

Real-World MySQL Tuning

MySQL Conference 2007



Presented by

Raj Thukral <thukral@pythian.com>

DBA Team Lead

[The Pythian Group](#)

What's Real-World?



- Default MySQL setup
- Runs fine in dev
- Application launched with default setup
- Application grows
- Performance Issues
- Enter Pythian, or, Real-World Tuning

Constraints



- Database in production
- Can not change data model or SQL quickly
- Limited downtime windows
- Need to tune quickly, effectively
- Need to get results
- It has to be done yesterday!

Outline



- Parameters, Status
- General parameters
- MyISAM parameters related to performance
- InnoDB parameters
- Tuning methodologies
- Questions?

Let's Start!



- `show (global) variables`
- `show (global)status`
- Filter by using like `'%<name>%'`
- List of parameters at:
www.pythian.com/blogs/mysql_parameters

Some Examples



The full list (from my instance)
and a short version!

```
mysql> show variables like '%max_c%';
```

```
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| max_connect_errors | 10    |
| max_connections    | 100   |
+-----+-----+
2 rows in set (0.12 sec)
```

```
mysql>
```

Connecting and Caching



- **max_connections**
 - Corresponding status name: `max_used_connections`
 - Watch out for memory usage!
- **table_cache**
 - Corresponding status name: `open_tables` / `opened_tables`
 - Watch out for `open_files_limit` !
- **thread_cache_size**
 - Corresponding status name: `threads_cached`, `threads_created`

Connecting and Caching



- **max_connections**
 - Corresponding status name: `max_used_connections`
 - Watch out for memory usage!
- **table_cache**
 - Corresponding status name: `open_tables` / `opened_tables`
 - Watch out for `open_files_limit` !
- **thread_cache_size**
 - Corresponding status name: `threads_cached`, `threads_created`

The Query Cache



- `query_cache_size`
- Caches queries and results for frequently-run query
- Caveats: cannot use binds, underlying data must not change, overhead for all queries
- Corresponding status name: `qcache_hits`, `qcache_inserts`, `qcache_not_cached`, `qcache_queries_in_cache`

Buffers



- Difference between buffers and caches
 - Per-thread vs. global
 - Used for various purposes – sorts, table scans, joins
 - Watch for memory usage:
 - $(\text{thread_stack} + 2 * \text{net_buffer_length} + \text{join_buffer_size} + \text{read_buffer_size} + \text{sort_buffer_size} + \text{read_rnd_buffer_size} + \text{tmp_table_size} + \text{bulk_insert_buffer_size}) * \text{max_connections} (!)$
 - Over-committing memory – not all your connections will need all of these buffers concurrently. Or will they?

Buffer Zone



- `sort_buffer_size`
 - Used for `order by` or `group by` clauses.
 - Avoid disk-based sorts -- they're slow. Use a bigger sort buffer instead.
 - check status: `sort_merge_passes`, `sort_range`, `sort_rows`, `sort_scan`
- `join_buffer_size`
 - For joins that do not use indexes
 - Should optimize joins rather than increase this size, but increase if you have to
 - check status: `select_full_join`

Buffer Zone, p2



- `read_buffer_size`
 - for caching row data during a full table scan
 - size set is allocated for each table scanned per thread.
 - 1-2M
- `tmp_table_size`, and `max_heap_table_size`
 - these 2 determine the maximum tmp table size (memory table)
 - tmp tables are used for group by, sorts, etc.
 - Queries that use blob/text will spill to disk
 - check: `created_tmp_tables` and `created_tmp_disk_tables`

The Key to MyISAM



- **key_buffer_size**
- The single most important factor for MyISAM tuning. Caches Indexes
- Corresponding status name: `key_reads` / `key_read_requests` / `key_buffer_size`
- **Named key caches**
- Syntax is different: `set global mycache.key_buffer_size=16*1024*1024`
- Corresponding status name: none. Use `SELECT @@global.mycache.key_buffer_size;`
- Pin indexes in named cache on startup: `cache index <tablename> in mycache; load index into cache <tablename>.`

InnoDB



- **innodb_buffer_pool_size**
 - the granddaddy of caches
 - check `show engine innodb status\G`
- **innodb_log_file_size**
 - usually 25-50% of `buffer_pool_size`
 - sets how often InnoDB 'checkpoints' – i.e. commits modified contents of the buffer pool to disk
 - beware of setting these too big – recovery after a crash can take hours!
- **innodb_log_buffer_size**
 - size of transaction held in memory, reduces log file writes

InnoDB



Hit-or-Miss Parameters

- `innodb_thread_concurrency`
 - Let's increase it! No, let's decrease it! YMMV.
 - Needs a bounce
- `innodb_flush_method`
 - `fdatasync` (default)
 - `O_DSYNC` – slow!
 - `O_DIRECT` (linux) – recommended (bypasses OS cache overhead).
 - Not for windows (`async_unbuffered` only option for windows).

Tuning Tips



- Query-based

- See typical queries, tune for them (`show processlist`)
- Slow query log

- Holistic view

- Look at status output
- Disk and CPU usage
- Queries / Application
- Talk to the developers!

Deltas



- Check and store output of `show global status`
- Compare deltas over a period
- Check changes over time
- Sample scripts available at www.pythian.com/blogs/mysql_status_delta

Conclusion



- Major parameters
- How to tune
- Questions

Presented by
Raj Thukral <thukral@pythian.com>
DBA Team Lead
The Pythian Group