



How to Stop Hating MySQL: Fixing Common Mistakes and Myths

Presented by:
Sheeri K. Cabral

at PICC 2010



Pythian
love your data

Who I Am

- MySQL DBA
- MySQL User Group
- MySQL Podcast (OurSQL, on hiatus), videos (technocation.org)
- Lots of community stuff

Myths about MySQL

- Uses too much memory
- Slow
- I need more features!

More Myths about MySQL

- Don't use ENUM
- Schema changes take forever
- You have to restart to log
- No partitioning

www.pythian.com/blogs/1168/why-you-want-to-switch-to-mysql-51

<http://www.pythian.com/news/1414/new-in-mysql-51-sheeris-presentation/>

aka <http://bit.ly/mysql51upgrade>

MySQL uses too much memory!

- INFORMATION_SCHEMA.GLOBAL_VARIABLES (5.1)
- SHOW GLOBAL VARIABLES (5.0+)
- LIKE
 - '%cache%'
 - '%buffer%'

Common Mistake: Wasting Memory

- `bdb_cache_size`, `bdb_log_buffer_size`
- `query_cache_size`, if query cache is not used
- 32-bit operating system

Myth: Query Cache Is Faster!

- Use memcached for caching common queries
- `query_cache_type`
 - 1 or ON
 - 2 or DEMAND – `SQL_CACHE`
 - `SQL_NO_CACHE`
- Default `query_cache_size=0`

MyISAM Index Cache

- `key_buffer_size` – globally allocated on startup
- How much is being used?
 - STATUS variables
 - `Key_blocks_unused` vs. `Key_blocks_used`
 - $(\text{Key_blocks_}\% * \text{key_cache_block_size}) / \text{key_buffer_size}$

MyISAM Data Cache

- OS filesystem cache
- ISAM =
 - indexed
 - sequential
 - access
 - method

InnoDB Buffer Pool

- `innodb_buffer_pool_size` – globally allocated on startup
- How much is being used?
 - GLOBAL STATUS variables
 - `Innodb_buffer_pool_pages_free`
 - `Innodb_buffer_pool_pages_total`

Allocated Per-thread

- `binlog_cache_size` = 32 Kb
- `max_binlog_cache_size` = 4 Gb
- `net_buffer_length` = 16 Kb
 - 1 Mb max
 - `max_allowed_packet`

Per-Thread, Allocated As Needed

- `join_buffer_size` – memory buffer for joins not using indexes
- `read_buffer_size` – memory buffer for sequential table scans
- `read_rnd_buffer_size` – memory buffer for random table seeks

Per-Thread, Allocated As Needed

- `preload_buffer_size` – when pre-loading indexes
- `sort_buffer_size` – for sorting
- `myisam_sort_buffer_size`
 - Index sorting only – REPAIR, OPTIMIZE, creating indexes

Temporary Tables

- `tmp_table_size`
- `max_heap_table_size`
- Large rows, BLOBs written to disk

Performance Caches

- Global Variables
 - table_cache
 - STATUS variables Opened_tables
 - thread_cache_size
 - STATUS variables Connections and Threads_created

Size Matters

- Larger/fragmented data/indexes use more memory
- Larger/fragmented data/indexes take more time to search
- Clustered indexes
- `innodb_file_per_table`

Size Matters – Data Diet

- OPTIMIZE – how often?
- Purge/archive regularly
- Follow large deletes with OPTIMIZE

Size Matters – Schema Diet

- IP addresses
- TIMESTAMP vs DATETIME
- Strings vs. Numbers vs. ENUM/SET

Myth: MySQL is Slow

- Memory/disk tradeoffs
- RAID
- Temporary tables
 - Created_tmp_tables
 - Created_tmp_disk_tables

Disk I/O

- Many data changes
 - Binary logs
 - Data
 - Index
- INSERT DELAYED
- Batch update/deletes

Network

- Large queries vs. small
 - CPU compute time
 - Network traffic
 - Large data sets
- INSERT....ON DUPLICATE KEY UPDATE

Size Matters, So Does Performance

- Choosing Correct Table Types:
 - MyISAM
 - InnoDB
 - BLACKHOLE
 - MERGE
 - MEMORY
 - ARCHIVE

Query Optimization

- Use EXPLAIN

<http://dev.mysql.com/doc/refman/5.1/en/using-explain.html>

www.pythian.com/blogs/wp-content/uploads/explain-diagram.pdf

Know What You're Doing

- Subqueries
- VIEW
- TRIGGER

Know What You're Doing

- LIMIT
- Stored Procedures
 - compiled per thread
- Indexes
 - Selectivity
 - Functions
 - Overhead vs. utility

