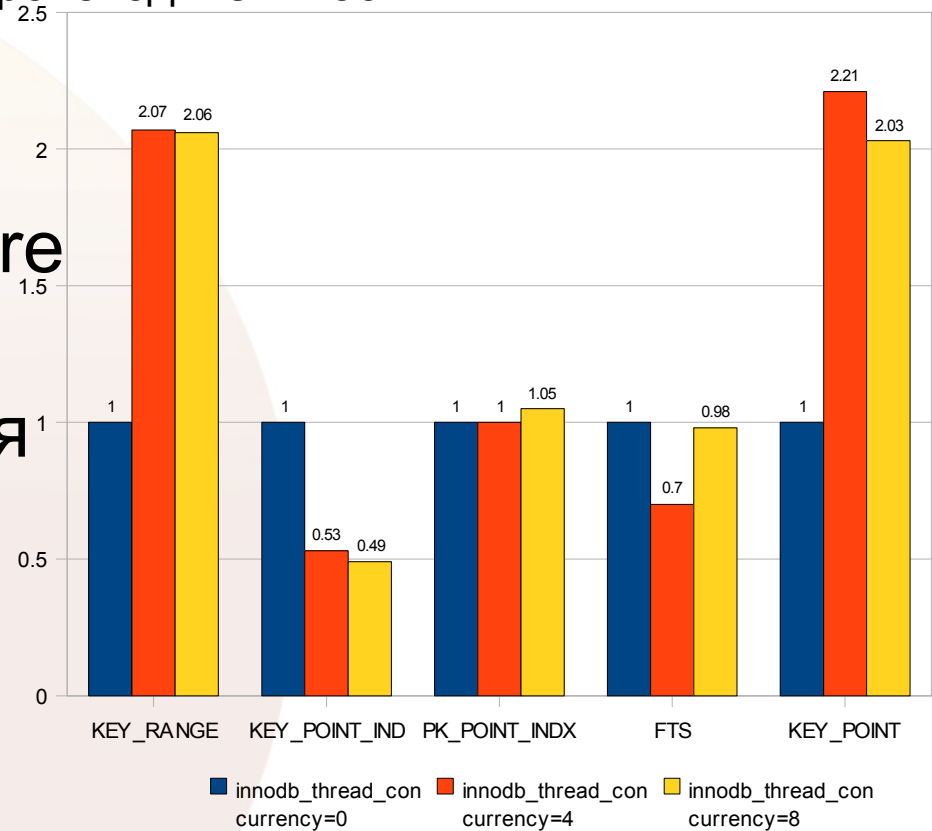
# M-ость разных версий

- Фактор масштабирования для 64 потоков
  - MySQL 5.0.51 лучшие результаты
  - Заметные улучшения относительно 5.0.22
  - 5.1 медленнее (возможно исправлено)
- Фактор масштабирования разных версий MySQL



