Optimizing MySQL Configuration for MySQL 5.6

Zabbix Conference 2014
Sep 12, 2014

Aurimas Mikalauskas
Architect, Percona

Agenda

- MySQL Configuration Tuning Basics
- What's new with MySQL 5.6
- Looking at Most Important Options

Things to Know About MySQL Configuration

- Default configuration is poor
  - MySQL does not scale it with server size
  - MySQL 5.6 default changes are not enough
- 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

Most Options Do Not Scale

- Going to Server with 8x memory you can't just multiply all configuration variables 8x
- 16GB of memory to 128GB of memory
  - **sort_buffer_size** 4MB to 32MB is bad idea.

Know Scope and Unit

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

Set Variables Locally

- Many variables are **SESSION**
  - Can be set for current session only

- Set variable value for session doing complex queries instead of setting it globally:

```
mysql> set session sort_buffer_size=16*1024*1024;
Query OK, 0 rows affected (0.00 sec)
```

Avoid Basic Mistakes

- Setting variables in the 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 options
  - Server reads **[mysqld]**, client **[mysql]**

Config Management Practices

- Keep Config files in sync on different servers
  - Out of Sync config files is a frequent cause of mistakes and confusion
- Keep record of changes
  - Config files under version control is great
  - At least keep your changes documented

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

```
pz@ubuntu:~$ vmstat 5
procs -----------memory---------- ---swap-- -----io---- -system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa
 1  0      0 2725708 253216 513572    0    0     1     1   20   22  0  0 100  0
 0  0      0 2725700 253216 513596    0    0     0     0   72   73  0  0 100  0
 0  0      0 2725700 253216 513596    0    0     0     3   70   74  0  0 99  1
 0  0      0 2725700 253216 513596    0    0     0     0   70   74  0  0 100  0
 0  0      0 2725700 253216 513596    0    0     0     0   70   74  0  0 100  0
 0  0      0 2725700 253216 513596    0    0     0     0   70   72  0  0 100  0
```

Agenda

- MySQL Configuration Tuning Basics
- What's new with MySQL 5.6
- Looking at Most Important Options

Better Defaults

- Changes to defaults values
  - Many Variables are now computed based on other variables
  - Most important changes
    - **innodb_file_per_table=1**
    - **innodb_buffer_pool_instances=8**
    - **innodb_log_file_size=48M**
    - **innodb_old_blocks_time=1000**
    - **innodb_stats_on_metadata=off**
    - **thread_cache_size=8+**
    - **max_allowed_packet=4M**

Auto Resizing Innodb Log files

- Cumbersome to change **innodb_log_file_size** before MySQL 5.6
- Now Automatic!