Check this stuff regularly

- You can automate with `mysqltuner 2.0`
- Ignite slides follow, so I'll go quickly :)

mysqltuner 1.0

- Author: Major Hayden
- Quick sanity check:
 - Status/system variables
 - Memory, Architecture, Tables
- Not all recommendations were relevant
 - e.g., “turn on query cache” if off

mysqltuner 1.0

- Hard-coded everything:
 - what, threshold, recommend, output
 - can skip some checks
- No offline mode
- mysqltuner 2.0:
 - Still a quick sanity check
 - Much more flexible
 - Guided by a config file
 - Can be remote or offline too

How it works

- Perl
- Gets information into a key-value hash
 - Currently status and system variables
 - e.g. key= “max_connections”, value=100
 - Can be anything
 - e.g. key=db.tbl, value=size (INFORMATION_SCHEMA)

How it works

- Many supported platforms
- Can get information from files
 - `--filelist vars.txt,status.txt,keyval.txt,...`
 - Does not login to database
 - Can easily be coded to get info from both

Config file parsing

- #Comment
- delimiter is |||
- label|||comparison and value|||expr|||output
- Label and expr value are always printed out
 - e.g. % slow queries: 13.25

How it works

- Parses config file
 - label|||comparison|||expr|||output

% slow queries|||>5|||Slow_queries/Questions * 100|||Too many slow queries, either increase long_query_time or optimize the queries so they are no longer slow.

- “Slow_queries” and “Questions” are replaced with hash values
- expr value is calculated
- comparison is made (to >5)
- “/” = bad key; do not overwrite keys

How it works

% slow queries|||>5|||Slow_queries/Questions * 100|||Too many slow queries, either increase long_query_time or optimize the queries so they are no longer slow.

- “Too many slow...” shown if value>5 and --recommend option
- comparison/value and expr are Perl

Supported Version|||ne "5"|||substr("version",0,1)|||version less than 5.x, upgrade!

mysq tuner Functions

- From mysq tuner 1.0:
 - &pretty_uptime (157961 → 1d 19h 52m 41s)
 - &hr_num (157691 → 158K)
 - in 2.0 → 158 Thousand
 - &hr_bytes (157691 → 154K)
 - in 2.0 → 154 Kb

mysqltuner Functions

- New in mysqltuner 2.0
 - `&hr_bytime`
 - for rates
- ie, `&hr_bytime(Slow_queries/Uptime)`
 - `&hr_bytime(12345/157961)`
 - `&hr_bytime(0.078)`
 - =4.69 per minute

More complex example

rate of sorts that cause temporary tables|||

=~ /second|minute/|||

&hr_bytime((Sort_merge_passes/Uptime_since_flush_status)|||

Too many sorts are causing temporary tables.
Consider increasing\nsort_buffer_size and/or
read_rnd_buffer_size

Printing out variables

```
sort_buffer_size|||
```

```
eq "0 bytes"|||
```

```
&hr_bytes(sort_buffer_size)|||
```

There is something wrong with the value of
sort_buffer_size

Printing out variables

```
sort_buffer_size|||
```

```
eq "0 bytes"|||
```

```
&hr_bytes(sort_buffer_size)|||
```

There is something wrong with the value of
sort_buffer_size

Output:

```
sort_buffer_size: 4 Mb (--output pretty, default)
```

```
sort_buffer_size,4 Mb (--output csv)
```

```
>> MySQLTuner 1.0.1 - Major Hayden <major@mhtx.net>
>> Bug reports, feature requests, and downloads at http://mysqmtuner.com/
>> Run with '--help' for additional options and output filtering
[OK] Logged in using credentials passed on the command line
```

----- General Statistics -----

```
[--] Skipped version check for MySQLTuner script
[OK] Currently running supported MySQL version 5.1.37-1ubuntu5-log
[OK] Operating on 64-bit architecture
```

----- Storage Engine Statistics -----

```
[--] Status: -Archive -BDB -Federated +InnoDB -ISAM -NDBCluster
[--] Data in MyISAM tables: 308M (Tables: 157)
[--] Data in InnoDB tables: 57G (Tables: 317)
[--] Data in BLACKHOLE tables: 0B (Tables: 2)
[!!] Total fragmented tables: 74
```

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

```
[--] Up for: 1d 18h 53m 2s (517K q [3.352 qps], 13K conn,
50M, RX: 4B)
[--] Reads / Writes: 1% / 99%
[--] Total buffers: 49.9G global + 48.4M per thread (1000 max threads)
[!!] Maximum possible memory usage: 97.2G (142% of installed RAM)
[OK] Slow queries: 0% (16/517K)
[OK] Highest usage of available connections: 0% (3/1000)
[OK] Key buffer size / total MyISAM indexes: 150.0M/60.3M
[OK] Key buffer hit rate: 98.7% (1M cached / 17K reads)
[!!] Query cache efficiency: 17.0% (16K cached / 99K selects)
[OK] Query cache prunes per day: 0
[OK] Sorts requiring temporary tables: 0% (0 temp sorts / 2 sorts)
```

```
[OK] Temporary tables created on disk: 11% (4K on disk /
37K total)
[OK] Thread cache hit rate: 99% (3 created / 13K
connections)
[OK] Table cache hit rate: 25% (522 open / 2K opened)
[OK] Open file limit used: 0% (369/200K)
[OK] Table locks acquired immediately: 100% (8M
immediate / 8M locks)
[!!] InnoDB data size / buffer pool: 57.4G/48.0G
```

