

MySQL状态变量Aborted\_connects与Aborted\_clients浅析

2018-06-21 00:25 潇湘隐者 阅读(1354) 评论(0) 编辑 收藏

关于MySQL的状态变量Aborted\_clients & Aborted\_connects分别代表的意义，以及哪些情况或因素会导致这些状态变量变化呢？下文通过实验测试来验证一下，首先我们来看看状态变量的描述：

Aborted Connect

Aborted Connect表示尝试连接到MySQL服务器失败的次数。这个状态变量可以结合host\_cache表和其错误日志一起来分析问题。引起这个状态变量激增的原因如下：

- 1、 客户端没有权限但是尝试访问MySQL数据库。
- 2、 客户端输入的密码有误。
- 3、 A connection packet does not contain the right information.
- 4、 超过连接时间限制，主要是这个系统变量connect\_timeout控制（mysql默认是10s，基本上，除非网络环境极端不好，一般不会超时。）

官方解释如下：

If a client is unable even to connect, the server increments the [Aborted\\_connects](#) status variable. Unsuccessful connection attempts can occur for the following reasons:

- A client attempts to access a database but has no privileges for it.
- A client uses an incorrect password.
- A connection packet does not contain the right information.
- It takes more than [connect\\_timeout](#) seconds to obtain a connect packet. See [Section 5.1.7, "Server System Variables"](#).

Aborted Clients:

Aborted Clients表示由于客户端没有正确关闭连接而中止的连接数。官方解释如下：

The number of connections that were aborted because the client died without closing the connection properly. See [Section B.5.2.10, "Communication Errors and Aborted Connections"](#).

当Aborted Clients增大的时候意味着有客户端某些原因断开连接或者被终止了，这种情况一般发生中。主要的可能性有：

20

About

个人简介: 网名潇湘隐者/潇湘剑客、英文名Kerry，兴趣广泛，广泛涉猎，个性随意。执意做一名会写代码的DBA，混迹于IT行业



个人微信公众号：



打赏

博客简介：

这里是潇湘隐者的一亩三分地，偶尔意兴阑珊的时候，整理打理下本“菜园”，本“菜园”主要关注各类数据库技术（ORACLE、MS SQL，MySQL，Hbase...）以及NOSQL、NET、JAVA等各类技术。在这里，记录着我成长、学习、工作过程的点点滴滴（笔记、总结、研究、问题解决...），本人对数据库管理、数据库开发、性能优化、数据挖掘、BI等有着浓厚的兴趣，欢迎大家一起探讨、研究各种技术话题！非诚勿扰。

联系方式： 邮箱地址: [kerry2008code@qq.com](mailto:kerry2008code@qq.com)

DBA—技术交流分享群: 335155934

DBA—技术交流分享群2: 294770674

加群请回答下面问题中任何一个：

- ORACLE：SGA由哪几部分组成？
- SQL SERVER: SQL锁类型有那些？

座右铭：命在于自造，运在于自改，福在于自求，境在于自选，友在于自择，习在于自改，悟在于自通，凡事在于自己。人生就是一场修行！走自己的道，修自己的行。脚踏实地、切忌浮躁，眼界决定高度，思路决定出路

年轻时的付出，都会是一种沉淀，它们会默默铺路，只为你成为更好的人。心简单，世界就简单，幸福才会生长；心自由，生活就自由，到哪都有快乐.回忆中，总有些瞬间，能温暖整个曾经...

昵称: [潇湘隐者](#)  
园龄: [10年5个月](#)  
荣誉: [推荐博客](#)  
粉丝: [2674](#)  
关注: [168](#)  
[+加关注](#)

SEARCH

最新随笔

- [SQL Server优化技巧——如何避免查询条件OR引起的性能问题](#)
- [MySQL能否授予查看存储过程定义权限给用户](#)
- [MySQL解惑——GROUP BY隐式排序](#)

- 连接。
- 1、 客户端程序在退出之前未调用mysql\_close（）正确关闭MySQL
- 2、 客户端休眠的时间超过了系统变量wait\_timeout和interactive\_timeout的值，导致连接被MySQL进程终止
- 3、 客户端程序在数据传输过程中突然结束

