O'REILLY

OSCON™
Open Source Convention

Обзор транзакционных Storage Engines с открытыми исходниками

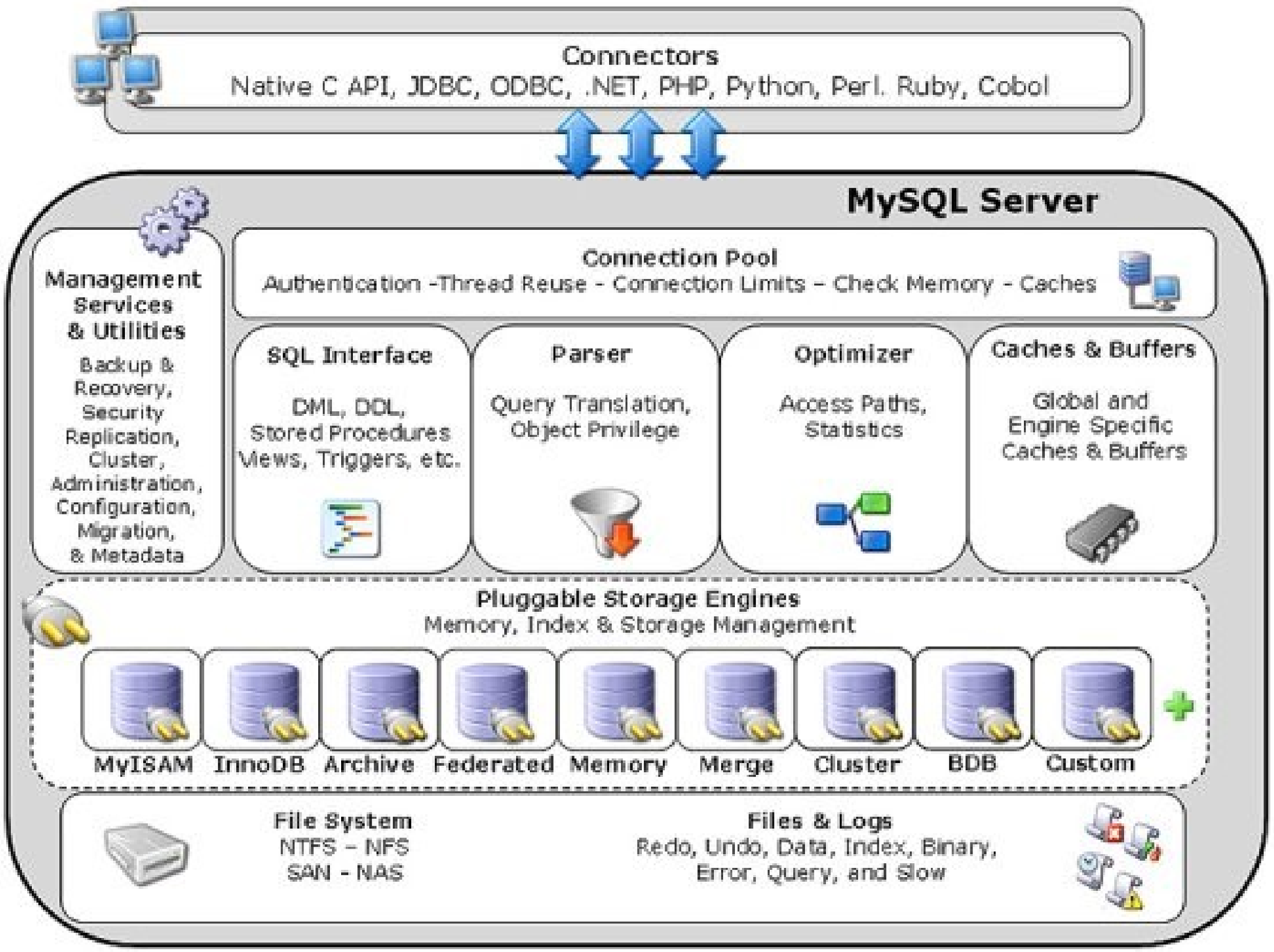
Петр Зайцев
Вадим Ткаченко

http://MySQLPerformanceBlog.com

О нас

- Основатели Percona Ltd.
  - Консультанты по производительности MySQL
- Авторы http://www.MySQLPerformanceBlog.com
  - Несколько лет работы в MySQL AB
    - Петр - руководитель "High Performance Group", Вадим – его правая рука
    - С давних пор пользователи MySQL для множества своих проектов

Архитектура MySQL plugin-ов



Транзакционные движки MySQL

- **BDB** - старый Storage Engine, удален в 5.1. Не тестировали
- **InnoDB** - "Наиболее популярный" (Единственный широко распространенный) Storage Engine от Innobase Oy.
- **SolidDB** - Storage Engine от Solid Information Technology
- **PBXT** - Storage Engine от SNAP Innovation (Paul McCullagh)
- **Falcon** - новый Storage Engine от MySQL AB, глава проекта Jim Starkey
- **NDB** - MySQL Cluster - это совсем другая история, он здесь не рассматривается

InnoDB

- http://www.innodb.com/
- Вполне готовый продукт. Разрабатывается Heikki Tuuri более 10 лет.
- Heikki искал пути улучшить традиционную производительность баз данных
- Приобретен Ораклом в конце 2005 года
- Единственный транзакционный storage engine, доступный в официальной версии MySQL 5.0

solidDB

- http://www.solidtech.com/solidDBforMySQL/
- Открыл исходники в 2006 году
- Существующий Storage Engine был интегрирован в MySQL
- Основное внимание надежности и многопроцессорной масштабируемости
- В настоящий момент доступен как готовый продукт.

PrimeBase XT (PBXT)

- http://www.primebase.com/xt/
- Разрабатывается в основном Paul McCullagh-ом с 2005 года
- Не портирован под MySQL, а написан изначально для работы с ним
- Много необычных архитектурных решений
- Только на 50% транзакционный
- Основное внимание – эффективной поддержке BLOB
- http://www.blobstreaming.org/

Falcon

- http://dev.mysql.com/doc/falcon/en/index.html
- Основан на движке "Netfrastructure", автор Jim Starkey
- Куплен MySQL AB в начале 2006
- "Облегченная архитектура"
- Основное внимание транзакционным потребностям Web-приложений, эффективное использование больших объемов памяти

O'REILLY

OSCON™
Open Source Convention

Архитектура и поведение

Архитектура InnoDB

- MVCC и очень эффективные блокировки на уровне строк
- Кластеризация по первичному ключу, запись в ту же страницу
- Дополнительные индексы без сжатия, с информацией о транзакции
- Один общий tablespace или tablespace для каждой таблицы
- Пессимистическая блокировка
- Незамедлительное определение Deadbck
- Гибкий Checkpointing
- "Двойная запись" для защиты от частичной записи страницы

InnoDB

- Определение `DEADLOCK`

- ```
  Session 1: BEGIN;
  Session 2: BEGIN;
  Session 1: UPDATE test SET name='random1-1' WHERE id=1;
  Session 2: UPDATE test SET name='random2-1' WHERE id=2;
  Session 1: UPDATE test SET name='random1-2' WHERE id=2;
  Session 2: UPDATE test SET name='random2-2' WHERE id=1;
  ```

- `InnoDB` определил `deadbck (Error 1213)` во второй сессии **незамедлительно**

- Пессимистическая блокировка :

- `UPDATE` одной и той же строки в двух транзакциях – вторая транзакция ждет `COMMIT/ROLLBACK` первой


  ```

Преимущества InnoDB

- Мощный MVCC
- Хорошая производительность на самых различных нагрузках
- Высокая стабильность
- Отличная защита данных
- Кластеризация по первичному ключу дает большие возможности для оптимизации
- Информация о транзакции в дополнительных индексах допускает быстрое сканирование только по индексу
- Используется адаптивный Hash-индекс и другие продвинутые технологии

Недостатки InnoDB

- Разработка в последние годы замедлилась
- Все еще проблемы с масштабируемостью на несколько процессоров
- По-прежнему не масштабитруется Auto-Increment и не работает Group Commit
- Требуется больше места для хранения данных, особенно для дополнительных индексов
  (После сравнения оказалось, что не намного больше)
- Недоработана интеграция с MySQL
  - Что нужно сделать, чтобы узнать сколько свободного места в Innodb tablespace ?

Архитектура SolidDB

- MVCC блокировка на уровне строк
- Кластеризация по первичному ключу
- Новые данные сохраняются на новых страницах
- для мультиверсионности используется "BonsaiTree"
- ОПТИМИСТИЧЕСКАЯ и ПЕССИМИСТИЧЕСКАЯ блокировки указываются на уровне таблицы
- Online Backup (не пригодный при создании Slave)
- Скоро должна появиться High Available-синхронная репликация

solidDB – пессимистическая блокировка

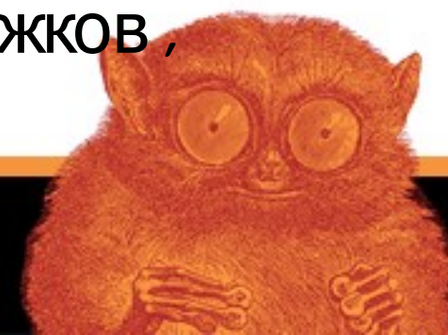
- DEADLOCK - DEADLOCK определился в первой сессии после 20 секунд ожидания
  - deadbck по таймауту
- UPDATE одной и той же строки – вторая сессия ждет окончания первой

solidDB – оптимистическая блокировка

- DEADLOCK - DEADLOCK определился во второй сессии немедленно, но произошла ошибка 1205 – Lock wait timeout exceeded
- UPDATE одной и той же строки:

