



PERCONA  
Performance Consulting Experts

# Performance improvements in MySQL 5.5

Percona Live

Feb 16, 2011 San Francisco, CA

By Peter Zaitsev

Percona Inc

# Performance and Scalability

- Talk about Performance, Scalability, Diagnostics in MySQL 5.5
  - All either directly seen as Performance or provide a pathway to getting one
- Major Areas of Improvements
  - SQL Level
  - Innodb
  - Windows
  - Performance Schema

# Improvements to What ?

- We compare MySQL 5.5 to MySQL 5.1 with Innodb Plugin
- MySQL 5.1 by default uses very old Innodb code which has a lot more problems
- Some articles out there compare MySQL 5.5 to MySQL 5.1 without plugin showing more gains

# LOCK\_open

- “Table Cache” Operations did not scale well in MySQL
  - A lot of operations had to be done under global lock making it hot spot
  - Linear Search used for some operations
- MySQL 5.5 reduces contention introducing MDL (Meta Data Locking) Subsystem
  - Both optimizations for number of operations and reduced contention

# Smaller Optimizations

- LOCK\_alarm mutex
  - Replaced by SO\_SNDTIMEO/SO\_RCVTIMEO on platforms what support it
    - Most current do
- LOCK\_thread\_count mutex
  - Reduced usage to connect/disconnect from 1-2 times per query
- THR\_LOCK\_charset
  - Previously used for character set setup.  
Removed

# InnoDB Optimizations

- A lot of Scalability improvements
- Important performance optimization for Purging, IO Subsystem and Change Buffering

# Improved Log Sys Mutex

- Served “double duty” protecting
  - Log records and LSN
  - Pages in buffer pool changed than MTR is committed
- Changed to 2 separate mutexes
- Contention can still be observed by heavy log write workloads

# Separate Flush\_list Mutex

- Buffer Pool operations and Flush List operations were bound on very hot buffer pool mutex
- Flush list has its own mutex reducing pressure on buffer pool mutex
  - Which remains rather hot

# Multiple Buffer Pool Instances

- “buffer pool mutex remains hot”
- But we can now create multiple buffer pools to relief this problem
  - **innodb\_buffer\_pool\_instances**
- Pages will be hashed across these buffer pools
  - No user level table mapping etc
- Some will contain hot pages and may get hot
- Many buffer pools can increase internal overhead

# Multiple Rollback Segments

- InnoDB used to have single rollback segment
  - Both contention spot and limit to 1023 concurrent transactions (or less)
- Now 128 Segments
  - Reducing contention and giving up to 128K transactions
- **innodb\_fast\_shutdown=0** needed for upgrade
- Old InnoDB Versions will benefit from this change
  - They can use more than one segment, just do not create more than one

# Asynchronous IO on Linux

- Before MySQL 5.5 InnoDB would use Native Async IO only on Windows
- On Linux it was “Simulated” by using helper threads
- Assumes O\_DIRECT
  - Good to use anyway
- More outstanding IO requests
- Pushes burden of merging io requests to Kernel
- Can make crash recovery up to 1.5x faster

# Change Buffering

- Insert Buffer existed in Innodb forever
  - Speeding up Inserts for IO bound workloads
- Updates and Deletes were a lot slower, unable to use this feature
- MySQL 5.5 support “delete mark” and “purge” change buffering
  - It handles update case as well as delete because secondary indexes are never updated in place
- Up to 40x performance improvement for large deletes

# Separate Purge Thread

- Before MySQL 5.5 purging of old records is done by main thread
  - Making it stall on other activities while doing purge
  - Making purging lag more often than it should
  - Very large “History Length” with purge thread unable to keep up a big problem
- **innodb\_purge\_threads=1** to use separate purge thread