官方文档B.5.2.10 Communication Errors and Aborted Connections的介绍如下：

If a client successfully connects but later disconnects improperly or is terminated, the server increments the [Aborted\\_clients](#) status variable, and logs an Aborted connection message to the error log. The cause can be any of the following:

- The client program did not call [mysql\\_close\(\)](#) before exiting.
- The client had been sleeping more than [wait\\_timeout](#) or [interactive\\_timeout](#) seconds without issuing any requests to the server. See [Section 5.1.7, "Server System Variables"](#).
- The client program ended abruptly in the middle of a data transfer.

Other reasons for problems with aborted connections or aborted clients:

- The [max\\_allowed\\_packet](#) variable value is too small or queries require more memory than you have allocated for [mysqld](#). See [Section B.5.2.9, "Packet Too Large"](#).
- Use of Ethernet protocol with Linux, both half and full duplex. Some Linux Ethernet drivers have this bug. You should test for this bug by transferring a huge file using FTP between the client and server machines. If a transfer goes in burst-pause-burst-pause mode, you are experiencing a Linux duplex syndrome. Switch the duplex mode for both your network card and hub/switch to either full duplex or to half duplex and test the results to determine the best setting.
- A problem with the thread library that causes interrupts on reads.
- Badly configured TCP/IP.
- Faulty Ethernet, hubs, switches, cables, and so forth. This can be diagnosed properly only by replacing hardware.

如上介绍所示，有很多因素引起这些状态变量的值变化，那么我们来一个个分析、演示一下吧。首先，我们来测试一下导致Aborted\_clients增加的可能因素

- 1、 客户端没有权限但是尝试访问MySQL数据库。

SQL Server解惑——为什么你的查询结果超出了查询时间范围
MySQL的统计信息学习总结
[翻译]——MySQL 8.0 Histograms
Centos 7修改hostname浅析
MySQL的sleep函数的特殊特现象
阿里云RDS for SQL Serrver关于权限的一个严重Bug
SQL Server获取索引创建时间&重建时间&重组时间

最新评论
Re:ORA-12514, TNS:listener does not currently know of service requested in connect descriptor案例2
@ impwang那有意思，还没有碰到过jre崩溃引起这样的案例！ ... -- 潇湘隐者
Re:ORA-27125: unable to create shared memory segment
博主明显是企业用户吧？我个人在centos7搭建的oracle 19c，也遇到了这个问题，而且只在创建新数据库时有问题，如果在已有的数据库创建插接式数据库则没问题。如果这是一个bug的话，不应该19c... -- impwang
Re:ORA-12514, TNS:listener does not currently know of service requested in connect descriptor案例2
在12c上遇到和博主一模一样的问题，tnsping、lsnrctl但是怎么排查都没有找到错误，最后查看netca，竟然是jre崩溃 -- impwang
Re:shell脚本执行时报"bad interpreter: Text file busy"的解决方法
我使用的是kill的方式解决的。 -- 大象无形01
Re:Linux命令学习总结：dos2unix - unix2dos
666 -- Janvn

日历	随笔档案
< 2019年12月 >	2019年11月(7)
日 一 二 三 四 五 六	2019年10月(7)
1 2 3 4 5 6 7	2019年9月(12)
8 9 10 11 12 13 14	2019年8月(14)
15 16 17 18 19 20 21	2019年7月(8)
22 23 24 25 26 27 28	2019年6月(10)
29 30 31 1 2 3 4	2019年5月(7)
5 6 7 8 9 10 11	2019年4月(6)
随笔分类	2019年3月(9)
.NET技术(10)	2019年2月(8)
C/C++(2)	2019年1月(5)
DBA思考系列(2)	2018年12月(11)
log4net(1)	2018年11月(11)
NoSQL(Neo4j)(1)	2018年10月(11)
Python学习笔记(11)	2018年9月(10)
Unix&Linux技术(198)	2018年8月(10)
Window 技术汇集(16)	2018年7月(10)
编程技巧(3)	2018年6月(13)
存储网络(4)	2018年5月(12)
个人小软件(1)	2018年4月(13)
工欲善其事必先利其器(17)	2018年3月(11)
管理之道(1)	2018年2月(2)
其它技巧(1)	2018年1月(2)
软件重构(4)	2017年12月(2)
设计模式(1)	2017年11月(5)
数据库技术(Azure)(3)	2017年10月(10)
数据库技术(MS SQL)(315)	2017年9月(14)
数据库技术(My SQL)(93)	2017年8月(15)
数据库技术(Oracle)(309)	2017年7月(9)
数据库建模(1)	2017年6月(14)
数据库重构(2)	2017年5月(11)
数据挖掘及BI(1)	2017年4月(6)
	2017年3月(12)

