

Top 15 MySQL parameters for beginners

Aleksandrs Asafovs
Lattelecom Tehnology
Riga,Latvia

Keywords:

Mysql, Security, Backup, Performance, Mysql beginners

Introduction

MySQL is the world's most popular open-source database and the #1 database for web-based applications. But unfortunately not all dba/unix administrators/programmers and other people who installing mysql are spending time to set few parameters, they are living all default parameters. But setting just few parameters can avoid you from having problems with performance, data consistency and recovery (just spending 20 minutes to configure you MySQL instance.) In the new version of MySQL, MySQL developers are trying to help inexperienced users setting default parameters, but MySQL team could not do all work for you. So which parameters are most critical and what they can impact? I will try to answer on this question in these presentation.

Backup

When I'm getting new mysql instance the first thing I'm thinking, is how to be sure that i will not lose client data. And there is only one way how to do this is, start making backup. There are not a lot of possibilities to achieve this:

- 1) mysqldump
- 2) filesystem backup
- 3) filesystem snapshot
- 4) Percona XtraBackup(GPL license)

Whatever tool you will choose, you will need binary logs to bring databases up to date from the point of the backup. Binary logs contain database changes. (insert,update,delete). The binary logs are enabled with the **log_bin** parameter (1 parameter). After we have enabled binary logs we need to tell mysql how often to expire it. By default mysql will not expire it, and your filesystem can become full. We can use parameter **expire_logs_days**(2 parameter) to expire binary logs. The best choose to store binary logs between last two full database backups. For example we are making backups every Sunday. Then we need to set this parameter to **expire_logs_days=8**. Also we need to backup ours binary log, to have possibility to recover then, if we will loose our database server.

Security

When we are sure that we can restore our system, we need to make database more secure.

Access to database

60% of mysql databases that I met every day do not need TCP/IP connection, because application that connects to database is located on the same server. In these situations you can disable TCP/IP connection and all connection will be made using named pipes or shared memory. To do this we need to change one of these parameters `skip_networking=1` or `skip_networking=1((3 parameter))`, both these configurations will have the same effect, mysql will disable TCP/IP connections. If for any reason we still need TCP/IP connection we can restrict user access by limiting from which IP/host they can connect. (`CREATE USER 'jeffrey'@'194.12.12.12' IDENTIFIED BY 'mypass';`) And changing default port that MySQL listens 3306 to any other available port (for example `port= 4212`) . More difficult to find database. 2800 3300 4000

Local security

Very often people are forgetting about `LOCAL INFILE(4 parameter)` parameter ,which allow MySQL to read files from the local file system, for example we can read `/etc/passwd` in unix (`SELECT load_file("/etc/passwd");`); there are two possible options for dba :

- Disable LOCAL in `load_data` statements
- limit import and export operations to work only with files in specified directory

If you already know that you and yours application will not use `load_data` statements disable it by setting `local_infile=0`. If you are planning to use import/export operation than set `secure_file_priv=/tmp/` ,where `/tmp` is a directory which will be used for storing files.

Users

Granting privileges in mysql is very dangerous, because when we granting privileges to user we can accidentally create user without password.

```
mysql> grant all privileges on *.* to 'tom'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql> select user,host,password from mysql.user;
+-----+-----+-----+
| user      | host      | password      |
+-----+-----+-----+
| root      | localhost | *94BDCEBE19083CE2A1F959FD02F964C7AF4CFC29 |
| root      | 127.0.0.1 | *94BDCEBE19083CE2A1F959FD02F964C7AF4CFC29 |
| root      | ::1       | *94BDCEBE19083CE2A1F959FD02F964C7AF4CFC29 |
| aasafovs  | %         | *94BDCEBE19083CE2A1F959FD02F964C7AF4CFC29 |
| jeffrey   | %         | *6C8989366EAF75BB670AD8EA7A7FC1176A95CEF4 |
| tom       | %         |                                     |
+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> exit
Bye
[root@bobamo etc]# mysql -u tom -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 89
Server version: 5.6.27 MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

(Picture 1)

Example we have a user Tomas and we would like to grant full privileges to this user because it is our new database administrator. Adding privileges to user Tomas we made a mistake and

write tom. And in this situation mysql creates user(because the is no such user) with no password and now everyone can connect to our system without password (Example picture 1). To avoid this situation we need to add one more variable to these parameter `sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER`.(**parameter 5**) After we change this parameter, when we will try to grant privileges to not existing user mysql we will receive a error.