# InnoDB Adaptive Flushing

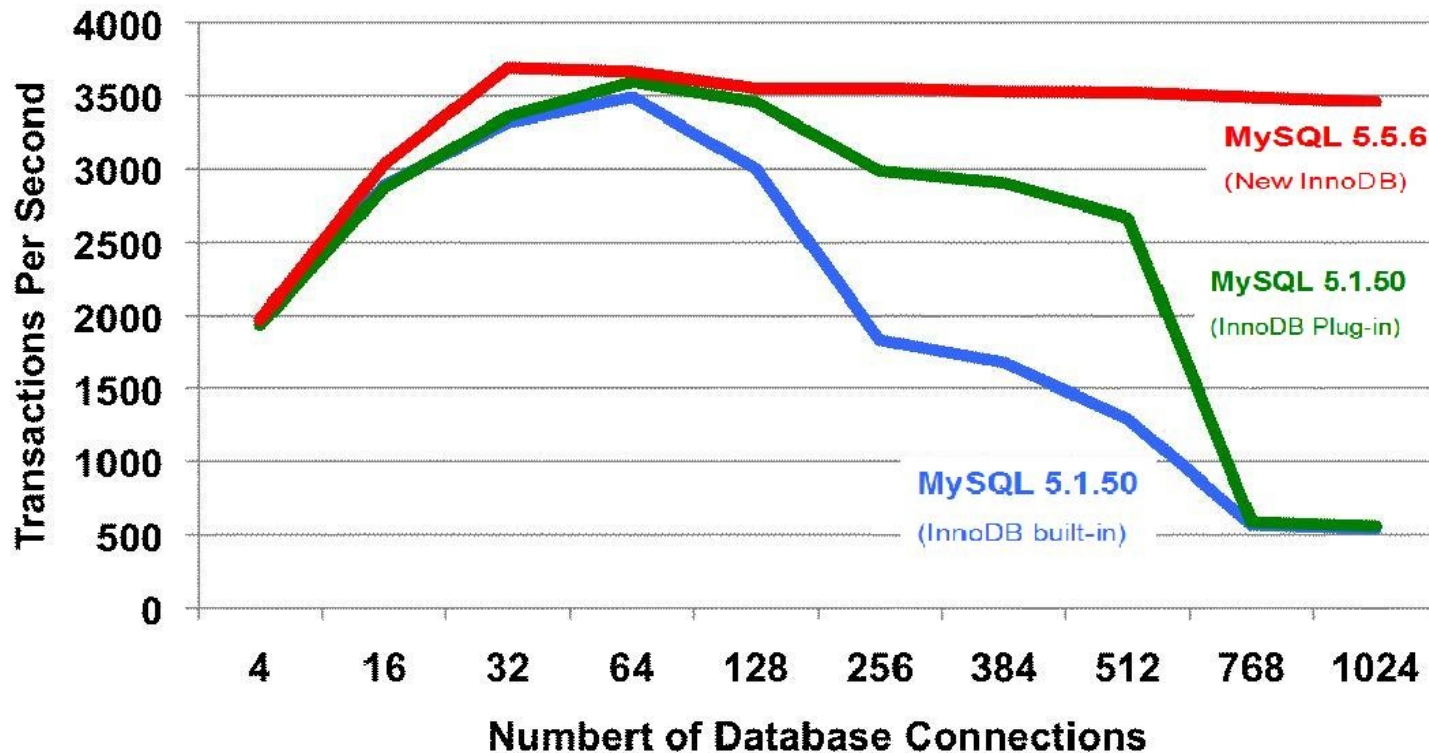
- Not a new feature in MySQL 5.5
- Was adjusted to deal with multiple buffer pools
- Getting constants small tunings
- MySQL 5.5 should be a lot “smoother” on many workloads
- This will likely continue to improve as MySQL 5.5 matures

# Windows Optimizations

- Using Native Windows calls instead of POSIX subsystem
- Use CriticalSection for mutexes and use native ConditionVariables in newer Windows OS
- Windows Optimizations graphs from Innodb Blog

