Optimizing MySQL Configuration

Peter Zaitsev,CEO
Percona Live, Washington DC
11 January 2012

Agenda

- Approach to getting great MySQL Configuration
- Types of Configuration Options
- Tools to Configure MySQL
- Looking at Most Important Options

But Before We start

- How many settings do you set in Config File ?

Also

- How useful do you think MySQL Default Configuration ?

Configuration Tuning Basics

- Understand what you're changing
  - Google Copy/Paste without thinking can be bad
- Avoid Obsessive Tuning Disorder
  - Setting 10 settings will give 95% of possible performance in 95% cases
- Beware of “Sample Configs” In MySQL distributions
  - They are pretty outdated
  - 2GB of memory is “huge” these days ?
    - Right for Cell Phone

Most Options do not Scale

- Going to Server with 4x memory you can't just multiply all configuration variables 2x

Know Scope and Unit

- sort_buffer_size=16G
  - Wrong! sort_buffer_size is set per connection
- table_cache=64M
  - Wrong! table_cache is set in elements not memory size.

Avoid Basic Mistakes

- Setting variables in wrong config file
  - /etc/mysql/my.cnf instead of /etc/my.cnf
    - These depend on Linux Distro, Beware
- Duplicating Options
  - Last option will override previously set
- Not knowing Synonyms
  - **table_cache** is same as **table_open_cache**
- Using wrong section for option
  - Server reads [mysqld], client [mysql]

Config Management Practices

- Keep Config Files in Sync on different servers
  - Out of Sync config files is frequent cause of mistakes and confusion
- Keep Record of Changes
  - Config files under version control is great

Why would set an option

- Sanity Check Options
  - Many do not impact Performance at all
- General Options to set for Hardware and Workload
  - Few of them
- Special Options
  - Defaults generally fine, unless your circumstances are very special

Do not let MySQL Swap

- Allocating too much memory and having MySQL swapping is a lot worse than not using all memory
- Monitor swapping (si/so from vmstat closely)
- Start with safe buffer values and increase them gradually if a lot of memory stays free

Automated Configuration Tuning

- Tools which claim to create best configuration by looking at status variables
  - Beware. Recommendations are often poor
- Advisory Tools
  - Tools which check your config file for typical mistakes and omissions
- Basic configuration creation tools
  - Do not claim to do magic but can get your started with better configuration than default

mysqtuner

-------- General Statistics --
[--] Skipped version check for MySQLTuner script
[OK] Currently running supported MySQL version 5.1.57-rel12.8-log
[OK] Operating on 64-bit architecture

-------- Storage Engine Statistics -------------------------------------
[--] Status: +Archive -BDB -Federated +InnoDB -ISAM -NDBCluster
[--] Data in MyISAM tables: 73G (Tables: 1282)
[--] Data in InnoDB tables: 1G (Tables: 338)
[--] Data in MEMORY tables: 0B (Tables: 2)
[!!] Total fragmented tables: 110

-------- Security Recommendations -------------------------------------
[!!] User '@' has no password set.

Mysqltuner (2)

-------- Performance Metrics ---

- [--] Up for: 157d 10h 0m 23s (533M q [39.219 qps], 8M conn, TX: 1202B, RX: 146B)
- [--] Reads / Writes: 97% / 3%
- [--] Total buffers: 4.3G global + 2.7M per thread (200 max threads)
- [OK] Maximum possible memory usage: 4.8G (40% of installed RAM)
- [!!] Slow queries: 7% (41M/533M)
- [OK] Highest usage of available connections: 54% (109/200)
- [OK] Key buffer size / total MyISAM indexes: 4.0G/1.3G
- [OK] Key buffer hit rate: 100.0% (45B cached / 6M reads)
- [!!] Query cache is disabled
- [OK] Sorts requiring temporary tables: 0% (44K temp sorts / 60M sorts)
- [!!] Joins performed without indexes: 255685
- [!!] Temporary tables created on disk: 41% (25M on disk / 61M total)
- [OK] Thread cache hit rate: 91% (791K created / 8M connections)
- [!!] Table cache hit rate: 2% (1K open / 59K opened)
- [OK] Open file limit used: 32% (2K/8K)
- [OK] Table locks acquired immediately: 99% (436M immediate / 436M locks)
- [!!] InnoDB data size / buffer pool: 1.1G/256.0M
-

mysqltuner(3)

