## 1.1 MySQL主主同步配置

1). 配置 /etc/my.cnf masterA配置文件

server-id=132

log-bin=mysql-bin

auto-increment-increment=2

auto-increment-offset=1

log-slave-updates

2). 重启mysql

/etc/init.d/mysqld restart

3). 配置 /etc/my.cnf masterB配置文件

server-id=133

log-bin=mysql-bin

auto-increment-increment=2

auto-increment-offset=2服务器都设定了一次自动增长值2，所以它们的起点必须得不同，这样才能避  
免两台服务器数据同步时出现主键冲突

log-slave-updates

4). 重启mysql

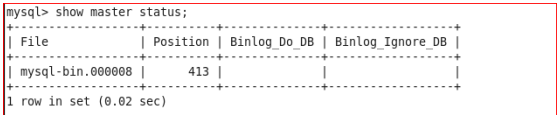
/etc/init.d/mysqld restart

5). 将master1设为master2的主服务器

在master1主机上创建授权账户，允许在master2主机上连接

grant replication slave on \*.\* to 'rep'@'master2IP' identified by 'password'；

查看master1的当前binlog状态信息

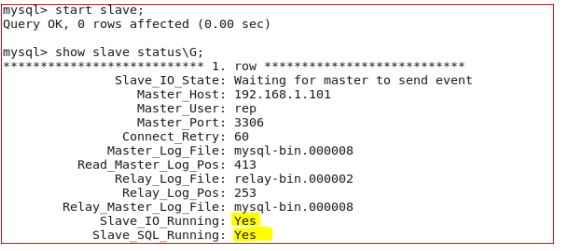


在master2上将master1设为自已的主服务器并开启slave功能。

change master to master\_host = 'master1Ip',master\_user='rep',master\_password='password',master\_log\_file='mysql-bin.000008',master\_log\_pos=413;

start slave;

查看从的状态，以下两个值必须为yes,代表从服务器能正常连接主服务器

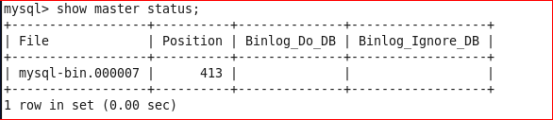


6). 将master2设为master1的主服务器

在master2主机上创建授权账户，允许在master1主机上连接

grant replication slave on \*.\* to 'rep'@'master1IP' identified by 'password'；

查看master2的当前binlog状态信息



在master1上将master2设为自已的主服务器并开启slave功能。

change master to master\_host = 'master2Ip',master\_user='rep',master\_password='password',master\_log\_file='mysql-bin.000008',master\_log\_pos=413;