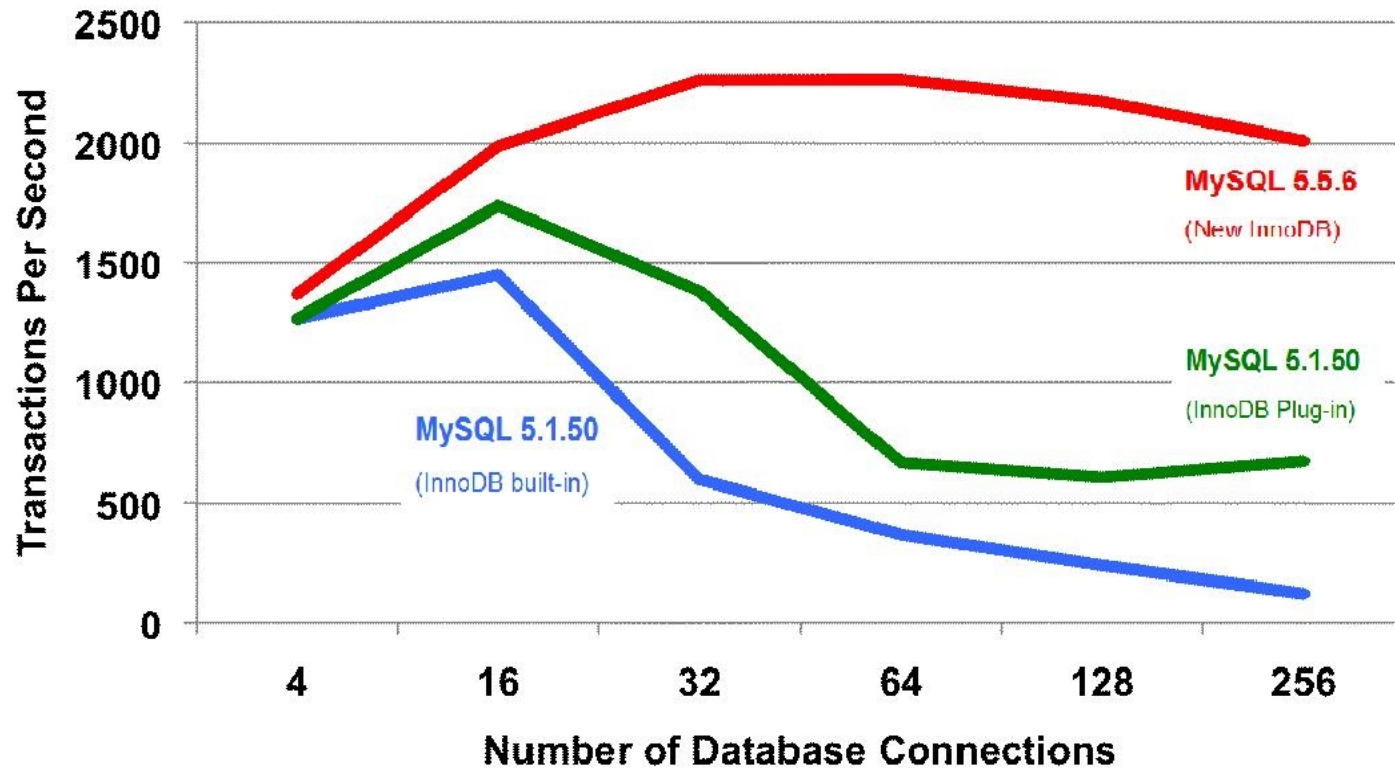
# Windows Reads

## MySQL 5.5 vs. 5.1 - Read Only



# Windows Writes

## MySQL 5.5 vs. 5.1 - Read Write



# Performance Schema

- Performance Schema includes many Innodb Probes
- Can be used by developers and by Users to better understand sources of Performance problems
- Hence contributing to further Innodb improvements

# Remaining issues in 5.5

- Per Index Lock
  - Restricts performance for high update workloads
  - Use Partitioning as partial Workaround
- Kernel Mutex
  - Seems to be significant focus for Innodb optimization in MySQL 5.6

# Remaining Issues MySQL 5.5

- Dropping Tablespace with Innodb File Per Table
  - Operation requires scan of LRU list
  - So the larger is buffer pool the slower it gets
- Log Mutex
  - Still causing contention
- Group Commit
  - Is “fixed” now only if you chose not to use **sync\_binlog=1** and so lose binary logs on crash

# What about Percona Server

- MySQL 5.5 and Percona Server 5.1 work on many similar bottlenecks, though approaches sometimes different
- Percona Server is focused on more simple changes to ensure shorter time to market
- MySQL 5.5 is often faster than Percona Server 5.1 on Multi Core CPUs with Many Connections
- Percona Server 5.5 Beta is available combining best changes from both code bases
  - <http://www.percona.com/software/>

