

关于MySQL慢日志，你想知道的都在这里

2017-07-20 11:21

阅读 5.1k

评论 0



作者介绍

邹鹏，现任职于腾讯云数据库团队，负责腾讯云数据库MySQL中间件研发，多年的数据库、网络安全研发经验，对云计算平台的网络、计算、存储、安全有着深入的了解，在MySQL的高可用、高可靠、中间件方面有丰富的经验。

```
# Time: 170524 18:21:04
# User@Host: user[user] @ [192.168.0.7] Id: 2058705
# Query_time: 6.977880 Lock_time: 0.000056 Rows_sent: 2 Rows_examined:
3961943
SET timestamp=1495621264;
SELECT column1,column2,COUNT(1)
FROM db.table
WHERE column1 ='20170524' AND column2 = 'xxx'
GROUP BY column1,column2;
```

目录：

1. 什么是慢日志？
2. 什么情况下产生慢日志？
3. 慢日志相关参数
4. 慢日志输出内容
5. 慢日志分析工具
6. 慢日志的清理与备份

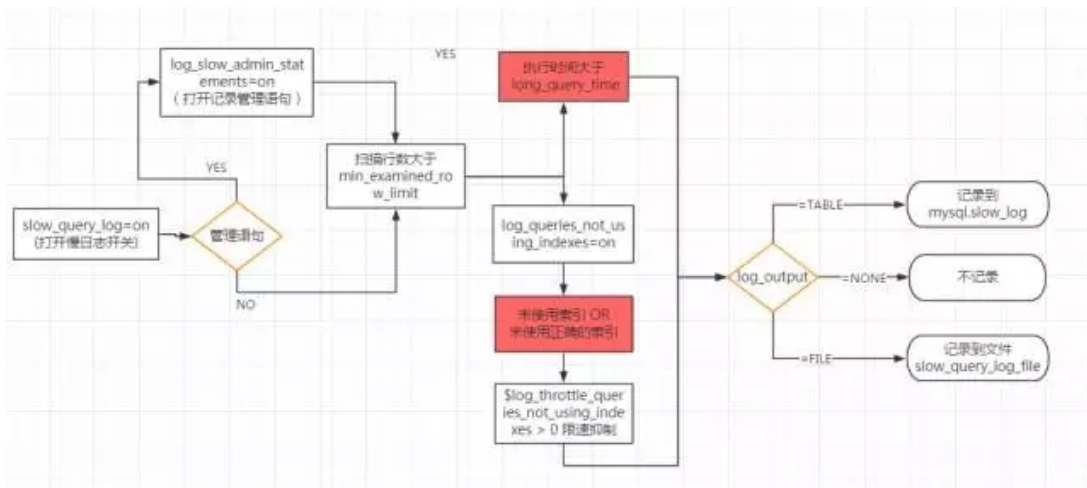
一、什么是慢日志？

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中响应时间超过阈值的语句，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或多或少会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。

- 5.6官方说明：<https://dev.mysql.com/doc/refman/5.6/en/slow-query-log.html>
- 5.7官方说明：<https://dev.mysql.com/doc/refman/5.7/en/slow-query-log.html>

二、什么情况下产生慢日志？



看图说话，有很多开关影响着慢日志的生成，相关的参数后面会挨个说明。从上图可以看出慢日志输出的内容有两个，第一执行时间过长（大于设置的long_query_time阈值）；第二未使用索引，或者未使用最优的索引。

这两种日志默认情况下都没有打开，特别是未使用索引的日志，因为这一类的日志可能会有很多，所以还有个特别的开关log_throttle_queries_not_using_indexes用于限制每分钟输出未使用索引的日志数量。

关键代码如下：

```

// (5.6.34) 代码路径 /sql/sql_parse.cc : 1812
bool log_slow_applicable(THD *thd)
{
    if (thd->enable_slow_log)
    {
        bool warn_no_index= ((thd->server_status &
            (SERVER_QUERY_NO_INDEX_USED |
            SERVER_QUERY_NO_GOOD_INDEX_USED)) &&
            opt_log_queries_not_using_indexes &&
            !(sql_command_flags[thd->lex->sql_command] &
            CF_STATUS_COMMAND));

        bool log_this_query= ((thd->server_status & SERVER_QUERY_WAS_SLOW) ||
            warn_no_index) &&
            (thd->get_examined_row_count() >=
            thd->variables.min_examined_row_limit);

        bool suppress_logging= log_throttle_qni.log(thd, warn_no_index);
        if (!suppress_logging && log_this_query)
            DEBUG_RETURN(true);
    }
    DEBUG_RETURN(false);
}
  
```

Slow log调用栈（MySQL 5.6.34）：

```

sql_cache.cc: 1003: 3: | | >Query_cache::end_of_result
sql_cache.cc: 1007: 3: | | <Query_cache::end_of_result
1007sql_parse.cc: 1937: 3: | | >log_slow_statement
sql_parse.cc: 1862: 4: | | | >log_slow_applicable
sql_parse.cc: 1891: 4: | | | <log_slow_applicable 1891
sql_parse.cc: 1906: 4: | | | >log_slow_do
sql_parse.cc: 611: 4: | | | THD::enter_stage:
sql_profile.cc: 348: 5: | | | >PROFILING::status_change
sql_profile.cc: 354: 5: | | | <PROFILING::status_change 354
log.cc: 1999: 5: | | | >MYSQL_QUERY_LOG::write
mf_iocache.c: 1740: 6: | | | | >my_b_flush_io_cache
mf_iocache.c: 1741: 6: | | | | enter: cache: 0x2ca7f50
my_write.c: 45: 7: | | | | | >my_write
my_write.c: 47: 7: | | | | | my: fd: 31 Buffer: ...
my_write.c: 115: 7: | | | | | <my_write 115
mf_iocache.c: 1809: 6: | | | | <my_b_flush_io_cache 1809
log.cc: 2117: 5: | | | | <MYSQL_QUERY_LOG::write 2117
sql_parse.cc: 1918: 4: | | | <log_slow_do 1918
sql_parse.cc: 1942: 3: | | <log_slow_statement 1942