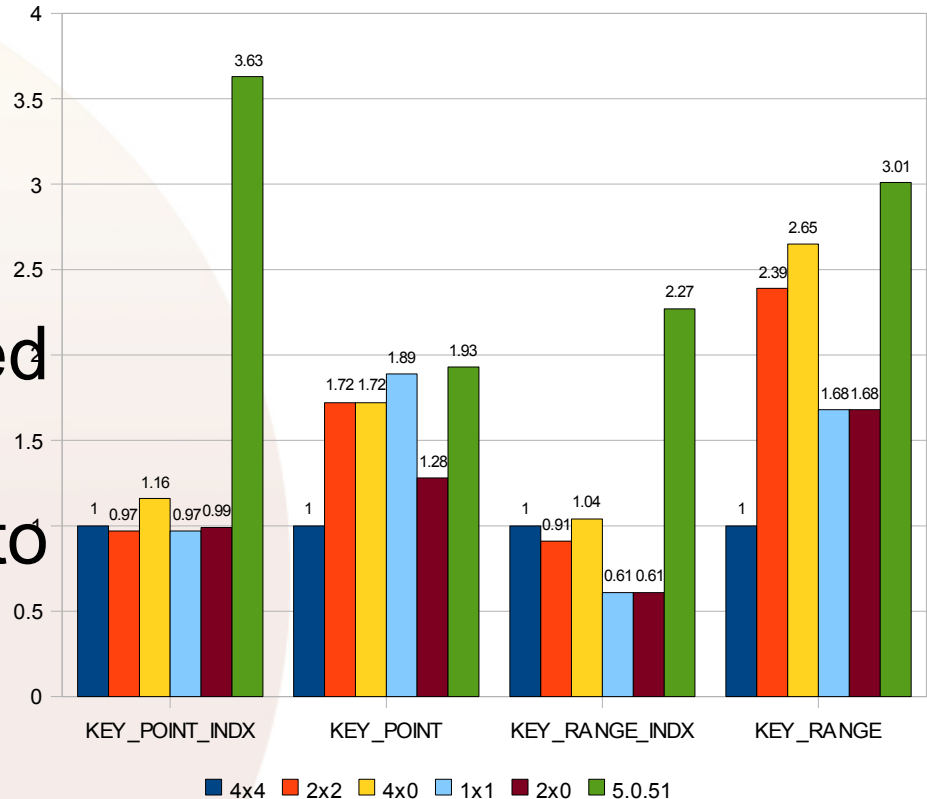
# innodb\_thread\_concurrency

- Производительность для 64 потоков
  - MySQL 5.1.23  
innodb\_thread\_concurrency=0 - база
  - Нет лучшего значения для всех типов нагрузки
- Как innodb\_thread\_concurrency влияет на производительность



# Fixing scaling by CPU Affinity

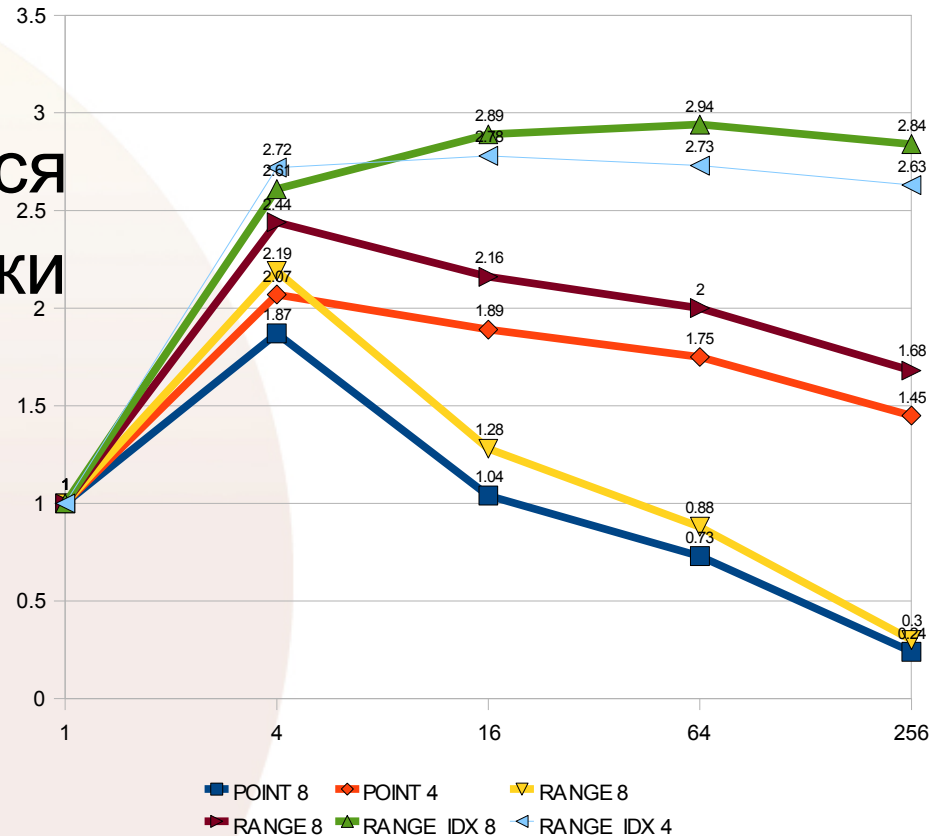
- Performance for 16 threads
  - MySQL 5.1.23
    - 5.0.51a for comparison
  - Workloads which scaled worse on 5.1.23
  - Trying to bind MySQL to specific CPU Cores
  - Restricting can help scaling
- How binding to CPUs affects performance



