MySQL Performance Tuning for Non-DBAs

Tom Clark
entropymedia

We all know how this story goes...

we build a nifty web application*...

*using MySQL


Slashdot
NEWS FOR NERDS. STUFF THAT MATTERS.
Stories
Recent
Popular
Search
tclark
Slashdot
Apple
Ask Slashdot
Book Reviews
Games
Hardware
Idle
Interviews
Slashdot is powered by your submissions, so send in your scoop
Ask Slashdot: Affordably Aggregating ISP Connections?
Posted by timothy on Thursday October 15, @07:01PM
from the glob-glob-glob dept.
An anonymous reader writes
"Has anyone setup a system to aggregate multiple ISP connections to form a high ban
interesting, but it doesn't look like it has been widely adopted. Multi-Link PPP appears
find any good guides for setting up both sides of the connection for a site-to-site link. T

we start to get some real traffic...

and we find that we have some performance issues.

and we find that we have some performance issues.

Now what?

application code

database design

web server

database server

operating system

hardware

application code

database design

web server

database server

operating system

application code

database design

web server

database server

database design

web server

database server

web server

database server

database server

Some bad news

- The default MySQL configuration is not right for your application.
- The configurations found on many hosting services is often worse.
- There is no one size fits all (or even most) MySQL configuration.

Some good news

There are opportunities to get real performance gains by tuning our MySQL servers

Of all of the options we considered, we are likely to get the biggest performance gains for the least effort by tuning our servers

Some platform notes

- 64 bit Linux, Solaris
  - Kernel threads
  - More memory for caches, buffers
- Plenty* of RAM

*For some value of plenty

Storage engines

- MyISAM
  - Fast for applications that primarily use SELECTs
- InnoDB
  - Fast for applications with more INSERT/UPDATE/DELETEs
  - More sensitive to tuning than MyISAM

Mixing storage engines?

Complicates server administration

Complicates configuration

Complicates performance analysis

Server configuration options

Based on version 5.1 documentation

Generally set in /etc/mysql/my.cnf

Our strategy:

- Suggest a starting point
- Check relevant server status fields
- Adjust as necessary

General Options

table_open_cache
(prev. table_cache)

- Start at 1024 – value depends on number of tables and number of connections
- Check value of Opened_tables in status
- If the value is high or rapidly increasing, increase value
- If the value of Open_tables is consistently low, you may be able to decrease value

thread_cache_size
(prev. thread_cache)

- Set to 16-64 to start
- Check Threads_created/Connections
- Goal is to cache enough threads so that they do not need to be created in normal operation

sort_buffer_size

- Important only if you use ORDER BY, GROUP BY in many queries.
- Leave at default if you only use simple queries
- Experiment by adjusting the value and testing – setting it too high can hurt performance

query_cache_size

- Query cache stores results of SELECT statements
- Defaults to 0 – no caching
- Set to 32-512 MB to enable
- Monitor “Qcache%” variables to tune further.

Query caching doesn't always help

- Monitor query cache use by comparing Qcache_hits and Qcache_inserts
- If your hit rate is poor, you may be better off disabling query caching

query_cache_min_res_unit

- Qcache_free_blocks – high values indicate fragmentation, decrease query_cache_min_res_unit
- If your queries return large values, try increasing query_cache_min_res_unit

MyISAM Options

key_buffer_size

- 25-40% of available memory
- To high a setting will may hurt performance
- Check Key_reads/Key_read_requests
  value should be < .01
- If you have a small data set, consider
  reducing

InnoDB Options

innodb_buffer_pool_size

~ 10% larger than your data or

70% of available memory

Use SHOW INNODB STATUS and look for BUFFER POOL AND MEMORY section

innodb_log_file_size

- Set up to 64-512MB
- Important for write-intensive loads
- Larger log file sizes increase recovery time after a crash
- If you change this setting, you must remove old log files before restarting the server

innodb_log_buffer_size

- 1-16 MB
- Value depends on write load and size of transactions
- Buffer is flushed once per second

innodb_flush_log_at trx_commit

Default is 1 - every transaction commit flushed to disk – best data integrity & slowest performance

2 – log buffer written to file with each trx & flushed to disk once per second

0 – log buffer written to file once per second & flushed to disk once per second

Additional resources

MySQLTuner: http://blog.mysqltuner.com

Nagios plugin: check_mysql_health

MySQL Performance Blog:
http://www.mysqlperformanceblog.com

Final thoughts

Every case is a special case.

Any tuning advice should be regarded as a starting point.

Database usage profiles and size may change over time. Continue to monitor performance and adjust as needed.