

How To Improve MySQL Performance With Tuning

Posted January 15, 2020

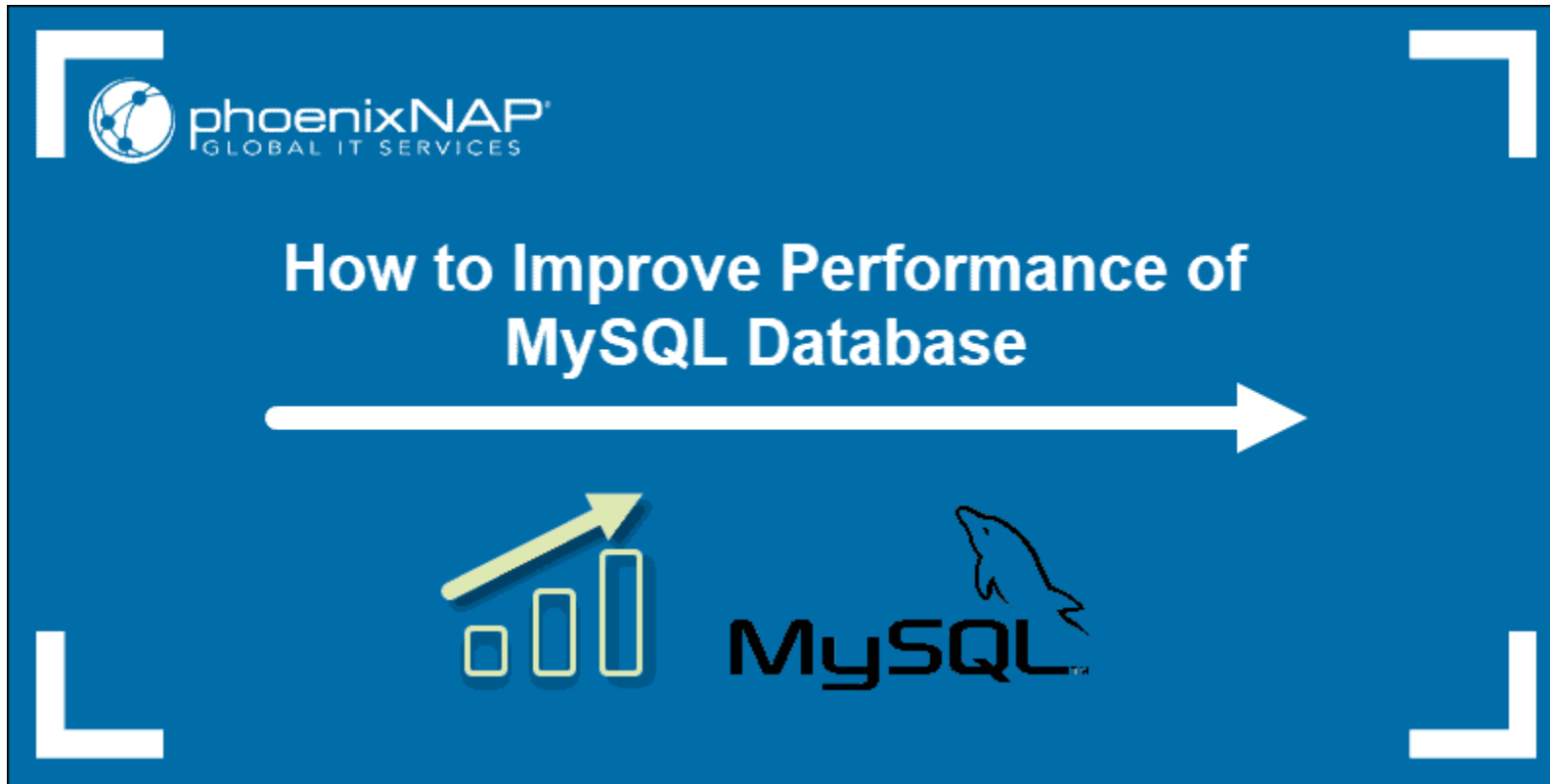
[Home](#) / [SysAdmin](#) / [How to Improve MySQL Performance With Tuning](#)

≡ Contents



Introduction

MySQL is a popular open-source database application that stores and structures data in a way that is meaningful and readily accessible.