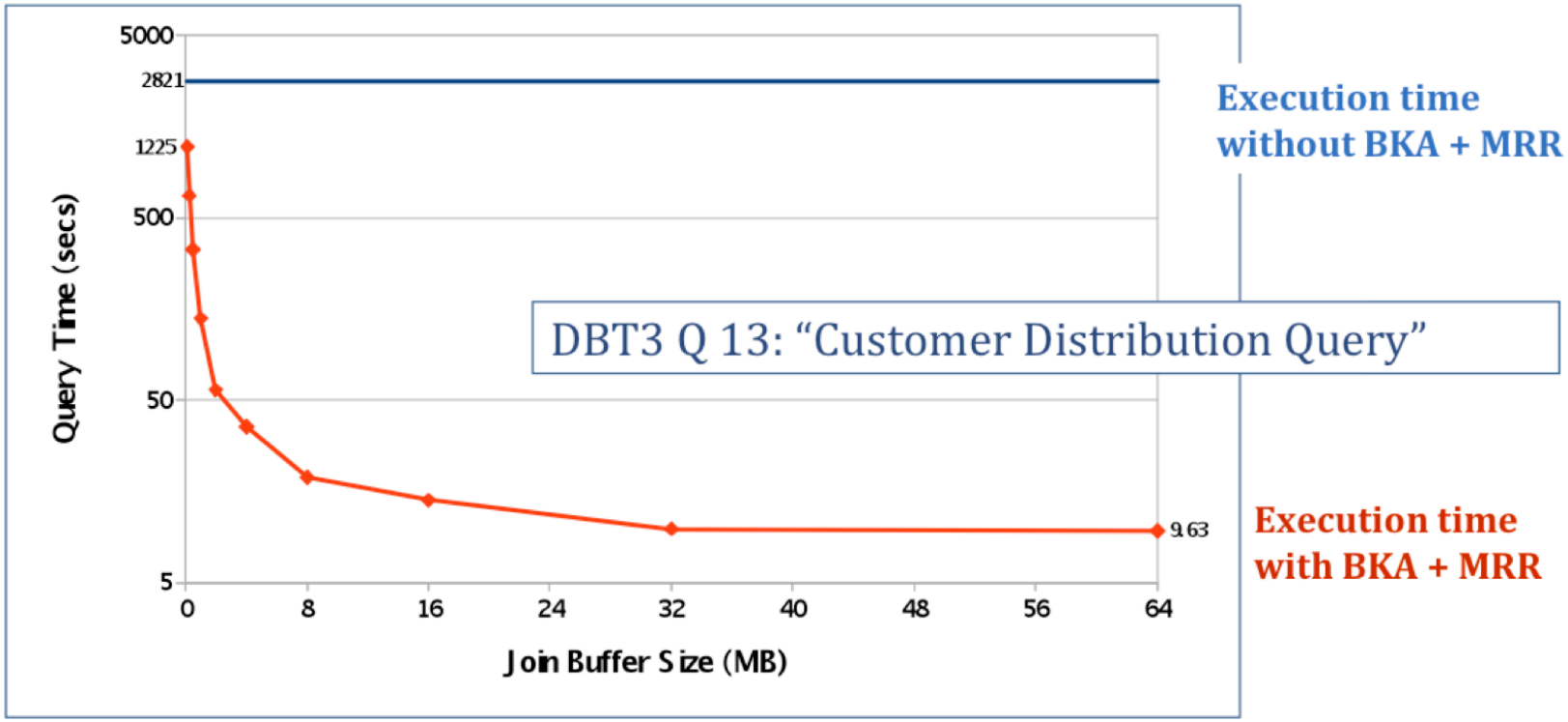
```
InnoDB: Resizing redo log from 2*3072 to 2*32768 pages, LSN=1626007
InnoDB: Starting to delete and rewrite log files.
InnoDB: Setting log file ./ib_logfile101 size to 512 MB
InnoDB: Progress in MB: 100 200 300 400 500
InnoDB: Setting log file ./ib_logfile1 size to 512 MB
InnoDB: Progress in MB: 100 200 300 400 500
InnoDB: Renaming log file ./ib_logfile101 to ./ib_logfile0
InnoDB: New log files created, LSN=1626007
```

Improved Performance Schema

- "Mixed" Configuration mode
  - Configuration file and tables
- **skip-performance-schema**
  - Disable if not using performance schema
- Limits have to be set statically
  - **performance_schema_events_stages_history_long_size=10000**
- Can configure instruments/consumers
  - **performance-schema-instrument='wait/synch/cond/%=counted'**
- Check out Webinar on Performance Schema configuration and usage
  - http://bit.ly/ZWhQVi

New Meaning for some variables

- **join_buffer_size** used for BKA



MySQL 5.6 top options to consider

- innodb_io_capacity=2000
- innodb_io_capacity_max=6000
- innodb_lru_scan_depth=2000
- relay-log-info-repository=TABLE
- master-info-repository=TABLE
- table_open_cache_instances = 16
- join_buffer_size=16M
- innodb_checksum_algorithm=crc32
- innodb_flush_neighbors=0
- innodb_monitor_enable = '%'

Agenda

- MySQL Configuration Tuning Basics
- What's new with MySQL 5.6
- Looking at Most Important Options

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
- **pt-mext** from Percona Toolkit 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
  - Auto set in MySQL 5.6
  - Otherwise 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.
- **table_open_cache_instances=16**
  - If have heavy contention on table cache

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 number of tables + 10% unless 50K+ tables
  - **meta_data_locks_hash_instances=256**
    - When we have contention on meta data locks
    - Only helps with multi table workload

General Options

- **back_log**
  - Need adjustment if many connections/sec
  - MySQL 5.6 Auto tunes but might not go high enough
  - 2048 is reasonable value
    - Note OS TCP/IP stack might need configuration
- **max_allowed_packet**
  - Limits maximum size of query
  - Limits internal string variable size
  - 16MB is a good value
- **max_connect_errors**
  - Can cause “Host Blocked” error messages
    - Especially running on unreliable networks
  - 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 a real performance killer with slow drives.
- **sync_relay_log**
  - Same for relay log starting MySQL 5.6
- **expire_log_days**
  - Purge old binary logs after this number of days
  - 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
  - Beware 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 above 512MB

