<https://www.cnblogs.com/shengdimaya/p/6959021.html>

1、介绍

什么是半同步复制？我们知道在默认情况下，MySQL的复制是异步的，这意味着主服务器及其从服务器是独立的。异步复制可以提供最佳的性能，因为主服务器在将更新的数据写入它的二进制日志（Binlog）文件中后，无需等待验证更新数据是否已经复制到从服务器中，就可以自由处理其它进入的事务处理请求。但这也同时带来了很高的风险，如果在主服务器或从服务器端发生故障，会造成主从数据的不一致，甚至在恢复时造成数据丢失。

半同步复制是从MySQL5.5开始引入了一种半同步复制功能，该功能可以确保主服务器和访问链中至少一台从服务器之间的数据一致性和冗余。在这种配置结构中，一台主服务器和其许多从服务器都进行了配置，这样在复制拓扑中，至少有一台从服务器在父主服务器进行事务处理前，必须确认更新已经收到并写入了其中继日志（Relay Log）。当出现超时，源主服务器必须暂时切换到异步复制模式重新复制，直到至少有一台设置为半同步复制模式的从服务器及时收到信息。

mysql> INSTALL PLUGIN rpl\_semi\_sync\_master SONAME 'semisync\_master.so';
mysql> INSTALL PLUGIN rpl\_semi\_sync\_slave SONAME 'semisync\_slave.so';

SELECT PLUGIN\_NAME, PLUGIN\_STATUS FROM INFORMATION\_SCHEMA.PLUGINS WHERE PLUGIN\_NAME LIKE '%semi%';

**rpl\_semi\_sync\_master\_enabled** ：主库是否打开半同步复制

**rpl\_semi\_sync\_master\_timeout** ：毫秒为单位，当主库等待从库ACK的实践超过这个值，就会自动转化为异步复制

**rpl\_semi\_sync\_master\_trace\_level** ：master的trace 级别，分为四个（1,16,32,64），分别记录不同的信息，32能够输出更详细的网络延迟等信息，也是默认值

**rpl\_semi\_sync\_master\_wait\_for\_slave\_count** ：至少有N个slave接收到日志，一主多从的情况下只要有一个slave的ACK返回给了主库，就会进行commit

**rpl\_semi\_sync\_master\_wait\_no\_slave**：默认为ON，当半同步复制转换为异步复制后，如果从库的日志追赶上了主库，会自动转换为半同步复制，设置为OFF的话就不会再进行转换。

**rpl\_semi\_sync\_slave\_enabled** ：从库是否打开半同步复制功能

**rpl\_semi\_sync\_slave\_trace\_level** ：trace 级别

**rpl\_semi\_sync\_master\_wait\_point** ：这是MySQL5.7新增的功能，可以设置两个值AFTER\_SYNC 和AFTER\_COMMIT，AFTER\_COMMIT的模式下master将每个事务写入binlog ,传递到slave 刷新到磁盘(relay log)，同时主库提交事务。master等待slave 反馈收到relay log，只有收到ACK后master才将commit OK结果反馈给客户端。AFTER\_SYNC 情况下master 将每个事务写入binlog , 传递到slave 刷新到磁盘(relay log)。master等待slave 反馈接收到relay log的ack之后，再提交事务并且返回commit OK结果给客户端。 即使主库crash，所有在主库上已经提交的事务都能保证已经同步到slave的relay log中。我们推荐使用默认AFTER\_SYNC 的情况，这样可以提高性能，减少等待时间。