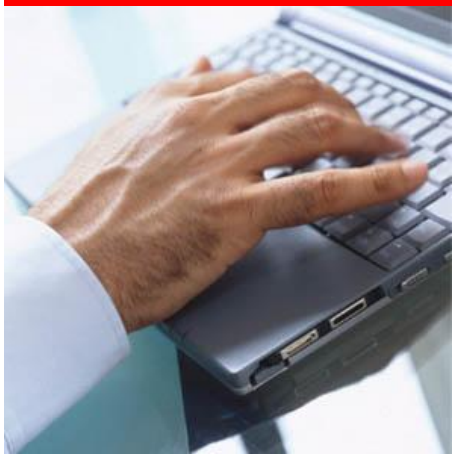
InnoDB: Past, Present, and Future

Alibaba Developer Conference, July 7, 2012

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.
The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda



- What is InnoDB?
- InnoDB: Past
- InnoDB: Present
- InnoDB: Future

What is InnoDB?

InnoDB Overview

- The most popular transactional storage engine for MySQL, distributed and supported by MySQL since 2001
- Follows the ACID model, with transactions featuring commit, rollback, and crash-recovery capabilities
- MVCC (multiversion concurrency control)
- Rowlock
- FOREIGN KEY referential-integrity constraints.
- Innovative features make it faster, such as
  - Insert buffering
  - Adaptive hash index

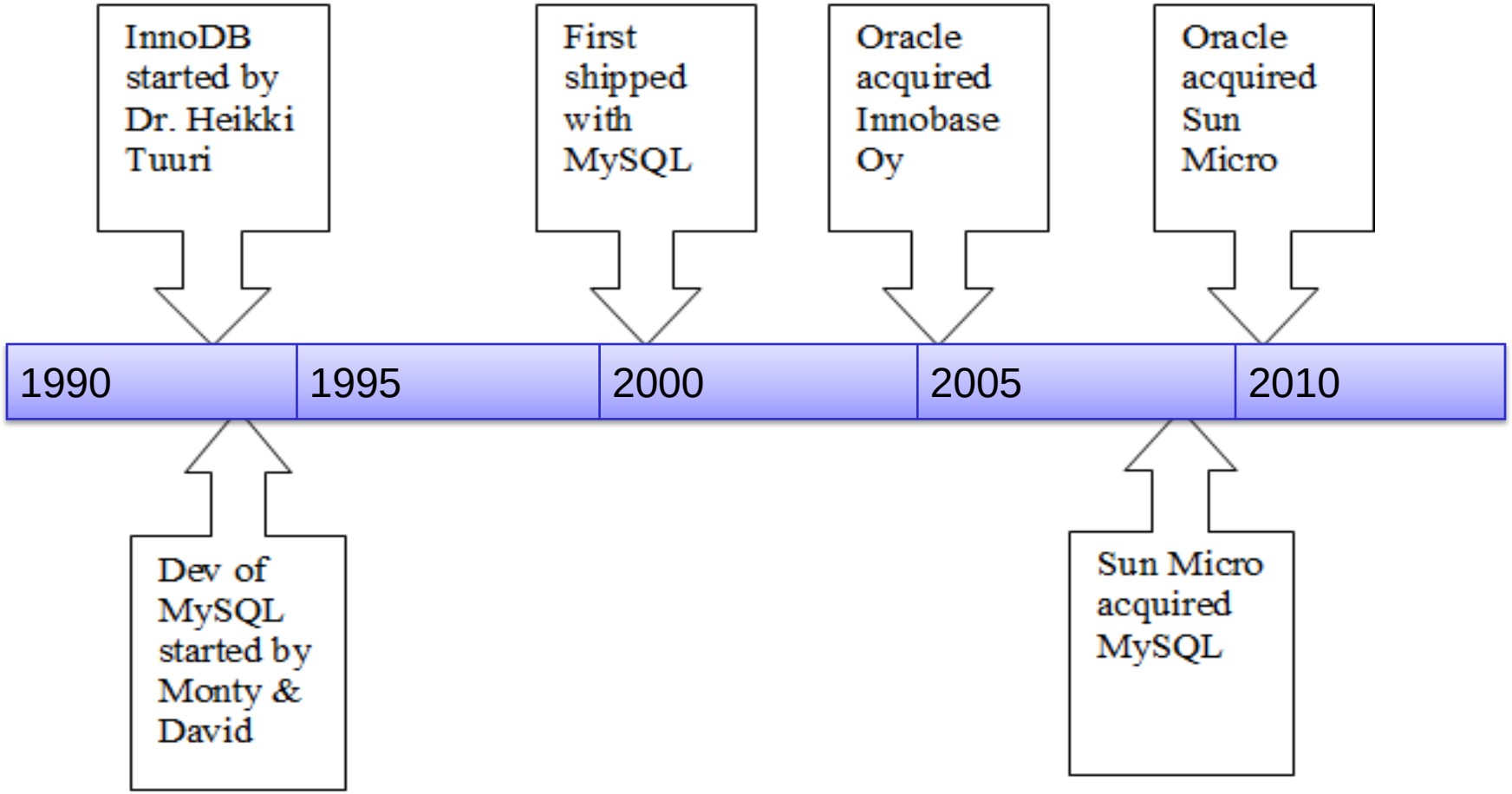
InnoDB Users

InnoDB: Past

Brief InnoDB History

- Started by Dr. Heikki Tuuri in January 1994
- Innobase Oy was founded in 1995
- First shipped with MySQL in March 2001 as open source
- Oracle acquired Innobase Oy in October 2005
- Sun announced the acquisition of MySQL AB in January 2008
- Oracle announced the acquisition of Sun in April 2009
- InnoDB Plugin 1.0 for MySQL 5.1 GA in April 2010

InnoDB Timeline



InnoDB Key Characteristics

- Row-level locking
- Multi-version concurrency control
- Efficient indexing (covering indexes)
- Fuzzy checkpoint
- Adaptive hash indexing
- Table compression
- Group commit
- Fast DDL operations
- Insert buffering

