# MySQL慢查询日志总结 - 潇湘隐者 - 博客园

http://www.cnblogs.com/kerrycode/p/5593204.html

**慢查询日志概念**

     MySQL的慢查询日志是MySQL提供的一种日志记录，它用来记录在MySQL中响应时间超过阀值的语句，具体指运行时间超过long\_query\_time值的SQL，则会被记录到慢查询日志中。long\_query\_time的默认值为10，意思是运行10S以上的语句。默认情况下，Mysql数据库并不启动慢查询日志，需要我们手动来设置这个参数，当然，如果不是调优需要的话，一般不建议启动该参数，因为开启慢查询日志会或多或少带来一定的性能影响。慢查询日志支持将日志记录写入文件，也支持将日志记录写入数据库表。

官方文档，关于慢查询的日志介绍如下（部分资料，具体参考官方相关链接）：

The slow query log consists of SQL statements that took more than long\_query\_time seconds to execute and required at least min\_examined\_row\_limit rows to be examined. The minimum and default values of long\_query\_time are 0 and 10, respectively. The value can be specified to a resolution of microseconds. For logging to a file, times are written including the microseconds part. For logging to tables, only integer times are written; the microseconds part is ignored.

By default, administrative statements are not logged, nor are queries that do not use indexes for lookups. This behavior can be changed usinglog\_slow\_admin\_statements and log\_queries\_not\_using\_indexes, as described later.

## 慢查询日志相关参数

MySQL 慢查询的相关参数解释：

slow\_query\_log    ：是否开启慢查询日志，1表示开启，0表示关闭。

log-slow-queries  ：旧版（5.6以下版本）MySQL数据库慢查询日志存储路径。可以不设置该参数，系统则会默认给一个缺省的文件host\_name-slow.log

slow-query-log-file：新版（5.6及以上版本）MySQL数据库慢查询日志存储路径。可以不设置该参数，系统则会默认给一个缺省的文件host\_name-slow.log

long\_query\_time ：慢查询阈值，当查询时间多于设定的阈值时，记录日志。

log\_queries\_not\_using\_indexes：未使用索引的查询也被记录到慢查询日志中（可选项）。

log\_output：日志存储方式。log\_output='FILE'表示将日志存入文件，默认值是'FILE'。log\_output='TABLE'表示将日志存入数据库，这样日志信息就会被写入到mysql.slow\_log表中。MySQL数据库支持同时两种日志存储方式，配置的时候以逗号隔开即可，如：log\_output='FILE,TABLE'。日志记录到系统的专用日志表中，要比记录到文件耗费更多的系统资源，因此对于需要启用慢查询日志，又需要能够获得更高的系统性能，那么建议优先记录到文件。

## 慢查询日志配置

默认情况下slow\_query\_log的值为OFF，表示慢查询日志是禁用的，可以通过设置slow\_query\_log的值来开启，如下所示：

mysql> show variables  like '%slow\_query\_log%';

+---------------------+-----------------------------------------------+

| Variable\_name       | Value                                         |

+---------------------+-----------------------------------------------+

| slow\_query\_log      | OFF                                           |

| slow\_query\_log\_file | /home/WDPM/MysqlData/mysql/DB-Server-slow.log |

+---------------------+-----------------------------------------------+

2 rows in set (0.00 sec)

mysql> set global slow\_query\_log=1;

Query OK, 0 rows affected (0.09 sec)

mysql> show variables like '%slow\_query\_log%';

+---------------------+-----------------------------------------------+

| Variable\_name       | Value                                         |

+---------------------+-----------------------------------------------+

| slow\_query\_log      | ON                                            |

| slow\_query\_log\_file | /home/WDPM/MysqlData/mysql/DB-Server-slow.log |

+---------------------+-----------------------------------------------+

2 rows in set (0.00 sec)

mysql>

使用set global slow\_query\_log=1开启了慢查询日志只对当前数据库生效，如果MySQL重启后则会失效。如果要永久生效，就必须修改配置文件my.cnf（其它系统变量也是如此）。例如如下所示：

mysql> show variables like 'slow\_query%';

+---------------------+-----------------------------------------------+

| Variable\_name       | Value                                         |

+---------------------+-----------------------------------------------+

| slow\_query\_log      | OFF                                           |

| slow\_query\_log\_file | /home/WDPM/MysqlData/mysql/DB-Server-slow.log |

+---------------------+-----------------------------------------------+

2 rows in set (0.01 sec)