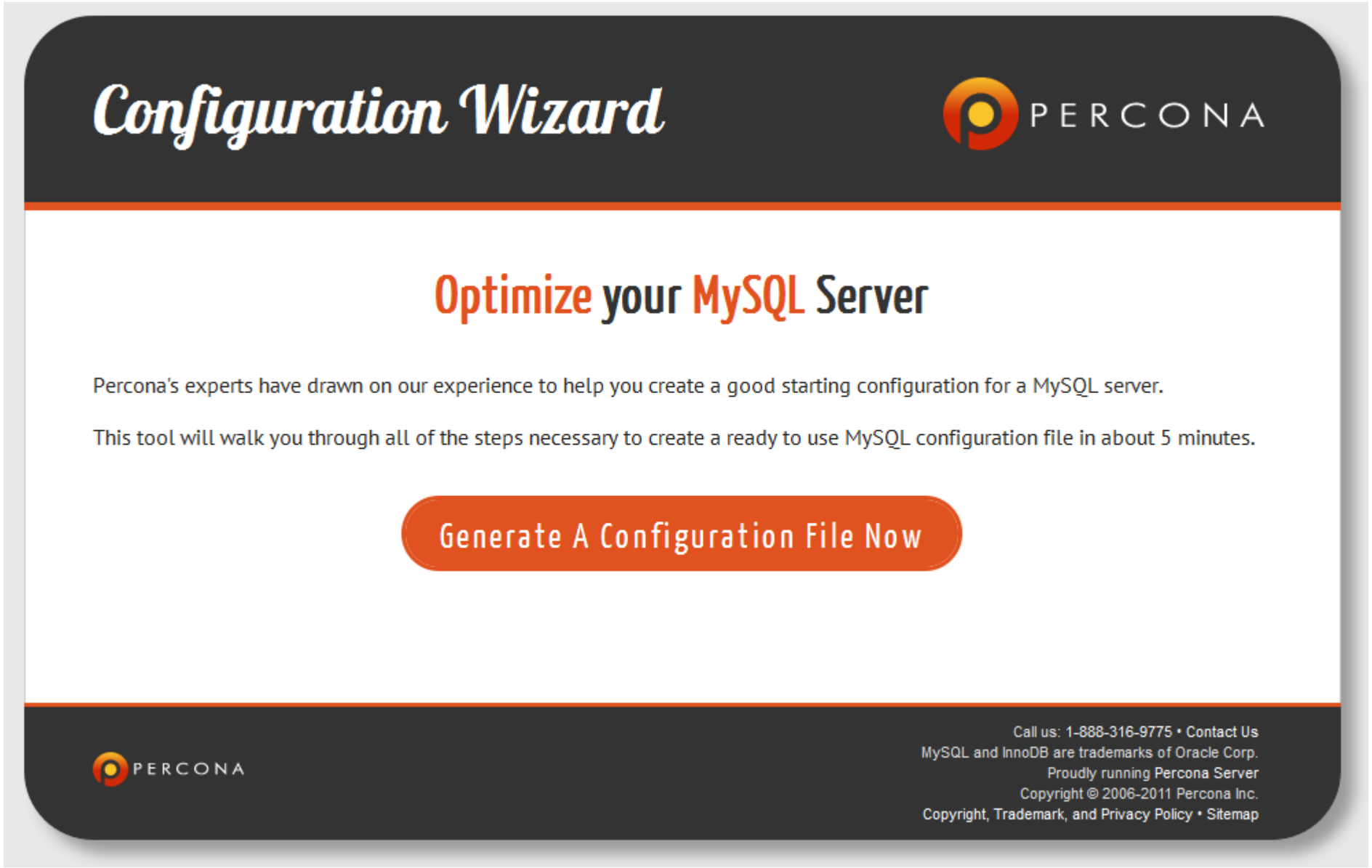
-------- Recommendations ---

- General recommendations:
- Run OPTIMIZE TABLE to defragment tables for better performance
- Adjust your join queries to always utilize indexes
- When making adjustments, make tmp_table_size/max_heap_table_size equal
- Reduce your SELECT DISTINCT queries without LIMIT clauses
- Increase table_cache gradually to avoid file descriptor limits
- Variables to adjust:
- query_cache_size (>= 8M)
- join_buffer_size (> 128.0K, or always use indexes with joins)
- tmp_table_size (> 16M)
- max_heap_table_size (> 16M)
- table_cache (> 4096)
- innodb_buffer_pool_size (>= 1G)