```
SESSION 1: BEGIN;
SESSION 2: BEGIN;
SESSION 1: UPDATE test SET name = 'md' WHERE id=2;
SESSION 2: UPDATE test SET name = 'md' WHERE id=2;
```

- Во второй сессии мы получили:
  ERROR 1205 (HY000): Lock wait timeout exceeded; try restarting transaction
- Это нормальное поведение для оптимистических движков, но может вызвать проблемы в Web-приложениях.



Преимущества и недостатки SolidDB

- Ограниченно используется на практике
- Из всех рассмотренных `storage engines` по архитектуре наиболее близок к `Innodb`
- Выбор между оптимистической и пессимистической блокировкой удобен для некоторых приложений
- Нет немедленного определения `deadbck`
- Пока что возможно скачать только как отдельную инсталляцию (даже не как `plugin`)

Архитектура PBXT

- MVCC блокировка на уровне строк
- транзакции в пределах одной базы данных
- нет реальной durability, слабый crash recovery
- оптимистическая блокировка
- однократная запись в данные, последовательная запись в лог
- никогда не происходит update данных
- кеш данных + кеш индекса
- эффективная поддержка BLOB

PBXT

- `DEADLOCK` произошел во второй сессии, ошибка 1213
- одновременный `UPDATE` одной и той же строки – оптимистическая блокировка, во второй сессии получаем ошибку:

```
ERROR 1020 (HY000): Record has changed since last
read in table 'test2'
```

Преимущества и недостатки PBXT

- До сих пор не популярен для реального использования (мы пытались, но обнаружили слишком много багов)
- Очень хорошая производительность для некоторых задач
- Эффективно организовано хранилище данных, сравнимо с MyISAM
- Ориентирован на эффективную поддержку BLOB, имеет дополнительные возможности типа Bbb Streaming
- По-прежнему в основном разрабатывается одним человеком
- Большой список ToDo, многие возможности не завершены, включая Recovery
- Потенциально большие накладные расходы при применении процедуры удаления старых записей

Архитектура Falcon

- MVCC, блокировка на уровне строк (на практике, а не в теории)
- пессимистическая блокировка
- нет кластеринга по первичному ключу
- кеш строк (кешируются только нужные вам строки)
- "оптимальный" просмотр индекса
- "сжатие данных" - Nulls, пустые строки
- из-за структуры индекса, всегда необходимо читать строку из данных

Falcon

- DEADLOCK:

  Во второй сессии была получена ошибка:
  ERROR 1020 (HY000): Record has changed since last read in table 'test2'

  Ann Harrison сказала, что Falcon не осуществляет проверку циклов на графе блокировок незамедлительно во время ожидания блокировки строки, а делает это периодически

- UPDATE:
  Вторая сессия ждет

Преимущества и недостатки Falcon

- Выводы делать рано – это все еще версия Alpha с большим количеством багов
- Активная поддержка со стороны MySQL AB
- Высокая скорость разработки – баги исправляются быстро, за последние 3 месяца очень возросла производительность
- Хорошая интеграция в MySQL, например таблицы с данными о производительности
- Нет поддержки кластеризации по первичному ключу или по covering index
- Различие архитектурных решений может затруднять переход из Innodb (хотя логически поведение стало более похожим)

O'REILLY

OSCON™
Open Source Convention

Бывает ложь, большая ложь и **Benchmark**-и

Benchmarks – необходимые пояснения

- Данные тесты могут не отображать производительность вашего приложения
- Мы использовали ранние версии Falcon и PBXT. Их производительность может измениться к моменту завершения их разработки.
- У нас не так много опыта в настройке Falcon, PBXT и Solid под MySQL, поскольку они не очень распространены
- Мы провели меньше тестов, чем хотели бы – слишком много времени потратили на выявление багов и внесение исправлений

Benchmark-и

- Read-Only на типичной таблице для web-приложений
- DBT2 – эмуляция TPC-C
- DellDVD Store – эмуляция сайта e-commerce
- Sysbench – транзакции OLTP
- Sqlbench - малый объем данных, один пользователь, примеры распространенных запросов

Спецификация сервера

- Dell PowerEdge 2950
- CentOS release 4.5
- 4 CPU

