



PERCONA
Performance Consulting Experts

Introduction to Innodb Performance Tuning

April 22, 2009

Percona Performance
Conference

Santa Clara, CA

by Peter Zaitsev, Percona Inc

Where tune Innodb

- MySQL Settings
- Schema
- OS/Hardware
- Application

MySQL Settings

- Very important to adjust them for Innodb
 - Defaults are impractical
- Innodb does not do any “self tuning” for size of the system

innodb_buffer_pool_size

- By far most important settings
- InnoDB caches data and indexes here
 - And keeps additional structures like locks, adaptive hash buffer
- Allocate memory for OS and other buffers and use the rest
- Note InnoDB allocates some 10% more than specified due to overhead
- Watch for small amount of innodb tables
 - Having buffer pool much larger than data is a waste

innodb_log_file_size

- Important for write intensive workloads
- Larger log files = less data flushes
- Larger log files = longer recovery time
- Ways to practically define it
 - Looking at time it takes to recover
 - Looking at amount of data written to the logs
 - **Innodb_os_log_written**

innodb_flush_log_at_trx_commit

- Defines your transaction durability
 - 1 - “fsync” log at each transaction commit
 - If power goes down data is safe with good hardware
 - 2 - only “flush” log to operating system cache
 - If MySQL crashes data is safe
 - 0 - do not bother flushing
 - InnoDB will flush once per second by background thread
 - Likely data loss in case of MySQL crash
- Also check **sync-binlog**
 - If you want to safe replication or point in time recovery

innodb_flush_method

- How InnoDB will flush data to the disk
- “default” - use fsync()
 - Caused “double buffering” data in OS cache and buffer pool
- O_DIRECT
 - Bypass OS cache – avoids double buffering
 - Restricts parallel request execution
 - Can be slow if RAID does not have BBU
 - Some filesystems as ext3 has inode level locking
 - Parallel IO to the same file may be restricted.

Other Options

- **innodb_file_per_table=1**
 - Keep each InnoDB table in separate file.
- **innodb_thread_concurrency**
 - Restrict number of threads in InnoDB kernel
 - 0 or 8-16 can be good values
- **innodb_log_buffer_size**
 - Watch amount of data written in the log
 - Flushed each second and on transaction commit
 - 4MB is large enough unless working with huge blobs.

XtraDB options

- XtraDB – InnoDB Based Storage engine by Percona
 - <http://www.percona.com/percona-lab.html>
- **innodb_extra_undoslots=1**
 - Support more than 1024 concurrent transactions
- **innodb_read_io_threads,innodb_write_io_threads**
 - Use more threads for background IO
- **innodb_io_capacity**
 - Set capacity of IO subsystem
- **innodb_adaptive_checkpoint**
 - Even out checkpointing process; avoid stalls.



Schema

Clustering by Primary Key

- Close PK values are in the same page
 - Clustering is per page only, not global
- Lookups by PK are fastest
- Primary key ranges are very fast
 - Design your schema to get advantage of this
 - Using (user_id,message_id) as a key instead of auto_increment message_id
- “Random” primary keys such as md5() are bad.

Indexes

- InnoDB tables are typically much larger than MyISAM
 - InnoDB plugin offers compression
- Indexes are especially increased
 - No prefix compression (unlike MyISAM)
 - Have transaction visibility information
- Keep Primary key short
 - Indexes refer data by Primary Keys
- UNIQUE indexes are expensive
 - Do not use Insert buffer

Blobs

- Beware of large blobs
- Rows over 8K has to have some data stored outside of data page.
- At least 16K will be allocated for each object
- Row fragmentation
- Increased space usage
- Storing compressed data often useful
- Blobs will not be read if they are not in select list.

OS/Hardware

- Linux the most commonly used platform
 - Sun did plenty of work with Solaris
- Chose recent OS version
 - Optimizations; improvements in hardware support.
- Fast CPUs, Large caches
- Efficient use of multi cores may vary
 - Depends on workload; though improved a lot
- 64bit hardware and OS is a must
 - If you're working with any reasonable data size

OS/Hardware

- Raid with BBU is very important
 - Especially in “durable” configuration
- Invest in memory
 - Many problems are gone just if you have enough memory
- Flash Storage/SSD starts to look very attractive
- RAID10 is best for heavy read/write
- Consider XFS if running Linux
 - Ext3 has IO serialization issues

Optimizing Applications

- **Use Transactions**
 - Otherwise they are implicit per statement
 - Though do not keep them open for too long
- **Get benefit of MVCC**
 - Selects do not block anything
 - You do not have to worry about table locks
- **Do not use LOCK TABLES**
 - This is most likely MyISAM legacy and not needed.

Application Optimizations

- Keep updates fully indexed
 - All rows which are traversed will be locked
- Watch out for deadlocks
 - Handle them in the applications
 - Keep same locking order to minimize
 - Consider using `SELECT ... FOR UPDATE` to prelock data
 - `SELECT ... GET_LOCK('mylock')` is often helpful
- Be careful with foreign keys
 - Require more indexes than you may need otherwise
 - Can result in tricky unexpected locking



The End