----- Recommendations -----

General recommendations:

Run OPTIMIZE TABLE to defragment tables for better performance

Reduce your overall MySQL memory footprint for system stability

Variables to adjust:

*** MySQL's maximum memory usage is dangerously

TX: high ***

*** Add RAM before increasing MySQL buffer variables

query_cache_limit (> 4M, or use smaller result sets)

innodb_buffer_pool_size (>= 57G)

OK Logged in using credentials passed on the command line

Uptime in seconds: 157961.00

Uptime: 1d 19h 52m 41s

Questions: 519525.00

% slow queries: 0.00

slow query rate: 0.00 per day

Long query time: 1.00

Slow query logging: ON

% reads: 1.00

% writes: 99.00

qps: 3.29

Queries: 3.29 per second

Connections: 13 Thousand

Bytes sent: 50 Million

Bytes received: 4 Billion

Supported Version: 5

Release Series: 5.1

Minor Version: 37

Distribution: (Ubuntu)

Distribution: (Ubuntu)

MySQL Architecture: x86_64

Query cache efficiency (%): 17.27

% query cache used: 99.95

Query cache low memory prunes: 0.00 per day

Query cache size: 512.0 Mb

Query cache min result size: 4.0 Mb

Total sorts: 3.00

% sorts that cause temporary tables: 0.00

rate of sorts that cause temporary tables: 0.00 per day

sort_buffer_size: 16.0 Mb

read_rnd_buffer_size: 16.0 Mb

Sort rows: 8.18 per hour

rate of joins without indexes: 13.68 per minute

rate of reading first index entry: 1.92 per minute

rate of reading fixed position: 14.49 per minute

rate of reading next table row: 34.78 per second

tmp_table_size-max_heap_table_size: 0.00

tmp_table_size: 784.0 Mb

max_heap_table_size: 784.0 Mb

% temp disk tables: 11.80

temp disk rate: 1.72 per minute

temp table rate: 12.84 per minute

MyISAM key buffer size: 150.0 Mb

max % MyISAM key buffer ever used: 11.26

% MyISAM key buffer used: 29.29

% index reads from memory: 98.71

table open cache size (5.1+): 2000.00

rate of table open: 47.86 per hour

% open files: 0.18

OK Logged in using credentials passed on the command line

Uptime in seconds: 157961.00

Uptime: 1d 19h 52m 41s

Questions: 519525.00

% slow queries: 0.00

slow query rate: 0.00 per day

Long query time: 1.00

Slow query logging: ON

% reads: 1.00

% writes: 99.00

qps: 3.29

Queries: 3.29 per second

Connections: 13 Thousand

Bytes sent: 50 Million

Bytes received: 4 Billion

Supported Version: 5

Release Series: 5.1

Minor Version: 37

Distribution: (Ubuntu)

Distribution: (Ubuntu)

MySQL Architecture: x86_64

Query cache efficiency (%): 17.27

% query cache used: 99.95

Query cache low memory prunes: 0.00 per day

Query cache size: 512.0 Mb

Query cache min result size: 4.0 Mb

Total sorts: 3.00

% sorts that cause temporary tables: 0.00

rate of sorts that cause temporary tables: 0.00 per day

sort_buffer_size: 16.0 Mb

read_rnd_buffer_size: 16.0 Mb

Sort rows: 8.18 per hour

rate of joins without indexes: 13.68 per minute

rate of reading first index entry: 1.92 per minute

rate of reading fixed position: 14.49 per minute

rate of reading next table row: 34.78 per second

tmp_table_size-max_heap_table_size: 0.00

tmp_table_size: 784.0 Mb

max_heap_table_size: 784.0 Mb

% temp disk tables: 11.80

temp disk rate: 1.72 per minute

temp table rate: 12.84 per minute

MyISAM key buffer size: 150.0 Mb

max % MyISAM key buffer ever used: 11.26

% MyISAM key buffer used: 29.29

% index reads from memory: 98.71

table open cache size (5.1+): 2000.00

rate of table open: 47.86 per hour

% open files: 0.18

rate of open files: 8.41 per hour
Immediate table locks %: 100.00
Table lock wait rate: 0.00 per day
thread cache: 8.00
Total threads created: 3.00
thread cache hit rate %: 0.00
Threads that are slow to launch: 0.00
Slow launch time: 2.00
% connections used: 0.30

Max connections used: 3.00
Max connections limit: 1000.00
% aborted connections: 0.00
rate of aborted connections: 0.00 per day
% aborted clients: 0.09
rate of aborted clients: 6.56 per day
Is InnoDB enabled?: YES
% innodb log size: 2.08
MyISAM concurrent inserts: 2.00

RECOMMENDATIONS:

There are too many joins without indexes -- this means that joins are doing full table scans.

