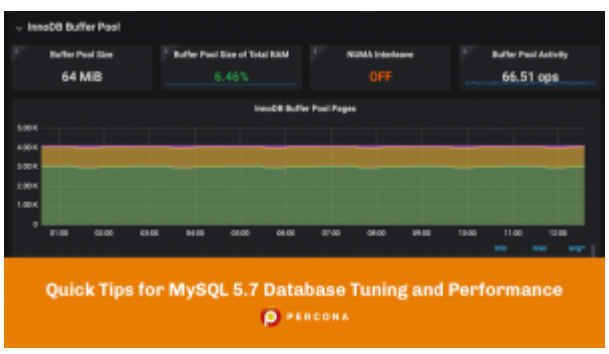
Tips for MySQL 5.7 Database Tuning and Performance

Back to the Homepage

01 Apr 2020

By Brian Sumpter Insight for DBAs, Monitoring, MySQL, Percona Software insight for DBAs, Monitoring, MySQL, Percona Software

0 Comments



While there is no magic bullet for MySQL 5.7 database tuning, there are a few areas that can be focused on upfront that can dramatically improve the performance of your MySQL installation. While much information has been published on this topic over the years, I wanted to break down some of the more critical settings that anyone can implement – with no guesswork required.

Depending on the version of MySQL you are running, some of the default values used in this post may differ from your install, but the premise is still largely the same. We will focus on MySQL 5.7 for the purposes of this article.

Initial MySQL performance tuning can be broken down to the following categories:

- Tuning for your hardware
- Tuning for best performance/best practices
- Tuning for your workload

MySQL 5.7 Database Tuning

Tuning For Your Hardware

Depending on the hardware you have installed MySQL on, some variables need to be set based on the machine (or VM) specifications. The following variables are largely dependent on your hardware:

innodb_buffer_pool_size

- Generally, set to 50% – 70% of your total RAM as a starting point.
- It does not need to be set any larger than the total database size.

innodb_log_file_size

- This is generally set between 128M – 2G.
- It does not need to be any larger than the buffer pool.

innodb_flush_log_at_trx_commit

- Setting to “1” (default in 5.7) gives the most durability.
- Setting to “0” or “2” will give more performance, but less durability.

innodb_flush_method

- Setting this to O_DIRECT will avoid the performance penalty from double buffering.

Tuning For Best Performance / Best Practices

innodb_file_per_table

- Setting this to “ON” will generate an independent InnoDB table space for every table in the database.

innodb_stats_on_metadata

- Setting this to “OFF” avoids unnecessary updating of InnoDB statistics and can greatly improve read speeds.

innodb_buffer_pool_instances

- A best practice is to set this to “8” unless the buffer pool size is < 1G, in which case set to “1”.