```
model name      : Intel(R) Xeon(R) CPU           5148 @
2.33GHz
stepping        : 6
cpu MHz         : 2327.529
cache size      : 4096 KB
```

- 16 GB RAM
- RAID 10 (6 10K RPM 3.5" SAS hard drives)

Версии MySQL

- Конечно, на производительность влияет не только storage engine, но и версия MySQL, однако мы не смогли заставить все storage engine работать с одной и той же версией.
- InnoDB и PBXT
  5.1.19
- Falcon
  6.0.1-alpha, bk tree от 10 июля 2007
- SolidDB
  5.0.41-0073

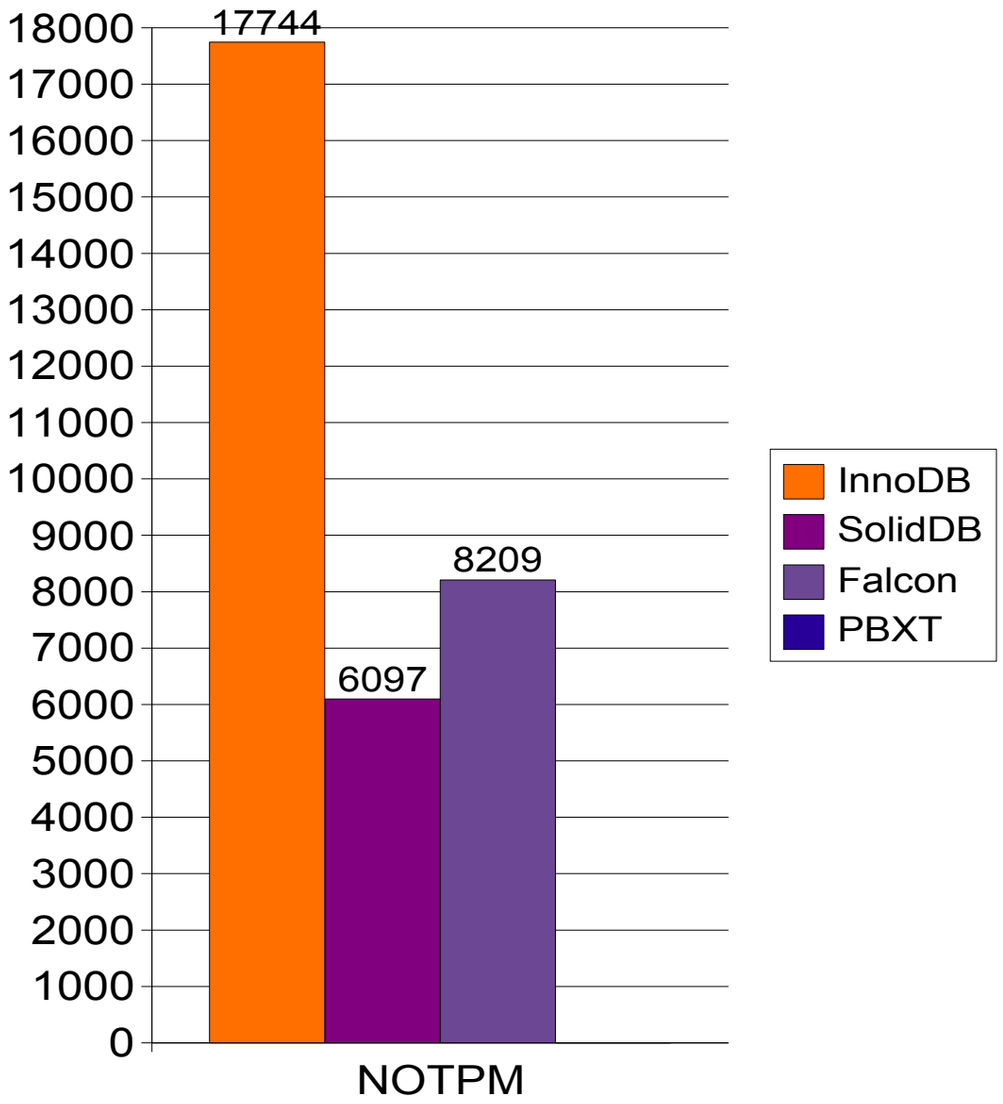
Engines-параметры

- 12 GB RAM для буферов
- InnoDB **--innodb_buffer_pool_size=12G**
  --innodb_flush_method=O_DIRECT
  --innodb-log-file-size=100M
- SolidDB **--soliddb-cache-size=12G**
- Falcon
  --falcon_min_record_memory=2G
  --falcon_max_record_memory=4G
  --falcon_page_cache_size=8G
- PBXT
  pbxt_index_cache_size=8G
