



下载APP



19 | 日志（上）：系统出现问题，如何及时发现？

2021-04-22 朱晓峰

MySQL 必知必会

[进入课程 >](#)**讲述：朱晓峰**

时长 10:18 大小 9.44M



你好，我是朱晓峰。

我们曾经开发过一个数据库应用系统，但是却突然遭遇了数据库宕机。在这种情况下，定位宕机的原因就非常关键，毕竟，知道了问题，才能确定解决方案。

这时，我们就想到了查看数据库的错误日志，因为日志中记录了数据库运行中的诊断信息，包括了错误、警告和注释等信息。从日志中，我们发现，原来某个连接中的 SQL 操作发生了死循环，导致内存不足，被系统强行终止了。知道了原因，处理起来也就比较轻松了，系统很快就恢复了运行。



除了发现错误，日志在数据复制、数据恢复、操作审计，以及确保数据的永久性和一致性等方面，都有着不可替代的作用，对提升你的数据库应用的开发能力至关重要。

今天，我就结合超市项目的实际案例，给你讲解一下怎么通过查看系统日志，来了解数据库中实际发生了什么，从而快速定位原因。

MySQL 的日志种类非常多，包括通用查询日志、慢查询日志、错误日志、二进制日志、中继日志、重做日志和回滚日志，内容比较多，而且都很重要，所以我们来花两节课的时间学习一下。

这节课，我会先具体讲一讲通用查询日志、慢查询日志和错误日志。

通用查询日志

通用查询日志记录了所有用户的连接开始时间和截止时间，以及发给 MySQL 数据库服务器的所有 SQL 指令。当我们的数据发生异常时，开启通用查询日志，还原操作时的具体场景，可以帮助我们准确定位问题。

举个小例子，在超市项目实施的过程中，我们曾遇到过这样一件事：超市经营者月底查账的时候发现，超市的 1 号门店在 12 月 1 日销售了 5 件化妆品，但是当天对应的历史库存并没有减少。化妆品的金额都比较大，库存不对的话，会在报表查询中产生巨额差异，触发到报警机制，对超市经营者的决策产生影响。超市经营者找到我们，对系统的可靠性提出质疑。


我们对系统进行了仔细检查，没有发现数据问题。可是商品确实卖出去了，当天的历史库存也确实没有消减。这个时候，我们想到了检查通用查询日志，看看当天到底发生了什么。

查看之后，我们就复原了当天的情况：12 月 1 日下午，门店的收银台销售了 5 件化妆品，但是由于网络故障，流水没有及时上传到总部。12 月 1 日晚上 11:59，总部的历史库存被保存下来，但是因为没有收到门店的流水，所以没有消减库存。12 月 2 日上午，门店的网络恢复了，流水得以上传总部，这个时候，对应化妆品的库存才被消减掉。

这样，我们就确定了故障的原因，也就是超市的网络问题，而系统本身是没有问题的。

你看，通用查询日志可以帮助我们了解操作发生的具体时间和操作的细节，对找出异常发生的原因极其关键。

下面我来具体介绍一下控制通用查询日志的系统变量。通过这些变量，你会清楚怎么控制通用查询日志的开启和关闭，以及保存日志的文件是哪个。

 复制代码

```
1 mysql> SHOW VARIABLES LIKE '%general%';
2 +-----+
3 | Variable_name | Value |
4 +-----+
5 | general_log   | OFF   | -- 通用查询日志处于关闭状态
6 | general_log_file | GJTECH-PC.log | -- 通用查询日志文件的名称是GJTECH-PC.log
7 +-----+
8 2 rows in set, 1 warning (0.00 sec)
```


在这个查询的结果中，有 2 点需要我们注意一下。

1. 系统变量 `general_log` 的值是 `OFF`，表示通用查询日志处于关闭状态。在 MySQL 中，这个参数的默认值是关闭的。因为一旦开启记录通用查询日志，MySQL 会记录所有的连接起止和相关的 SQL 操作，这样会消耗系统资源并且占用磁盘空间。我们可以通过手动修改变量的值，在需要的时候开启日志。
2. 通用查询日志文件的名称是 `GJTECH-PC.log`。这样我们就知道在哪里可以查看通用查询日志的内容了。

