

10_RDMBS201mysql 主从同步

一 主从同步概述

1.1 mysql 主从同步

实现数据自动同步的服务结构

主服务器:接受客户端访问连接

从服务器:自动同步主服务器数据

1.2 拓扑结构

主服务器:192.168.4.51, 从服务器:192.168.4.52; 客户端:192.168.4.50



1.3 主从同步原理

1.3.1 Master 启用 binlog 日志

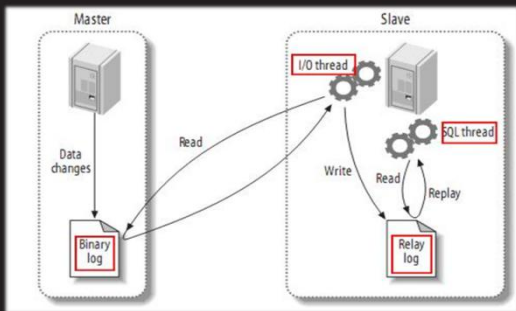
1.3.2 slave

slave_I0:复制 master 主机 binlog 日志文件里的 SQL 命令到本机的 relay-log 文件里

slave_SQL:执行本机 relay-log 文件里的 SQL 语句,实现与 Master 数据一致.

主从同步原理

知识讲解



二 构建主从同步

2.1 构建思路

2.1.1 配置主服务器：启用 binlog 日志\授权用户\查看 binlog 日志信息

2.1.2 配置从服务器

设置 `server_id`

确保从服务器数据与主服务器数据一致

指定主库信息

启动 slave 程序：启动后产生 I/O thread 线程和 SQL thread 线程

查看状态信息

2.2 配置主服务器

2.2.1 修改主配置文件,启用 binlog 日志,并重启 mysqld 服务

```
mysql51 ~]# vim /etc/my.cnf
```

```
[mysqld]
```

```
log_bin=/myload/master51 #启用 binlog 日志,默认路径/var/lib/mysql
```

```
server_id=51
```

```
mysql51 ~]# chown -R mysql:mysql /myload #这步很重要
```

```
mysql51 ~]# systemctl restart mysqld
```

2.2.2 授权用户

授权用户 replication slave 权限,用户名自定义,客户端地址允许从库连接,密码自定义,此用户用于指定主库信息

```
51 mysql> grant replication slave on *.* to repluser@"%" identified  
by "123456";
```

```
51 mysql> show grants for repluser@"%"; #查看确认授权完成
```

2.2.3 查看日志信息(系统状态)

```
51 mysql> show master status\G;
```

```
***** 1. row *****
```

```
File: master51.000001
```

```
...
```

2.3 配置从服务器

2.3.1 修改主配置文件 my.cnf, 指定 server_id, 不允许与主库 server_id 值相同, 并重启 mysqld 服务

```
mysql52 ~]# vim /etc/my.cnf
```

```
[mysqld]
```

```
server_id=52    #指定 id
```

```
mysql52 ~]# systemctl restart mysqld
```

2.3.2 确保与主服务器数据一致

2.3.2.1 在主服务器上备份数据, 将备份文件拷贝给从服务器

```
mysqldump -uroot -p 密码 --master-data 数据库名 > /allbak.sql
```

```
mysql51 ~]# mysqldump -uroot -p123456 --master-data db5 >  
/allbak.sql
```

```
mysql51 ~]# scp /allbak.sql root@192.168.4.52:/root
```

--master-date 用于向 allbak.sql 文件中写入 master 信息, 见 2.3.2.3

2.3.2.2 从服务器使用备份文件恢复数据

```
mysql -uroot -p 密码 数据库名 < /root/allbak.sql
```

```
52 mysql> create database db5;
```

```
mysql52 ~]# mysql -uroot -p123456 db5 < /root/allbak.sql
```

2.3.2.3 在从服务器查看备份文件中的 binlog 日志信息

```
mysql52 ~]# vim /root/allbak.sql    或
```

```
mysql52 ~]# grep -i master /root/allbak.sql
```

```
CHANGE MASTER TO
```

```
MASTER_LOG_FILE='master51.000001', MASTER_LOG_POS=441;
```

2.3.3 指定主库信息

```
52 mysql> show slave status\G;  #先查看一下从服务器状态
```

```
52 mysql> change master to
```

```
-> master_host="192.168.4.51",
```

```
-> master_user="repluser",
```

```
-> master_password="123456",
```

```
-> master_log_file="master51.000001",
```

```
-> master_log_pos=441;
```

2.3.4 启动 slave 程序

```
52 mysql> start slave;
```

2.3.5 查看 I/O thread 线程和 SQL thread 线程

```
52 mysql> show slave status\G;
```

```
***** 1. row *****
```

```
Slave_IO_State: Waiting for master to send event
```

```
Master_Host: 192.168.4.51  #主库 ip 地址
```

```
Master_User: repluser
```

```
.....
```

Slave_IO_Running: Yes #IO 线程已运行

Slave_SQL_Running: Yes #SQL 线程已运行

2.4 从服务器相关文件

数据库目录 /var/lib/mysql/

master.info 主库信息

relay-log.info 中继日志信息

主机名-relay-bin.xxxxxx 中继日志

主机名-relay-bin.index 中继日志索引文件

2.5 如何把 52 还原成独立 mysql 服务器

删除 2.4 内的 4 个文件,重启 mysqld.