  pbxt_record_cache_size=4G

Детали конфигурации DBT2

- DBT2
  - http://osdldbt.sourceforge.net/
- 10 одновременных пользователей (по 2 на каждый процессор и диск)
- "Нулевые задержки" для полной нагрузки MySQL сервера
- В конфигурации 400W объем памяти уменьшен до 4G блокированием 12GB памяти, чтобы эмулировать IO bound-нагрузку.
- Объем буферов уменьшен до 2GB

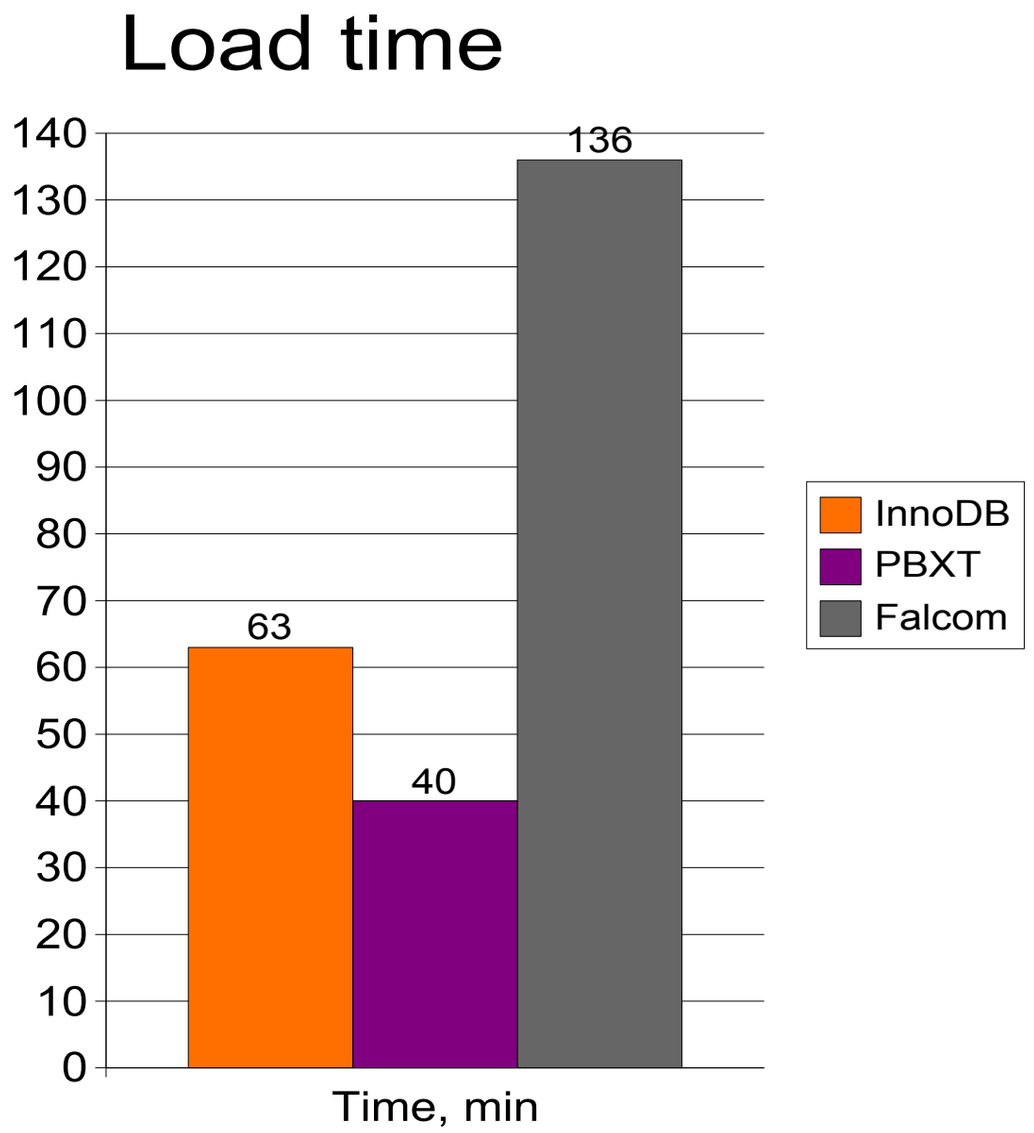
DBT2 – 10 warehouses



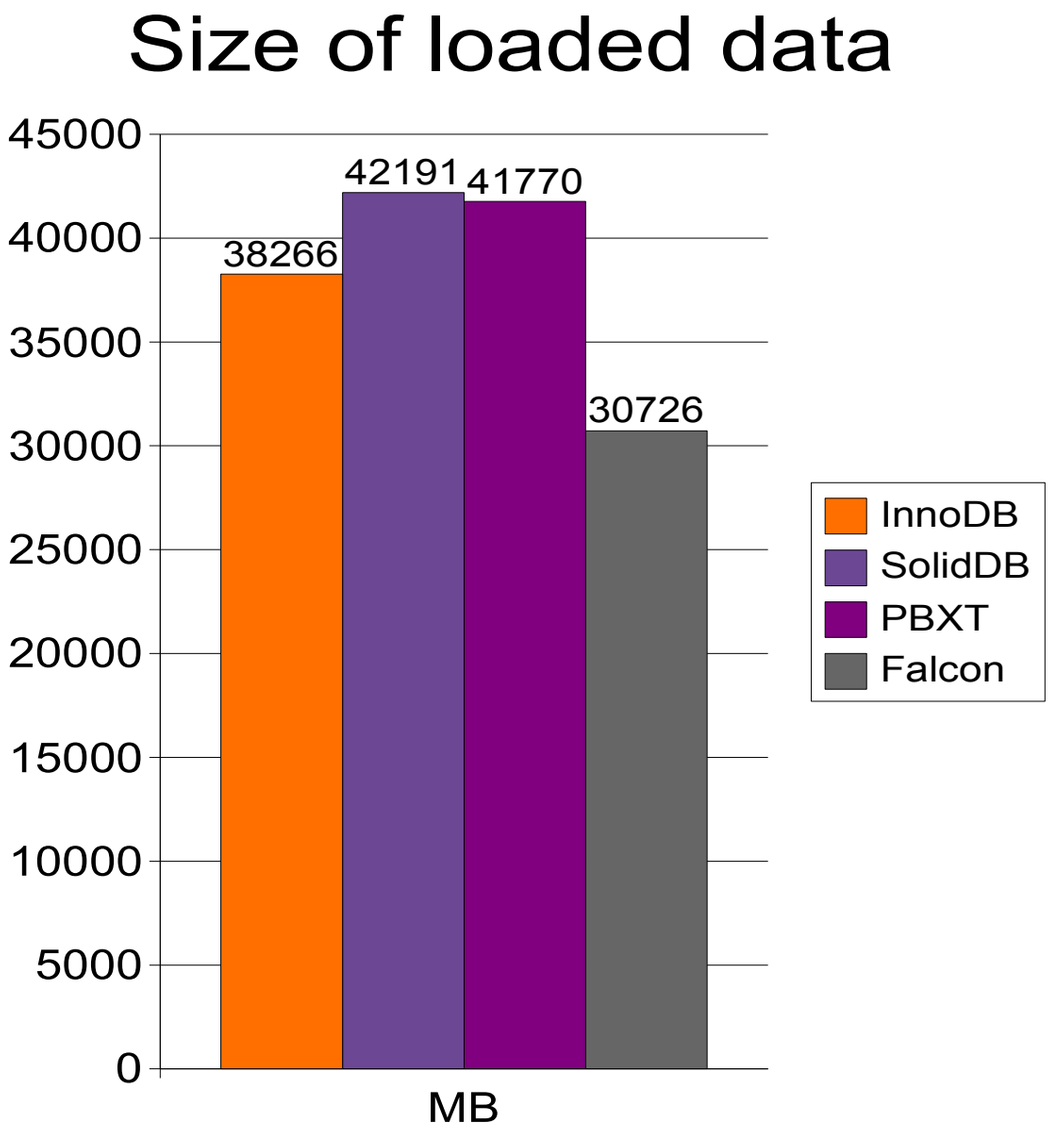
- 10 warehouses, 10 пользователей (объем данных ~ 700M)
- Результаты – количество New OrderTransaction в минуту, чем больше, тем лучше
- PBXT упал
- Старая версия Falcon показала результат ~1100 NOTPM
- Значительное улучшение !

DBT2 – 400 warehouses

- Объем данных ~ 29GB
- SolidDB упал через 336 минут
- Логи транзакций в SolidDB не запрещались, чтобы получить сравнимые результаты.