其实这里所说的没有权限，个人理解是：**客户端使用没有授权的账号访问数据库**。打个比方，你尝试用账号kkk访问MySQL数据库，其实你也不知道数据库是否存在这个用户，实际上不存在这个用户。

实验对比测试前，先将状态变量清零。

```
mysql> flush status;
Query OK, 0 rows affected (0.01 sec)
mysql> show status like 'Abort%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Aborted_clients | 0 |
| Aborted_connects | 0 |
+-----+-----+
2 rows in set (0.01 sec)

mysql>
mysql> select host,user from mysql.user;
+-----+-----+
| host | user |
+-----+-----+
| % | mydba |
| % | root |
| % | test |
| 127.0.0.1 | root |
| 192.168.% | mydbadmin |
| 192.168.103.18,192.168.103.22 | LimitIP |
| ::1 | root |
| db-server.localdomain | root |
| localhost | backuser |
| localhost | root |
+-----+-----+
```

在本机的SecureCRT的另外一个窗口，使用不存在的账号kkk访问MySQL后，你会发现状态变量Aborted\_connects变为1了。

```
[root@DB-Server ~]# mysql -u kkk -p
Enter password:
ERROR 1045 (28000): Access denied for user 'kkk'@'localhost' (using
password: YES)
```

```
mysql> show status like 'Abort%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Aborted_clients | 0 |
| Aborted_connects | 0 |
+-----+-----+
2 rows in set (0.00 sec)

mysql> show status like 'Abort%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Aborted_clients | 0 |
| Aborted_connects | 1 |
+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

也有可能，这个账号本身存在，但是只允许特定IP地址才能访问，实际环境中，可能是有人在进行尝试暴力破解。可能性非常多。我们来测试一下限制IP访问的情况