查看从的状态，以下两个值必须为yes,代表从服务器能正常连接主服务器

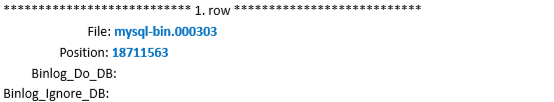
Slave\_IO\_Running:Yes

Slave\_SQL\_Running:Yes

## 1.2 MySQL主从复制状态检查及异常处理

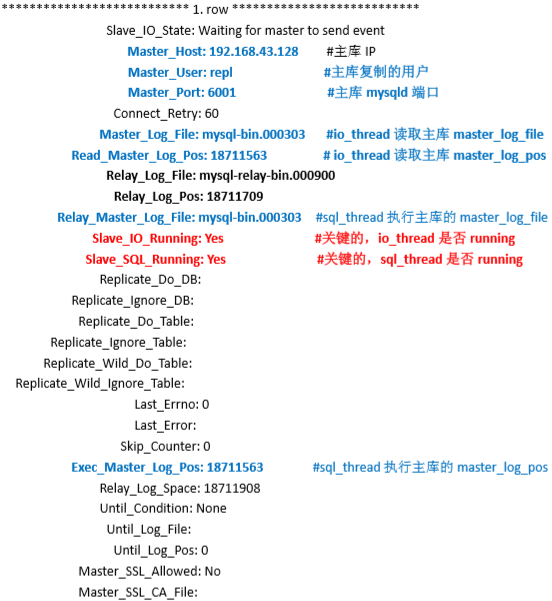
1)、主从复制状态检查

show master status\G



在从库上主要是使用以下命令查看从库与主库的同步状态：

show slave status\G



2)、IO\_thread 异常

IO\_thread 异常，状态往往是 Slave\_IO\_Running: Connecting 或 NO。

IO\_thread 是向 Master 发送请求读取 master binlog，如果处于 Connecting 状态，说明无

法正确地与 Master 进行连接，可能的原因有：

a、网络不通（是否打开防火墙）

b、复制用户的密码不对

c、指定的 master\_port 端口不对

d、master 上的 mysql-bin.xxxxxx 被误删

e、主库磁盘空间满了

通过 show slave status\G 可以看到相关错误信息，例如：

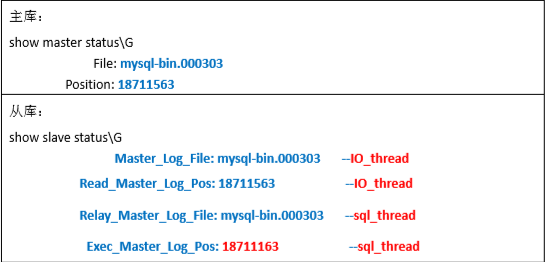


3)、sql\_thread 异常

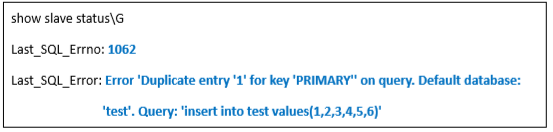
sql\_thread 发生异常，状态就会变为 Slave\_SQL\_Running: NO。

sql\_thread 发生异常的情况非常多，发生异常后，需要通过以下方法排查和解决：

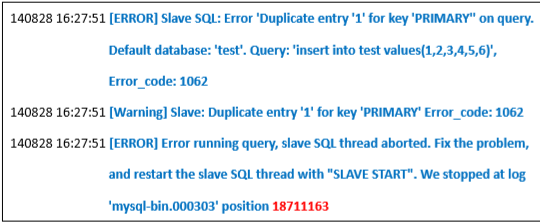
1. 对比主库和从库的二进制日志的情况：



1. 通过 show slave status\G 查看错误信息：



1. 通过错误日志查看错误信息：



根据这些报错信息，往往就能够定位到发生异常的原因。如果我们了解产生异常的具体

事件，而且能够掌控，可以通过设置 sql\_slave\_skip\_counter 参数来跳过当前错误。

set global sql\_slave\_skip\_counter=1;

或者使用 slave\_skip\_errors 参数（read only variable），指定跳过某种类型的错误：

参数文件中设置：

slave\_skip\_errors=1062 #跳过 1062 错误

遇到错误时，不要一通百度后，然后根据看起来很类似的操作直接来进行操作。因为网

上大部分解决 sql\_thread 异常的方法是：

a、直接 set global sql\_slave\_skip\_counter=n; （n 设置很大的值，即：跳过所有错误），

b、设置 slave\_skip\_errors=all; 跳过所有类型的错误

c、直接查看主库的 binlog，然后在从库上直接执行 change master to。

这些方法都会导致主从数据不一致。

如果发现从库与主库差异太大，无法通过手动操作或数据修改重新建立同步。可以参考

上述"MySQL 主从复制搭建" 重新搭建从库。

## 1.3 MySQL常见故障及处理方法

1).【错误过程】：MySQL 主从中断,1236报错

【造成原因】：

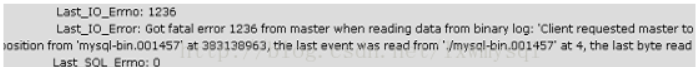
sync\_binlog=0时 ，master binlog文件的flush log buffer（这个buffer是由于binlog文件的os buffer)  到disk是依赖于OS本身的，但Slave IO 线程在读取master dump 线程的位置，

一般是直接读取log buffer的，这个位置，可能远远大于binlog文件实际大小。所以当主机宕机后，binlog buffer未刷盘，当Master主机再次启动后，此时从库的binlog pos已经比实

际的binlog文件大小还大了。

【解决办法】：

如果innodb\_flush\_log\_at\_trx\_commint=1的话，这一般对主从不会有影响的，直接做change master to到当下一个binlog。如果是其他值2或0，可能有点小差别。



2).【错误过程】：MySQL 主从中断,1032报错

【造成原因】：

由于主上的binlog-format是mixed，这样在主从同步修改数据的时候会使用rowed模式，直接修改对应数据，如果此时主从数据不一致的话，就会出现找不到记录的1032的错误。

【解决办法】：

保持主从数据一致性



3).【错误过程】：MySQL 出现大量Too many connections

【造成原因】：

查询发现这算属MySQL的一个bug，不管连接是通过hosts还是ip的方式，MySQL都会对DNS做反查，IP到DNS，由于反查的接续速度过慢（不管

是不是isp提供的dns服务器的问题或者其他原因），大量的查询就难以应付，线程不够用就使劲增加线程，但是却得不到释放，所以MySQL会“假死”。

【解决办法】：

在配置文件加上以下参数即可

--skip-name-resolve，不要解析正检查客户端连接的主机名。只使用IP号。如果你使用该项， 授权表中的所有Host列值必须为IP号或localhost。