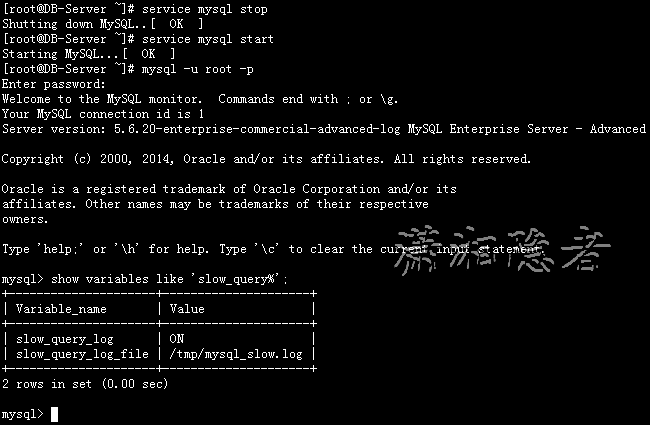
mysql>

[](http://images2015.cnblogs.com/blog/73542/201606/73542-20160617103909885-101540424.png)

修改my.cnf文件，增加或修改参数slow\_query\_log 和slow\_query\_log\_file后，然后重启MySQL服务器，如下所示

slow\_query\_log =1

slow\_query\_log\_file=/tmp/mysql\_slow.log

[](http://images2015.cnblogs.com/blog/73542/201606/73542-20160617103912370-1034590693.png)

mysql> show variables like 'slow\_query%';

+---------------------+---------------------+

| Variable\_name       | Value               |

+---------------------+---------------------+

| slow\_query\_log      | ON                  |

| slow\_query\_log\_file | /tmp/mysql\_slow.log |

+---------------------+---------------------+

2 rows in set (0.00 sec)

mysql>

关于慢查询的参数slow\_query\_log\_file ，它指定慢查询日志文件的存放路径，系统默认会给一个缺省的文件host\_name-slow.log（如果没有指定参数slow\_query\_log\_file的话）

mysql> show variables like 'slow\_query\_log\_file';

+---------------------+-----------------------------------------------+

| Variable\_name       | Value                                         |

+---------------------+-----------------------------------------------+

| slow\_query\_log\_file | /home/WDPM/MysqlData/mysql/DB-Server-slow.log |

+---------------------+-----------------------------------------------+

1 row in set (0.00 sec)

那么开启了慢查询日志后，什么样的SQL才会记录到慢查询日志里面呢？ 这个是由参数long\_query\_time控制，默认情况下long\_query\_time的值为10秒，可以使用命令修改，也可以在my.cnf参数里面修改。关于运行时间正好等于long\_query\_time的情况，并不会被记录下来。也就是说，在mysql源码里是判断大于long\_query\_time，而非大于等于。从MySQL 5.1开始，long\_query\_time开始以微秒记录SQL语句运行时间，之前仅用秒为单位记录。如果记录到表里面，只会记录整数部分，不会记录微秒部分。

mysql> show variables like 'long\_query\_time%';

+-----------------+-----------+

| Variable\_name   | Value     |

+-----------------+-----------+

| long\_query\_time | 10.000000 |

+-----------------+-----------+

1 row in set (0.00 sec)

mysql> set global long\_query\_time=4;

Query OK, 0 rows affected (0.00 sec)

mysql> show variables like 'long\_query\_time';

+-----------------+-----------+

| Variable\_name   | Value     |

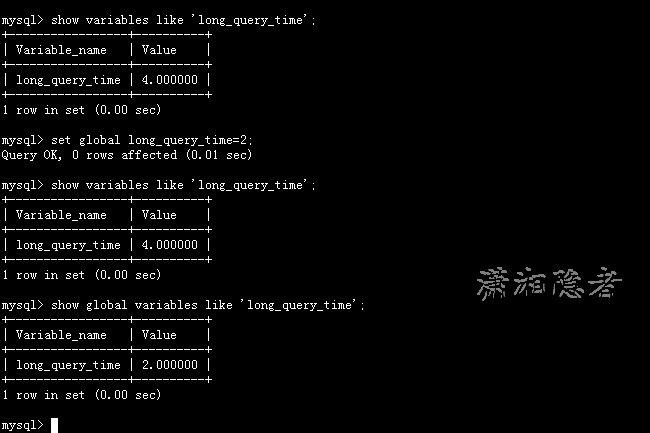
+-----------------+-----------+

| long\_query\_time | 10.000000 |

+-----------------+-----------+

1 row in set (0.00 sec)

如上所示，我修改了变量long\_query\_time，但是查询变量long\_query\_time的值还是10，难道没有修改到呢？注意：使用命令 set global long\_query\_time=4修改后，需要重新连接或新开一个会话才能看到修改值。你用show variables like 'long\_query\_time'查看是当前会话的变量值，你也可以不用重新连接会话，而是用show global variables like 'long\_query\_time'; 如下所示：