```
mysql> grant all on MyDB.* to mydbadmin@'10.20.%'
identified by '123456';
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> show status like 'Abort%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
```

20

虚拟化技术(3)

杂想闲思录(14)

阅读排行榜

1. 深入理解Linux修改hostname(195465)

2. Linux 查看服务器开放的端口号(187910)

3. ORACLE基本数据类型总结(172389)

4. ORACLE VARCHAR2最大长度问题(151756)

5. MySQL ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)的真正原因(146578)

6. Crontab定时任务配置(143319)

7. Linux如何搜索查找文件里面内容(138298)

8. Linux mysql 5.6: ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)(136859)

9. Linux平台卸载MySQL总结(128450)

10. Linux如何查看JDK的安装路径(126348)

11. 查看mysql数据库版本方法总结(119817)

12. Linux的NTP配置总结(118601)

13. Linux如何查找大文件或目录总结(117297)

14. Linux查看系统开机时间(106388)

15. ORA-12514, TNS:listener does not currently know of service requested in connect descriptor案例2(104364)

文章分类

数据库学习

积分与排名

积分 - 1950365

排名 - 34

推荐排行榜

1. 你人生中的那口井挖了没有? (128)

2. SQL Server 中WITH (NOLOCK)浅析(91)

3. MS SQL 日常维护管理常用脚本(二)(54)

4. ORACLE基本数据类型总结(52)

5. 深入理解Linux修改hostname(49)

6. 重温SQL——行转列，列转行(47)

7. MySQL慢查询日志总结(40)

8. SQL SERVER中什么情况会导致索引查找变成索引扫描(33)

9. Linux如何查找大文件或目录总结(30)

10. ORACLE同义词总结(30)

11. Linux的NTP配置总结(29)

12. Linux 查看服务器开放的端口号(29)

2017年2月(8)

2017年1月(5)

2016年12月(9)

2016年11月(13)

2016年10月(10)

2016年9月(7)

2016年8月(24)

2016年7月(23)

2016年6月(17)

2016年5月(11)

2016年4月(13)

2016年3月(19)

2016年2月(5)

2016年1月(13)

2015年12月(15)

2015年11月(13)

2015年10月(12)

2015年9月(13)

2015年8月(21)

2015年7月(17)

2015年6月(11)

2015年5月(9)

2015年4月(11)

2015年3月(20)

2015年2月(10)

2015年1月(24)

2014年12月(6)

2014年11月(8)

2014年10月(14)

2014年9月(8)

2014年8月(8)

2014年7月(22)

2014年6月(9)

2014年5月(12)

2014年4月(13)

2014年3月(21)

2014年2月(4)

2014年1月(5)

2013年12月(7)

2013年11月(7)

2013年10月(10)

2013年9月(12)

2013年8月(24)

2013年7月(11)

2013年6月(9)

2013年5月(9)

2013年4月(8)

2013年3月(9)

2013年1月(2)

2012年12月(6)

2012年8月(1)

2012年7月(2)

2012年6月(3)

2012年5月(8)

2012年4月(5)

2011年7月(4)

2011年6月(4)

打赏

https://www.cnblogs.com/kerrycode/p/9206787.html3/13

Aborted_clients	0
Aborted_connects	0

2 rows in set (0.00 sec)

13. 记一次Linux服务器上查杀木马经历(27)

2011年5月(10)
14. ORACLE临时表总结(27)

2011年4月(6)
15. ORACLE的SQL JOIN方式小结(26)

2011年3月(4)

- 2010年12月(6)
- 2010年11月(1)
- 2010年10月(3)
- 2010年9月(6)
- 2010年8月(6)
- 2010年7月(5)
- 2010年5月(2)
- 2010年4月(3)
- 2010年3月(3)
- 2010年2月(3)
- 2010年1月(2)
- 2009年11月(3)
- 2009年10月(3)
- 2009年9月(1)
- 2009年8月(2)

打赏

如上所示，创建一个mydbadmin的行号，只允许10.20段的IP访问，然后我们从192.168段的IP访问MySQL数据库

```
# mysql -h 10.20.57.24 -u mydbadmin -p
Enter password:
ERROR 1045 (28000): Access denied for user
'mydbadmin'@'192.168.7.208' (using password: YES)
```

此时，状态变量Aborted\_connects就变为1了。

```
mysql> grant all on MyDB.* to mydbadmin@'10.20.%' identified by '123456';
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> show status like 'Abort%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Aborted_clients | 0     |
| Aborted_connects | 0     |
+-----+-----+
2 rows in set (0.00 sec)

mysql> show status like 'Abort%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Aborted_clients | 0     |
| Aborted_connects | 1     |
+-----+-----+
2 rows in set (0.00 sec)
```

2、客户端输入的密码有误或者根本就是尝试各个密码。(A client uses an incorrect password)

如下所示，使用test账号访问MySQL数据，但是输入了一个错误密码

```
[root@DB-Server ~]# mysql -u test -p
Enter password:
ERROR 1045 (28000): Access denied for user
'test'@'localhost' (using password: YES)
[root@DB-Server ~]#
```

你检查状态变量Aborted\_connects就会发现状态变量Aborted\_connects变为2了。

```
mysql> show status like 'Abort%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Aborted_clients | 0     |
| Aborted_connects | 2     |
+-----+-----+
2 rows in set (0.00 sec)
```

20

3: A connection packet does not contain the

这个比较容易构造，可以对MySQL的端口进行端口测试（ping 端口），因为psping的包不包含正确的信息（right information），测试之前，先将状态变量清空。

```
mysql> flush status;
Query OK, 0 rows affected (0.00 sec)
mysql> show status like 'abort%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Aborted_clients | 0     |
| Aborted_connects | 0     |
+-----+-----+
2 rows in set (0.00 sec)
```

在客户端对MySQL服务所在的主机进行端口连通性验证（psping）

