13 Tips for Tuning and Optimizing MySQL and MariaDB

MySQL and MariaDB are the most widely used relational database management systems (RDMS) when it comes to website hosting and CMS systems such as Joomla, WordPress, Drupal, and Typo 3. In this article, I will explain how to speed up and optimize your MySQL and MariaDB database server.

Store MySQL data in the separate partitions

When it comes to the point of optimization and assurance, it is always the best idea to store the database data in a separate volume. The volumes are specifically for fast storage volumes such as SSD, NVMe. Even if your system will fail, you will have your database safe. As the partition volume is made up of fast storage volumes, the performance will be faster.

Set the maximum number of MySQL connections

MySQL/MariaDB uses an instruction **max_connections** which specifies how many concurrent connections are currently allowed in the server. Too many connections result in high memory consumption as well as high CPU load. For small websites, the connections can be specified to 100-200 and larger ones may need 500-800 and more. The **max_connections** can be dynamically changed using the SQL query. In this example, I have set the value to 200.

```
$ mysql -u root -p
```

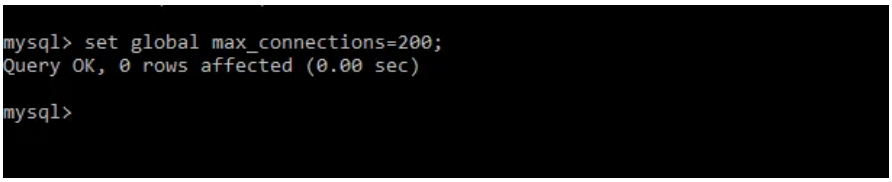
Search
Search

About This Site
Vitux.com aims to become a Linux compendium with lots of unique and up to date tutorials.

Latest Tutorials
Linux Basics: 3 Ways to find your local IP Address in Debian 11
How to Install Cockpit on Rocky Linux 8
How to Make a User an Administrator in Debian 11
How to View the Network Routing Table in Ubuntu Linux
How to Install VMware Tools on Debian 11

```
mysql> set global max_connections=200;
```

Output :



Enable MySQL slow query log

Logging queries that take a very long time to execute makes it easier to troubleshoot the database problems. The slow query log can be enabled by adding the following lines in the MySQL/MariaDB configuration file.

```
slow-query-log=1
slow-query-log-file= /var/lib/mysql/mysql-slow-query.log
long-query-time=1
```

Where the first variable enables the slow query log

The second variable defines the log file directory

The third variable defines the time to complete a MySQL query

Restart the MySQL/MariaDB service and monitor the log

```
$ systemctl restart mysql
```

```
$ systemctl restart mariadb
```

```
$ tail -f /var/lib/mysql/mysql-slow-query.log
```

Set the maximum packet allowed by MySQL

Data is split into the packets in MySQL. Max_allowed_packet defines the maximum size of the packets that can be sent. Setting up the max_allowed_packet too low can cause the query to be too slow. It is recommended to set the packet value to the size of the largest packet.

Setting up the Temporary table capacity

Tmp_table_size is the maximum space used for the built-in-memory table. If the size of the table exceeds the limit specified, it will be converted to a MyISAM table on disk. In MySQL/MariaDB, you can add the following variables in the configuration file to set up Temporary table size. It is recommended to set this value on the server 64M per GB of memory.

```
[mysqld]

tmp_table_size=64M
```

Restart the mysql service

```
$ systemctl restart mysql
```

```
$ systemctl restart mariadb
```

Set up the maximum memory table capacity.

Max_heap_table_size is the variable used in MySQL to configure maximum memory table capacity. The size of the maximum memory table capacity should be the same as the temporary table capacity to avoid disk writes. It is recommended to set this value on the server to 64M per GB of memory. Add the following line in the MySQL configuration file and restart the service.

```
[mysqld]

max_heap_table_size=64M
```

To apply the changes, restart the database server.

```
$ systemctl restart mysql
```

```
$ systemctl restart mariadb
```

Disable DNS reverse lookup for MySQL

When a new connection is received, MySQL/MariaDB will perform a DNS lookup to resolve the user's IP address. This may cause a delay when the DNS configuration is invalid or there is a problem with the DNS server. To disable the DNS lookup, add the following line in the MySQL configuration file and restart the MySQL service.

```
[mysqld]

skip-name-resolve
```