4)、主从复制延迟

主从复制延迟，可能的原因有：

a、主从同步延迟与系统时间的关系，查看主从两台机器间系统时间差

b、主从同步延迟与压力、网络、机器性能的关系，查看从库的 io，cpu，mem 及网络 压力

c、主从同步延迟与 lock 锁的关系（myisam 表读时会堵塞写），尽量避免使用 myisam 表。一个实例里面尽量减少数据库的数量。

d、主从复制发生异常而中断，过很久之后才发现复制异常。可通过查看 master 与 slave 的

status 估算相差的日志。如果相差太大，则可以考虑重做从库。

## MySQL主从数据一致性检查

1).pt-table-checksum安装

安装工具的服务器需要能够连接mysql的主从

1.软件下载：

wget https://www.percona.com/downloads/percona-toolkit/3.0.3/binary/redhat/6/x86\_64/percona-toolkit-3.0.3-1.el6.x86\_64.rpm

2.安装该工具依赖的软件包：

yum install perl-IO-Socket-SSL perl-DBD-MySQL perl-Time-HiRes -y

3、软件安装

rpm -ivh percona-toolkit-3.0.3-1.el6.x86\_64.rpm

如果提示需要依赖，那么使用下列方式安装

yum localinstall percona-toolkit-3.0.3-1.el6.x86\_64.rpm

4、查看是否安装成功

rpm -qa | grep percona-toolkit

2).数据库添加check用户

在master和slave创建账号：

mysql>GRANT SELECT,PROCESS,SUPER,REPLICATION SLAVE,CREATE,DELETE,INSERT,UPDATE ON \*.\* TO 'USER'@'HOST' identified by 'PASSWORD';

mysql>flush privileges;

3).执行命令

#pt-table-checksum --nocheck-replication-filters --no-check-binlog-format --replicate=cnt.checksums --databases=cnt --tables=ldb\_alipay\_order h=Master\_HOST,u=root,p=123456,P=3306

--no-check-replication-filters 表示不需要检查 Master 配置里是否指定了 Filter。 默认会检查，如果配置了 Filter，如 replicate\_do\_db,replicate-wild-ignore-table,binlog\_ignore\_db 等，在从库checksum就与遇到表不存在而报错退出，所以官方默认是yes（--check-replication-filters）但我们实际在检测中时指定--databases=，所以就不存在这个问题，干脆不检测。

--no-check-binlog-format 不对binlog的格式进行检查

--replicate-check-only 只显示主从不一致部分,此参数不会生成新的checksums数据，只会根据checksums表已经有的数据来显示

--databases=，-d：要检查的数据库，逗号分隔。 --databases-regex 正则匹配要检测的数据库，--ignore-databases[-regex]忽略检查的库。Filter选项。

--tables=，-t：要检查的表，逗号分隔。如果要检查的表分布在不同的db中，可以用--tables=dbname1.table1,dbnamd2.table2的形式。

同理有--tables-regex，--ignore-tables，--ignore-tables-regex。--replicate指定的checksum表始终会被过滤。

结果：

[root@ws02 scripts]# pt-table-checksum --nocheck-replication-filters --no-check-binlog-format --replicate=cnt.checksums --databases="test" h=192.168.1.1,u=test,p="test",P=3308

TS ERRORS DIFFS ROWS CHUNKS SKIPPED TIME TABLE

03-30T13:31:48 0 0 607 1 0 0.208 test.tb\_key

03-30T13:31:48 0 0 253 1 0 0.220 test.tb\_player

03-30T13:31:49 0 0 6 1 0 0.204 test.tb\_con

03-30T13:31:49 0 0 6116 4 0 0.704 test.tb\_dev

03-30T13:31:49 0 0 5645 1 0 0.221 test.tb\_com

03-30T13:31:50 0 0 17 1 0 0.213 test.tb\_class

03-30T13:31:50 0 0 6 1 0 0.196 test.tb\_banj

说明：

TS ：完成检查的时间

ERRORS ：检查时候发生错误和警告的数量

DIFFS ：0表示一致，1表示不一致。当指定–no-replicate-check时，会一直为0，当指定–replicate-check-only会显示不同的信息

ROWS ：表的行数

CHUNKS ：被划分到表中的块的数目

SKIPPED ：由于错误或警告或过大，则跳过块的数目

TIME ：执行的时间

TABLE ：被检查的表名