```
C:\Users>psping 10.20.57.24:3306

PsPing v2.10 - PsPing - ping, latency, bandwidth measurement
Copyright (C) 2012-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

TCP connect to 10.20.57.24:3306:
5 iterations (warmup 1) ping test:
Connecting to 10.20.57.24:3306 (warmup): from 192.168.125.44:57696: 12.56ms
Connecting to 10.20.57.24:3306: from 192.168.125.44:57697: 12.64ms
Connecting to 10.20.57.24:3306: from 192.168.125.44:57698: 16.32ms
Connecting to 10.20.57.24:3306: from 192.168.125.44:57699: 12.84ms
Connecting to 10.20.57.24:3306: from 192.168.125.44:57701: 12.19ms

TCP connect statistics for 10.20.57.24:3306:
Sent = 4, Received = 4, Lost = 0 (0% loss),
Minimum = 12.19ms, Maximum = 16.32ms, Average = 13.50ms
```

如上所示，psping测试后，Aborted\_connects变成了5，如果继续进行psping测试，那么这个状态变量就会继续增长。

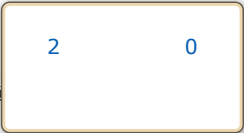
```
mysql> show status like 'abort%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Aborted_clients | 0     |
| Aborted_connects | 0     |
+-----+-----+
2 rows in set (0.00 sec)

mysql> show status like 'abort%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Aborted_clients | 0     |
| Aborted_connects | 5     |
+-----+-----+
2 rows in set (0.00 sec)
```

另外，如果超过max\_connect\_error的限制后，某一个客户端持续访问MySQL，这个是否会引起状态变量Aborted\_connects变化呢，实验测试的答案是不会。有兴趣的可以验证一下，很奇怪，网上有不少文章都说如果连接数满了，也会导致Aborted\_connects状态变量增加，实际上这个是不会引起状态变量Aborted\_connects变化的。

4、超过连接时间限制，主要是这个参数connect\_timeout控制（mysql默认是10s，基本上，除非网络环境极端不好，一般不会超时。）

首先在一台MySQL数据库服务器上执行下面命令，我命令模拟构造出复杂环境下的网络传输延时案例，如



```
# tc qdisc add dev eth0 root netem delay 11000ms
```

在另外一台MySQL服务器ping这台MySQL服务器，如下所示，你会看到网络时延为11秒

```
# ping 10.20.57.24
PING 10.20.57.24 (10.20.57.24) 56(84) bytes of data.
64 bytes from 10.20.57.24: icmp_seq=1 ttl=61 time=11001 ms
64 bytes from 10.20.57.24: icmp_seq=2 ttl=61 time=11001 ms
64 bytes from 10.20.57.24: icmp_seq=3 ttl=61 time=11001 ms
64 bytes from 10.20.57.24: icmp_seq=4 ttl=61 time=11001 ms
64 bytes from 10.20.57.24: icmp_seq=5 ttl=61 time=11001 ms
```

此时访问MySQL数据库，由于网络时延为11秒，超出了系统变量connect\_timeout的10秒，就会出现下面错误，此时状态变量Aborted\_connects的值变化！

```
# mysql -h 10.20.57.24 -u test -p
Enter password:
ERROR 2013 (HY000): Lost connection to MySQL server at 'reading
authorization packet', system error: 0
```

那么如何区分状态变量Aborted Connect是那个引起的呢？单从状态变量本身是无法区分的，但是可以结合performance\_schema.host\_cache来稍微做判别、甄别。

COUNT\_NAMEINFO\_PERMANENT\_ERRORS IP到主机名称DNS解析期间的永久性错误数。

COUNT\_AUTHENTICATION\_ERRORS 验证失败导致的错误数量  
SUM\_CONNECT\_ERRORS: 被视为“阻塞”的连接错误的数量（根据max\_connect\_errors系统变量进行评估）。只有协议握手错误才会被计数，只有通过验证（HOST\_VALIDATED = YES）的主机才会被计数

#### 1、客户端没有权限但是尝试访问MySQL数据库。

每次都会引起COUNT\_AUTHENTICATION\_ERRORS增1，第一次会引起COUNT\_NAMEINFO\_PERMANENT\_ERRORS也增1

#### 2、客户端输入的密码有误

每次都会引起COUNT\_AUTHENTICATION\_ERRORS增1，第一次会引起COUNT\_NAMEINFO\_PERMANENT\_ERRORS也增1

其实对于与1和2，两者无法判别，最简单有效的将系统变量log\_warnings设置为2，然后分析、查看错误日志信息：

