

# MariaDB SQL- Tuning

After setting up mariadb and tweaking its basic configuration, one may want to check the structures and sql- queries for performance.

It is not really well documented, how mariadb will output what exactly is the cause for bad performance. In phpMyAdmin there is a tab in the Server-Menu named Status where there is an Advisor, which may give hints for what may be wrong. But after that, there is no way to find the Statements, causing the trouble. Furtheron, many statements beeing problematik may not cause Errors or very slow behaviour, but are doing the „wrong things“ really fast. Having bad Statements that are executing fast is not a good thing, as strange may happen (e.g. deadlocking when the size of data is growing). So one should have a look at bad statements.

## Introducing the Slow Log

For all those things, one can use the slow log - even if the name appears to only be useful for long running transacations.

There are three types of bad statement, that can be analysed with the slow log, each described in a seperate section. They cannot be analysed all in all without having big impact to the load of the server.

## Setup Logging

Todos here: General logging - paths and setup logrotate

Keep in mind, that changes are lost each restart of mariadb if not set in configuration.

/etc/logrotate.d/mariadb

```
# This logname can be set in /etc/my.cnf
# by setting the variable "log-error"
# in the [mysqld] section as follows:
#
# [mysqld]
# log-error=/var/log/mysqld.log
#
# If the root user has a password you have to create a
# /root/.my.cnf configuration file with the following
```

```
# content:
#
# [mysqladmin]
# password = <secret>
# user= root
#
# where "<secret>" is the password.
#
# ATTENTION: This /root/.my.cnf should be readable ONLY
# for root !
/var/log/mysql/*.log {
# create 600 mysql mysql
su mysql mysql
notifempty
daily
rotate 3
missingok
compress
postrotate
# just if mariadb is really running
if test -x /usr/bin/mysqladmin && \
/usr/bin/mysqladmin -pMYSECRETPASSWORD ping &>/dev/null
then
/usr/bin/mysqladmin -pMYSECRETPASSWORD --local flush-error-log \
flush-engine-log flush-general-log flush-slow-log
ret=$?
if test $ret -ne 0
then
echo "/etc/logrotate.d/mariadb failed, probably because">&2
echo "the root account is protected by password.">&2
echo "See comments in /etc/logrotate.d/mariadb on how to fix this">&2
exit $ret
fi
fi
endscript
}
```

## Method 1: Slow Queries

This is ordinary. TODO: Explain long\_query value...

## Method 2: Queries causing Disk- Access

This is the real fun. There may be queries, that are executed fast, but are causing temporary files written to disk each execution. The programmer itself has no influence to this behaviour, as mariadb is doing it internally.

In the Advicor you can find: „Many temporary tables are being written to disk instead of being kept in memory.“

Analyse it with those settings in /etc/my.cnf:

```
log-slow-verbosity=explain
log-slow-disabled_statements=admin
log-slow-admin-statements=off
long-query-time=0
log-slow-filter=filesort_on_disk,tmp_table_on_disk
```

As you can see, there is a trick: by setting long-query-time to 0, every statement will be logged AND explained. Be cautious to always apply a filter to that logging, as otherwise ALL statements would be analysed.

## Method 3: Queries having no index

For troubleshooting queries without an index, you can also use:

```
log-slow-verbosity=explain
log-slow-disabled_statements=admin
log-slow-admin-statements=off
long-query-time=1000
log-slow-filter=not_using_index
```

Todo: explain more

From:

<http://dokuwiki.obel1x.de/> - **obel1x.de**

Permanent link:

[http://dokuwiki.obel1x.de/content:mariadb\\_tuning](http://dokuwiki.obel1x.de/content:mariadb_tuning)

Last update: **2022/12/25 23:32**