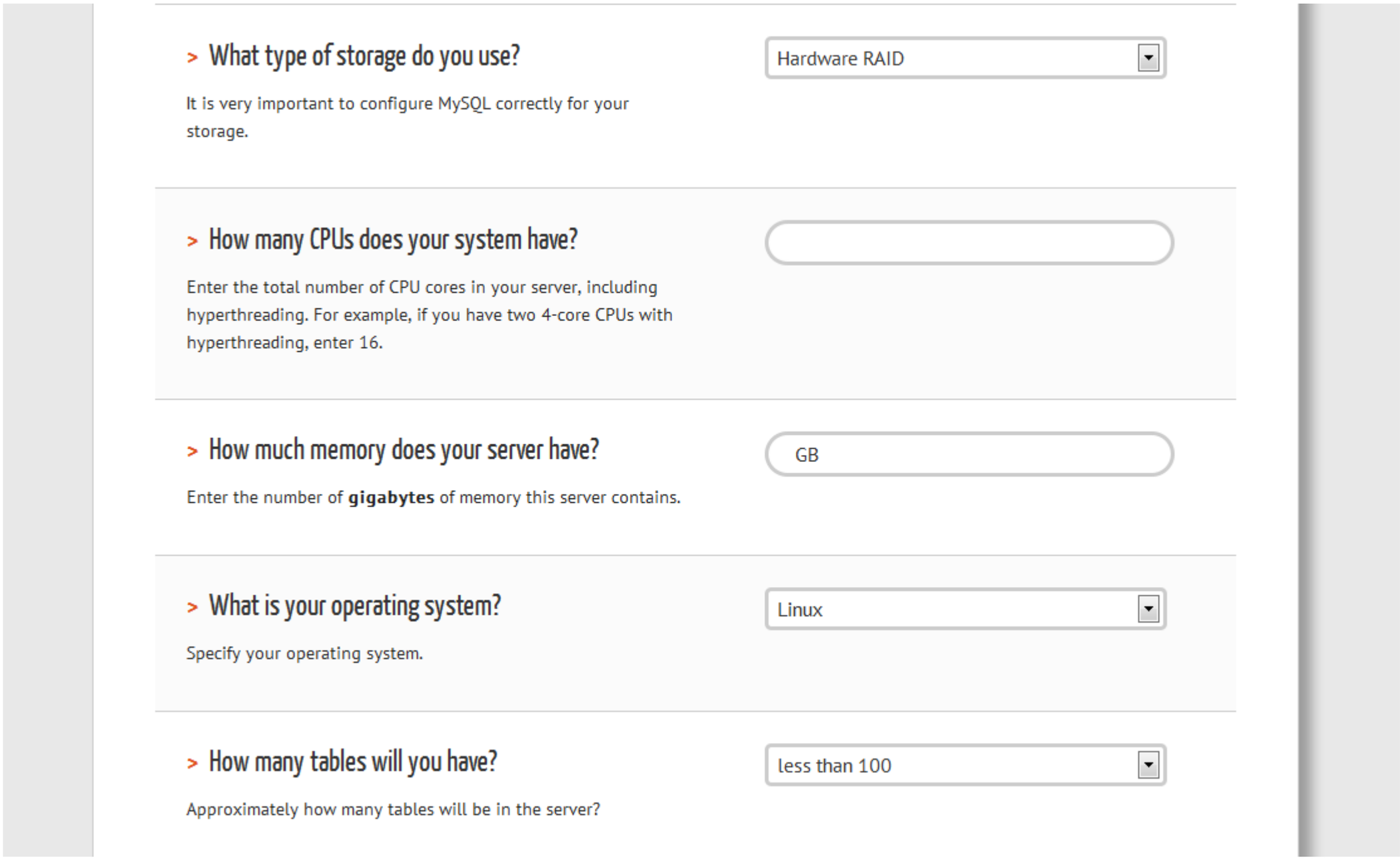
pt-variable-advisor

- # WARN innodb_flush_log_at_trx_commit-1: InnoDB is not configured in strictly ACID mode.
-
- # NOTE innodb_max_dirty_pages_pct: The innodb_max_dirty_pages_pct is lower than the default.
-
- # NOTE log_warnings-2: Log_warnings must be set greater than 1 to log unusual events such as aborted connections.
-
- # NOTE max_connect_errors: max_connect_errors should probably be set as large as your platform allows.
-
- # WARN old_passwords: Old-style passwords are insecure.
-
- # WARN slave_net_timeout: This variable is set too high.
-
- # NOTE innodb_data_file_path: Auto-extending InnoDB files can consume a lot of disk space that is very difficult to reclaim later.
-
- # WARN myisam_recover_options: myisam_recover_options should be set to some value such as BACKUP,FORCE to ensure that table corruption is noticed.
-
- # WARN sync_binlog: Binary logging is enabled, but sync_binlog isn't configured so that every transaction is flushed to the binary log for durability.
-

tools.percona.com



Tools.percona.com



Lets look at the options now

- Different classes of options:
  - General Options
  - MyISAM
  - Innodb
  - Visibility and Logging

Getting Status Variables

- We refer to **SHOW GLOBAL STATUS** output in many descriptions
- Percona Toolkit too pt-mext is helpful
- **pt-mext -r -- mysqladmin ext -i100 -c4**

Aborted_clients	128	0	0
Aborted_connects	909	0	0
Binlog_cache_disk_use	3	0	0
Binlog_cache_use	262857	0	0