```
mysql> set global log_warnings=2;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

那么此时1和2都会记录到错误日志里面去，然后你可以通过分析错误日志，结合状态变量Aborted Connect来分析，如下测试案例所示：

```
2018-06-20 22:44:16 18026 [Warning] IP address '192.168.xxx.xxx'
could not be resolved: Name or service not known
2018-06-20 22:44:16 18026 [Warning] Access denied for user
'kkkk'@'192.168.xxx.xxx' (using password: YES)
2018-06-20 22:45:18 18026 [Warning] Access denied for user
'test'@'192.168.xxx.xxx' (using password: YES)
```

2 0

#### 3、A connection packet does not contain information



每次引起COUNT\_HANDSHAKE\_ERRORS增1,  
每次引起SUM\_CONNECT\_ERRORS增1

```
C:\Users>psping 10.20.57.24:3306

PsPing v2.10 - PsPing - ping, latency, bandwidth
measurement utility

Copyright (C) 2012-2016 Mark Russinovich

Sysinternals - www.sysinternals.com

TCP connect to 10.20.57.24:3306:

5 iterations (warmup 1) ping test:

Connecting to 10.20.57.24:3306 (warmup): from
192.168.103.34:55327: 1.93ms

Connecting to 10.20.57.24:3306: from
192.168.103.34:55328: 10.08ms

Connecting to 10.20.57.24:3306: from
192.168.103.34:55329: 3.35ms

Connecting to 10.20.57.24:3306: from
192.168.103.34:55330: 3.71ms

Connecting to 10.20.57.24:3306: from
192.168.103.34:55331: 2.32ms

TCP connect statistics for 10.20.57.24:3306:

    Sent = 4, Received = 4, Lost = 0 (0% loss),

    Minimum = 2.32ms, Maximum = 10.08ms, Average =
4.87ms
```

打赏

```
mysql> select * from performance_schema.host_cache\G;
***** 1. row *****
                IP: 192.168.103.34
                HOST: NULL
                HOST VALIDATED: YES
                SUM_CONNECT_ERRORS: 5
                COUNT_HOST_BLOCKED_ERRORS: 0
                COUNT_NAMEINFO_TRANSIENT_ERRORS: 0
                COUNT_NAMEINFO_PERMANENT_ERRORS: 0
                COUNT_FORMAT_ERRORS: 0
                COUNT_ADDRINFO_TRANSIENT_ERRORS: 0
                COUNT_ADDRINFO_PERMANENT_ERRORS: 0
                COUNT_FCRDNS_ERRORS: 1
                COUNT_HOST_ACL_ERRORS: 0
                COUNT_NO_AUTH_PLUGIN_ERRORS: 0
                COUNT_AUTH_PLUGIN_ERRORS: 0
                COUNT_HANDSHAKE_ERRORS: 5
                COUNT_PROXY_USER_ERRORS: 0
                COUNT_PROXY_USER_ACL_ERRORS: 0
                COUNT_AUTHENTICATION_ERRORS: 0
                COUNT_SSL_ERRORS: 0
                COUNT_MAX_USER_CONNECTIONS_ERRORS: 0
                COUNT_MAX_USER_CONNECTIONS_PER_HOUR_ERRORS: 0
                COUNT_DEFAULT_DATABASE_ERRORS: 0
                COUNT_INIT_CONNECT_ERRORS: 0
                COUNT_LOCAL_ERRORS: 0
                COUNT_UNKNOWN_ERRORS: 0
                FIRST_SEEN: 2018-06-20 12:24:41
                LAST_SEEN: 2018-06-20 12:24:45
                FIRST_ERROR_SEEN: 2018-06-20 12:24:41
                LAST_ERROR_SEEN: 2018-06-20 12:24:45
```

