

停从库,为啥主库报错[ERROR]mysqld:Got an error reading communication packe

原创 大大刺猬 大大刺猬 2025年04月15日 18:10 上海

现象

做自动化重启的时候, 会检查相关日志信息, 发现停止从库时, 主库的日志里面有报错如下(5.7):

```
2025-04-15T06:48:37.968266Z 0 [ERROR] /soft/mysql_3308/mysqlbase/mysql/bin/mysqld: Got an error reading communication packets
```



对应8.0环境信息是:

```
2025-04-15T09:05:10.040068Z 0 [ERROR] [MY-013129] [Server] A message intended for a client cannot be sent there as no client-session is attached. Therefore, we're sending the information to the error-log instead
: MY-001158 - Got an error reading communication packets
```

报错代码: MY-001158

问: 这个信息有影响吗?

答: 没得

分析 & 验证

通常 Got an error reading communication packets 信息我们是不需要分析的, 就是连接断开了而已, 但是这里被分类为ERROR了. 也就需要瞅瞅为啥了.

先看报错内容

先看报错内容: Got an error reading communication packets, 该信息为连接断开了, 比如我们把某个连接的进程给kill了(是kill的客户端连接进程,而非服务端连接) 比如:

```

15:10:06 [root@ddcw21 ~]#mysql -h127.0.0.1 -P3314 -p123456 -e 'select sleep(100000)' &
[1] 18103
15:10:08 [root@ddcw21 ~]#mysql: [Warning] Using a password on the command line interface can be insecure.
15:10:09 [root@ddcw21 ~]#kill `pidof mysql`
15:10:13 [root@ddcw21 ~]#
[1]+  Terminated                  mysql -h127.0.0.1 -P3314 -p123456 -e 'select sleep(100000)'
15:10:22 [root@ddcw21 ~]#
15:10:24 [root@ddcw21 ~]#tail -1 /data/mysql_3314/mysqllog/dblogs/mysql3314.err
2025-04-15T07:10:18.657541Z 87 [Note] [MY-010914] [Server] Aborted connection 87 to db: 'unconnected' user: 'root' host: 'loca
lhost' (Got an error reading communication packets).
15:10:29 [root@ddcw21 ~]#

```

但这个是属于 [Note] 啊, 和我们看到的 [ERROR] 不一样. 而且还多了一部分连接信息...

我们还可以使用telnet来模拟, 这样就没有连接信息了. -_-

```

15:13:40 [root@ddcw21 ~]#nohup telnet 127.0.0.1 3314 2>&1 &
[2] 18309
15:13:56 [root@ddcw21 ~]#nohup: ignoring input and appending output to 'nohup.out'

[2]-  Exit 1                  nohup telnet 127.0.0.1 3314 2>&1
15:13:57 [root@ddcw21 ~]#tail -1 /data/mysql_3314/mysqllog/dblogs/mysql3314.err
2025-04-15T07:13:56.599911Z 89 [Note] [MY-010914] [Server] Got an error reading communication packets
15:14:05 [root@ddcw21 ~]#

```

但始终是NOTE级别.

能看到错误码MY-010914 是因为是8.0环境, 5.7是连报错码都没有的. 而且和我们文章开头的报错码也不一样.

看看是谁输出的这个信息

那我们看看是哪个对象输出的这部分信息, 还是老方法, 在源码里面使用 grep 找. 先看对应的报错名字:

```
grep -r 'Got an error reading communication packets' sql/share/errmsg-utf8.txt -C 10
```

```

15:20:55 [root@ddcw21 mysql-server-5.7]#grep -r 'Got an error reading communication packets' sql/share/errmsg-utf8.txt -C 10
rum "Nu s-a putut decompresa pachetul de comunicatie (communication packet)"
rus "Невозможно распаковать пакет, полученный через коммуникационный протокол"
serbian "Ne mogu da dekompresujem mrežne pakete"
spa "No puedo descomprimir paquetes de comunicación"
swe "Kunde inte packa up kommunikationspaketet"
ukr "Не змогу декompresувати комунікаційний пакет"
ER_NET_READ_ERROR 08501
cze "Zjistil na chybu při čtení komunikačního packetu"
dan "Fik fejlmeddelelse ved læsning af kommunikations-pakker (communication packets)"
nla "Fout bij het lezen van communicatiepakketten"
eng "Got an error reading communication packets"
est "Viga andmepaketi lugemisel"
fre "Erreur de lecture des paquets reçus"
ger "Fehler beim Lesen eines Kommunikationspakets"
hun "Hiba a kommunikációs adatcsomagok olvasása során"
ita "Rilevato un errore ricevendo i pacchetti di comunicazione"
jpn "パケットの受信でエラーが発生しました。"
kor "통신 패킷을 읽는 과정에서 통신 오류가 발생했습니다."
por "Obteve um erro na leitura de pacotes de comunicação"
rum "Eroare obținută citind pachetele de comunicație (communication packets)"
rus "Получена ошибка в процессе получения пакета через коммуникационный протокол"

