

之前给大家讲完了MySQL传统的主从复制搭建方式，其实一般大家在生产中都会采用半同步的复制模式，但是其实除了那种传统搭建方式之外，还有一种更加简便一些的搭建方式，就是GTID搭建方式，今天就给大家讲讲GTID的搭建方式。

首先在主库进行配置：

```
gtid_mode=on  
enforce_gtid_consistency=on  
log_bin=on  
server_id=单独设置一个  
binlog_format=row
```

接着在从库进行配置：

```
gtid_mode=on  
enforce_gtid_consistency=on  
log_slave_updates=1  
server_id=单独设置一个
```

接着按照之前讲解的步骤在主库创建好用于复制的账号之后，就可以跟之前一样进行操作了，比如在主库dump出来一份数据，在从库里导入这份数据，利用mysqldump备份工具做的导出，备份文件里会有SET @@GLOBAL.GTID_PURGED='*'一类的字样，可以照着执行一下就可以了。

接着其余步骤都是跟之前类似的，最后执行一下show master status，可以看到executed_gtid_set，里面记录的是执行过的GTID，接着执行一下SQL：select * from gtid_executed，可以查询到，对比一下，就会发现对应上了。

那么此时就说明开始GTID复制了。

其实大家会发现无论是GTID复制，还是传统复制，都不难，很简单，往往这就是比较典型的MySQL主从复制的搭建方式了，然后大家可以自行搜索一下MyCat中间件或者是Sharding-Sphere的官方文档，其实也都不难，大家照着文档做，整合到Java代码里去，就可以做出来基于主从复制的读写分离的效果了。

那些中间件都是支持读写分离模式的，可以仅仅往主库去写，从从库去读，这都没问题的。

如果落地到项目里，那么就完成了一个主从架构以及读写分离的架构了，此时按照我们之前所说的，如果说你的数据库之前对一个库的读写请求每秒总共是2000，此时读写分离后，也许就对主库每秒写TPS才几百，从库的读QPS是1000多。

那么万一你要是从库的读QPS越来越大，达到了每秒几千，此时你是不是会压力很大？没关系，这个时候你可以给主库做更多的从库，搭建从库，给他挂到主库上去，每次都在凌晨搞，先让系统停机，对外不使用，数据不更新。

接着对主库做个dump，导出数据，到从库导入数据，做一堆配置，然后让从库开始接着某个时间点开始继续从主库复制就可以了，一旦搭建完毕，就等于给主库挂了一个新的从库上去，此时继续放开系统的对外限制，继续使用就可以了，整个过程基本在1小时以内。

如果在凌晨比如2点停机1小时，基本对业务是没有影响的。

好，那么到此为止，主从复制这块就初步的算讲完了，下讲给大家介绍一下主从复制的延迟问题如何解决。

End