#### 4、超过连接时间限制

如果是超时引起，那么就会出现下面状况：

每次引起SUM\_CONNECT\_ERRORS增1，

2 0

每次引起COUNT\_HANDSHAKE\_ERRORS增1

第一次会引起COUNT\_NAMEINFO\_PERMANENT\_ERRORS增1

注意：3与4不会写入错误日志，3与4的区别可以通过COUNT\_NAMEINFO\_PERMANENT\_ERRORS的值来区别。

```
mysql> select * from performance_schema.host_cache\G;
***** 1. row *****
IP: 192.168.7.
HOST: NULL
HOST_VALIDATED: YES
SUM_CONNECT_ERRORS: 2
COUNT_HOST_BLOCKED_ERRORS: 0
COUNT_NAMEINFO_TRANSIENT_ERRORS: 0
COUNT_NAMEINFO_PERMANENT_ERRORS: 1
COUNT_FORMAT_ERRORS: 0
COUNT_ADDRINFO_TRANSIENT_ERRORS: 0
COUNT_ADDRINFO_PERMANENT_ERRORS: 0
COUNT_FCRDNS_ERRORS: 0
COUNT_HOST_ACL_ERRORS: 0
COUNT_NO_AUTH_PLUGIN_ERRORS: 0
COUNT_AUTH_PLUGIN_ERRORS: 0
COUNT_HANDSHAKE_ERRORS: 2
COUNT_PROXY_USER_ERRORS: 0
COUNT_PROXY_USER_ACL_ERRORS: 0
COUNT_AUTHENTICATION_ERRORS: 0
COUNT_SSL_ERRORS: 0
COUNT_MAX_USER_CONNECTIONS_ERRORS: 0
COUNT_MAX_USER_CONNECTIONS_PER_HOUR_ERRORS: 0
COUNT_DEFAULT_DATABASE_ERRORS: 0
COUNT_INIT_CONNECT_ERRORS: 0
COUNT_LOCAL_ERRORS: 0
COUNT_UNKNOWN_ERRORS: 0
```

潇湘隐者

打赏

下面我们来实验测试一下状态变量Aborted Clients的变化因素，

1、客户端程序在退出之前未调用mysql\_close（）正确关闭MySQL连接。

在实验前，使用flush status清理一下状态变量

```
mysql> flush status;
Query OK, 0 rows affected (0.00 sec)

mysql> show status like 'Abort%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Aborted_clients | 0 |
| Aborted_connects | 0 |
+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

写一个简单的Python测试脚本python\_mysql.py，如下所示，将关闭数据库连接的地方dbcon.close注释掉，

```
import mysql.connector

try:
    dbcon=mysql.connector.connect(
        host='127.0.0.1',
        user='root' ,
        passwd='xxxxxxx',
        database='information_schema'
    )
```

2 0



```

cursor= dbcon.cursor()
sql_tex='select count(*) from MyDB.test'
cursor.execute(sql_tex)
dtlist= cursor.fetchall()
print dtlist
except mysql.connector.Error as e:

    print('operation the sql fail!{0}'.format(e))

finally:
    cursor.close;
    # dbcon.close;

```

然后执行一下脚本，检查状态变量Aborted\_clients，然后发现状态变量Aborted\_clients的值增1了。

```

[root@DB-Server kerry]# python python_mysql.py
[(99999,)]

```

```

mysql> flush status;
Query OK, 0 rows affected (0.00 sec)

mysql> show status like 'Abort%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Aborted_clients | 0 |
| Aborted_connects | 0 |
+-----+-----+
2 rows in set (0.00 sec)

mysql> show status like 'Abort%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Aborted_clients | 1 |
| Aborted_connects | 0 |
+-----+-----+
2 rows in set (0.00 sec)

mysql>

```

打赏

2、客户端休眠的时间超过了系统变量wait\_timeout和interactive\_timeout的值，导致连接被MySQL进程终止

