

mysql多源复制

5.7新特性多主一从架构的好处：

一、在从服务器进行数据汇总，如果我们的主服务器进行了分库分表的操作，为了实现后期的一些数据统计功能，往往需要把数据汇总在一起再统计。

二、如果我们想在从服务器时对主服务器的数据进行备份，在MySQL 5.7之前每一个主服务器都需要一个从服务器，这样很容易造成资源浪费，同时也加大了DBA的维护成本，但MySQL 5.7引入多源复制，可以把多个主服务器的数据同步到一个从服务器进行备份。

环境：

Master_1 : 192.168.10.128

Master_2 : 192.168.10.129

Slave : 192.168.10.130

1. 分别在Master_1, Master_2上导出需要同步的数据

在Master_1：

```
mysqldump -uroot -p123456 --master-data=2 --single-transaction --databases --add-drop-database master01 > master01.sql
```

在Master_2：

```
mysqldump -uroot -p123456 --master-data=2 --single-transaction --databases --add-drop-database master02 > master02.sql
```

分别把备份scp到Slave：

```
scp -P22 master01.sql 192.168.10.130:/data/service/mysql/
```

```
scp -P22 master02.sql 192.168.10.130:/data/service/mysql/
```

2. 在Master_1和Master_2上创建复制账号

在Master_1：

```
grant replication slave on *.* to 'repl'@'192.168.10.130' identified by '123456';
```

在Master_2：

```
grant replication slave on *.* to 'repl'@'192.168.10.130' identified by '123456';
```

3. 分别Slave上把Master_1和Master_2的数据导入Slave服务器，在导入前先修改MySQL存储master-info和relay-info的方式，即从文件存储改为表存储，在my.cnf里添加以下选项：

```
master_info_repository=TABLE
```

```
relay_log_info_repository=TABLE
```

or 在线修改：

```
stop slave;
```

```
set global master_info_repository='TABLE';
```

```
set global relay_log_info_repository='TABLE';
```

数据导入：

```
mysql -uroot -p < master01.sql
```

```
mysql -uroot -p < master02.sql
```

分别找出Master_1和Master_2的binlog位置和POS位置：

```
cat master01.sql | grep CHANGE MASTER
```

```
cat master02.sql | grep CHANGE MASTER
```

4. 登录Slave进行同步操作，分别change master到两台master服务器，

```
change master to master_host='192.168.10.128', master_user='repl', master_password='123456', master_log_file='Master_1-bin.000001', master_log_pos=1539 for channel 'Master_1';
```

```
change master to master_host='192.168.10.129', master_user='repl', master_password='123456', master_log_file='Master_2-bin.000003', master_log_pos=630 for channel 'Master_2';
```

启动slave：

```
start slave;
```

or

```
start slave for channel 'Master_1';
```

```
start slave for channel 'Master_2';
```

查看复制源的同步状态：

```
show slave status for channel 'Master_1'\G
```

```
show slave status for channel 'Master_2'\G
```

or 通过查看performance_schema相关的表查看同步状态，

```
select * from performance_schema.replication_connection_status;
```

5. 验证数据是否同步

在Master_1上插入两条数据：

```
insert into tb1(name) values ('user1'), ('user2');
```

在Master_2上插入两条数据：

```
insert into tb2(name) values ('user3'), ('user4');
```

回到Slave上查看是否正常把数据同步过来了：

```
show databases;
```

```
select * from master01.tb1;
```

```
select * from master02.tb2;
```

参考：<http://www.cnblogs.com/xuanzhi201111/p/5151666.html>