General Options

- **sort_buffer_size**
  - In memory buffer used for sorting
  - Watch **sort_merge_passes**
  - Consider setting for session for large queries
  - Default lowered to 256K on MySQL 5.6
    - Large values hurt performance of small queries
- **join_buffer_size**
  - Helps performance of Joins with no indexes
    - Better get rid of such Joins!
  - Also used with BKA in MySQL 5.6
  - 8MB can be reasonable value
- **default_storage_engine**
  - Use this engine for tables if not specified
- Also check **default-tmp-storage-engine**

General Options

- **read_rnd_buffer_size**
  - Buffer for reading rows in sorted offer
  - Specifies maximum value
  - Values around 16MB often make sense
  - Also used as a buffer for MRR in 5.6
  - Do not mix with **read_buffer_size**
- **Tmpdir**
  - Specify location of temporary directory
  - Tmpfs often good choice unless very large temporary space is needed.
  - May cause Innodb to refuse using native AIO, when it is not worth it
  - **tmpdir=/dev/shm**

Optimizer Switch

- Many optimizer specific options are moved to **optimizer_switch**
- Defaults generally good. Might need to adjust if getting bad query plans

```
mysql> SELECT @@optimizer_switch\G
 *************************** 1. row ***************************
 @@optimizer_switch: index_merge=on,index_merge_union=on, index_merge_sort_union=on,
 index_merge_intersection=on, engine_condition_pushdown=on, index_condition_pushdown=on,
 mrr=on,mrr_cost_based=on, block_nested_loop=on,batched_key_access=off, materialization=on,
semijoin=on,loosescan=on, firstmatch=on, subquery_materialization_cost_based=on, use
_index_extensions=on
```

MyISAM

- **MyISAM ? What MyISAM**
  - You do not want to use MyISAM with MySQL 5.6
  - It will be still used for “mysql” database and on disk TMP tables
- **key_buffer_size=32M**
  - May be larger if large TMP tables are used
- **myisam_recover=BACKUP,FORCE**
  - In case tables in “mysql” database get corrupted

Innodb – Memory Settings

- **innodb_buffer_pool_size**
  - The most important setting. Often 80%+ of memory is allocated here.
- **innodb_buffer_pool_instances**
  - Reduce contention. Default of 8 in MySQL 5.6
- **innodb_log_buffer_size**
  - Buffer for log files. Good Values 4MB-128MB
  - Not only reduce writes but help contention
- **innodb_change_buffer_max_size**
  - Control size of Insert buffer. Default is ¼ 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_buffer_pool_dump_at_shutdown**
- **innodb_buffer_pool_dump_now**
  - Save and restore Buffer Pool for faster warmup
- **innodb_io_capacity**
  - Controls Innodb Assumption about Disk Performance. Increase for faster drives. Default of 200 is quite low.
- **Innodb_io_capacity_max** bursts in MySQL 5.6

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_neighbors=0**
  - Can give better performance especially for SSDs

Innodb Undo Table Spaces

- Store Undo records in Separate Tablespace
- Initializing Innodb Instance
  - **innodb_undo_tablespaces=2**
    - One or more dedicated tablespaces
- Can play at run time
  - **innodb_undo_logs=8**
  - **innodb_undo_directory=/fast/storage**
    - Has to be durable !

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 a good choice. Default in MySQL 5.6

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
  - Default in MySQL 5.6

Other Innodb Options

- **innodb_file_format**
  - Which file format Innodb will use
  - “Antelope” is default legacy format
  - “Barracuda” allows use of 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 (Default in MySQL 5.6)
    - » Innodb will still refresh stats when table changes significantly
- **innodb_sync_array_size=16**

Visibility Options

- **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=1**
  - Get advanced table and index usage statistics in Percona Server and MariaDB

Summary

- Many options to chose from!
- 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.

Learning More

- Percona Webinars http://bit.ly/19QaWoj
- Percona Training
  - http://www.percona.com/products/mysql-training
- Percona Live London
  - Nov 3-4, London
    - Tens of sessions and tutorials
  - http://www.percona.com/live/london-2014

Thank You!

Aurimas Mikalauskas

aurimas@percona.com