Restart the service:

```
$ systemctl restart mysql
```

```
$ systemctl restart mariadb
```

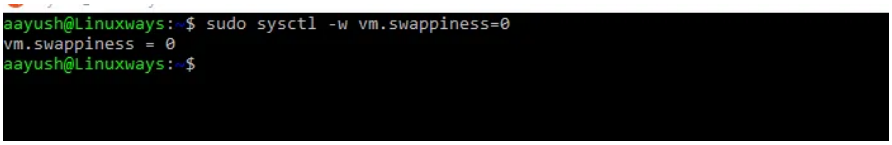
Avoid Using Swappiness in MySQL

The Linux Kernel moves part of the memory to a special partition of the disk called "swap" space when the system runs out of the physical memory. In this condition, the system writes information to the disk rather than freeing some

memory. As the system memory is faster than the disk storage it is recommended to disable the swappiness. Swappiness can be disabled by using the following command.

```
$ sysctl -w vm.swappiness=0
```

Output :



Increase InnoDB buffer pool size

MySQL/MariaDB has an InnoDB engine that has a buffer pool to cache and index data in the memory. Buffer pool helps MySQL/MariaDB queries be executed comparatively faster. Choosing the proper size of the InnoDB buffer pool requires some knowledge of system memory. The best idea is to set the value of the InnoDB buffer pool size to 80% of the RAM.

Example.

- System Memory = 4GB
- Buffer Pool size = 3.2GB

Add the following line in the MySQL configuration file and restart the service

```
[mysqld]

Innodb_buffer_pool_size 3.2G
```

Restart the database:

```
$ systemctl restart mysql
```

```
$ systemctl restart mariadb
```

Dealing with Query cache size

Query cache directive in MySQL/MariaDB is used to cache all the queries which keep on repeating with the same data. It is recommended to set the value to 64MB and increase in time for small websites. Increasing the value of query cache size to GB's is not recommended as it may degrade the database performance. Add the following line in the my.cnf file.

```
[mysqld]

query_cache_size=64M
```

Check idle connections

Resources are consumed by idle connections so it needs to be terminated or refreshed if possible. These connections remain in the "sleep" state and may stay for a long period of time. Check the idled connections using the following command.

```
$ mysqladmin processlist -u root -p | grep
"Sleep"
```

The query will list out the processes that are in the sleep state. Generally in PHP, the event can occur when using mysql_pconnect. This opens the MySQL connection, executes the queries, removes authentications, and leaves the connection open. Using **wait_timeout** directive, the idle connections can be interrupted. The default value for **wait_timeout** is 28800 seconds which can be decreased to a minimum time range like 60 seconds. Add the following line in the my.cnf file

```
[mysqld]

wait_timeout=60
```

Optimize and repair MySQL database

If the server is shut down unexpectedly then there is a chance that tables in MySQL/MariaDB can crash. There are other possible reasons to get the database table crash such as accessing the database while the copy process is running, the file system is suddenly crashed. In this situation, we have a special tool called "**mysqlcheck**" which checks, repairs, and optimizes all tables in the databases.

Use the following command to perform the repair and optimization activities.

For all databases:

```
$ mysqlcheck -u root -p --auto-repair --check --optimize --all-databases
```

For specific database:

```
$ mysqlcheck -u root -p --auto-repair --check --optimize dbname
```

Replace dbname with your database name

1. Check MySQL/MariaDB performance using testing tools

It is best practice to check the MySQL/MariaDB database performance regularly. This will make it easy to get the performance report and point of improvement. There are many tools available among which mysqltuner is the best one.

Run the following command to download the tool

```
$ wget https://github.com/major/MySQLTuner-perl/tarball/master
```

Untar the file

```
$ tar xvzf master
```

Go to the project directory and execute the following script.

```
$ cd major-MySQLTuner-perl-7aa57fa
```

```
$ ./mysqltuner.pl
```

Output:

Conclusion

In this article, we have learned how to optimize MySQL/MariaDB using different techniques. Thank you for reading.

Vitux June 10, 2022 CentOS, Debian, Linux, Ubuntu

← How to install and use PhotoRec to recover deleted files in Ubuntu 20.04

How to Install and Configure ELK Stack on Ubuntu and Debian →