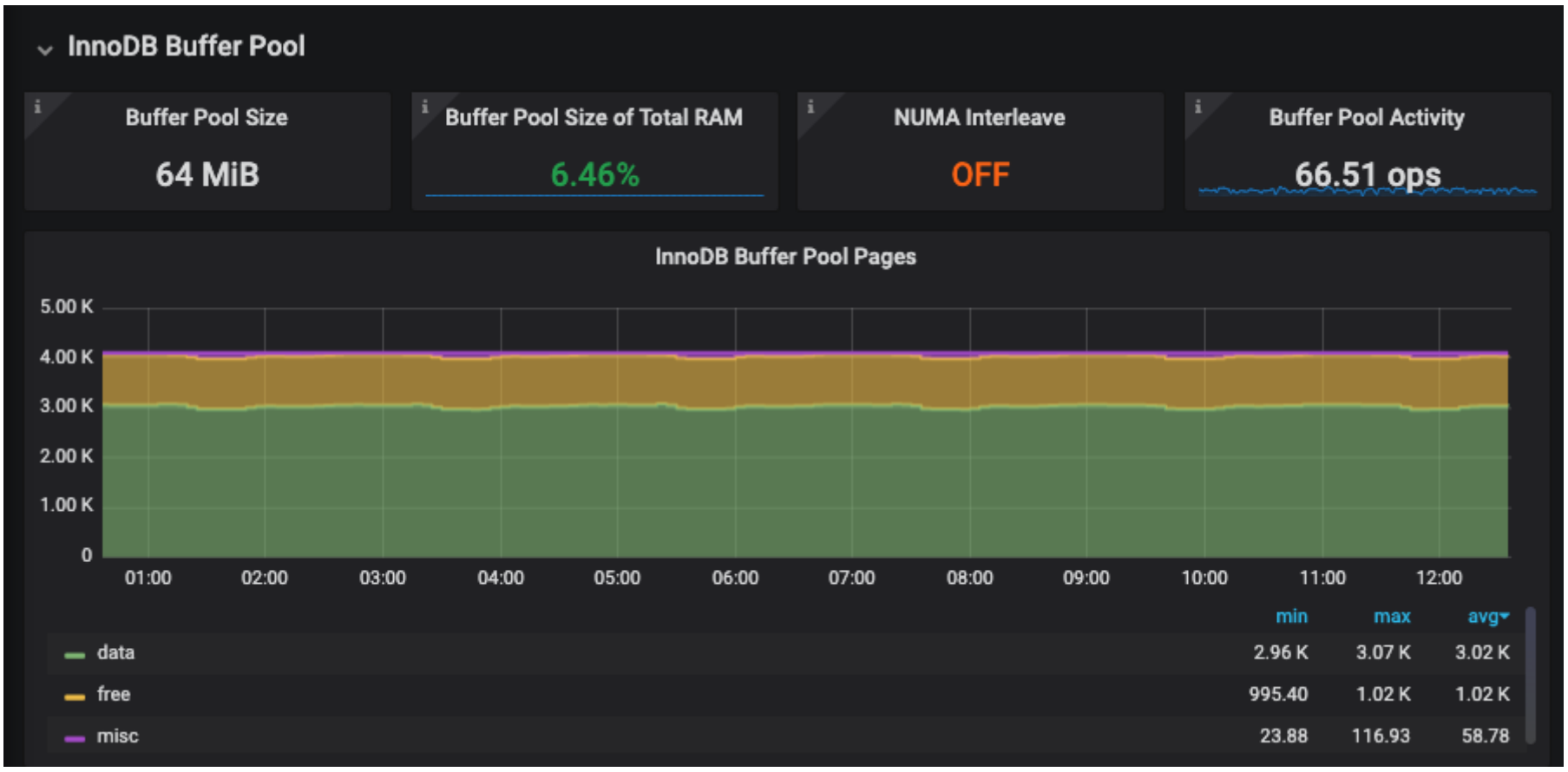
query_cache_type & query_cache_size

- Setting both of these to “0” will entirely disable the query cache.