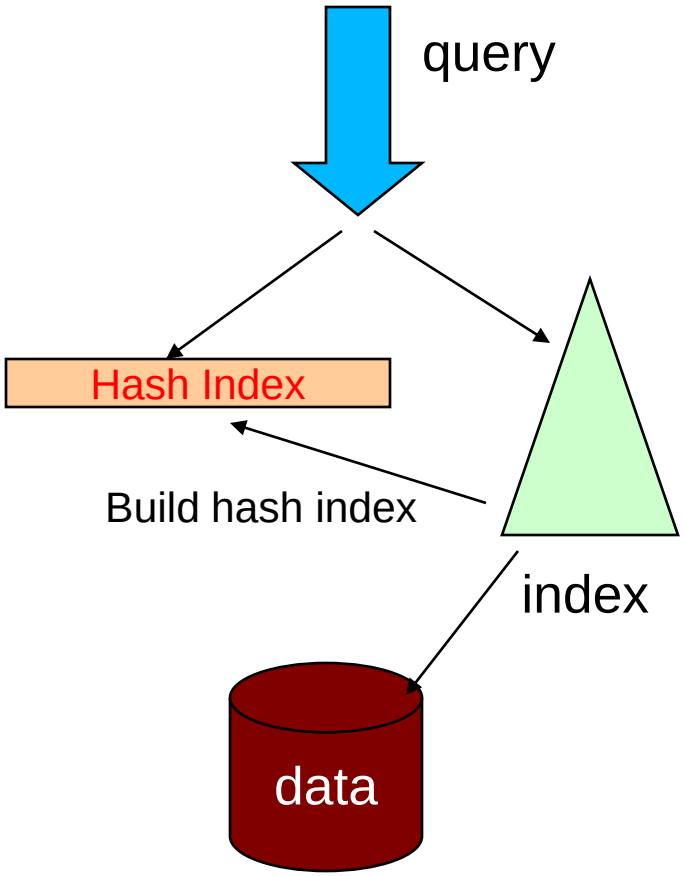
InnoDB Key Characteristics

- ACID-compliant transactions
- Two phase commit
- Automatic crash recovery
- Doublewrite buffer
- Referential integrity
- Online backup with MySQL Enterprise Backup
- Well written, well tested code

Unique InnoDB Features

- Insert Buffering
  - A special index in the InnoDB system tablespace that buffers modifications to secondary indexes when the leaf pages are not in the buffer pool
  - When the leaf page is loaded to the buffer pool, any buffered changes will be merged to it, so that users never see unmerged changes.
  - it trades random I/O with a larger amount of sequential I/O, it speeds up operation on hard disks, where random access is much slower than sequential access.
  - Shows approximately 7.2 X faster performance than server without the feature
  - On solid-state storage, where there is little difference between sequential and random access times. Change buffering may still be useful if writes to solid-state storage are expensive

Unique InnoDB Features



- Adaptive Hash Index
  - Automatically creates hash index on prefix of key for frequent queries
  - Approximate it to in memory database (IMDB).
  - InnoDB allocates innodb_buffer_pool_size/64 to the adaptive hash index at startup (so if you have 64G buffer pool, the adaptive hash index would be 1G).
  - Performance study (sysbench) shows it gives:
  - 2X better Read/Write
  - 5X better on index joins

InnoDB: Present

MySQL 5.5

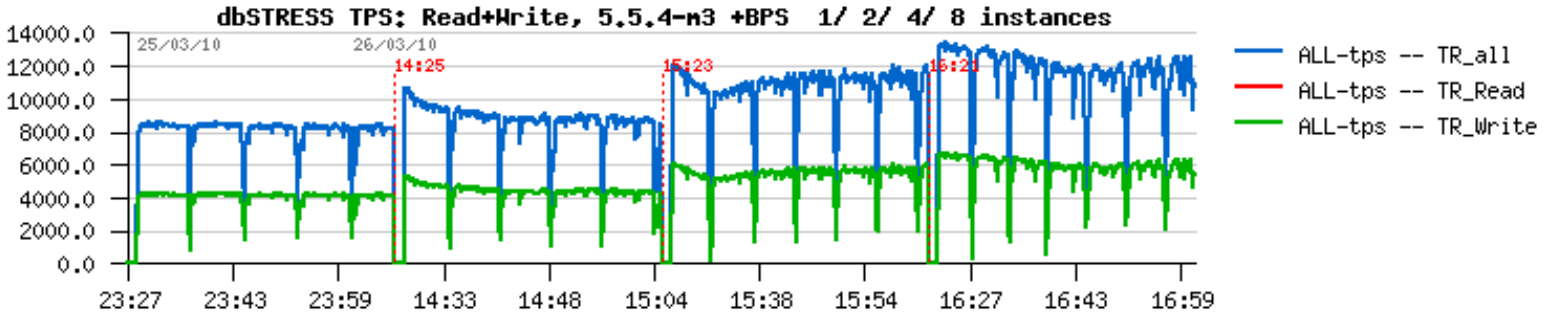
- The current GA version
- InnoDB becomes the default storage engine
- First major release after InnoDB team joined MySQL engineering
- Many new features are co-developed by InnoDB team and MySQL server, such as multiple buffer pool, performance schema, and more.
- InnoDB Hot Backup becomes part of MySQL Enterprise offering

InnoDB Features in MySQL 5.5

- Performance and scalability
  - Multiple buffer pool instances
  - Improved Crash Recovery Performance
  - Extended insert/change buffering with delete and Purge Buffering
  - Support Native AIO on linux
  - Multiple rollback segments(128)
  - Separate purge thread from master thread
  - Separate flush list mutex
    - One per buffer pool
    - Separate mutex for enforcing the flush list order