Bytes_received	146518902681	580976	459113
Bytes_sent	1202983049426	1417886	1018617

General Options

- **max_connections**
  - How many connections to allow ? Watch **max_used_connections** status value
- thread_cache
  - Cache to prevent excessive thread creation
  - 50-100 is good value. Watch **threads_created**
- **table_cache/table_open_cache**
  - Cache of opened table instances
  - Single table may have multiple entries
  - Watch opened_tables status value
  - Start with 4096
    - MySQL will only use as needed anyway.

General Options

- **open_files_limit**
  - MyISAM tables require up to 2 file handlers
  - Each connection is file handler too
  - Safe to set to 65535 in most systems
- **table_definition_cache**
  - Cache table definitions (CREATE TABLE)
  - Only one entry per table
  - Watch **Opened_table_definitions**
  - Set to cover all tables unless 50K+ tables

General Options

- **back_log**
  - Need adjustment if many connections/sec
  - 2048 is reasonable value
- **max_allowed_packet**
  - Limits maximum size of query
  - Limits internal string variable size
  - 16MB is a good value
- **max_connect_errors**
  - Prevent password brute force attack
  - Can cause “Host Blocked” error messages
  - Value around 1000000 is good

General Options

- **skip_name_resolve**
  - Avoid DNS lookup on connection. Faster and Safer
  - Do not use host names in **GRANT**s
- **old_passwords**
  - Should NOT be enabled. Will cause insecure password hash to be used.