```

mysql> show global variables
like 'interactive_timeout';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| interactive_timeout | 28800 |
+-----+-----+
1 row in set (0.00 sec)

mysql> show global variables like 'wait_timeout';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wait_timeout | 28800 |
+-----+-----+
1 row in set (0.00 sec)

mysql>

```

将全局系统变量interactive\_timeout 和wait\_timeout设置为4秒

```

mysql> set global interactive_timeout=4;
Query OK, 0 rows affected (0.00 sec)

mysql> set global wait_timeout=4;
Query OK, 0 rows affected (0.00 sec)

mysql> show status like 'Abort%';
+-----+-----+

```

2 0

Variable_name	Value
Aborted_clients	0
Aborted_connects	0

2 rows in set (0.00 sec)

然后在客户端连接到MySQL数据库，不做任何操作，过来4秒后，你去操作就会出现错误"ERROR 2013 (HY000): Lost connection to MySQL server during query"

```
# mysql -h 10.20.57.24 -u test -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ;
or \g.
Your MySQL connection id is 43
Server version: 5.6.20-enterprise-commercial-advanced-log MySQL Enterprise Server - Advanced Edition (Commercial)

Copyright (c) 2000, 2018, 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> select current_user();
ERROR 2013 (HY000): Lost connection to MySQL server during query
mysql>
```

打赏

在MySQL服务器你就会看到状态变量Aborted\_clients变为1了。

```
mysql> show status like 'Abort%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Aborted_clients | 1     |
| Aborted_connects | 0     |
+-----+-----+
2 rows in set (0.00 sec)
```

还有其他一些原因(客户端异常中断或查询超出了max\_allowed\_packet值)由于不方便构造，在此略过。另外，其实我们还可以通过tcpdump抓包工具来追踪分析。下面举个例子（这里

简单介绍一下tcpdump, 后续文章再做展开分析）

在MySQL服务器使用tcpdump抓包

```
[root@DB-Server ~]# tcpdump -i eth0 port 3306 -s 1500 -w tcpdump.log
```

然后在另外一台MySQL服务器，使用不存在的账号或数据库

```
# mysql -h 10.20.57.24 -u kkk -p
```

2	0
---	---

```
Enter password:
ERROR 1045 (28000): Access denied for user 'kkk'@'192.168.7.208'
(using password: YES)
# mysql -h 10.20.57.24 -u test -p
Enter password:
ERROR 1045 (28000): Access denied for user 'test'@'192.168.7.208'
(using password: YES)
[root@GETLNX28 ~]#
```

执行完命令后，你可以使用CTRL + C结束抓包分析，然后查看分析。如下截图所示：

```
[root@DB-Server ~]# tcpdump -i eth0 port 3306 -s 1500 -w
tcpdump.log
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture
size 1500 bytes
28 packets captured
28 packets received by filter
0 packets dropped by kernel
[root@DB-Server ~]# strings tcpdump.log
```

```
_client_name
libmysql
_pid
18921
_client_version
5.7.21 _platform
x86_64
program_name
mysqlW'+[
4I2@
A9Rs(
WgcW'+[C>
A9Rs(
WgcK
#28000Access denied for user 'kkk'@'192.168.7.208' (using password: YES)W'+[h>
4I4@
WgcW'+[0A
4@/@
W'+[nA
4I5@
WgdW'+[
4@0@
Z'+[p8
Z'+[
Z'+[
```

```
mysql_native_password
Linux
_client_name
libmysql
_pid
18922
_client_version
5.7.21 _platform
x86_64
program_name
mysqlZ'+[
Z'+[NA
#28000Access denied for user 'test'@'192.168.7.208' (using password: YES)Z'+[
Z'+[[]D
<Z'+[
Z'+[
```

## 参考资料：

<https://dev.mysql.com/doc/refman/8.0/en/communication-errors.html>

<http://www.olivierdoucet.info/blog/2012/05/08/customer-case-finding-cause-of-max-user-connections/>

作者：潇湘隐者

出处：<http://www.cnblogs.com/kerrycode/>

2

0

打赏