Multiple Buffer Pool Instances

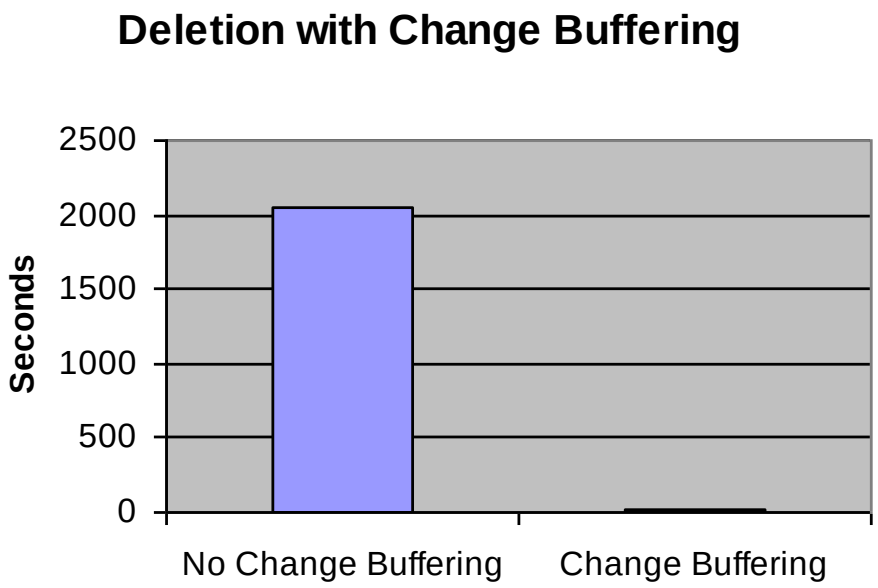
- –innodb-buffer-pool-instances=x



Extended InnoDB Change Buffering – Benchmark

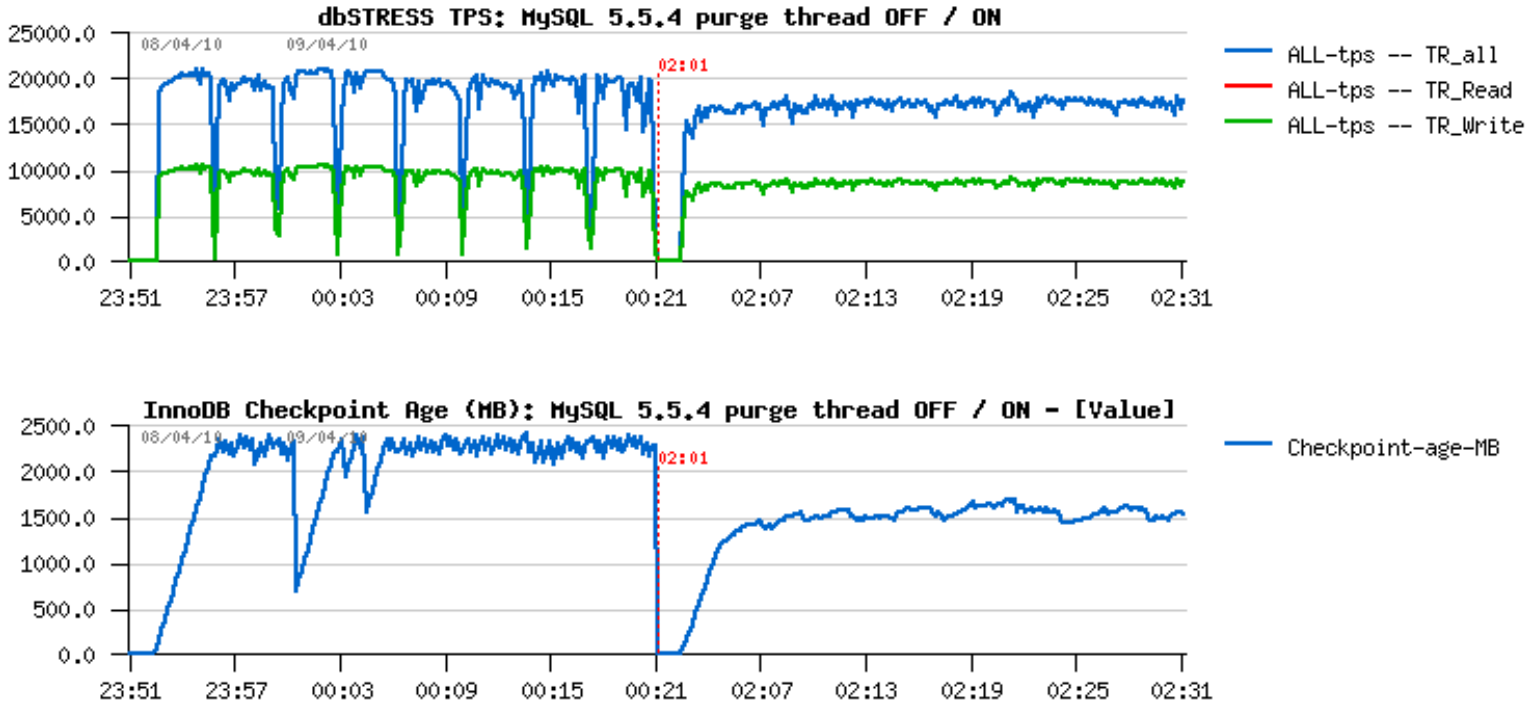
- Table with 5 Million rows
- Six secondary indexes
- Table size 3G
- Buffer Pool 1G
- Bulk delete of 100,000 rows

	100K Row Deletion (seconds)	Deletion Rate (rows/second)
Without Change Buffering	**2041**	**50**
With Change Buffering	**14.54**	**8000**