# Quadcore vs Dual Core

- MySQL 5.1.23rc
- Нагрузки которые плохо масштабируются
- 2 из 3х типов нагрузки лучше ведут себя на 2\*Dual Core системе

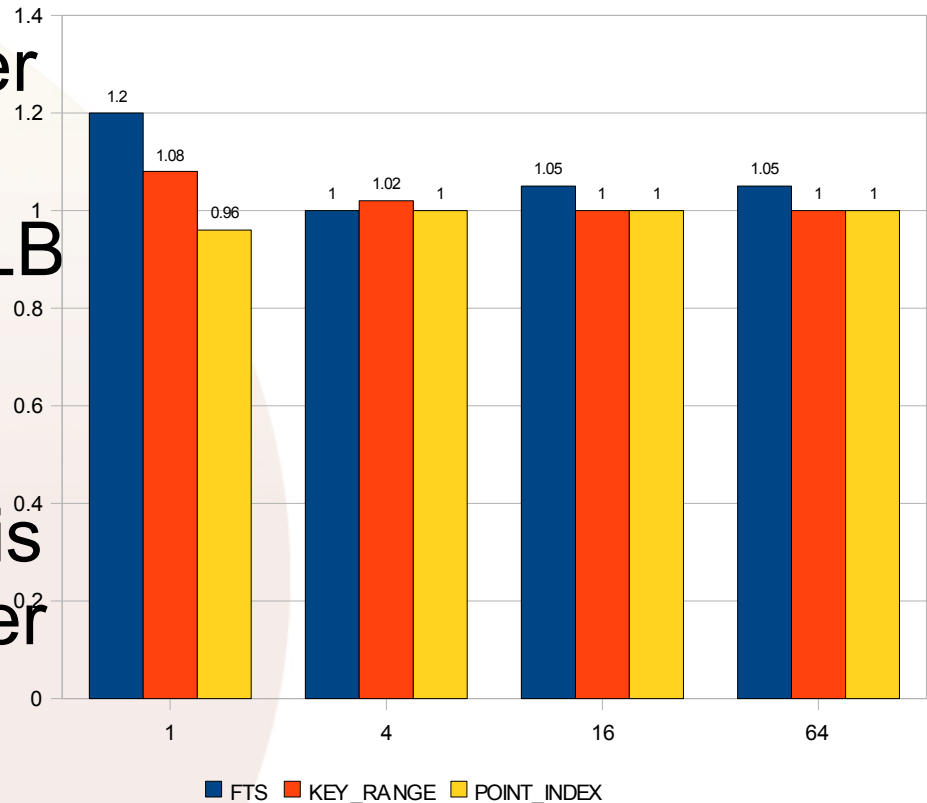
- Масштабируемость и число ядер



# Large Pages

- MySQL can use Large Pages for Innodb Buffer Pool
- Huge Pages reduce TLB cache misses
- Non Swapable
- Mediocre results for this workload, may be better in case of skewed working set