下面我们来看看如何开启通用查询日志，把所有连接的起止和连接的 SQL 操作都记录下来。这个操作可以帮助我们追踪 SQL 操作故障的原因。

开启通用查询日志

我们可以通过设置系统变量的值，来开启通用查询日志，并且指定通用查询日志的文件夹和文件名为 `"H:\mytest.log"`。这个操作如下：

 复制代码

```
1 mysql> SET GLOBAL general_log = 'ON';
2 Query OK, 0 rows affected (0.00 sec)
3
4 mysql> SET @@global.general_log_file = 'H:\mytest.log';
5 Query OK, 0 rows affected (0.02 sec)
```

为了确认我们的设定是否已经生效，我们再来查询一下通用查询日志的状态：

[复制代码](#)

```
1 mysql> SHOW VARIABLES LIKE '%general%';
2 +-----+-----+
3 | Variable_name | Value |
4 +-----+-----+
5 | general_log   | ON    | -- 通用查询日志开启
6 | general_log_file | H:\mytest.log | -- 日志名称也改过了
7 +-----+-----+
8 2 rows in set, 1 warning (0.00 sec)
```

结果显示，通用查询日志已经开启，文件是“H:\mytest.log”，这就意味着我们的操作成功了。

查看通用查询日志

通用查询日志都是文本型数据，可以用记事本打开。下面我们就用记事本打开我电脑上的通用查询日志，实际看一看通用查询日志的内容，包括都有哪些连接，什么时候登录了数据库，都做了哪些操作等信息。

[复制代码](#)

```
1 2021-04-05T06:39:53.621980Z 28 Connect zhangsan@localhost on using SSL/TLS -
2 2021-04-05T06:39:53.622085Z 28 Connect Access denied for user 'zhangsan'@'loca
3 2021-04-05T06:40:02.522303Z 29 Connect zhangsan@localhost on using SSL/TLS
4 2021-04-05T06:40:02.522913Z 29 Query select @@version_comment limit 1
5 2021-04-05T06:40:14.211511Z 29 Query SELECT *
6 FROM demo.invcount -- 查询数据表demo.invcount内容
7 2021-04-05T06:40:37.647625Z 29 Query UPDATE demo.invcount
8 SET plquant = - 5 -- 更新数据表demo.invcount
9 WHERE itemnumber = 1
10 2021-04-05T06:41:15.047067Z 29 Query SELECT *
11 FROM demo.goodsmaster -- 查询数据表demo.goodsmaster
```

在通用查询日志里面，我们可以清楚地看到，账号“zhangsan”是什么时间登录的服务，登录之后做了什么 SQL 操作，针对的是哪个数据表等信息。

删除通用查询日志

当用户对数据库的操作比较频繁时，通用查询日志文件会不断变大。为了节省磁盘空间，我们可以移除旧的日志文件，创建新的日志文件，来对通用查询日志文件进行维护。

第一步，关闭通用查询日志：

[复制代码](#)

```
1 mysql> SET GLOBAL general_log = 'OFF'; -- 关闭通用查询日志
2 Query OK, 0 rows affected (0.01 sec)
3
4 mysql> SHOW VARIABLES LIKE '%general_log%'; -- 查看通用查询日志状态
5 +-----+-----+
6 | Variable_name | Value |
7 +-----+-----+
8 | general_log   | OFF   |
9 | general_log_file | H:mytest.log |
10 +-----+-----+
11 2 rows in set, 1 warning (0.00 sec)
```

第二步，把通用查询日志文件 “H:\mytest.log” 移至备份文件夹，空出磁盘 H 的空间。


第三步，开启通用查询日志：

[复制代码](#)

```
1 mysql> SET GLOBAL general_log = 'ON';
2 Query OK, 0 rows affected (0.01 sec)
3
4 mysql> SHOW VARIABLES LIKE '%general_log%';
5 +-----+-----+
6 | Variable_name | Value |
7 +-----+-----+
8 | general_log   | ON    |
9 | general_log_file | H:mytest.log |
10 +-----+-----+
11 2 rows in set, 1 warning (0.00 sec)
```