```

三、慢日志相关参数

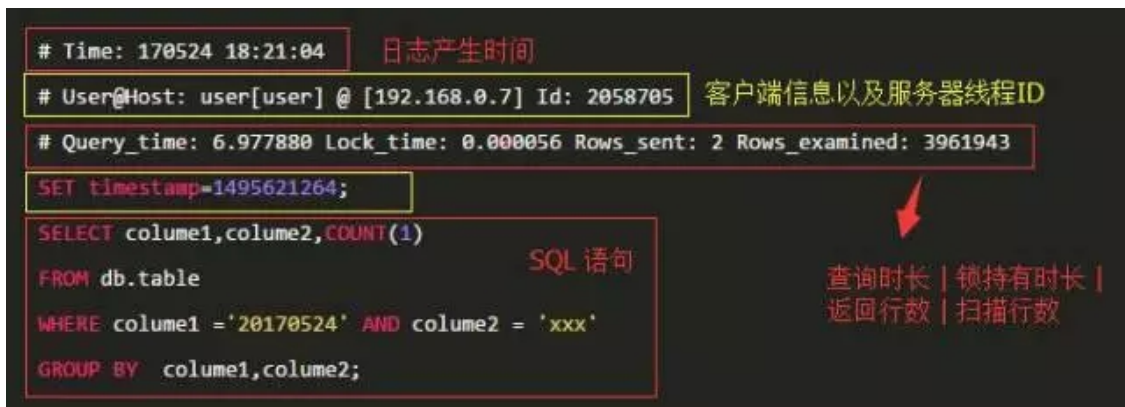
配置参数	系统变量	作用域	取值	默认
--slow-query-log	slow_query_log	全局	OFF ON	OFF
无	log_slow_admin_statements	全局	OFF ON	OFF
--min-examined-row-limit	min_examined_row_limit	全局 会话	[0-xxx]	0
--long-query-time	long_query_time	全局 会话	[0-xxx]	10
--log-queries-not-using-indexes	log-queries-not-using-indexes	全局	OFF ON	OFF
无	log_throttle_queries_not_using_indexes	全局	[0-xxx]	0
--log_output	log_output	全局	[TABLE FILE NONE]	FILE
--slow_query_log_file	slow_query_log_file	全局	string	host_name-slow.log
--log-short-format		全局	TRUE FALSE	FALSE

以上应该是最完整的和慢日志相关的所有参数，大多数参数都有前置条件，所以在使用的时候可以参照上面的流程图。

5.6官方文档：

- 1、<https://dev.mysql.com/doc/refman/5.6/en/server-system-variables.html>
- 2、<https://dev.mysql.com/doc/refman/5.6/en/server-options.html>

四、慢日志输出内容



第一行：标记日志产生的时间，准确说是SQL执行完成的时间点，改行记录每一秒只打印一条。

第二行：客户端的账户信息，两个用户名（第一个是授权账户，第二个为登录账户），客户端IP地址，还有mysqld的线程ID。

第三行：查询执行的信息，包括查询时长，锁持有时长，返回客户端的行数，扫描行数。通常我需要优化的就是最后一个内容，尽量减少SQL语句扫描的数据行数。

第四行：通过代码看，貌似和第一行的时间没有区别。

第五行：最后就是产生慢查询的SQL语句。

–log-short-format=true：

如果mysqld启动时指定了–log-short-format参数，则不会输出第一、第二行。

log-queries-not-using-indexes=on

log_throttle_queries_not_using_indexes > 0：

如果启用了以上两个参数，每分钟超过log_throttle_queries_not_using_indexes配置的未使用索引的慢日志将会被抑制，被抑制的信息会被汇总，每分钟输出一次。

格式如下：

```
# Time: 170526 11:26:10
# User@Host: [] @ [] Id: 38
# Query_time: 0.021872 Lock_time: 0.008620 Rows_sent: 0 Rows_examined: 0SET
timestamp=1495769170;
throttle: 14 'index not used' warning(s) suppressed.;
```

五、慢日志分析工具

1. 官方自带工具：mysqldumpslow
2. 开源工具：mysqslsla

3. percona-toolkit: 工具包中的pt-query-digest工具可以分析汇总慢查询信息，具体逻辑可以看SlowLogParser这个函数。

总的来说，MySQL的日志内容本身不复杂，上面3个工具都是用perl脚本实现，代码行数不超过200行，有兴趣的同学也可以自己尝试着解析下。

详情可以参阅下这篇文章：

- 《MySQL 慢查询设置和分析工具》：
https://flyerboy.github.io/2016/12/23/mysql_slow/

以上工具可以支撑慢日志的常用统计，但是当我们需要做到SQL级别的统计时，我们还需要取解析SQL把参数提取出来。

六、慢日志的清理与备份

删除：直接删除慢日志文件，执行flush logs（必须的）。

备份：先用mv重命名文件（不要跨分区），然后执行flush logs（必须的）。

另外修改系统变量slow_query_log_file也可以立即生效；

执行flush logs，系统会先close当前的句柄，然后重新open；mv, rm日志文件系统并不会报错，具体的原因可以Google下linux i_count i_nlink；

文章来自微信公众号：DBAplus社群