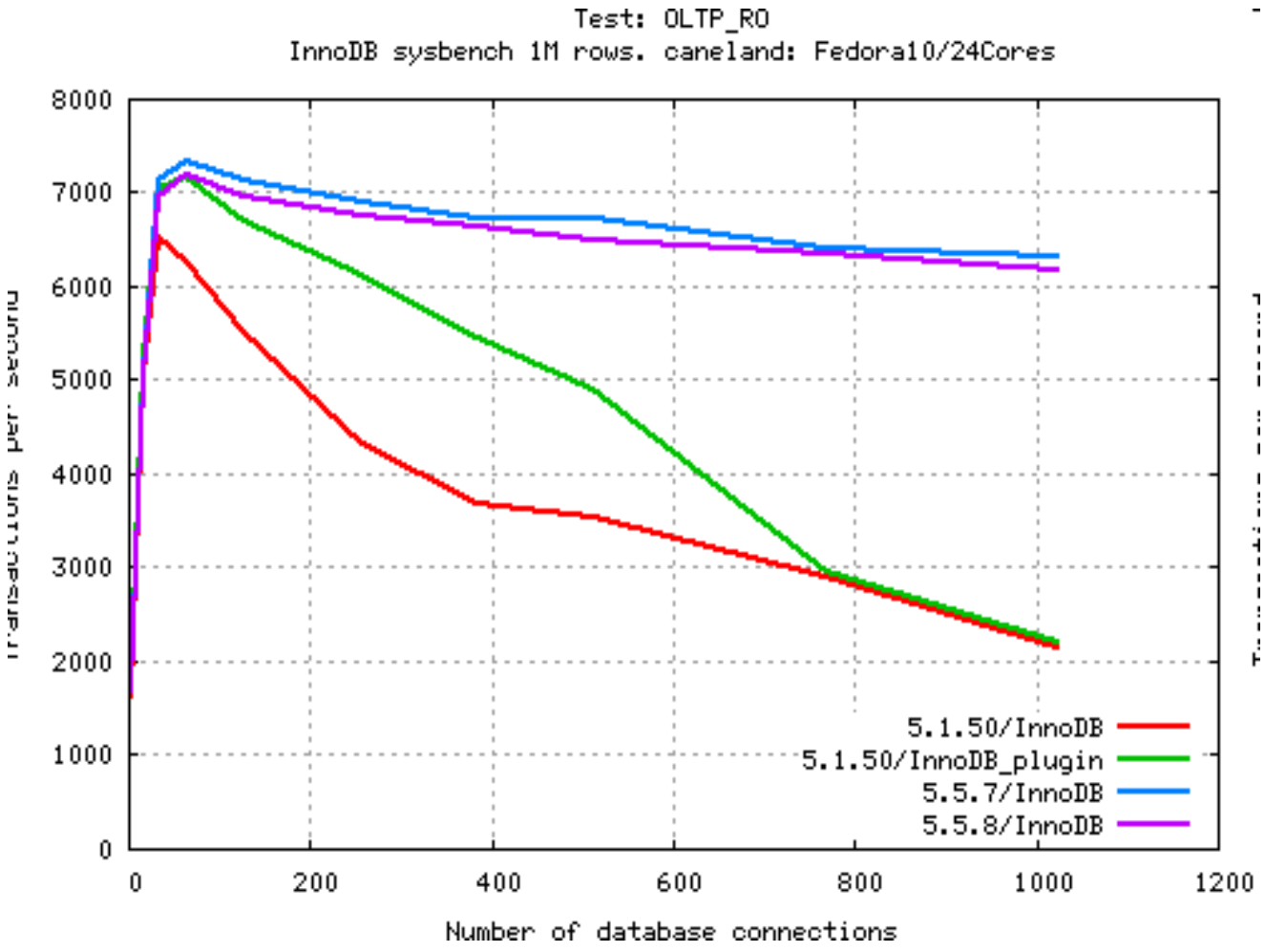


Improved Purge Scheduling – Benchmark

dbSTRESS: Read+Write & Purge Thread



MySQL 5.5 Benchmarks – Linux



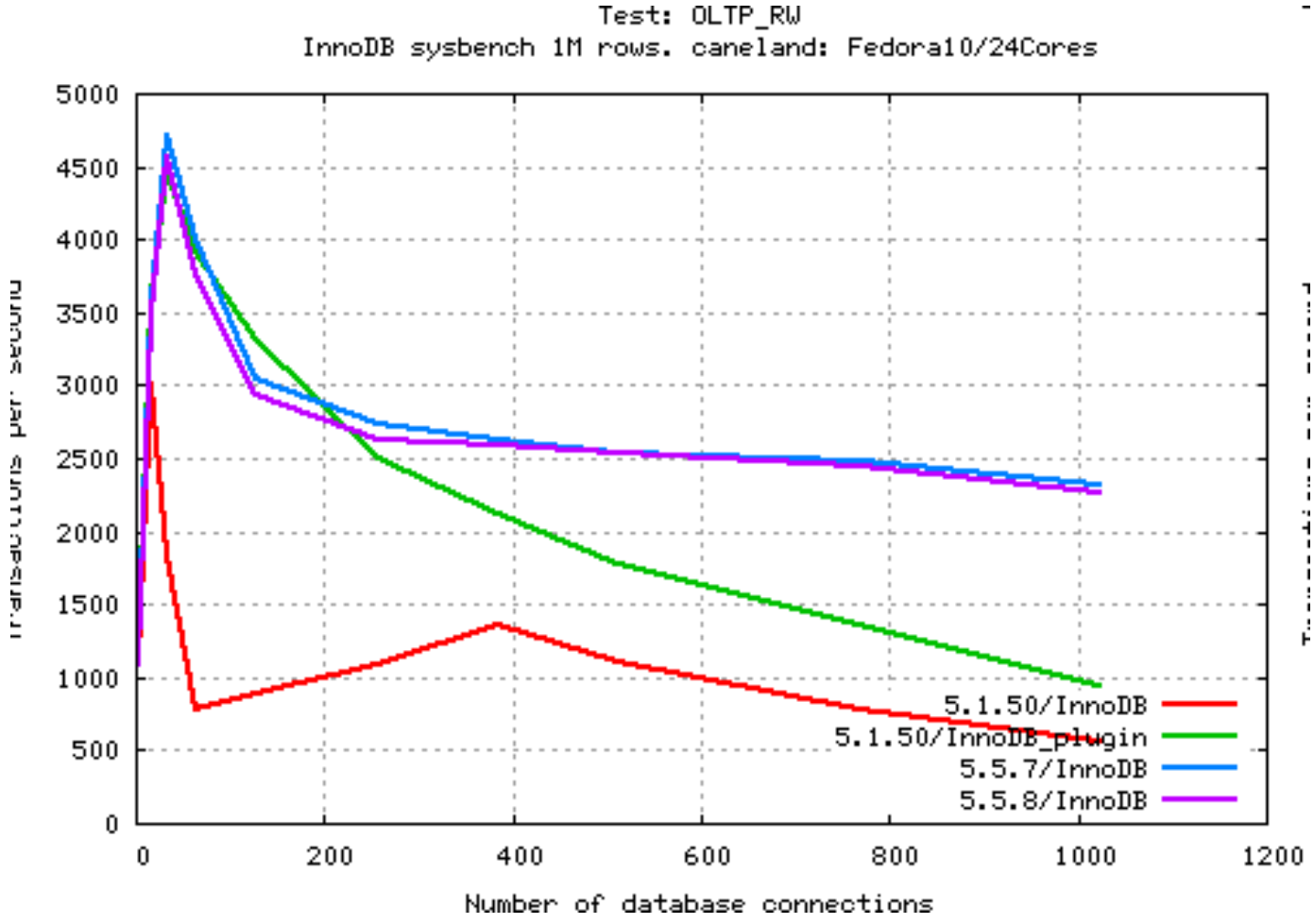
MySQL 5.5.8
(InnoDB 1.1)

MySQL 5.1.50
(InnoDB Plug-in)

MySQL 5.1.50
(InnoDB built-in)

Intel Xeon X7460 x86_64
4 CPU x 6 Cores/CPU
2.66 GHz, 32GB RAM
Fedora 10

MySQL 5.5 Benchmarks – Linux



MySQL 5.5.8
(InnoDB 1.1)

MySQL 5.1.50
(InnoDB Plug-in)

MySQL 5.1.50
(InnoDB built-in)

Intel Xeon X7460 x86_64
4 CPU x 6 Cores/CPU
2.66 GHz, 32GB RAM
Fedora 10

More InnoDB Features in MySQL 5.5

- Monitoring & Diagnostics
  - Performance schema for InnoDB
  - Improved InnoDB transaction reporting
  - Log start and end of InnoDB buffer pool initialization to the error log
- UTF-32 support
- Significantly reduced the number of kernel objects (Windows)

Performance Diagnosis: Performance Schema in InnoDB

- Performance Schema in InnoDB
  - 46 mutexes
  - 14 rwlocks
  - 8 types of threads
  - 3 types of I/O (data, log, tmpfile)
- What do you get from Performance Schema
  - Mutexes / rwlock usage statistics and current status
  - IO statistics
  - Active running threads