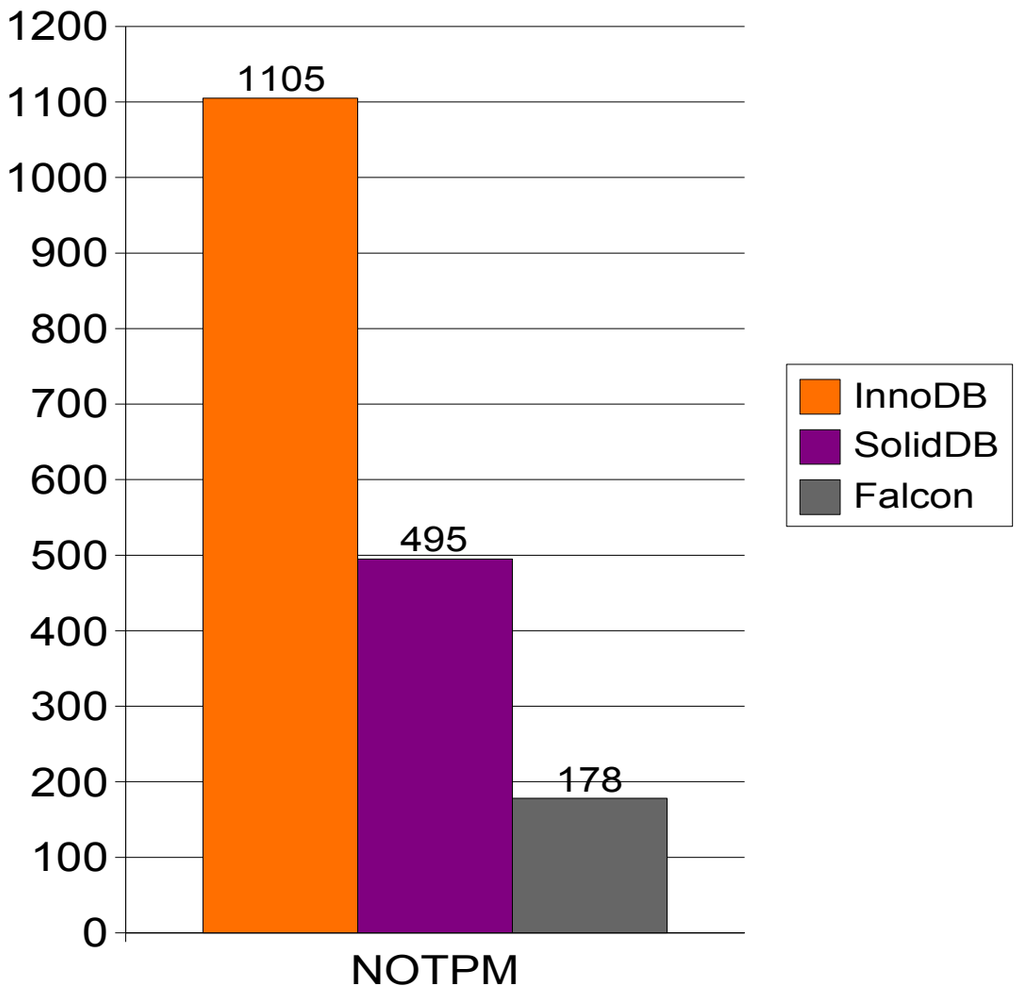


DBT2,400W ,Объем данных



- Неожиданно большой объем у PBXT
- SolidDB – таблицы были загружены в MyISAM и затем сконвертированы в SolidDB. Иначе он падал

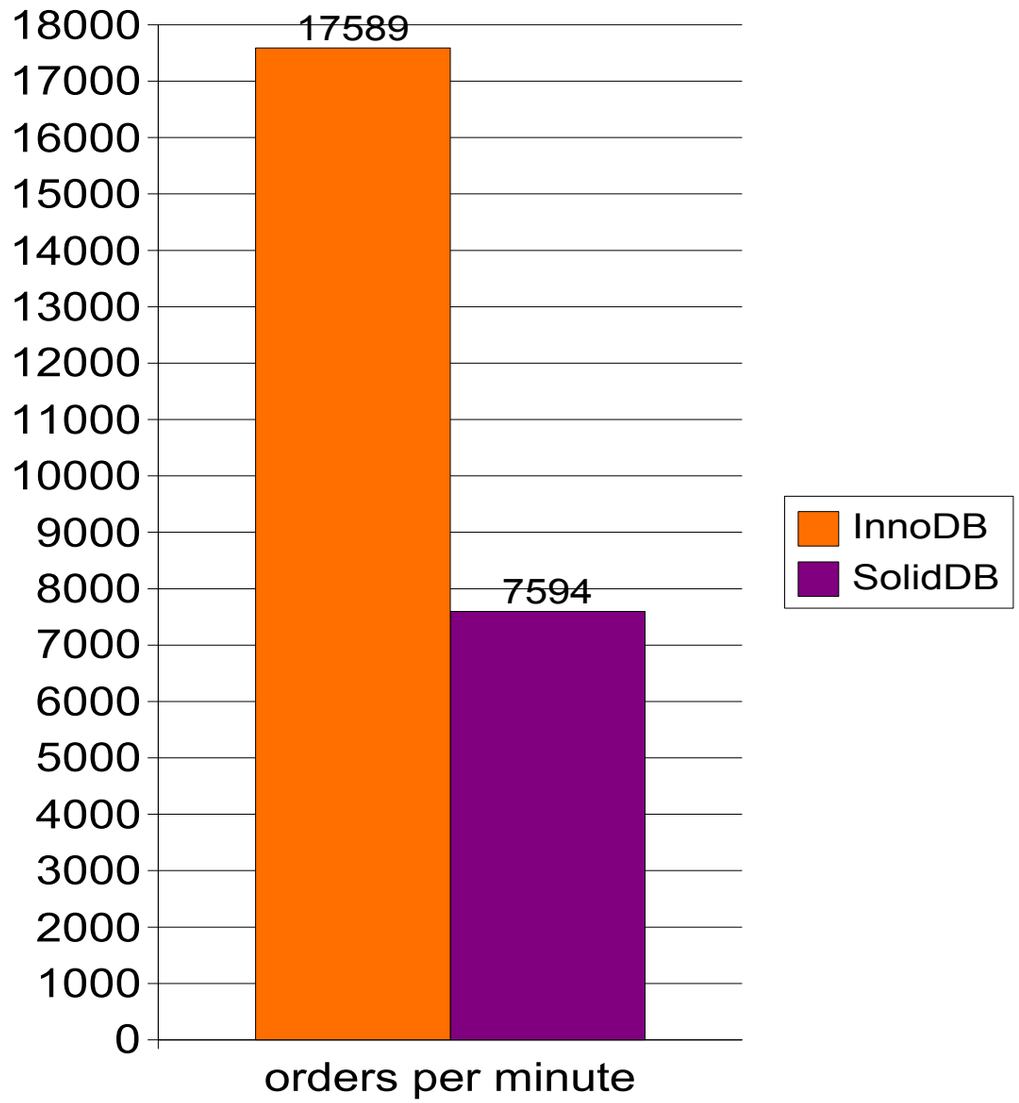
DBT2,400W ,Результаты



- PBXT упал
- Результаты – количество New Order Transaction в минуту, чем больше ,тем лучше

DellDVD Store

- Объем данных средний 1 GB
  2,000,000 Пользователей
  100,000 Продуктов
- Falcon – упал
- PBXT – много ошибок
- Результаты – количество New Orders в минуту, чем больше, тем лучше



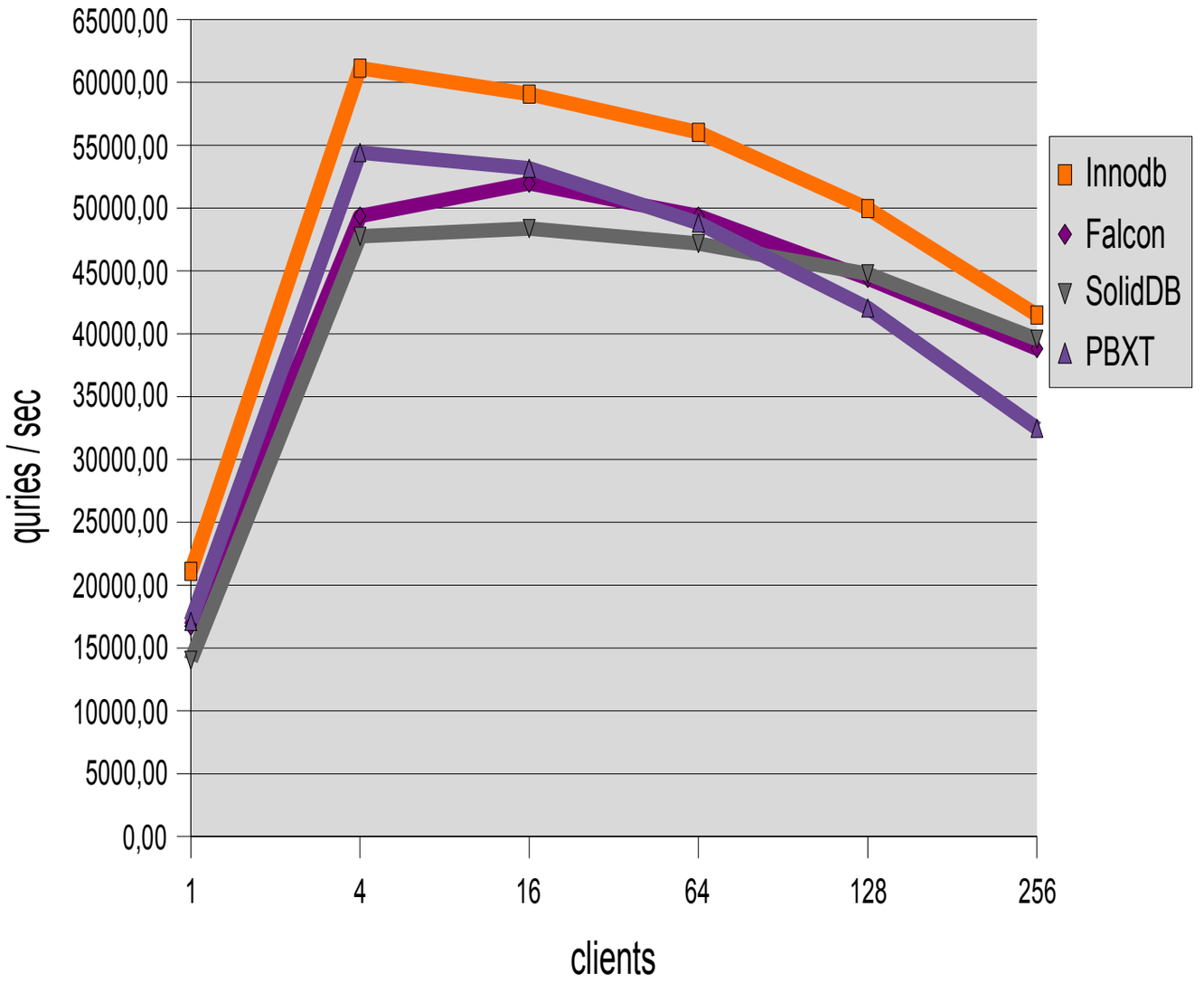
sysbench

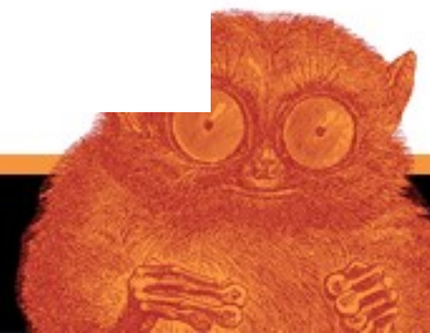
- В этом тесте использовалась старая версия `Falcon`. Новая падала :(
- Несколько `READ-ONLY` запросов к обычной таблице для `Web`-приложений – информация о `useraccount`:

```
CREATE TABLE IF NOT EXISTS sbtest (
id int(10) unsigned NOT NULL auto_increment,
name varchar(64) NOT NULL default '',
email varchar(64) NOT NULL default '',
password varchar(64) NOT NULL default '',
dob date default NULL,
address varchar(128) NOT NULL default '',
city varchar(64) NOT NULL default '',
state_id tinyint(3) unsigned NOT NULL default '0',
zip varchar(8) NOT NULL default '',
country_id smallint(5) unsigned NOT NULL default '0',
PRIMARY KEY  (id),
KEY `country_id` (country_id,state_id,city)
)
```

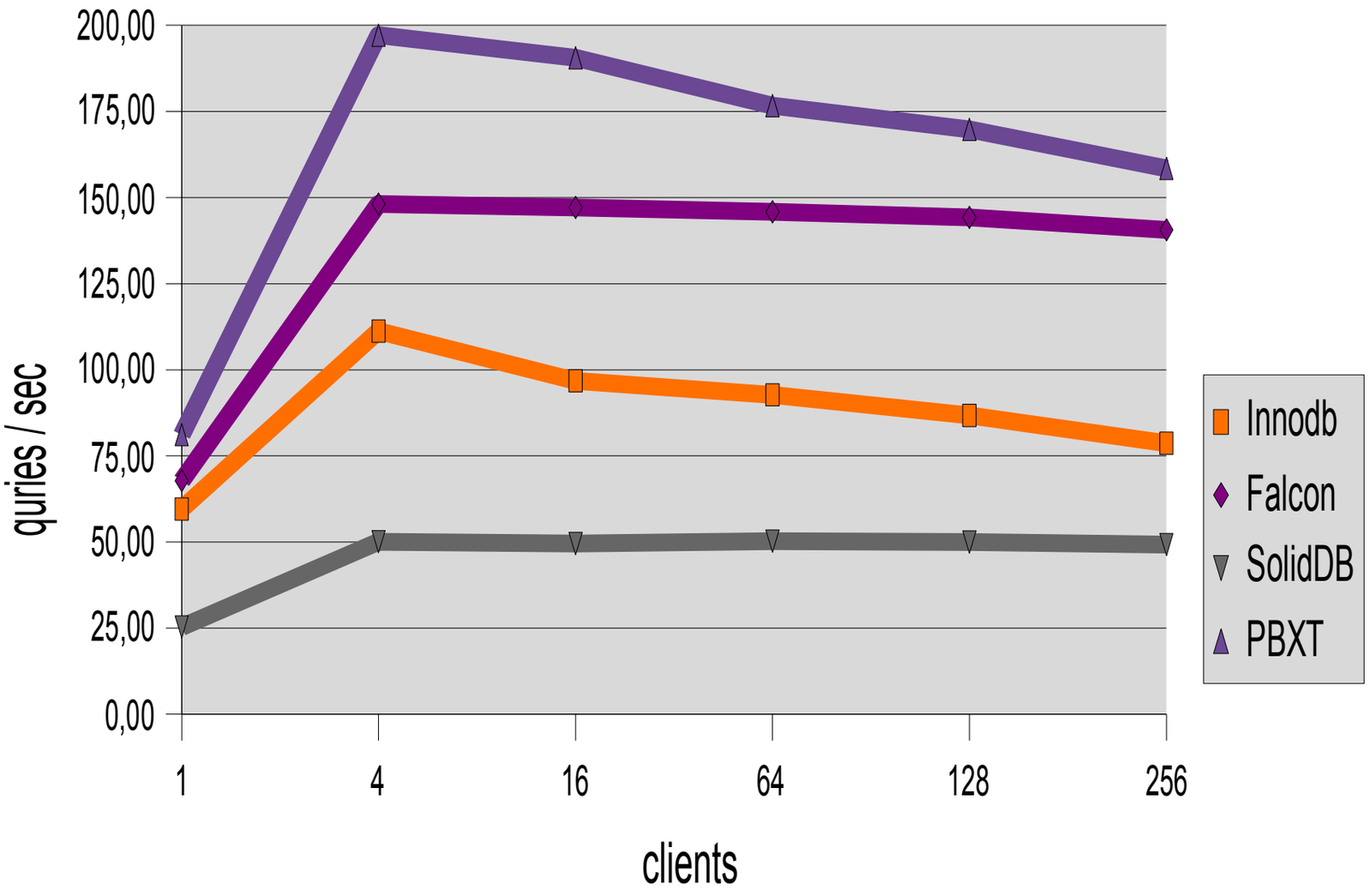
sysbench, выборка по первичному ключу

- `SELECT name FROM sbtest WHERE id=?`
- Innodb и Solid имеют преимущество, благодаря кластеризации по `PK`



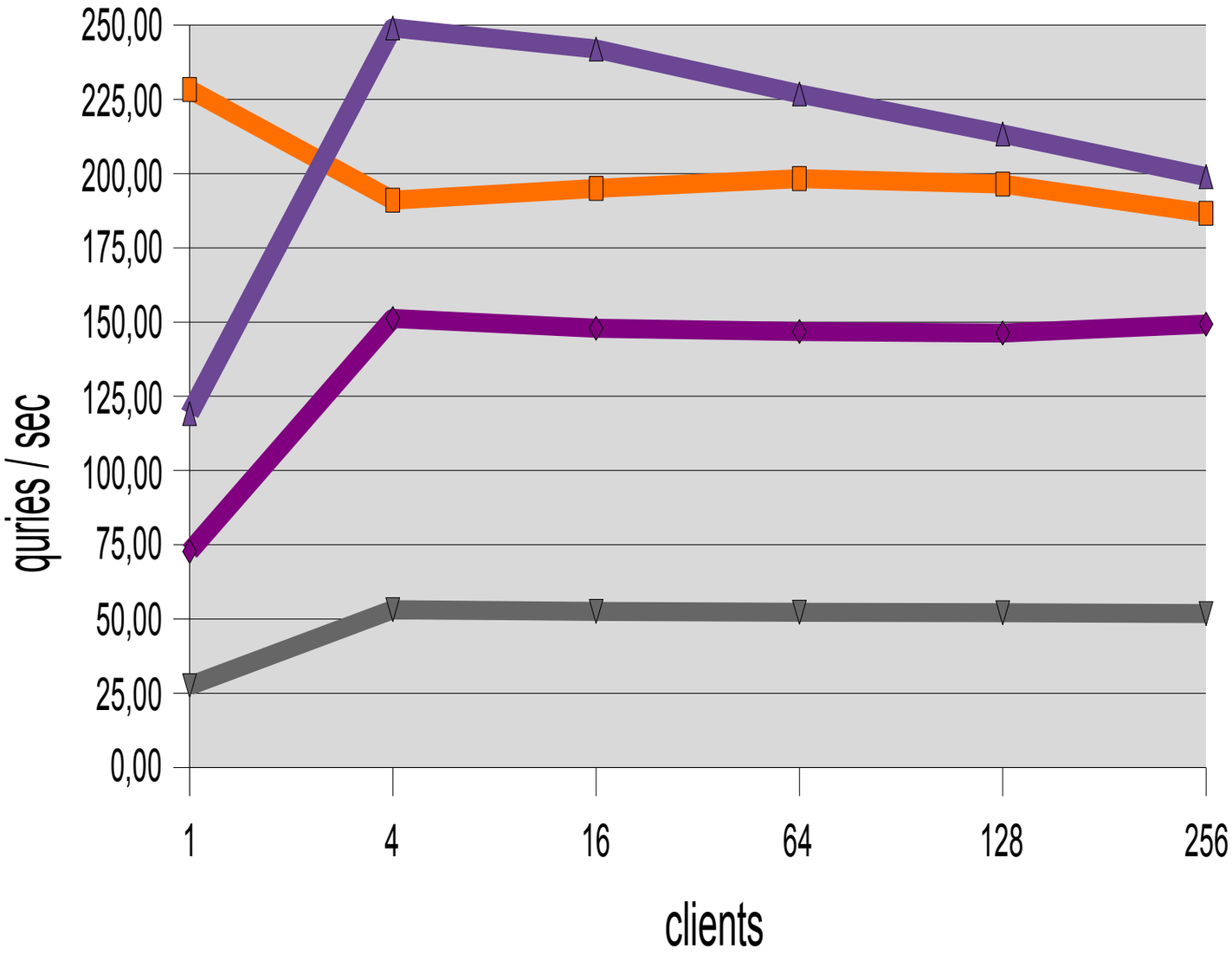


sysbench, выборка по индексу



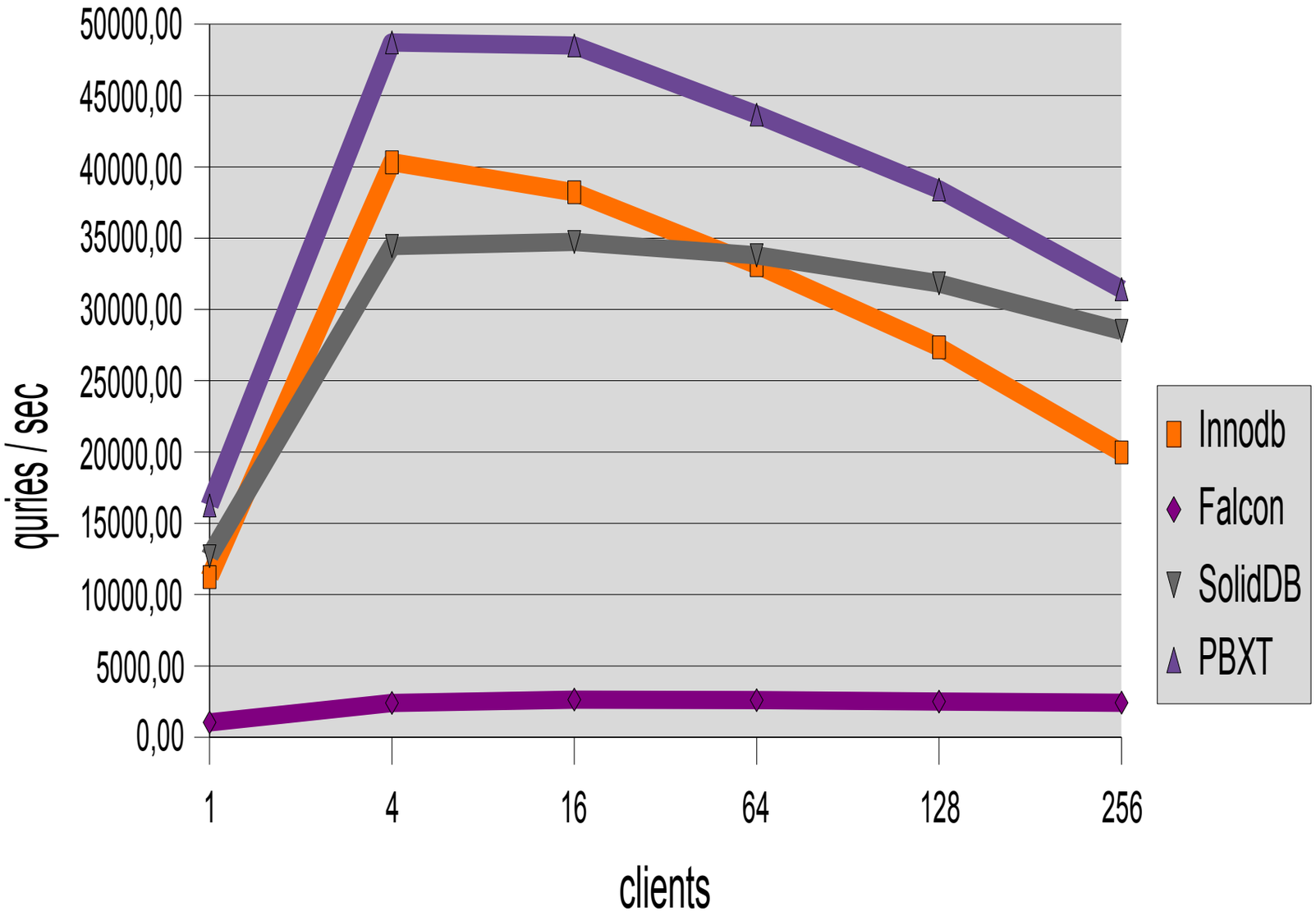
- SELECT name FROM sbtest WHERE country_id=?
- PBXT – лучший результат
- Falcon второе место

sysbench, выборка по covered index



- SELECT state_id FROM sbtest WHERE country_id=?
- PBXT – по-прежнему лучший результат
- Falcon не использует covered index

sysbench, выборка по индексу, LIMIT 20



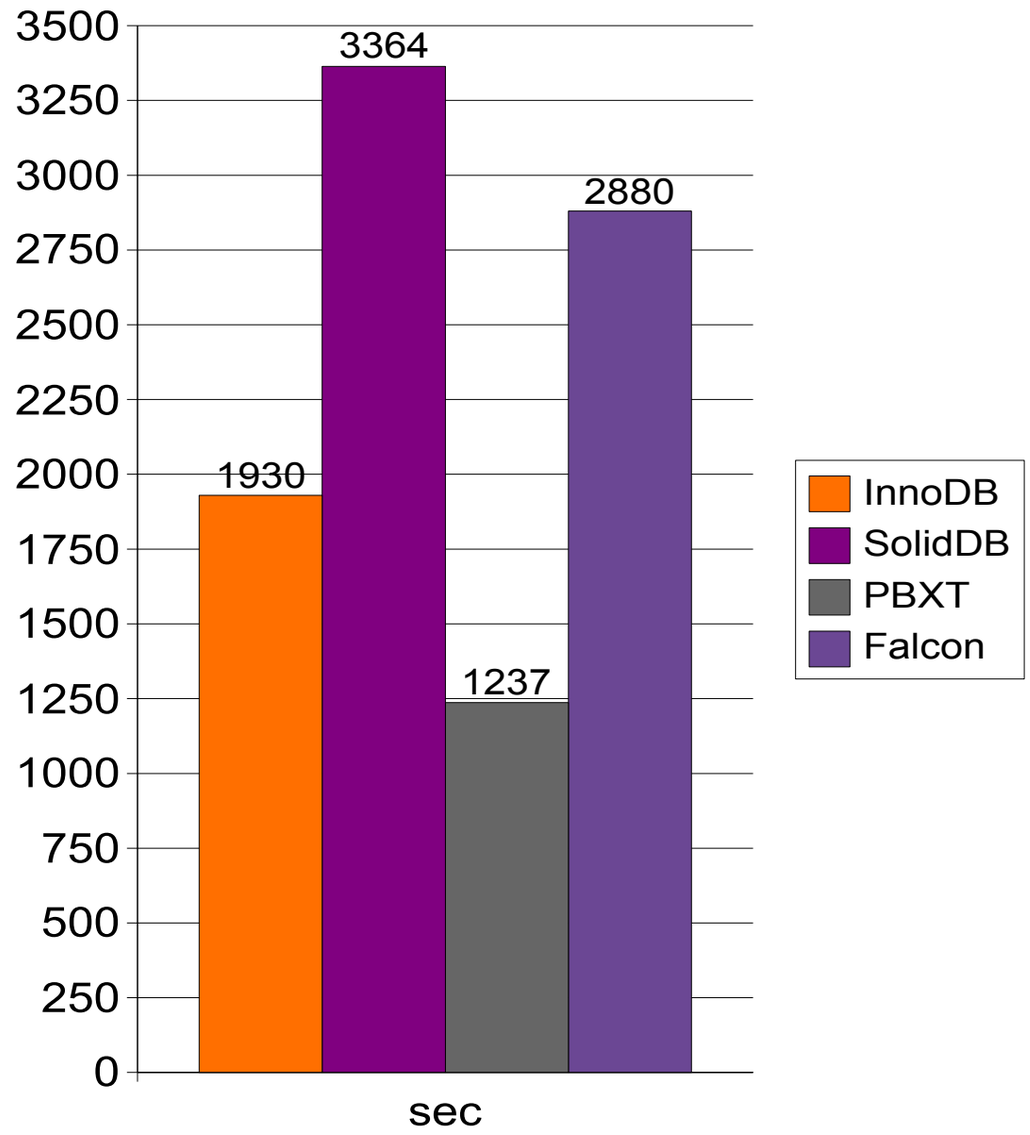
- SELECT name FROM sbtest WHERE country_id=? LIMIT 20
- Falcon не оптимизирует использование Limit
- Innodb плохо масштабируется

Sysbench OLTP

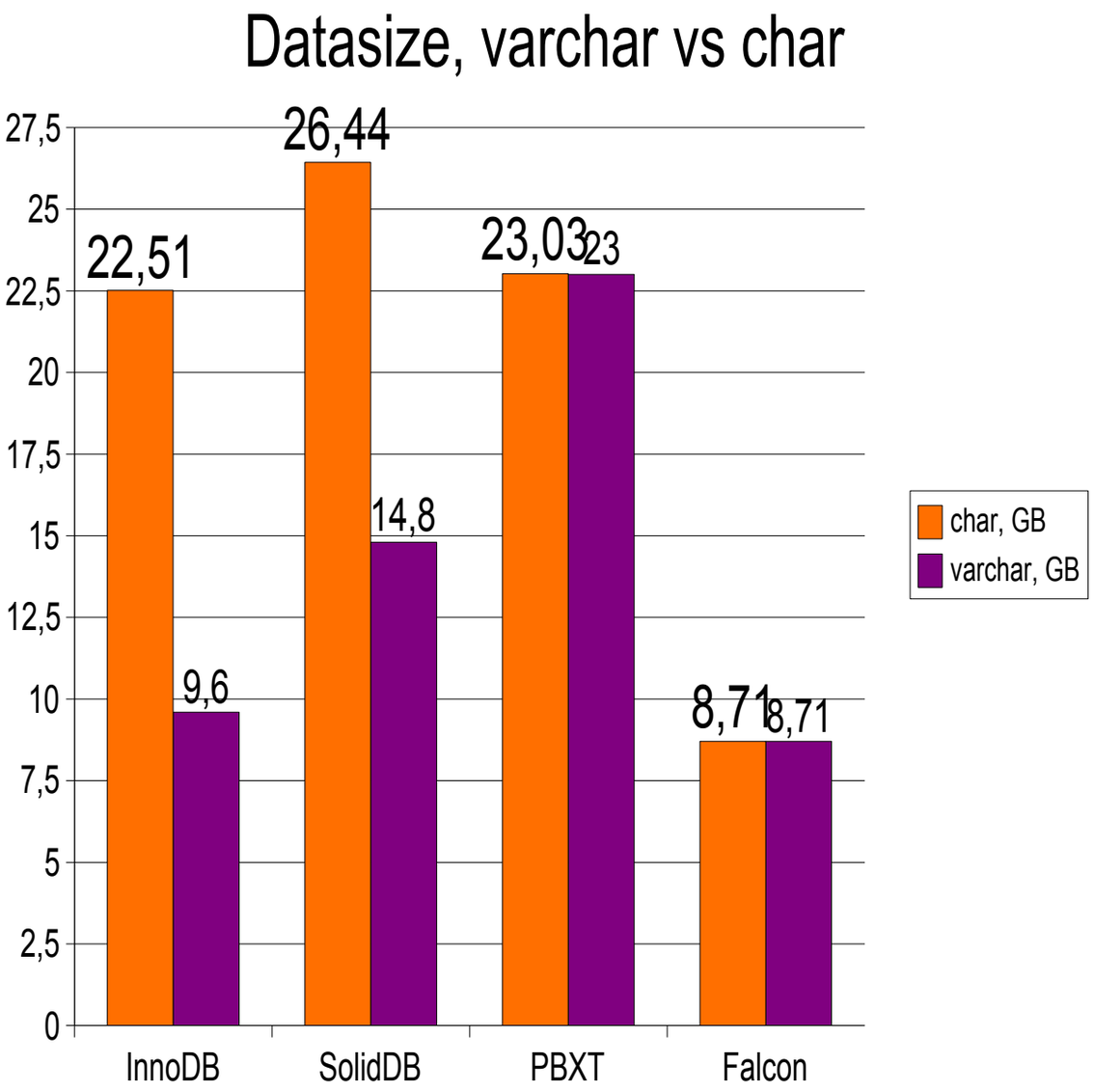