```
mysql> grant all privileges on *.* to 'tom'@'%';
ERROR 1133 (42000): Can't find any matching row in the user table
```

Data Integrity

We are not expecting that some mysql parameters may cause problems with data integrity. Example: we creating a innodb table with 2 fields and inserting two rows in to the table, then I decided to add auto increment to table `doag_users` field `id` and add one more row with data, after that I decided to check my data, and I realized that my data is changed . My row”0, Tom” was changed to “2, Tom” so mysql decided to change `id` field 0 to 2.

```
mysql> use doag;
Database changed
mysql> create table doag_users (id int not null primary key,user VARCHAR(10)) engine=innodb;
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO doag_users VALUES (0,'Tom');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO doag_users VALUES (1,'Jeff');
Query OK, 1 row affected (0.04 sec)

mysql> select * from doag_users;
+----+-----+
| id | user |
+----+-----+
| 0  | Tom  |
| 1  | Jeff |
+----+-----+
2 rows in set (0.01 sec)

mysql> alter table doag_users modify id int not null auto_increment, auto_increment=2;
Query OK, 2 rows affected (0.04 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> INSERT INTO doag_users VALUES (3,'Kate');
Query OK, 1 row affected (0.01 sec)

mysql> select * from doag_users;
+----+-----+
| id | user |
+----+-----+
| 1  | Jeff |
| 2  | Tom  |
| 3  | Kate |
+----+-----+
3 rows in set (0.00 sec)
```

All this we can prevent by add one more variable to `SQL_MODE=NO_AUTO_VALUE_ON_ZERO`(**parameter 6**)

```
mysql> create table doag_users (id int not null primary key,user VARCHAR(10)) engine=innodb;
Query OK, 0 rows affected (0.01 sec)

mysql> INSERT INTO doag_users VALUES (0,'Tom');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO doag_users VALUES (1,'Jeff');
Query OK, 1 row affected (0.01 sec)

mysql> select * from doag_users;
+----+-----+
| id | user |
+----+-----+
| 0  | Tom  |
| 1  | Jeff |
+----+-----+
2 rows in set (0.00 sec)

mysql> alter table doag_users modify id int not null auto_increment, auto_increment=2;
Query OK, 2 rows affected (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> INSERT INTO doag_users VALUES (3,'Kate');
Query OK, 1 row affected (0.00 sec)

mysql> select * from doag_users;
+----+-----+
| id | user |
+----+-----+
| 0  | Tom  |
| 1  | Jeff |
| 3  | Kate |
+----+-----+
3 rows in set (0.00 sec)
```

Performance shema

If you don't know what is performance shema and not planning to use it, disable it, because it is taking from 3 to 19 percent (my calculations) of yours resources. To disable it set `performance_schema=0`(**parameter 8**). But my recommendation will be spent some time to understand what is it performance shema? What kind data it has? How you can get a lot of useful information from it? Such as i(and don't forget to disable collection information that you are not planning to use.)

Innodb

Than main parameter in the Innodb engine and probably I a whole mysql server is `innodb_buffer_pool_size`(**parameter 8**) than main role for the last 10 year is add 70-80% of your RAM to this parameter. The percent is depend of

Others

max_allowed_packet = 16M default

```
mysql> show global variables like 'max_allowed_packet';
```

```
+-----+-----+
| Variable_name | Value |
```

```
+-----+-----+
| max_allowed_packet | 4194304 |
+-----+-----+
1 row in set (0.00 sec)
```

Conclusions:

In mysql 5.6 is more than 400 parameters, but you don't need to set up all these parameters. After mysql installation, you will need to set only few parameters and this will help you.

Contact address:

Aleksandrs Asafovs
Lattelecom Tehnology
Kleistu iela 5,
LV-1010, Rīga, Latvia

Phone: +371 28353668
Email: aleksandrs.asafovs@lattelecom.lv