Performance Schema in InnoDB

- Find out aggregated information from SUMMARY Tables

```
mysql> SELECT EVENT_NAME, COUNT_STAR, SUM_TIMER_WAIT, AVG_TIMER_WAIT
-> FROM EVENTS_WAITS_SUMMARY_BY_EVENT_NAME
-> WHERE EVENT_NAME like "%innodb%"
-> order by COUNT_STAR DESC;
```

EVENT_NAME	COUNT_STAR	SUM_TIMER_WAIT	AVG_TIMER_WAIT
buf_pool_mutex	1925253	264662026992	137468
buffer_block_mutex	720640	80696897622	111979
kernel_mutex	243870	44872951662	184003
purge_sys_mutex	162085	12238011720	75503
trx_undo_mutex	120000	11437183494	95309
rseg_mutex	102167	14382126000	140770
fil_system_mutex	97826	15281074710	156206
log_sys_mutex	80034	35446553406	442893
dict_sys_mutex	80003	6249472020	78115

InnoDB Metrics Monitor Table

- A light weight Monitoring system
- Work on System Resource counting as well as performance related counting
- Simple infrastructure that is extensible
- Information available to DBAs and Users to tune server configuration
- No measurable performance impact

MySQL Enterprise Edition

- Oracle Premier Lifetime Support Services
  - MySQL Consultative Support Services
- MySQL Enterprise Security
  - External Authentication for Windows and PAM
  - Integrates MySQL apps with existing infrastructures
- MySQL Enterprise Scalability
  - MySQL Thread Pool
  - Improves sustained performance/scale as connections grow
  - **20x scale** improvement in Sysbench OLTP RW benchmarks
- MySQL High Availability
  - Oracle VM Template for MySQL
  - Windows Clustering
- Oracle Product Integrations/Certifications
  - Manage MySQL with Oracle tools that are already in use

MySQL Enterprise Edition

- MySQL Enterprise Monitor
  - Global Monitoring of all MySQL and Cluster servers
  - Supports Thread Pool and Ext Auth commercial extensions
  - Supports MySQL Enterprise Backup Operations
  - Query Analyzer
  - Replication Monitor
  - Advisors and Alerts (SMTP, SNMP)
  - Integrated with MOS
- MySQL Enterprise Backup
  - Online Backup of InnoDB
  - Integrated with Oracle Secure Backup

InnoDB: Future

InnoDB Features in MySQL 5.6.2

- Kernel mutex split
- Multi threaded purge
- Configurable data dictionary cache
- InnoDB persistent statistics
- MRR/ICP support
- More counters in InnoDB Metrics Table
- I_S for InnoDB system tables
- InnoDB: Use rw_locks for page_hash
- Add 'page_cleaner' thread to flush dirty pages

InnoDB Features in MySQL 5.6.2

- kernel mutex splits into different mutexes for each of their functionalities
  - Server mutex – general mutex for global structures
  - Lock wait mutex - mutex protects the data structures required by a thread when it has to wait for a database lock.
  - Lock system mutex - This mutex protects the locking sub-system, in a sense this is the new kernel mutex. It has very tight coupling with transactions.
  - Transaction system read-write lock – protect structures for transaction creation and freeing, and read view creation
  - Read view list mutex - This mutex covers the operations on the trx_sys_t::view_list.

InnoDB Features in MySQL 5.6.2

- Multi threaded purge
  - The purge thread is responsible for removing delete marked records that will not be seen by any active transaction. It purges the records from the table and removes the relevant undo entries from the history list (a.k.a rollback segment).
  - Create several threads to do the InnoDB purge, Innodb-purge-threads := 0-32
- Use rw_locks for page_hash
  - In 5.5 we introduced multiple buffer pools to reduce contention on the buf_pool::mutex. In 5.6 to reduce this contention further we introduced a fix where page_hash in each buffer pool is protected by an array of mutexes. Each mutex protects a segment of page_hash. Now the mutex changes to rwlock, so that non-blocking S-lock can be held for most look-ups

InnoDB Features in MySQL 5.6.2

- Reduce contention during file extension
  - file extension in innodb does IO while holding fil_system::mutex. This blocks all other IO activity on all datafiles. Add new flags and mechanism to remove the mutex dependency during file extension.
- MRR/ICP support for InnoDB
  - MRR stands for multi-range read. It scans one or more index ranges used in a query, sorts the associated disk blocks for the row data, then reads those disk blocks using larger sequential I/O requests.

InnoDB Features in MySQL 5.6.2

- MRR/ICP support for InnoDB (contd.)
  - ICP stands for index condition pushdown - Instead of fetching entire rows to evaluate against a set of WHERE clauses, ICP sends those clauses to the storage engine, which can prune the result set by examining index tuples
- Monitoring & diagnostics
  - InnoDB Information Schema Metrics Table
  - Information schema system tables for InnoDB
  - Information schema table for InnoDB buffer pool
  - Dump all buffer pool page info

InnoDB Features in MySQL 5.6.3

- Improve LRU flushing
- Improve thread scheduling
- Reduce contention during file extension
- Increase max size of redo log files
- Improve deadlock detection performance
- Dump/Restore buffer pool for fast start up
- Use hardware checksum
- Separate tablespace(s) for the InnoDB undo log

InnoDB Features in MySQL 5.6.3

- Dump/Restore buffer pool for fast start up
  - innodb-buffer-pool-dump-now=ON
  - Innodb-buffer-pool-load-at-startup=ON
  - Dumping a 28G buffer pool on a running system took < 1s
- Improve thread scheduling
  - Take into consideration all waits including IO.
  - Atomics where possible to manage the scheduling

InnoDB feature in MySQL 5.6.3

- Improved CRC32 for page checksums
  - overhead of existing checksum in InnoDB is too high
  - new checksum code under BSD license that includes hardware-based implementation if it is supported by the CPU
  - innodb-checksum-algorithm := string
    - crc32, strict_crc32
      - uses CPU instructions to calculate CRC32 if the CPU supports them and falls back to a manual CRC32 calculation if the CPU does not have support for it.
      - Not backward compatible with versions < 5.6.3
    - innodb, strict_innodb
    - none, strict_none