- Объем данных
  100,000,000 строк
  ~25GB
- Равномерное распределение
- I/O-bound нагрузка
- read/write транзакции
- Объем доступной памяти уменьшен путем блокирования 12GB из 16GB

Sysbench OLTP, время загрузки данных

- Используем multi-value INSERTs вместо LOAD DATA INFILE
- Известно, что Innodb при загрузке данных работает медленнее, чем MyISAM. Однако Solid и Falcon оказались еще медленнее.



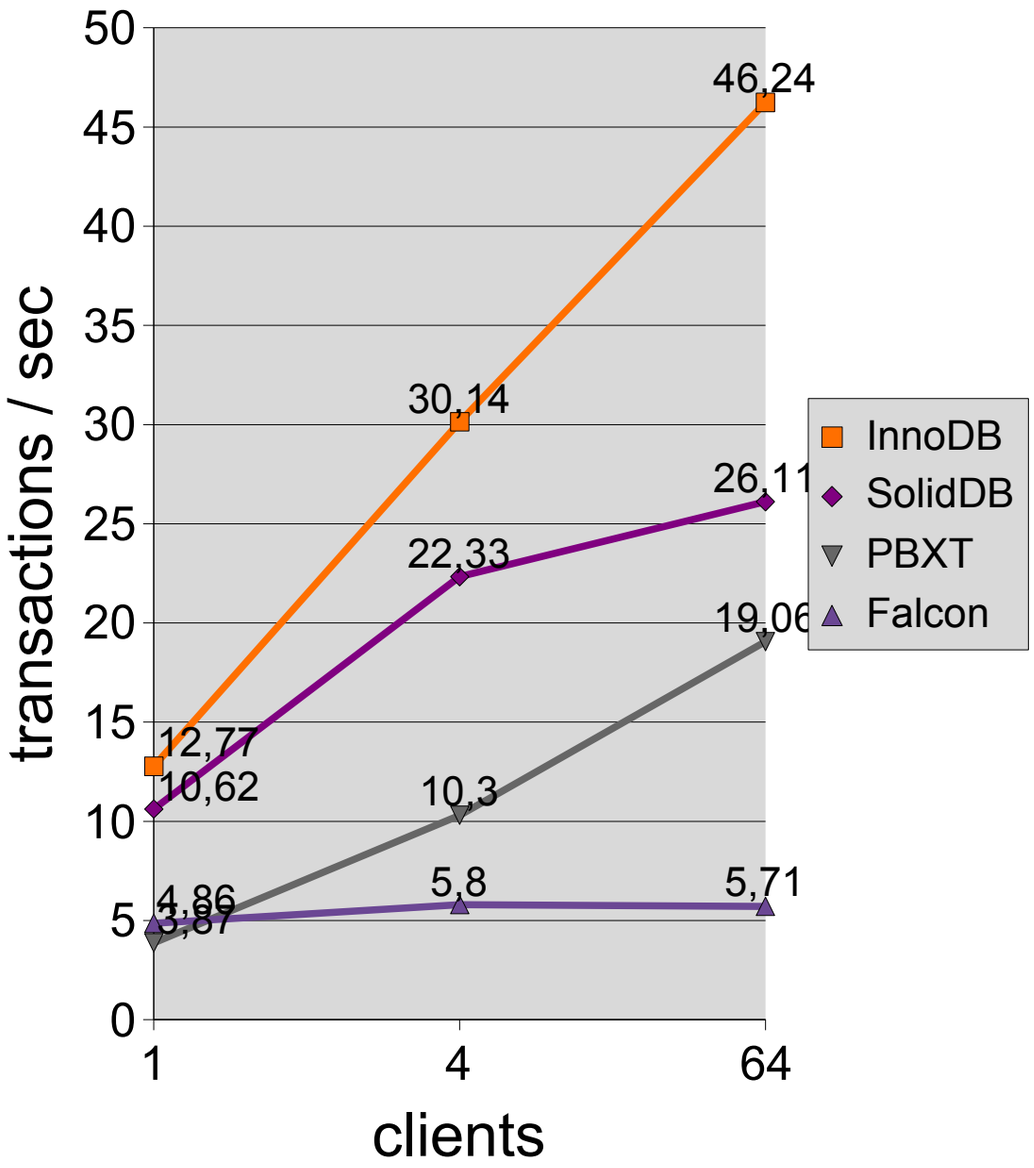
Sysbench OLTP, объем данных



- Сравним хранение в таблице данных типа `char` и типа `varchar`
- `Falcon` в обоих случаях использует динамическую длину строки
- `PBXT` неожиданно демонстрирует одинаково огромный объем для всех данных

Sysbench OLTP, результаты

I/O bound



- Объем памяти 4GB, буферов - 2GB
- Innodb и SolidDB показывают лучшие результаты, благодаря кластерингу по первичному ключу
- Все, кроме Falcon, хорошо масштабируются для IO bound-нагрузки, с этим количеством жестких дисков

O'REILLY OSCON™ Open Source Convention

Избранные результаты sqlbench

- единичная операция повторяется N раз, общее время в секундах. чем меньше результат, тем лучше

Операция	1	2	3
	innodb_	pbxt_fa	soliddb
alter_table_add (100)	8.00	3.00	32.00
count (100)	12.00	8.00	28.00
count_distinct (1000)	6.00	8.00	74.00
count_distinct_2 (1000)	11.00	11.00	16.00
count_group_on_key_parts (1000)	7.00	10.00	83.00