[](http://images2015.cnblogs.com/blog/73542/201606/73542-20160617103913557-1030002498.png)

在MySQL里面执行下面SQL语句，然后我们去检查对应的慢查询日志，就会发现类似下面这样的信息。

mysql> select sleep(3);

+----------+

| sleep(3) |

+----------+

|        0 |

+----------+

1 row in set (3.00 sec)

[root@DB-Server ~]# more /tmp/mysql\_slow.log

/usr/sbin/mysqld, Version: 5.6.20-enterprise-commercial-advanced-log (MySQL Enterprise Server - Advanced Edition (Commercial)). started with:

Tcp port: 0  Unix socket: (null)

Time                 Id Command    Argument

/usr/sbin/mysqld, Version: 5.6.20-enterprise-commercial-advanced-log (MySQL Enterprise Server - Advanced Edition (Commercial)). started with:

Tcp port: 0  Unix socket: (null)

Time                 Id Command    Argument

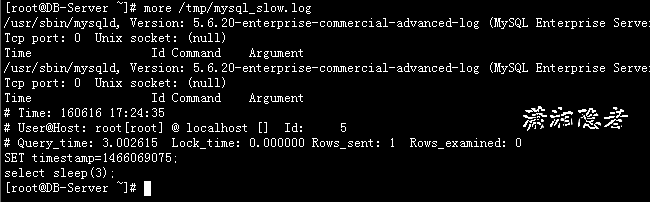
# Time: 160616 17:24:35

# User@Host: root[root] @ localhost []  Id:     5

# Query\_time: 3.002615  Lock\_time: 0.000000 Rows\_sent: 1  Rows\_examined: 0

SET timestamp=1466069075;

select sleep(3);

[](http://images2015.cnblogs.com/blog/73542/201606/73542-20160617103136057-1241105211.png)

log\_output 参数是指定日志的存储方式。log\_output='FILE'表示将日志存入文件，默认值是'FILE'。log\_output='TABLE'表示将日志存入数据库，这样日志信息就会被写入到mysql.slow\_log表中。MySQL数据库支持同时两种日志存储方式，配置的时候以逗号隔开即可，如：log\_output='FILE,TABLE'。日志记录到系统的专用日志表中，要比记录到文件耗费更多的系统资源，因此对于需要启用慢查询日志，又需要能够获得更高的系统性能，那么建议优先记录到文件。

mysql> show variables like '%log\_output%';

+---------------+-------+

| Variable\_name | Value |

+---------------+-------+

| log\_output    | FILE  |

+---------------+-------+

1 row in set (0.00 sec)

mysql> set global log\_output='TABLE';

Query OK, 0 rows affected (0.00 sec)

mysql> show variables like '%log\_output%';

+---------------+-------+

| Variable\_name | Value |

+---------------+-------+

| log\_output    | TABLE |

+---------------+-------+

1 row in set (0.00 sec)

mysql> select sleep(5) ;

+----------+

| sleep(5) |

+----------+

|        0 |

+----------+

1 row in set (5.00 sec)

mysql>

mysql> select \* from mysql.slow\_log;

+---------------------+---------------------------+------------+-----------+-----------+---------------+----+----------------+-----------+-----------+-----------------+-----------+

| start\_time          | user\_host                 | query\_time | lock\_time | rows\_sent | rows\_examined | db | last\_insert\_id | insert\_id | server\_id | sql\_text        | thread\_id |

+---------------------+---------------------------+------------+-----------+-----------+---------------+----+----------------+-----------+-----------+-----------------+-----------+

| 2016-06-16 17:37:53 | root[root] @ localhost [] | 00:00:03   | 00:00:00  |         1 |             0 |    |              0 |         0 |         1 | select sleep(3) |         5 |

| 2016-06-16 21:45:23 | root[root] @ localhost [] | 00:00:05   | 00:00:00  |         1 |             0 |    |              0 |         0 |         1 | select sleep(5) |         2 |

+---------------------+---------------------------+------------+-----------+-----------+---------------+----+----------------+-----------+-----------+-----------------+-----------+

2 rows in set (0.00 sec)

mysql>

系统变量log-queries-not-using-indexes：未使用索引的查询也被记录到慢查询日志中（可选项）。如果调优的话，建议开启这个选项。另外，开启了这个参数，其实使用full index scan的sql也会被记录到慢查询日志。