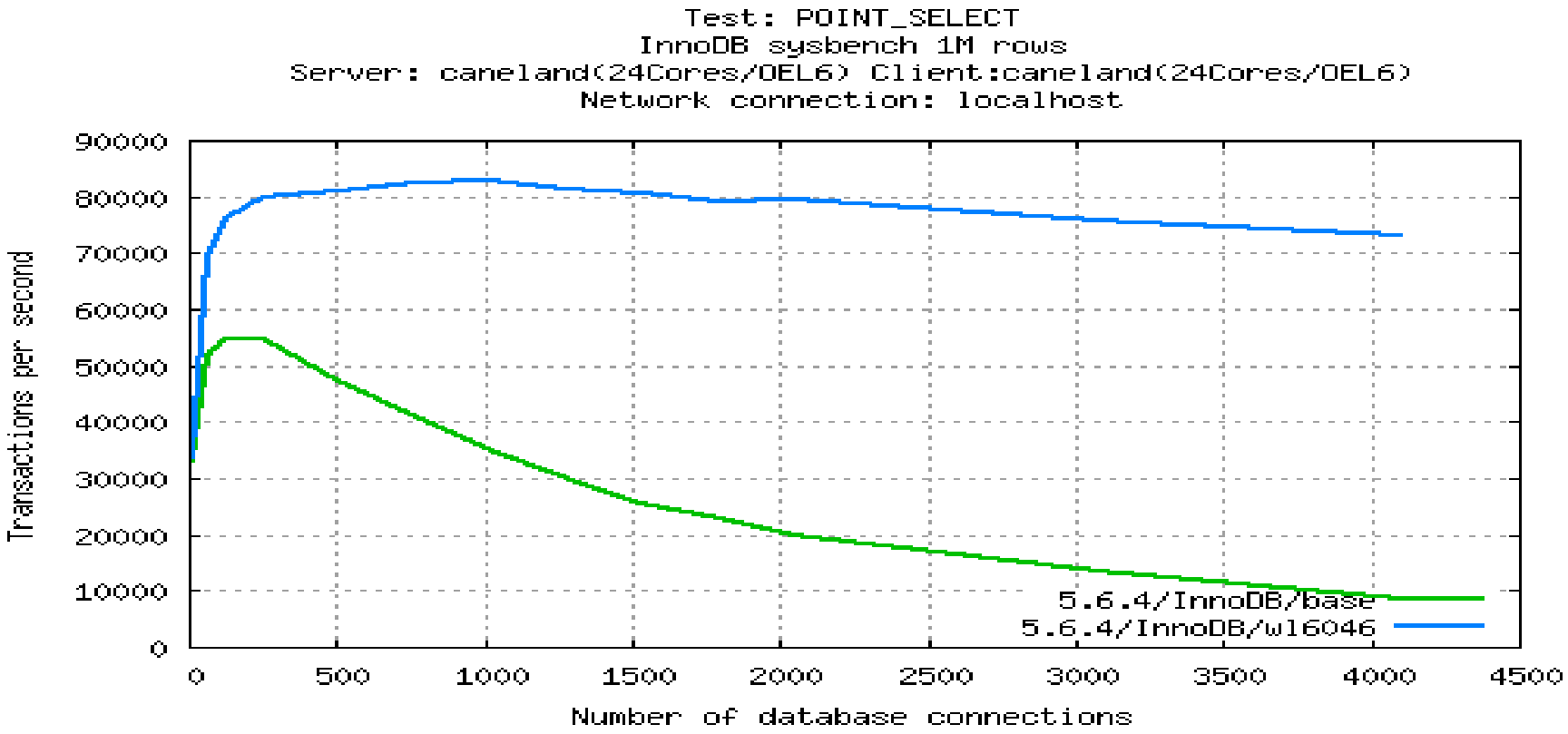
InnoDB Features in MySQL 5.6.4

- InnoDB full-text search
- Support 4k, 8k page sizes
- Special handling of read-only transactions

Special Handling of RO Transactions

- The read view is required for MVCC to work.
- Creating snapshot read views (for MVCC) is expensive, especially as the number of active transactions in the system increases.
- To reduce this overhead we can filter out those transaction ids that we know will not modify any records a.k.a READ-ONLY transactions.
- Read-only transactions
  - AUTOCOMMIT NON-LOCKING READ-ONLY SELECTs
  - General read only: Implement START TRANSACTION READ ONLY / READ WRITE;

MySQL 5.6 June – Sysbench RO Point Select



InnoDB Features in MySQL 5.6.6

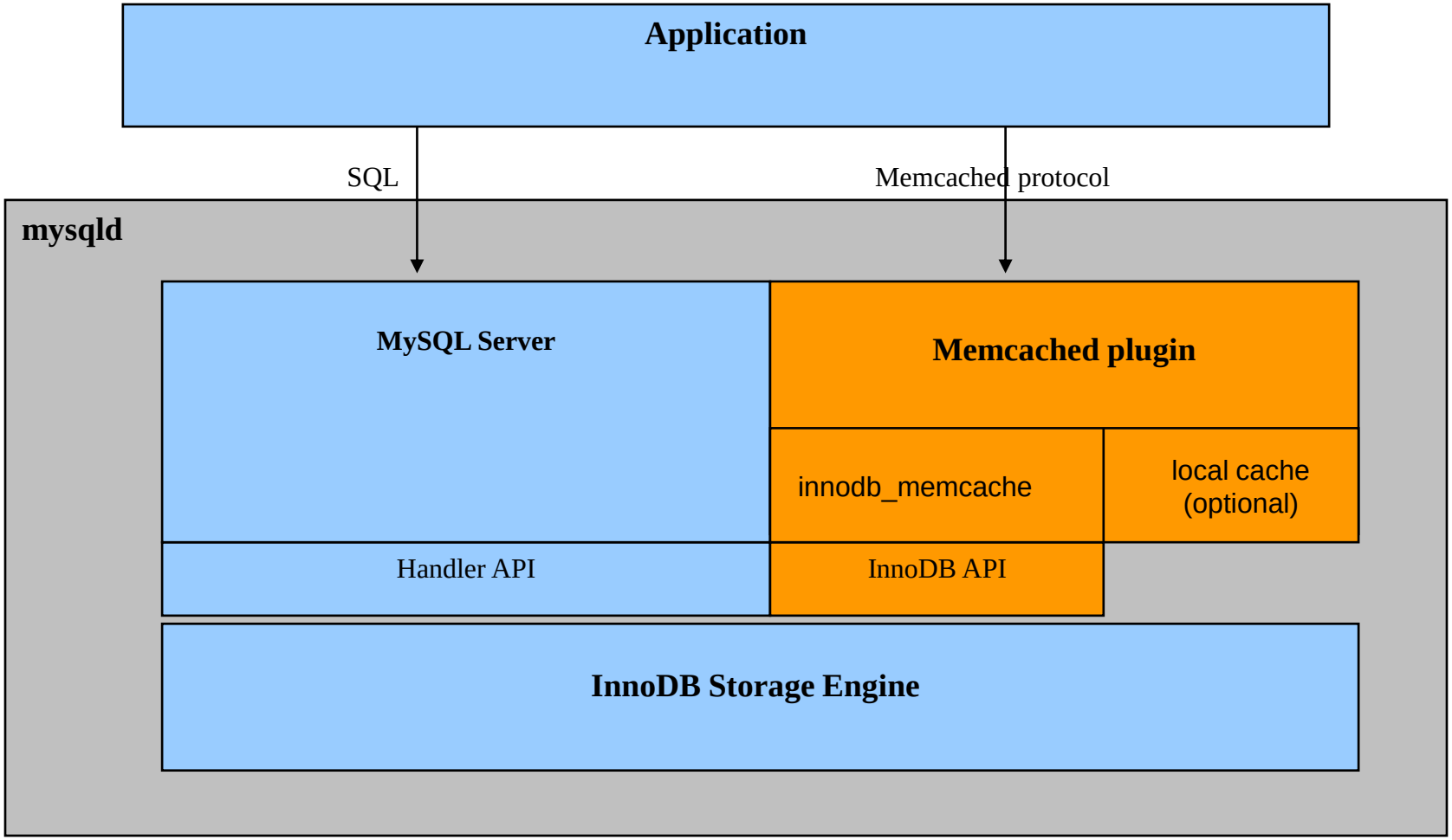
- Online Operations
  - Add Index
  - Add / Drop Foreign Key
  - Add / Drop Column
  - Rename Table
  - Rename Column
- Direct Access to InnoDB via Memcached
- Performance Enhancements
- Optimization for RO Transactions
- Transportable Tablespaces
- Tune InnoDB Persistent Statistics
- Tune Adaptive Flushing
- Improved Neighbor Flushing
- InnoDB Compression Enhancements
  - Configurable compression level
  - Optional to log compressed page images
  - Dynamic padding