count_on_key (50100)	70.00	94.00	210.00
delete_all_many_keys (1)	17.00	2.00	28.00
insert (350768)	6.00	5.00	21.00
outer_join (10)	14.00	7.00	61.00
select_key2_return_prim (200000)	30.00	29.00	25.00
select_many_fields (2000)	8.00	6.00	5.00
update_big (10)	18.00	56.00	727.00
update_of_key_big (501)	19.00	6.00	165.00
update_of_primary_key_many_keys (256	44.00	17.00	55.00
update_with_key_prefix (100000)	**19.00**	**8.00**	**10.00**

Выводы

- Все рассмотренные storage engines, кроме InnoDB, на данный момент слишком нестабильны для реального использования. Наиболее близок к завершению SolidDB.
- InnoDB лидирует в большинстве тестов
- Falcon-у необходимо решить проблемы с оптимизацией запросов с LIMIT и масштабированием IO bound
- PBXT и Falcon в определенных тестах показывают лучший результат
- SolidDB в настоящий момент аутсайдер в вопросах производительности
- Когда все storage engines будут готовы, тесты необходимо будет провести еще раз.

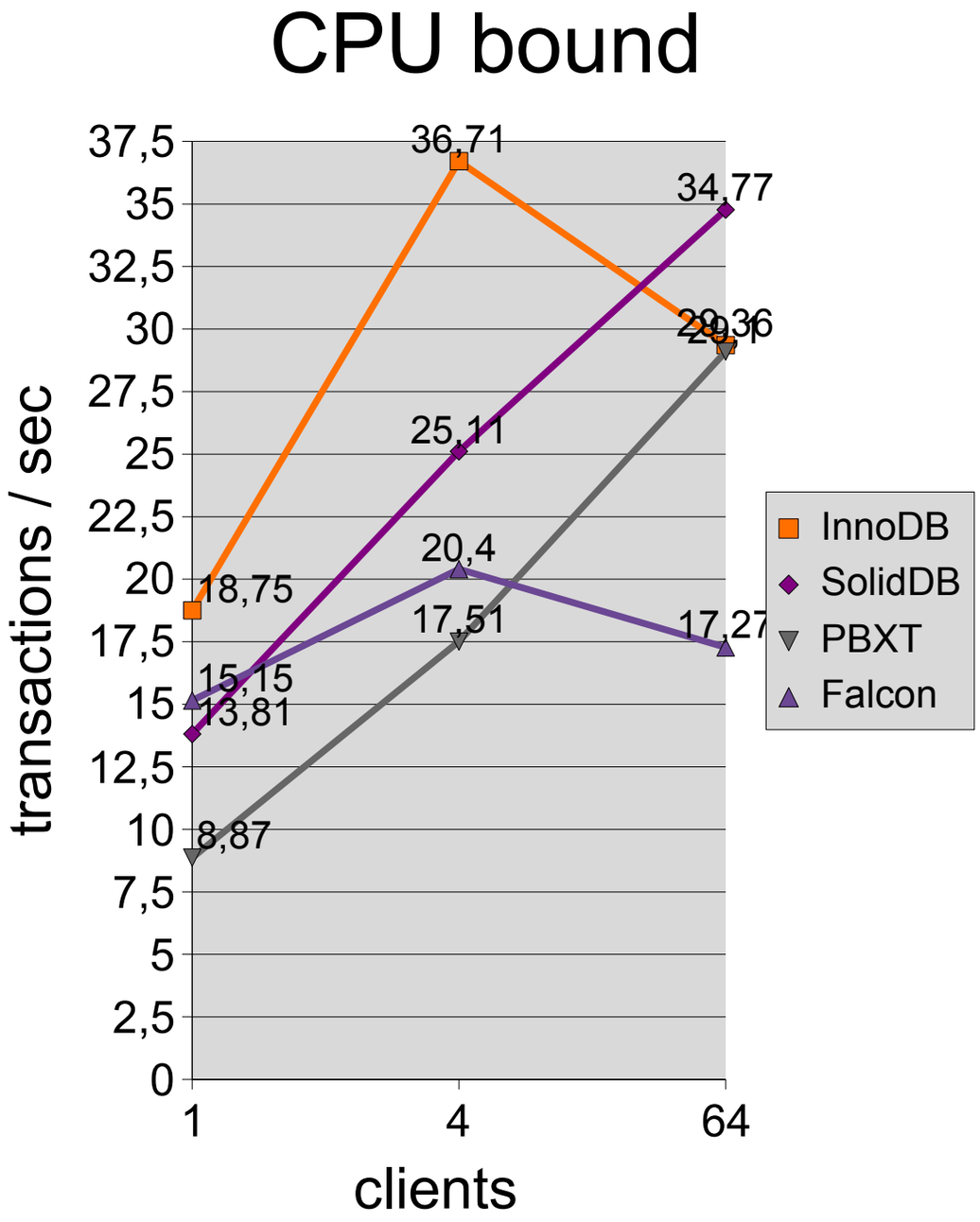
O'REILLY

OSCON™ Open Source Convention

Заключение

- Спасибо за внимание!
- Слайды будут опубликованы на http://www.mysqlperformanceblog.com/
- Будем рады ответить на ваши вопросы
- Возможны консультации по настройке производительности MySQL
- http://www.mysqlperformanceblog.com/mysql-consulting/

Sysbench OLTP, результаты, графики



- Объем данных соизмерим с объемом памяти