This option does not necessarily mean that no index is used. For example, a query that uses a full index scan uses an index but would be logged because the index would not limit the number of rows.

mysql> show variables like 'log\_queries\_not\_using\_indexes';

+-------------------------------+-------+

| Variable\_name                 | Value |

+-------------------------------+-------+

| log\_queries\_not\_using\_indexes | OFF   |

+-------------------------------+-------+

1 row in set (0.00 sec)

mysql> set global log\_queries\_not\_using\_indexes=1;

Query OK, 0 rows affected (0.00 sec)

mysql> show variables like 'log\_queries\_not\_using\_indexes';

+-------------------------------+-------+

| Variable\_name                 | Value |

+-------------------------------+-------+

| log\_queries\_not\_using\_indexes | ON    |

+-------------------------------+-------+

1 row in set (0.00 sec)

mysql>

系统变量log\_slow\_admin\_statements表示是否将慢管理语句例如ANALYZE TABLE和ALTER TABLE等记入慢查询日志

mysql> show variables like 'log\_slow\_admin\_statements';

+---------------------------+-------+

| Variable\_name             | Value |

+---------------------------+-------+

| log\_slow\_admin\_statements | OFF   |

+---------------------------+-------+

1 row in set (0.00 sec)

mysql>

系统变量log\_slow\_slave\_statements 表示

By default, a replication slave does not write replicated queries to the slow query log. To change this, use thelog\_slow\_slave\_statements system variable.

When the slow query log is enabled, this variable enables logging for queries that have taken more than long\_query\_time seconds to execute on the slave. This variable was added in MySQL 5.7.1. Setting this variable has no immediate effect. The state of the variable applies on all subsequent START SLAVE statements.

参数--log-short-format

The server writes less information to the slow query log if you use the --log-short-format option.

|  |  |  |
| --- | --- | --- |
|  **Command-Line Format** | --log-short-format | |
| **Permitted Values** | **Type** | boolean |
| **Default** | FALSE |

另外，如果你想查询有多少条慢查询记录，可以使用系统变量。

mysql> show global status like '%Slow\_queries%';

+---------------+-------+

| Variable\_name | Value |

+---------------+-------+

| Slow\_queries  | 2104  |

+---------------+-------+

1 row in set (0.00 sec)

mysql>

## 日志分析工具mysqldumpslow

在生产环境中，如果要手工分析日志，查找、分析SQL，显然是个体力活，MySQL提供了日志分析工具mysqldumpslow

查看mysqldumpslow的帮助信息：

[root@DB-Server ~]# mysqldumpslow --help

Usage: mysqldumpslow [ OPTS... ] [ LOGS... ]

Parse and summarize the MySQL slow query log. Options are

  --verbose    verbose

  --debug      debug

  --help       write this text to standard output

  -v           verbose

  -d           debug

  -s ORDER     what to sort by (al, at, ar, c, l, r, t), 'at' is default

                al: average lock time

                ar: average rows sent

                at: average query time

                 c: count

                 l: lock time

                 r: rows sent

                 t: query time

  -r           reverse the sort order (largest last instead of first)

  -t NUM       just show the top n queries

  -a           don't abstract all numbers to N and strings to 'S'

  -n NUM       abstract numbers with at least n digits within names

  -g PATTERN   grep: only consider stmts that include this string

  -h HOSTNAME  hostname of db server for \*-slow.log filename (can be wildcard),

               default is '\*', i.e. match all

  -i NAME      name of server instance (if using mysql.server startup script)

  -l           don't subtract lock time from total time

-s, 是表示按照何种方式排序，

c: 访问计数

l: 锁定时间

r: 返回记录

t: 查询时间

al:平均锁定时间

ar:平均返回记录数

at:平均查询时间

-t, 是top n的意思，即为返回前面多少条的数据；

-g, 后边可以写一个正则匹配模式，大小写不敏感的；

比如

得到返回记录集最多的10个SQL。

mysqldumpslow -s r -t 10 /database/mysql/mysql06\_slow.log

得到访问次数最多的10个SQL

mysqldumpslow -s c -t 10 /database/mysql/mysql06\_slow.log

得到按照时间排序的前10条里面含有左连接的查询语句。

mysqldumpslow -s t -t 10 -g “left join” /database/mysql/mysql06\_slow.log

另外建议在使用这些命令时结合 | 和more 使用 ，否则有可能出现刷屏的情况。

mysqldumpslow -s r -t 20 /mysqldata/mysql/mysql06-slow.log | more

**参考资料：**

https://dev.mysql.com/doc/refman/5.7/en/slow-query-log.html

https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html#sysvar\_long\_query\_time