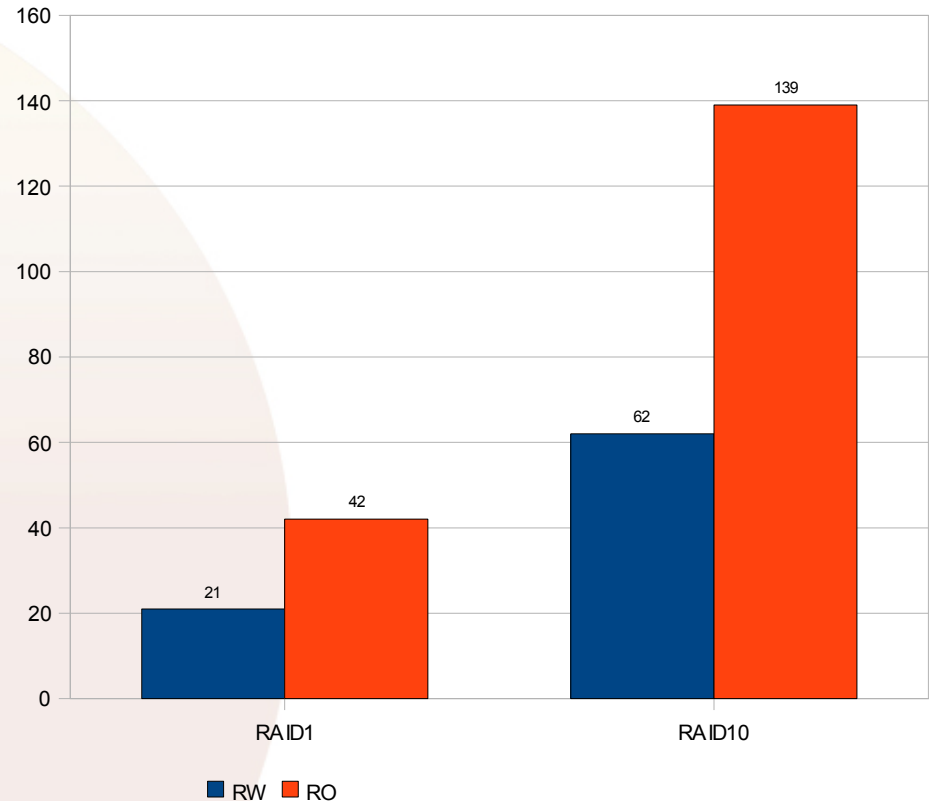
- Performance effect of using Large Pages



# Innodb – Масштабируемость IO

- Dell PowerEdge 2950, Perc6, CentOS 5.0
- RAID1 и RAID10 (6 disk)
- SysBench тест