The rate of reading the first index entry is high; this usually indicates frequent full index scans.

The rate of reading data from a fixed position is high; this indicates many queries need to sort results and/or do a full table scan, including join queries that do not use indexes.

The rate of reading the next table row is high; this indicates many queries are doing full table scans.

Too many temporary tables are being written to disk. Increase `max_heap_table_size` and `tmp_table_size`.

Too many intermediate temporary tables are being created; consider increasing `sort_buffer_size` (sorting), `read_rnd_buffer_size` (random read buffer, ie, post-sort), `read_buffer_size` (sequential scan).

MyISAM key buffer (index cache) % used is low. You may need to decrease the size of `key_buffer_size`, re-examine your tables to see if indexes have been removed, or examine queries and expectations about what indexes are being used.

MyISAM key buffer (index cache) % used is low. You may need to decrease the size of `key_buffer_size`, re-examine your tables to see if indexes have been removed, or examine queries and expectations about what indexes are being used.

Size of the table cache

InnoDB log file size is not an appropriate size, in relation to the InnoDB buffer pool. Consider changing either `\ninnodb_log_file_size` or `innodb_buffer_pool_size`

Limitations

- No grouping in output
- Check one item at a time
 - but item is a Perl expr

Limitations

- Limited configurable logic flow
- Still not a DBA-in-a-box

README

Pythian's MySQL Tuner (<http://launchpad.net/mysqltuner>). MySQL Tuner name used with the permission of the original mysqltuner creator, Major Hayden.

This version of mysqltuner uses a configuration file where you can change the thresholds, add new variables/calculations to check, and custom tune the recommendations output.

There is a default config file in this package, to use it, run:

```
mysqltuner.pl --config tuner-default.cnf
```

If you are not using MySQL 5.1, use tuner-default_pre_5_1.cnf

Output:

By default, mysqltuner.pl does not output recommendations, just the label and value. To see the recommendations at the end of the output, use the --recommend option.

The default output is the "pretty" mode, which is of the format:

label: value

By specifying **--output <type>**, you can change the format of the output.

Current output types are "pretty" and "csv".

The "csv" output mode is of the format:

label,value

Connecting:

By default, mysqltuner.pl will ask for a username and password and connect to the mysqld instance on localhost. You can use the following options to mysqltuner.pl to set the mysqld instance to connect to:

--host <hostname> MySQL host (default: localhost)
--port <port> Port to use for connection (default: 3306)
--socket <socket> Socket to connect to for data retrieval
--user <username> Username to use for authentication
--pass <password> Password to use for authentication

Remote operation:

By default, mysqltuner.pl will try to get local information for memory, swap, and whether the operating system is 32 or 64-bit. If you are connecting to a remote mysqld instance, you should set the following three options:

--forcemem <size> Use this amount of RAM in Mb instead of getting local memory size
--forceswap <size> Amount of swap memory configured in megabytes
--forcearch 32|64 Architecture of operating system (32 or 64-bit)

Offline operation:

mysqltuner.pl can work in an offline mode by using information inside of files, instead of querying the server itself. T

o use offline mode, first obtain the information desired in one or more files, such as:

```
mysql -e "SHOW GLOBAL VARIABLES" > vars.txt
```

```
mysql -e "SHOW GLOBAL STATUS" > status.txt
```

The files must contain 2 fields (variable name and value) separated by whitespace (ie, a space or tab).

You should also use --forcemem, --forceswap and --forcearch when operating in offline mode (see Remote operation) to avoid spurious errors. The --filelist option takes one or more filenames (comma-separated).

Sample offline operation:

```
mysqltuner.pl --config tuner-default.cnf --forcemem 16384 \  
--forceswap 2048 --forcearch 64 --filelist vars.txt,status.txt
```

Help:

The following options may be able to help if you get stuck:

--help Shows this help message
--debug debugging output will be shown

How to get

- mysqltuner 1.0
 - `wget mysqltuner.pl`
- mysqltuner 2.0
 - <https://launchpad.net/mysqltuner>
 - project is open to contributions, bugs, ideas
 - “mysqltuner” name used with permission

More Features: Do Not Want!

- Look into Drizzle, a MySQL fork
- <http://www.drizzleproject.org>

Feedback

- Questions?
- Comments?
- Why else do you hate MySQL?

cabral@pythian.com