InnoDB Features in MySQL 5.6.6

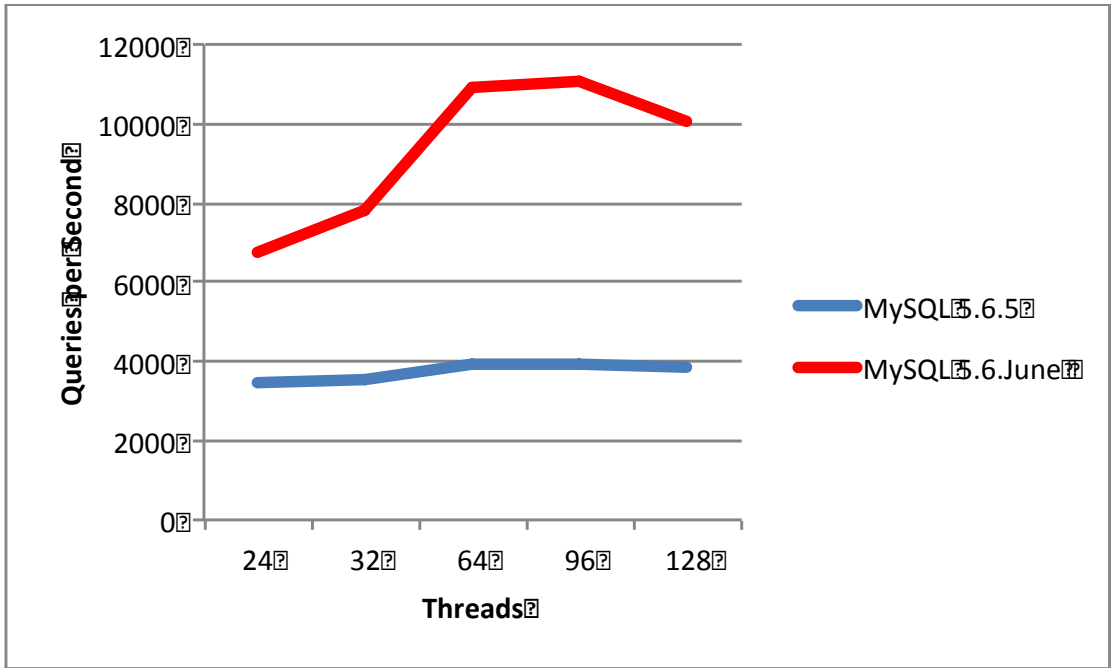
CREATE INDEX *index_name* ON *table name*, ALGORITHM=INPLACE

	Concurrent user	Source (table)	(cluster) Index	Metadata lock
Pre-Prepare Phase	Concurrent Select, Delete, Insert, Update	Check we support this Online DDL		Shared Metadata Lock
Prepare phase	Concurrent Read Only	Create temporary table for new cluster index	Allocate Log (files), Log starting	Metadata Lock that blocks Write
Build phase	Concurrent Select, Delete, Insert, update	Data scan Sort/merge Index build	**DML Logging. And Apply log at the end of create index**	Shared Metadata Lock
Final phase	No Concurrent DML allowed	Old table dropped (if create primary)	System table (Metadata) update	Exclusive Metadata Lock