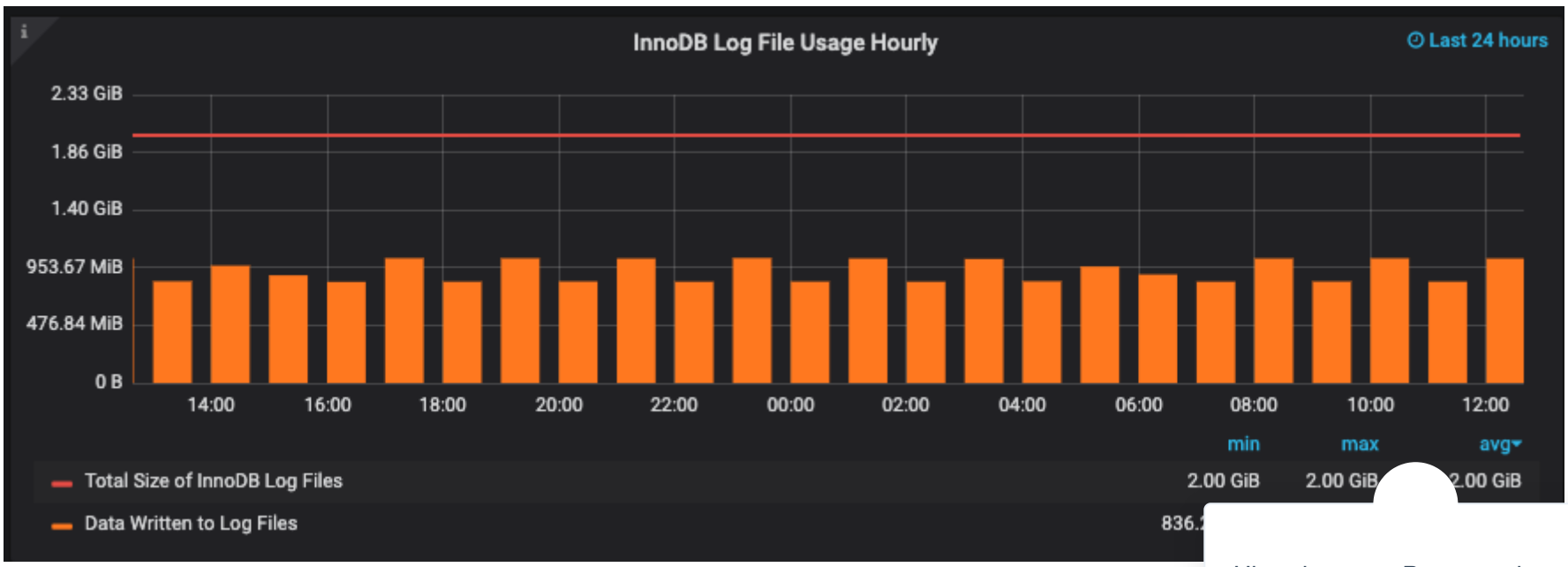
Tuning For Your Workload

To tune further, more information will be required. The best way to gather this information is to install a MySQL monitoring/graphing tool like Percona Monitoring and Management (PMM). Once you have a tool installed, we can dive into the individual metrics and start customizing based on the data.

I would recommend starting with one of the most impactful variables – the innodb_buffer_pool_size. Compare the RAM and number of free pages on your instance to the total buffer pool size. Based on these metrics, you can determine if you need to increase or decrease your overall buffer pool size setting.



Next, take a look at your metrics for the InnoDB Log File usage. The rule of thumb is that your log files should hold approximately one hour of data. If you see that the data written to the log files hourly exceeds the total size of the log files, you would want to increase the innodb_log_file_size variable and restart MySQL. You could also verify with “SHOW ENGINE INNODB STATUS;” via the MySQL CLI to assist in calculating a good InnoDB log file size.



In Summary

While this is by no means a comprehensive article on MySQL 5.7 database tuning, the suggestions above s
hanging fruit and get your system closer to an ideal setup. As with all database tuning, your process should be an ongoing one based on current information.

- Examine the settings proposed above, and implement if they make sense for your environment/workload.

- Install a good monitoring tool to give insight into the database (Percona Monitoring and Management is our suggestion).
- Stay current on your monitoring graphs to determine other areas where you may need to tune.

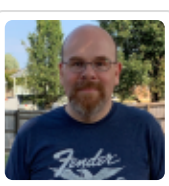
Related

MySQL 101: Parameters to Tune for MySQL Performance
June 30, 2020
In "Insight for DBAs"

MySQL Performance Tuning Settings
January 28, 2014
In "Insight for DBAs"

MySQL 5.7 Performance Tuning After Installation
October 12, 2016
In "MySQL"

Author



Brian Sumpter

Share this post

Leave a Reply

Enter your comment here...

CATEGORIES

MySQL(3355)

Insight for DBAs(1528)

Percona Software(1479)

Percona Events(870)

MongoDB(549)

Insight for Developers(472)

Benchmarks(340)

Percona Live(332)

Webinars(287)

Cloud(276)

PostgreSQL(176)

Monitoring(164)

MariaDB(157)

Percona Services(138)

ProxySQL(127)

Security(125)

Hardware and Storage(105)

Storage Engine(52)

Database Trends(46)

Percona Announcements(10)

Percona Blog RSS Feed

UPCOMING WEBINARS

The Open Source Alternative to Paying for MongoDB
Why PostgreSQL Is Becoming A Migration Target For Enterprise
How To Measure Linux Performance Wrong
Converting MongoDB to Percona Server for MongoDB
Moving MongoDB to the Cloud: Strategies and Points To Consider

All Webinars »

Hi, welcome to Percona - home of the database performance experts.