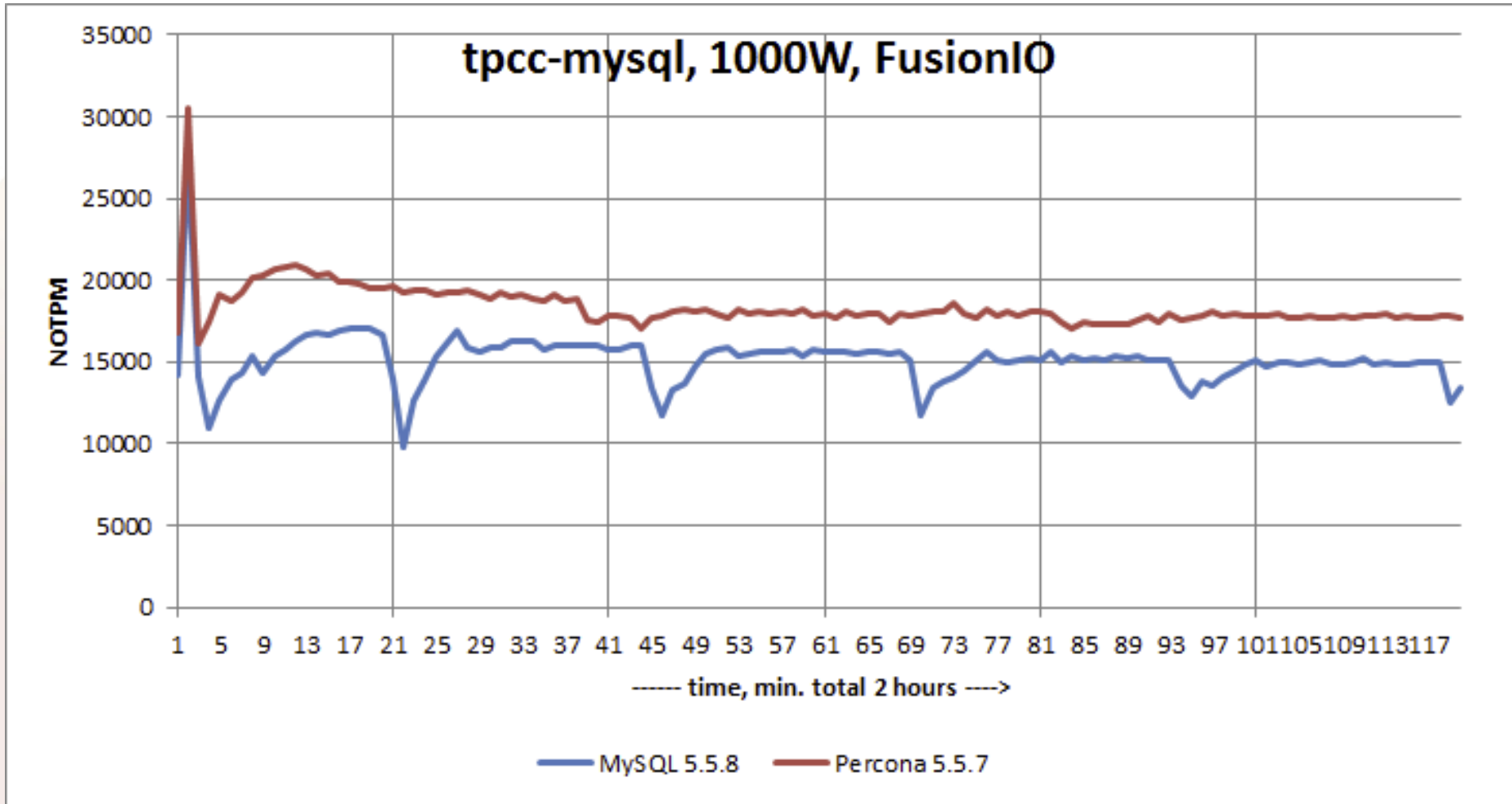
# Ideas Exchange

- MySQL Changes are released under GPL
- We review developments in current and next version and use the code/ideas if it gives good performance
- Same goes back
  - We're making our code available for Oracle to integrate, though no changes were taken directly after Ken Jacobs left
- Focus on “short time to market” allow to see safe parts of the future today

# LOCK\_open

- “Table Cache” Operations did not scale well in MySQL
  - A lot of operations had to be done under global lock making it hot spot
  - Linear Search used for some operations
- MySQL 5.5 reduces contention introducing MDL (Meta Data Locking) Subsystem
  - Both optimizations for number of operations and reduced contention

# MySQL 5.5 vs Percona Server 5.5



# Questions ?

- Thanks for coming !
- Contact at [pz@percona.com](mailto:pz@percona.com)
- Do not forget to come to Ignite show at 7PM