这个时候，你会发现，MySQL 已经给我们准备好了一个新的通用查询日志文件 “H:\mytest.log”，并且记录了我们第一个查询的语句：“SHOW VARIABLES LIKE '%general_log%'”。

文件内容如下：

 复制代码

```
1 C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld.exe, Version: 8.0.23 (MySQL
2 TCP Port: 3306, Named Pipe: MySQL
3 Time Id Command Argument
4 2021-04-05T07:02:03.007394Z 30 Query SHOW VARIABLES LIKE '%general_log%'
```

总之，开启了通用查询日志之后，如果遇到用户对数据产生质疑的情况，我们就可以通过查看通用查询日志，还原当时的场景，快速定位并解决问题。


慢查询日志

慢查询日志用来记录执行时间超过指定时长的查询。它的主要作用是，帮助我们发现那些执行时间特别长的 SQL 查询，并且有针对性地进行优化，从而提高系统的整体效率。当我们的数据库服务器发生阻塞、运行变慢的时候，检查一下慢查询日志，找到那些慢查询，对解决问题很有帮助。

慢查询日志是由 MySQL 的配置文件进行控制的。下面我先简单介绍一下 MySQL 的配置文件。

在 MySQL 的安装目录中（C:\ProgramData\MySQL\MySQL Server 8.0），我们可以找到 MySQL 的配置文件“my.ini”。这个文件是一个文本格式的文件，可以直接用记事本打开来阅读。

我们来看看配置文件中关于慢查询日志变量的相关设定：

 复制代码

```
1 slow-query-log=1 -- 表示开启慢查询日志，系统将会对慢查询进行记录。
2
3 slow_query_log_file="GJTECH-PC-slow.log" -- 表示慢查询日志的名称是"GJTECH-PC-slow
4
5 long_query_time=10 -- 表示慢查询的标准是查询执行时间超过10秒
```

除了刚刚的这些变量，控制慢查询日志的还有一个系统变量：min_examined_row_limit。这个变量的意思是，查询扫描过的最少记录数。这个变量和查询执行时间，共同组成了判别一个查询是否是慢查询的条件。如果查询扫描过的记录数大于等于这个变量的值，并且

查询执行时间超过 `long_query_time` 的值，那么，这个查询就被记录到慢查询日志中；反之，则不被记录到慢查询日志中。

如果要查看当前这个系统变量的值，我们就可以用下面的代码：

[复制代码](#)

```
1 mysql> show variables like 'min%';
2 +-----+-----+
3 | Variable_name | Value |
4 +-----+-----+
5 | min_examined_row_limit | 0 |
6 +-----+-----+
7 1 row in set, 1 warning (0.00 sec)
```

这个值默认是 0。与 `long_query_time=10` 合在一起，表示只要查询的执行时间超过 10 秒钟，哪怕一个记录也没有扫描过，都要被记录到慢查询日志中。你也可以根据需要，通过修改 “my.ini” 文件，来修改查询时长，或者通过 SET 指令，用 SQL 语句修改 “min_examined_row_limit” 的值。

只是你要注意，如果修改了 MySQL 的配置文件 “my.ini”，就需要重启服务器，这样才能使修改生效。

来看一个例子：之前我运行的一个慢查询，被记录到了慢查询日志中。这个例子记录了一个运行时间超过 10 秒的慢查询的发生时间、连接所属的用户、执行的时长、锁表的时长和扫描过的记录数等相关信息。

[复制代码](#)

```
1 C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld.exe, Version: 8.0.23 (MySQL
2 TCP Port: 3306, Named Pipe: MySQL
3 Time Id Command Argument
4 # Time: 2021-03-25T07:20:33.412260Z      -- 执行开始时间
5 # User@Host: root[root] @ localhost [::1] Id: 13  -- 用户
6 # Query_time: 10.166435 Lock_time: 0.000000 Rows_sent: 0  Rows_examined: 0
7 use demo;
8 SET timestamp=1616656823;
```