NoSQL to InnoDB with memcached

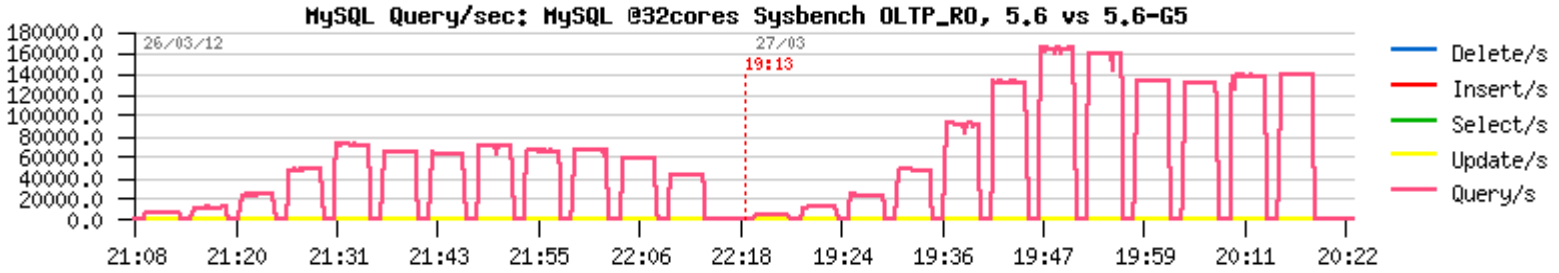


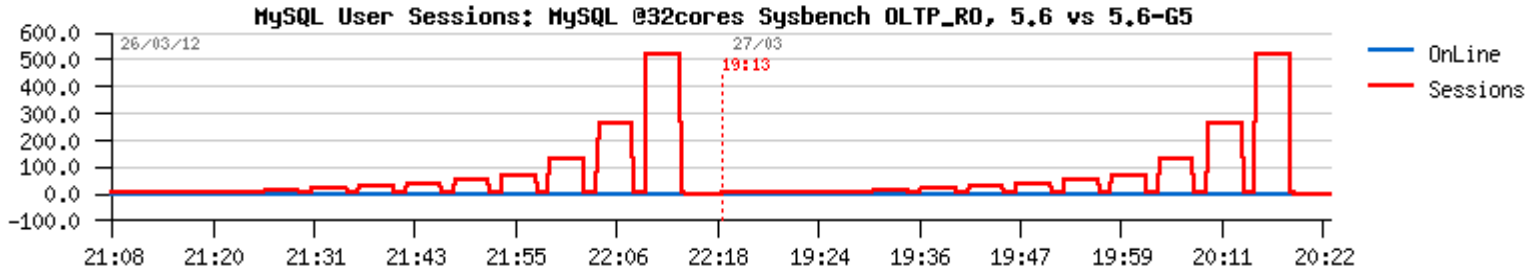
MySQL 5.6 June – Sysbench OLTP_RW



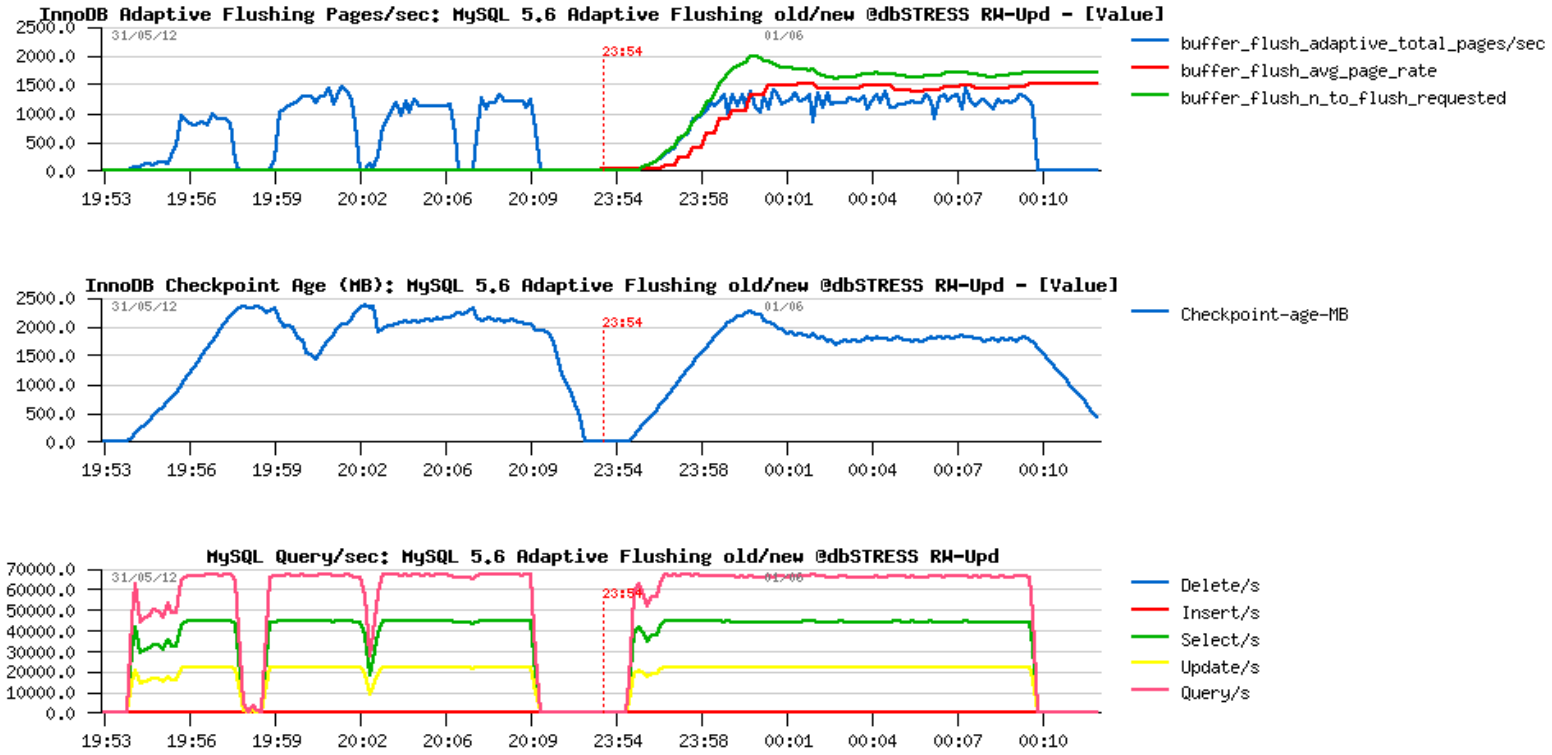
- **Up to 2.8x** higher performance
  - Removal of LOCK_open
  - Removal of CPU cache sharings
  - InnoDB flushing
- Sysbench R/W
  - 8 x Socket / 6-core Intel Xeon 7540, 2GHz
  - 512GB RAM
  - SSD

MySQL 5.6 June – Sysbench OLTP_RO

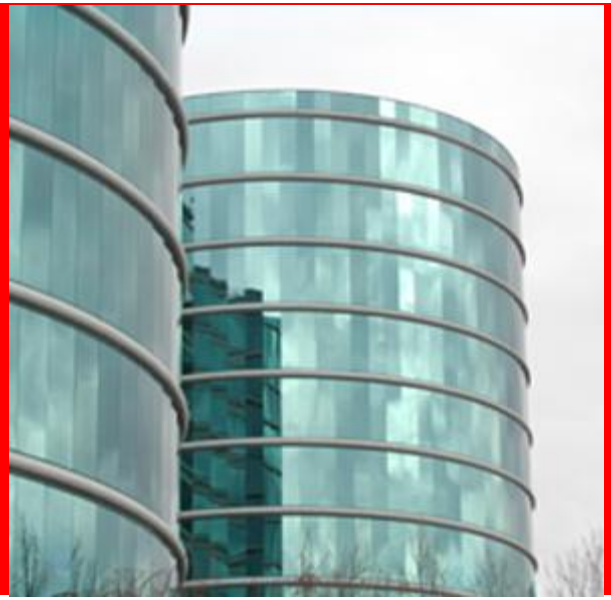




MySQL 5.6 June – Adaptive Flushing



ORACLE®



Thanks for attending!