```

发现是叫 ER_NET_READ_ERROR , 还挺顾名思义的.

再看看是哪些地方有匹配的:

```
grep -r ER_NET_READ_ERROR
```

```
15:22:27 [root@ddcw21 mysql-server-5.7]#grep -r ER_NET_READ_ERROR
plugin/semisync/semisync_master_ack_receiver.cc:         else if (net.last_errno == ER_NET_READ_ERROR)
sql/log_event.cc:   case ER_NET_READ_ERROR:
sql/net_serv.cc:       net->last_errno= ER_NET_READ_ERROR;
sql/share/errmsg-utf8.txt:ER_NET_READ_ERROR FROM PIPE 08501
sql/share/errmsg-utf8.txt:ER_NET_READ_ERROR 08501
storage/ndb/nodejs/Adapter/impl/common/MysqlErrToSQLStateMap.js:  "ER_NET_READ_ERROR FROM PIPE": 1154,
storage/ndb/nodejs/Adapter/impl/common/MysqlErrToSQLStateMap.js:  "ER_NET_READ_ERROR": 1158,
15:22:31 [root@ddcw21 mysql-server-5.7]#
```

看起来最像的就是半同步那的问题了.

```
do {
    net_clear(&net, 0);

    len= my_net_read(&net);
    if (likely(len != packet_error))
        repl_semisync.reportReplyPacket(slave_obj.server_id,
                                         net.read_pos, len);
    else if (net.last_errno == ER_NET_READ_ERROR)
        listener.clear_socket_info(i);
} while (net.vio->has_data(net.vio) && m_status == ST_UP);
```

具体的代码逻辑和怎么输出的内容我们就不看了.

日志输出并不在这里. 而是sql_print_error, 具体的可以看文末堆栈信息.

验证

我们直接来验证一下, 即存在半同步和不存在半同步时停止复制进程, error日志里面是否有输出.

存在半同步的情况

```
-- install plugin rpl_semi_sync_master soname 'semisync_master.so';
-- install plugin rpl_semi_sync_slave soname 'semisync_slave.so';
-- set global rpl_semi_sync_master_enabled = 1;
-- set global rpl_semi_sync_slave_enabled = 1;

-- 重启复制进程, 半同步才生效.
stop slave;
start slave;

-- 停止从库复制进程, 然后查看主库日志
stop slave;
```

```

2025-04-15T08:09:22.088463Z 0 [ERROR] /soft/mysql_3308/mysqlbase/mysql/bin/mysqld: Got an error reading communication packets
2025-04-15T08:09:22.102779Z 28 [Note] While initializing dump thread for slave with UUID <b68e2434-cd30-11ec-b536-000c2980c11e>, found a zombie dump thread with the same UUID. Master is killing the zombie dump thread(26).
2025-04-15T08:09:22.102860Z 28 [Note] Start binlog_dump to master_thread_id(28) slave_server(866003314), pos(m3308.001089, 154)
2025-04-15T08:09:22.102873Z 28 [Note] Start semi-sync binlog_dump to slave (server_id: 866003314), pos(m3308.001089, 154)
2025-04-15T08:09:22.103045Z 26 [Note] Stop semi-sync binlog_dump to slave (server_id: 866003314)
2025-04-15T08:09:23.090511Z 0 [ERROR] /soft/mysql_3308/mysqlbase/mysql/bin/mysqld: Got an error reading communication packets
2025-04-15T08:10:22.103595Z 28 [Note] Stop semi-sync binlog_dump to slave (server_id: 866003314)
2025-04-15T08:10:22.103728Z 28 [Note] Aborted connection 28 to db: 'unconnected' user: 'root' host: 'localhost' (failed on flush_net())