General Options

- **log_bin**
  - Enable for replication and point in time recovery
  - Set to “mysql-bin” to avoid default naming
- **sync_binlog**
  - Make Binlog durable. Set to 1 if have RAID with BBU or Flash
    - Can be really performance killer with slow drives.
- **expire_log_days**
  - Purge old binary logs
  - 14 (2 weeks) is a good value with weekly backups.

General Options

- **tmp_table_size**
- **max_heap_table_size**
  - Typically set to same value (workload based)
  - **Created_tmp_disk_tables** status variable
    - Bevare BLOB/TEXT fields cause on disk table with any size.
- **query_cache_size**
  - Enable query cache only if it is tested to provide significant gains
  - Often causes stalls and contention
  - Do not set over 512MB

General Options

- **sort_buffer_size**
  - In memory buffer used for sorting
  - Watch sort_merge_passes
  - Consider setting for session for large queries
  - Values up to 1MB are good default
  - Large values hurt performance of small queries
- **join_buffer_size**
  - Helps performance of Joins with no indexes
  - Better get rid of such Joins !
  - 8MB can be reasonable value
- **default_storage_engine**
  - Use this engine for tables if not specified

General Options

- **read_rnd_buffer_size**
  - Buffer for reading rows in sorted offer
  - Specifies Maximum Value
  - Values around 16MB often make sense
  - Do not mix with **read_buffer_size**
- **Tmpdir**
  - Specify location of temporary directory
  - Tmpfs often good choice unless very large temporary space is needed.

MyISAM options

- **key_buffer_size**
  - Cache MyISAM Indexes.
  - Does Not cache data.
  - Up to 30% of memory if using MyISAM only
- myisam_recover
  - Automatically repair corrupted MyISAM tables after crash. **BACKUP,FORCE** is a good value.
- **myisam_sort_buffer_size**
  - Buffer used for building MyISAM indexes by Sort. 8MB-256MB are good values.

MyISAM Options

- **low_priority_updates**
  - Allow higher concurrency for SELECTs
  - May starve update queries
- **bulk_insert_buffer_size**
  - Buffer to optimize Bulk Inserts
  - Values of ¼ of **key_buffer_size** make sense
  - Note it is per connection value
-

Innodb – Memory Settings

- **innodb_buffer_pool_size**
  - The most important setting. Often 80%+ of memory is allocated here.
- **innodb_buffer_pool_instances**
  - Reduce contention. Set to 4+ in MySQL 5.5+
- **innodb_log_buffer_size**
  - Buffer for log files. Good Values 4MB-128MB
  - Not only reduce writes but help contention
- **innodb_ibuf_max_size**
  - Control size of Insert buffer. Default is ½ of Buffer pool. Smaller values are good for SSD

Innodb IO Options

- **innodb_flush_log_at_trx_commit**
  - Control Durability
  - 1=flush and sync; 2=flush; 0=neither
- Innodb_flush_method
  - Controls how Innodb Performs IO
  - **O_DIRECT** good value for most servers
- **innodb_auto_lru_dump**
  - Percona Server Feature to warmup quickly
  - 300 (seconds) is a good value
- **innodb_io_capacity**
  - Controls Innodb Assumption about Disk Performance. Increase for faster drives.

Innodb IO Options

- **innodb_read_io_threads**
- **innodb_write_io_threads**
  - Control number of threads doing reads and writes
  - MySQL 5.5 has async IO so very high values might not be needed
  - 4 is good default. Higher for large IO systems.
- **innodb_flush_neighbor_pages**
  - Percona Server feature to control how flushing works
  - Disable (set to 0) for SSD

Other Innodb Options

- **innodb_log_file_size**
  - Size of redo log file. Larger logs better performance but longer recovery.
- **innodb_log_files_in_group**
  - Leave at 2 which is default.
- **innodb_file_per_table**
  - Store each Innodb table in separate file. Usually Good choice
- **innodb=force**
  - Enable so MySQL does not start if Innodb could not initialize. Otherwise it might start but error on access to all Innodb tables.