Prerequisites

- A Linux system with MySQL installed and running, [Centos](#) or [Ubuntu](#)
- An existing database

A vertical black advertisement for PhoenixNAP. At the top is the PhoenixNAP logo and "GLOBAL IT SERVICES". Below it, the text "FLEXSERVERS" is in red. The main headline "PUT THE PEDAL TO BARE METAL" is in large white letters. A red horizontal line separates the headline from the text "Vertical CPU scaling. Advanced performance. Excellent price." Below this is a glowing blue speedometer graphic. At the bottom is a red button with the text "ORDER NOW" in white.

System Performance Tuning for MySQL

At the system level, you'll adjust hardware and software options to improve MySQL performance.

1. Balance the Four Main Hardware Resources

Bare Metal Cloud is now live!

DEPLOY TODAY



Take a moment to evaluate your storage. If you're using traditional hard disk drives (HDD), you can upgrade to [solid-state drives \(SSD\)](#) for a [performance improvement](#).

Use a tool like **iotop** or **sar** from the **sysstat** package to monitor your disk input/output rates. If disk usage is much higher than usage of other resources, consider adding more storage or upgrading to faster storage.

Processor

Processors are usually considered the measure of how fast your system is. Use the [Linux top command](#) for a breakdown of how your resources are used. Pay attention to the MySQL processes and the percentage of processor usage they require.

Processors are more expensive to upgrade, but if your CPU is a bottleneck, an upgrade might be necessary.

Memory

Memory represents the total amount of RAM in your MySQL server. You can adjust the memory cache (more on that later) **to improve performance**. If you don't have enough memory, or if the existing memory isn't optimized, you can end up damaging your performance instead of improving it.

down your performance.

Network

It's important to [monitor network traffic](#) to make sure you have sufficient infrastructure to manage the load.

Overloading your network can lead to latency, dropped packets, and even server outages. Make sure you have enough [network bandwidth](#) to accommodate your normal levels of database traffic.

2. Use InnoDB, Not MyISAM

MyISAM is an older database style used for some MySQL databases. It is a less efficient database design. The newer **InnoDB** supports more advanced features and has in-built optimization mechanics.

InnoDB uses a clustered index and keeps data in pages, which are stored in consecutive physical blocks. If a value is too large for a page, InnoDB moves it to another location, then indexes the value. This feature helps keep relevant data in the same place on the storage device, meaning it takes the physical hard drive less time to access the data.

3. Use the Latest Version of MySQL

A part of the ongoing development includes performance enhancements. Some common performance adjustments may be rendered obsolete by newer versions of MySQL. In general, it's always better to use native MySQL performance enhancement over scripting and configuration files.

Software Performance Tuning

These methods involve tweaking the MySQL configuration files, writing more efficient database queries, and structuring the database to retrieve data more efficiently.



Note: When adjusting configuration settings, it's best to make small incremental adjustments. A major adjustment may overburden another value and degrade performance. Also, it is recommended that you make one change at a time and then test. It's easier to track errors or misconfigurations when you only change one variable at a time.

As with most software, not all tools work on all versions of MySQL. We will examine three utilities to evaluate your MySQL database and recommend changes to improve performance.

The first is [tuning-primer](#). This tool is a bit older, designed for MySQL 5.5 – 5.7. It can analyze your database and suggest settings to improve performance. For example, it may suggest that you raise the ***query_cache_size*** parameter if it feels like your system can't process queries quickly enough to keep the cache clear.

The second tuning tool, useful for most modern MySQL databases, is [MySQLTuner](#). This script (***mysqltuner.pl***) is written in Perl. Like tuning-primer, it analyzes your database configuration looking for bottlenecks and inefficiencies. The output shows metrics and recommendations:

Bare Metal Cloud is now live!

DEPLOY TODAY



At the top of the output, you can see the version of the MySQLTuner tool and your database.

The script works with MySQL 8.x. Log file recommendations are the first on the list, but if you scroll to the bottom, you can see general recommendations for improving MySQL performance.

use within the GUI.

5. Optimize Queries

A query is a coded request to search the database for data that matches a certain value. There are some query operators that, by their very nature, take a long time to run. If your environment relies on automated queries, they may be impacting performance. Check and [terminate MySQL processes](#) that may pile up in time.

6. Use Indexes Where Appropriate

Many database queries use a structure similar to this:

```
SELECT ... WHERE
```

These queries involve evaluating, filtering, and retrieving results. You can restructure these by adding [a small set of indexes](#) for the related tables. The query can be directed at the index to speed up the query.

Avoid using a function in the predicate of a query. For example:

```
SELECT * FROM MYTABLE WHERE UPPER(COL1)='123'Copy
```

The **UPPER** notation creates a function, which has to operate during the **SELECT** operation. This doubles the work the query is doing, and you should avoid it if possible.

