

MySQL基于GTID的主从复制优缺点及实践

参考：<http://cenalulu.github.io/mysql/mysql-5-6-gtid-basic/>

MySQL 5.6 的新特性之一，是加入了全局事务 ID (GTID) 来强化数据库的主备一致性，故障恢复，以及容错能力。

什么是GTID?

官方文档：<http://dev.mysql.com/doc/refman/5.6/en/replication-gtids.html>在这篇文档里，我们可以知道全局事务 ID 的官方定义是：GTID = source_id:transaction_id

MySQL 5.6 中，每一个 GTID 代表一个数据库事务。在上面的定义中，source_id 表示执行事务的主库 uuid (server_uuid)，transaction_id 是一个从 1 开始的自增计数，表示在这个主库上执行的第 n 个事务。MySQL 会保证事务与 GTID 之间的 1 : 1 映射。

原理：

从服务器连接到主服务器之后，把自己执行过的GTID(Executed_Gtid_Set)<SQL线程>、获取到的GTID(Retrieved_Gtid_Set)<IO线程>发给主服务器，主服务器把从服务器缺少的GTID及对应的transactions发过去补全即可。当主服务器挂掉的时候，找出同步最成功的那台从服务器，直接把它提升为主即可。

优点：(参考：<http://www.cnblogs.com/zhouljinyi/p/4717951.html>)

1. GTID能够保证每个MySQL实例事务的执行（不会重复执行同一个事务，并且会补全没有执行的事务）
2. 通过GTID可以极方便的进行复制结构上的故障转移，新主设置。

Server1(Master)崩溃，根据从show slave status获得Master_log_File/Read_Master_Log_Pos的值，Server2(Slave)已经跟上了主，Server3(Slave)没有跟上主。这时要是把Server2提升为主，Server3变成Server2的从。这时在Server3上执行change的时候需要做一些计算，这里就不做说明了，具体的说明见[高性能MySQL第10章](#)，相对来说是比较麻烦的。

这个问题在5.6的GTID出现后，就显得非常的简单。由于同一事务的GTID在所有节点上的值一致，那么根据Server3当前停止点的GTID就能定位到Server2上的GTID。甚至由于MASTER_AUTO_POSITION功能的出现，我们都不需要知道GTID的具体值，直接使用CHANGE MASTER TO MASTER_HOST='xxx'，MASTER_AUTO_POSITION命令就可以直接完成failover的工作。

3. GTID就是全局事务ID(global transaction identifier)，最初由google实现，官方MySQL在5.6才加入该功能。要是主从结构只有一台Master和一台Slave对于GTID来说就没有优势了，而对于2台主以上的结构优势异常明显，可以在数据不丢失的情况下切换新主。

缺点：(参考：<https://dev.mysql.com/doc/refman/5.6/en/replication-gtids-restrictions.html>)

1. 不支持非事务型引擎
2. 不支持 CREATE TABLE AS SELECT语句
3. 不支持创建和删除临时表
4. 不支持 sql_slave_skip_counter

预防：

为了防止某些语句导致基于GTID的复制失败，所有服务器都以以下模式开启：

--enforce-gtid-consistency

搭建：

主从环境的搭建和5.5没有什么区别，唯一需要注意的是：开启GTID需要启用这三个参数：

for MySQL 5.7

#GTID

gtid_mode = on (必选，开启GTID模式)

enforce_gtid_consistency = on (必选，保证GTID安全的参数)

log_slave_updates = on (可选，高可用切换，最好设置为on)

log_bin = on (可选，高可用切换，最好设置为on)

搭建步骤：(参考：<https://dev.mysql.com/doc/refman/5.6/en/replication-gtids-howto.html>)

1. Synchronize the servers :

on master and slave :

set @@global.read_only=on;

Then, allow the slave to catch up with the master.

2. Stop both servers :

mysqladmin -uusername -p shutdown

3. Restart both servers with GTIDs enabled :

on master :

```
mysqld_safe --gtid_mode=on --log-bin --log-slave-updates --enforce-gtid-consistency &  
on slave :  
mysqld_safe --gtid_mode=on --log-bin --log-slave-updates --enforce-gtid-consistency --skip-slave-start &  
4. Direct the slave to use the master :  
CHANGE MASTER TO  
  > MASTER_HOST = host,  
  > MASTER_PORT = port,  
  > MASTER_USER = user,  
  > MASTER_PASSWORD = password,  
  > MASTER_AUTO_POSITION = 1;  
START SLAVE;  
5. Disable read-only mode :  
set @@global.read_only=off;
```

参考 : http://blog.csdn.net/zhu_tianwei/article/details/53456943
<http://blog.csdn.net/winsonyuan/article/details/41646579>
<http://blog.csdn.net/celeste7777/article/details/48900101>