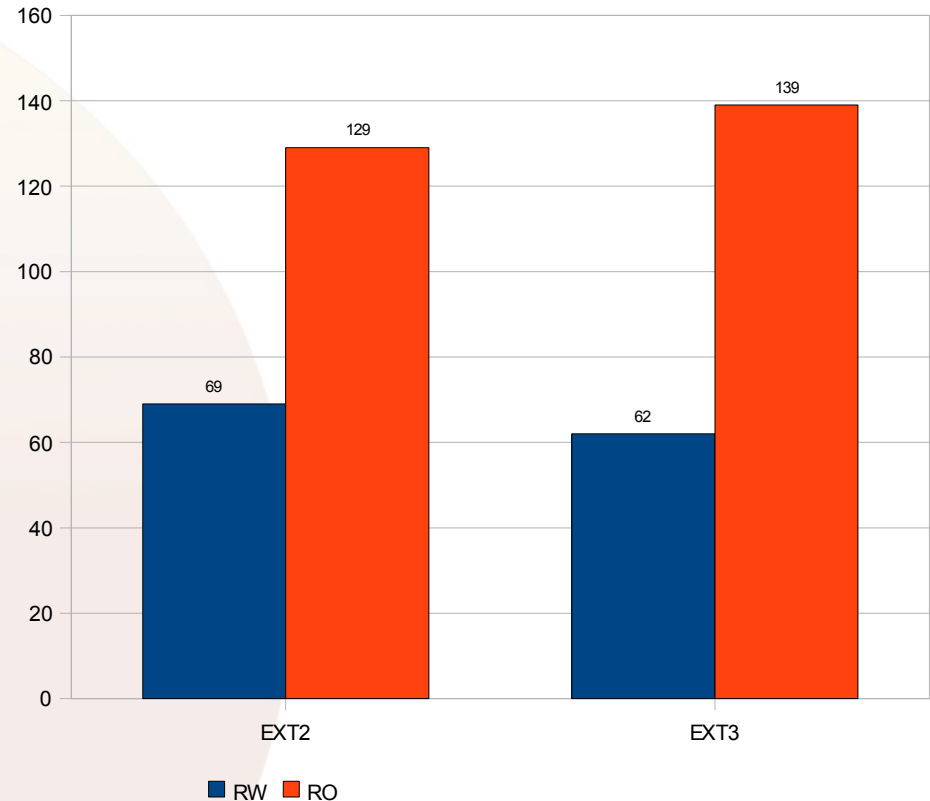
- Масштабируемость RAID



# EXT3 и EXT2

- То же самое железо
- RAID10
- Ext3 хуже на запись (оверхед на журналирование) но несколько лучше на чтение.

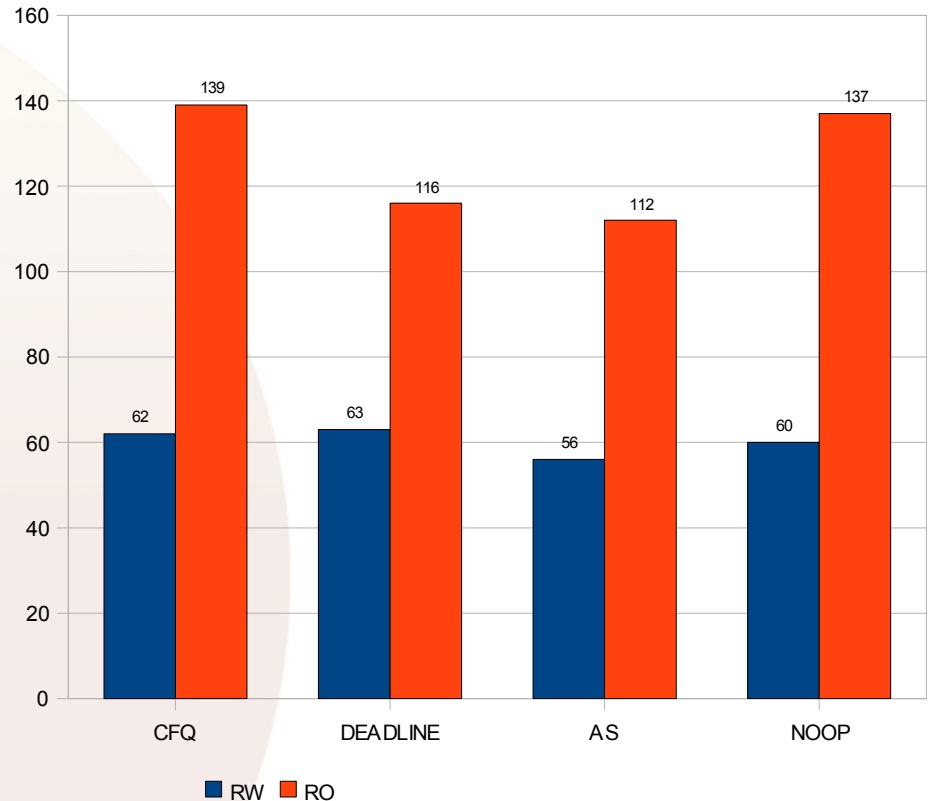
Производительность Файловых систем



# Linux планировщики I/O

- Улучшились за последние годы
  - Разница не такая большая как несколько лет назад
- CFQ (стандартный) дает лучшую производительность в этом случае

- Performance effect of using Large Pages



# Будущее Innodb

- Oracle продолжает вести работы
  - В страшной секретности и не раскрывает планов
- Community берет развитие в свои руки
  - Google (Mark Callaghan) выпустили патчи улучшающие производительность на ряде операций
    - На некоторых других производительность падает
- Yasufumi – множество патчей улучшающих масштабируемость
- Percona - Патчи/релизы
  - Улучшение производительности
  - Анализ производительности

# Спасибо что пришли !

- Время для вопросов
- Пишите
  - [pz@percona.com](mailto:pz@percona.com)
- Приходите к нам
  - За информацией: <http://www.mysqlperformanceblog.com>
  - За помощью: <http://www.percona.com>