8. Avoid % Wildcard in a Predicate

The **wildcard** % stands for zero or more characters. If your query looks like this:

```
SELECT * FROM MYTABLE WHERE COL1 LIKE '%123'Copy
```

The expression **'%123'** includes all values that end with **123**. The % sign stands for any type or number of characters preceding, including no characters. The query must now perform a full table scan to test for this condition.

9. Specify Columns in SELECT Function

10. Use **ORDER BY** Appropriately

The **ORDER BY** expression sorts results by the specified column. It can be used to sort by two columns at once. These should be sorted in the same order, ascending or descending. If you try to sort different columns in different order, it will slow down performance. You may combine this with an index to speed up the sorting.

11. **JOIN, UNION, DISTINCT**

Try to use an inner join whenever possible. An outer join looks at additional data outside the specified columns. That's fine if you need that data, but it's a waste of performance to include data that won't be required.

The **UNION** and **DISTINCT** commands are sometimes included in queries. Like an outer join, it's fine to use these expressions if they are necessary. However, they add additional sorting and reading of the database. If you don't need them, it's better to find a more efficient expression.

12. Use the **EXPLAIN** Function

Appending the **EXPLAIN** expression to the beginning of a query will read and evaluate the query. If there are inefficient expressions or confusing structures, **EXPLAIN** can help you find them. You can then adjust the phrasing of your query to avoid unintentional table scans or other performance hits.

13. MySQL Server Configuration

This configuration involves making changes to your **/etc/mysql/my.cnf** file. Proceed with caution and make minor changes at a time.

query_cache_size – Specifies the size of the cache of MySQL queries waiting to run. The recommendation is to start with small values around 10MB and then increase to no more than 100-200MB. With too many cached queries,

max_connection – Refers to the number of connections allowed into the database. If you're getting errors citing *"Too many connections,"* increasing this value may help.

innodb_buffer_pool_size – This setting allocates system memory as a data cache for your database. If you have large chunks of data, increase this value. Take note of the RAM required to run other system resources.

innodb_io_capacity – This variable sets the rate for input/output from your storage device. This is directly related to the type and speed of your storage drive. A 5400-rpm HDD will have a much lower capacity than a [high-end SSD or Intel Optane](#). You can adjust this value to better match your hardware.

Conclusion

You should now know how to improve MySQL performance and tune your database.

Look for bottlenecks (hardware and software), queries that are doing more work than needed, and consider using automated tools and the **EXPLAIN** function to evaluate your database.



Next you should also read

 [Databases, MySQL](#)

How To Find Duplicate Values in MySQL

March 4, 2020

Learn how to find the duplicate entries in your MySQL databases. The guide shows you how to use the GROUP BY,...

[READ MORE](#)

 [Databases, MySQL](#)

How to Find and Terminate / Kill MySQL Process

January 23, 2020

Killing a MySQL process can help you boost the performance of your server. By running a few commands, you can...

[READ MORE](#)

 [SysAdmin, Databases, MySQL](#)

Install and Get Started with MySQL Workbench on Ubuntu 18.04

November 19, 2019

Workbench is a visual tool for managing MySQL databases . Its graphical interface allows administrators and...

[READ MORE](#)

 [Databases, MySQL](#)

How to List All Users in a MySQL Database

November 18, 2019

This simple tutorial analyses the commands used to list all user accounts in MySQL. Learn about additional...

[READ MORE](#)

Goran Jevtic

Goran combines his passions for research, writing and technology as a technical writer at phoenixNAP. Working with multiple departments and on a variety of projects, he has developed extraordinary understanding of cloud and virtualization technology trends and best practices.

RECENT POSTS

How to Resolve the “cannot connect to the Docker daemon” Error

How to Configure Proxy Settings on Ubuntu 20.04

How to Install Helm on Ubuntu, Mac and Windows

CATEGORIES

SYSADMIN

VIRTUALIZATION

DEVOPS AND
DEVELOPMENT

SECURITY

BACKUP AND
RECOVERY

BARE METAL
SERVERS

WEB SERVERS

NETWORKING

DATABASES

COMPANY

ABOUT US

GITHUB

BLOG

RFP TEMPLATE

CAREERS

PRODUCTS

COLOCATION

SERVERS

CLOUD
SERVICES

SOLUTIONS

LOCATIONS

CONNECT

EVENTS

PRESS

CONTACT US