```

确实复现了。

再看看没得半同步的情况

```

set global rpl_semi_sync_slave_enabled = 0;
stop slave;
start slave;

```

-- 停止从库复制进程,然后查看主库日志

```
stop slave;
```

```

2025-04-15T08:12:33.941638Z 31 [Note] Start asynchronous binlog_dump to slave (server_id: 866003314), pos(m3308.001089, 154)
2025-04-15T08:12:33.941695Z 30 [Note] Stop semi-sync binlog_dump to slave (server_id: 866003314)

```

主库日志没有任何记录,就跟一个连接正常断开似的。

证明这部分ERROR信息确实是半同步插件"输出的".

处理方法

1. 忽略(假装没看到, 毕竟也不算啥问题.)
2. 匹配日志的时候去掉 Got an error reading communication packets 之类的信息. 这部分信息用处也不大.

去掉半同步这种操作是不建议的....

机智的小伙伴会发现: 既然是半同步输出的这部分信息, 那么如果先停止半同步, 然后停止复制进程, 是否就不会产生这个报错呢? 理论上是, 但不停止复制线程, 咋个停止半同步呢...

rpl_semi_sync_slave_enabled启动复制进程时才会看.

其它

如果有兴趣的, 还可以看下 ERROR 日志堆栈信息 (我这是 5.7 环境的, break sql_print_error), 如下:

```
(gdb) bt
#0  sql_print_error (format=0x14e82c4 "%s: %s") at /var/lib/pb2/sb_1-6473437-1647886122.65/mysql-5.7.38/sql/log.cc:2147
#1  0x0000000007eeb64 in my_message_sql (error=1158, str=0x7f6328836310 "Got an error reading communication packets",
MyFlags=<optimized out>) at /var/lib/pb2/sb_1-6473437-1647886122.65/mysql-5.7.38/sql/mysqld.cc:2293
#2  0x0000000000f78260 in my_error (nr=1158, MyFlags=0)
at /var/lib/pb2/sb_1-6473437-1647886122.65/mysql-5.7.38/mysys/my_error.c:222
#3  0x0000000000c93b6c in net_read_raw_loop (net=0x7f6328836740, count=4)
at /var/lib/pb2/sb_1-6473437-1647886122.65/mysql-5.7.38/sql/net_serv.cc:742
#4  0x0000000000c94508 in net_read_packet_header (net=0x7f6328836740)
at /var/lib/pb2/sb_1-6473437-1647886122.65/mysql-5.7.38/sql/net_serv.cc:794
#5  net_read_packet (net=0x7f6328836740, complen=0x7f63288366b8)
at /var/lib/pb2/sb_1-6473437-1647886122.65/mysql-5.7.38/sql/net_serv.cc:854
#6  0x0000000000c9468c in my_net_read (net=0x7f6328836740)
at /var/lib/pb2/sb_1-6473437-1647886122.65/mysql-5.7.38/sql/net_serv.cc:931
#7  0x00007f634bf06377 in Ack_receiver::run (this=0x7f634c10aac0 <ack_receiver>)
at /var/lib/pb2/sb_1-6473437-1647886122.65/mysql-5.7.38/plugin/semisync/semisync_master_ack_receiver.cc:279
#8  0x00007f634bf06699 in ack_receive_handler (arg=0x7f634c10aac0 <ack_receiver>)
at /var/lib/pb2/sb_1-6473437-1647886122.65/mysql-5.7.38/plugin/semisync/semisync_master_ack_receiver.cc:42
#9  0x0000000000143c2a4 in pfs_spawn_thread (arg=0x7f631801ad30)
at /var/lib/pb2/sb_1-6473437-1647886122.65/mysql-5.7.38/storage/perfschema/pfs.cc:2197
#10 0x00007f63a335fea5 in start_thread () from /lib64/libpthread.so.0
#11 0x00007f63a1e1796d in clone () from /lib64/libc.so.6
```

从堆栈信息也是能看出来主库ack_receiver的时候(my_net_read没收到信息), 然后就打印error信息了。

我们这里还能看到报错码1158, 但是error log里面却没得这部分了... -_- 有兴趣的自己去研究吧。