Other Innodb Options

- innodb_data_file_path
  - Settings for Innodb System Tablespace
  - Use one file. Limit growth as you can't shrink it
  - **ibdata1:10M:autoextend:max:10G**
- innodb_lock_wait_timeout
  - How long to wait for row level locks before bailing out ?
- **innodb_old_blocks_time**
  - Helps to make buffer pool scan resistant
  - Values around 1000 make sense

Other Innodb Options

- **innodb_file_format**
  - Which file format Innodb will use
  - “Antelope” is default legacy format
  - “Barracuda” allows to use new features like compression
- **innodb_stats_on_metadata**
  - Update statistics on meta data access
  - Such as Information_schema queries
  - Typically best disabled for more workloads
    - Set to 0
  - Innodb will still refresh stats when table changes significantly

Visibility Options

- **performance_schema**
  - Enable Performance Schema in MySQL 5.5+
  - Watch potential overhead.
- **log_slow_queries**
  - Enable Slow Query Log. Old but very helpful.
- **long_query_time**
  - Especially with long_query_time set to 0 periodically to get sample of the load
- **log_slow_verbosity=full**
  - Get a lot more data about queries in Percona Server

Visibility Options

- **low_warnings=2**
  - Get warnings about disconnects and other minor issues in error log.
  - More information but it can get spammy
- **userstat_running=1**
  - Get advanced Table and Index usage statistics in Percona Server and MariaDB

Summary

- Many Options to chose from !
- And these are selected “few”
- Close to 400 variables available In latest versions
- Remember in most cases you do not need to tune more than a few
- Consider starting with config file generated by http://tools.percona.com
  - At least it will show you which options to pay attention to first.

Main Body Slides

- Use this layout for the main body of your presentation
- Keep the following slides at the end of your presentation:
  - Percona Live DC Sponsors
  - Annual MySQL Conference & Expo
  - Percona Training (2 slides)
  - Your Contact Information
  - Percona Live Closing Slide with no text inserted for Q&A period

Percona Live DC Sponsors

Media Sponsor

Friends of Percona Sponsor

MySQL Conference & Expo 2012
Presented by Percona Live

The Hyatt Regency Santa Clara & Santa Clara Convention Center

April 10th-12th, 2012

Tutorials & Initial Breakout Sessions Announced

http://www.percona.com/about-us/pressreleases/percona-live-mysql-conference-and-expo-heads-to-santa-clara-california-april-10-12-2012/

Featured Speakers

Mark Callaghan (Facebook), Jeremy Zawodny (Craigslist), Marten Mickos (Eucalyptus Systems)

Sarah Novotny (Blue Gecko), Peter Zaitsev (Percona), Baron Schwartz (Percona)

Learn More at www.percona.com/live/mysql-conference-2012/

Want More MySQL Training?

Percona Training in Washington DC Next Week

January 16th-19th, 2012

MySQL Workshops

MySQL Developers Training - Monday

MySQL DBA Training - Tuesday

MySQL InnoDB / XtraDB - Wednesday

MySQL Operations – Thursday

Use Discount code DCPL30 to save 30%

Visit http://bit.ly/dc-training

Upcoming Percona MySQL Training

Washington, DC – January 16, 2012

Vancouver, Canada - February 6, 2012

Frankfurt, Germany - February 13, 2012

Irvine, CA – February 14, 2012

Denver, CO - February 20, 2012

San Francisco, CA - March 12, 2012

Austin, TX - March 19, 2012

Visit http://percona.com/training

Peter Zaitsev

pz@percona.com

We're Hiring!
www.percona.com/about-us/careers/

PERCONA LIVE

www.percona.com/live