通过这个慢查询日志的记录，我们就可以发现是哪个查询消耗了大量的系统资源，是哪个连接里面的查询，具体什么时间开始的。有了这些信息，我们就可以对慢查询进行分析，

决定优化的方式，避免出现同样的问题。

好了，到这里，通用查询日志和慢查询日志我就讲完了，咱们最后再来学习一种重要的日志：错误日志。

错误日志

错误日志记录了 MySQL 服务器启动、停止运行的时间，以及系统启动、运行和停止过程中的诊断信息，包括错误、警告和提示等。当我们的数据库服务器发生系统故障时，错误日志是发现问题、解决故障的首选。

错误日志默认是开启的。我们可以在 MySQL 的配置文件 “my.ini” 中配置它：

```
1 # Error Logging.  
2 log-error="GJTECH-PC.err"
```

[复制代码](#)

这段代码指定了错误日志的文件名。如果没有指定文件夹，默认就是数据目录：“C:\ProgramData\MySQL\MySQL Server 8.0\Data”。

下面我们查看一下错误日志的内容：

```
1 2021-02-28T08:07:07.228880Z 0 [System] [MY-010116] [Server] C:\Program Files\M  
2 2021-02-28T08:07:07.270982Z 1 [System] [MY-013576] [InnoDB] InnoDB initializat  
3 2021-02-28T08:07:08.116433Z 1 [System] [MY-013577] [InnoDB] InnoDB initializat
```

[复制代码](#)

可以看到，错误日志文件中记录了服务器启动的时间，以及存储引擎 InnoDB 启动和停止的时间等。

总结

今天这节课，我们学习了通用查询日志、慢查询日志和错误日志。

通用查询日志：可以记录所有连接的起始时间和终止时间，以及连接发送给数据库服务器的所有指令，对我们复原操作的实际场景、发现问题，甚至是对数据库操作的审计都有很大的帮助。

慢查询日志：可以记录运行时间和检查记录数超过指定值的查询，方便我们对查询进行优化。

错误日志：它记录了服务器启动、运行和停止过程中的诊断信息，方便我们了解服务器的状态，从而对服务器进行维护。

最后，我还是想提醒你一句，**千万不要小看日志**。很多看似奇怪的问题，答案往往就藏在日志里。很多情况下，只有通过查看日志才能发现问题的原因，真正解决问题。所以，一定要学会查看日志，养成检查日志的习惯。

思考题

请你思考一下：怎么设置开启慢查询日志，并且把慢查询日志的指定时长设置为 5 秒，最少扫描记录数为 1？

欢迎在留言区写下你的思考和答案，我们一起交流讨论。如果你觉得今天的内容对你有所帮助，也欢迎你分享给你的朋友或同事，我们下节课见。

提建议

更多课程推荐

深入浅出计算机组成原理

带你掌握计算机体系全貌

徐文浩

bothub 创始人



涨价倒计时 🕒

今日订阅 **¥89**，5月12日涨价至 **¥199**

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 18 | 权限管理：如何控制数据库访问，消除安全隐患？

下一篇 20 | 日志（下）：系统故障，如何恢复数据？

精选留言 (3)

写留言



不学完不改名

2021-04-25

general query log 目前在开发过程中用的很多，方便查看和分析系统都执行了那些 query 操作。

不过，它也有一定性能问题，一般并发高的生产环境中的机器也不建议开启。

展开 ∨



1



lesserror


2021-04-22

MySQL的日志类型大概可分为：通用查询日志、慢查询日志、错误日志、二进制日志、中继日志、重做日志和回滚日志。

对于任何一种日志来说，都可以准确的帮助我们定位和解决问题。所以MySQL的学习中，对于日志内容的学习是难点，但是不可获取。 ...

展开 ▾



Harry 
2021-04-22

很多看似奇怪的问题，答案往往就藏在日志里。

本节课成功引起了我对日志的兴趣。咱一定要学会查看日志，养成定期检查日志的习惯。

展开 ▾