2.6 排错

52 mysql> stop slave;

52 mysql> show slave status\G;

Last_IO_Error: #此处为 no,查看物理连接(ping) 检查授权用户

 #禁用防火墙,关闭 SELinux

 #检查 binlog 日志文件(日志名和 pos 节点)

 #connecting 为授权密码错误

Last_SQL_Error: #此处为 no,查看库\表或记录在本机是否存在

解决错误后

52 mysql> start slave;

终极办法,按 2.5 删除文件后重做.

2.7 验证

2.7.1 主服务器上添加验证用户

```
51 mysql> create database bbsdb;
```

```
51 mysql> grant all on bbsdb.* to yaya108@"%" identified  
by "123456";
```

```
51 mysql> grant all on db5.* to yaya108@"%" identified  
by "123456";
```

```
51 mysql> show grants for yaya108@"%";
```

2.7.2 50 上使用 yaya108 登录数据库服务器,并查看自身权限

```
mysql50 ~]# mysql -h192.168.4.51 -uyaya108 -p123456
```

```
50 mysql> show grants;
```

2.7.3 50 上使用 yaya108 往数据库写入数据

```
50 mysql> insert into db5.a values("xyy");
```

```
50 mysql> insert into db5.b values(120);
```

2.7.4 在从服务器 52 上用 root 登录,并查看 50 上 yaya108 写入数据库的数据

```
mysql52 ~]# mysql -uroot -p123456
```

```
52 mysql> select * from db5.a;
```

```
52 mysql> select * from db5.b;
```

三 主从同步模式

3.1 主从同步结构 之 结构类型

3.1.1 基本应用结构类型

单向复制结构类型： 一主 <-- 一从

3.1.2 扩展应用结构类型

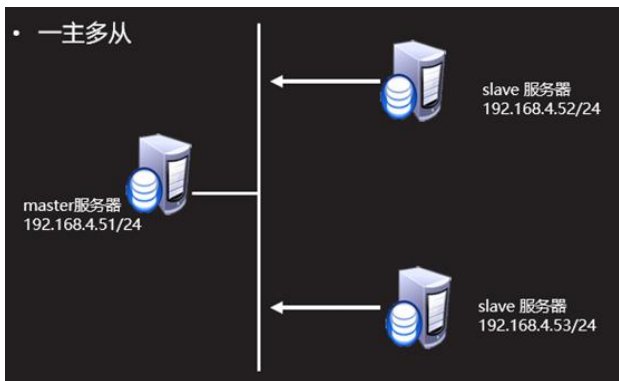
一主多从结构类型：从 <-- 主 --> 从



链式复制结构类型：主 <-- 从 <-- 从 或级连复制结构类型

互为主从结构类型：主 <--> 从

3.2 在二的基础上配置一主多从



3.2.1 修改配置文件并重启 mysqld 服务

```
mysql53 ~]# vim /etc/my.cnf

[mysqld]

server_id=53

mysql53 ~]# systemctl restart mysqld
```

3.2.2 确保与主服务器数据一致

```
mysql51 ~]# mysqldump -uroot -p123456 --master-data -A > /a.sql

mysql51 ~]# scp /a.sql root@192.168.4.53:/root

mysql51 ~]# mysql -uroot -p123456 -e "show master status;"

      master51.000001          2695

mysql53 ~]# mysql -uroot -p123456 -A < /root/a.sql
```

3.2.3 指定主库信息

```
53 mysql> show slave status;

      Empty set (0.00 sec)

53 mysql> change master to

-> master_host="192.168.4.51",

-> master_user="repluser",

-> master_password="123456",

-> master_log_file="master51.000001",

-> master_log_pos=2695;
```

3.2.4 启动 slave 进程

```
53 mysql> start slave;
```

3.2.5 查看 I/O thread 线程和 SQL thread 线程

```
53 mysql> show slave status\G;
```

```
***** 1. row *****  
  
Slave_IO_State: Waiting for master to send event  
  
Master_Host: 192.168.4.51  
  
.....  
  
Slave_IO_Running: Yes  
  
Slave_SQL_Running: Yes
```

3.2.6 验证

使用 yaya108 登录 50 上的数据库并插入数据,在 51 52 53 上查看插入的数据

```
mysql50 ~]# mysql -h192.168.4.51 -uyaya108 -p123456
```

```
50 mysql> show grants;    #yaya108 在 50 上登录 mysql 后查看自身权限
```

```
50 mysql> insert into db5.a values("wx");
```

```
mysql51 ~]# mysql -uroot -p123456 -e "select * from db5.a;"
```

```
mysql52 ~]# mysql -uroot -p123456 -e "select * from db5.a;"
```

```
mysql53 ~]# mysql -uroot -p123456 -e "select * from db5.a;"
```

3.3 配置主从从

master 服务器(192.168.4.53/24) --> slave 服务器(192.168.4.54/24) [同

时为 192.168.4.55 的主服务]-->slave 服务器(192.168.4.54/24)

3.3.1 配置主服务器 53

将 53 初始化

```
mysql53 ~]# systemctl stop mysqld #停止 mysqld 服务
```

```
mysql53 ~]# cd /var/lib/mysql #删除从服务器文件,见 2.4
```

```
mysql53 mysql]# rm -rf master.info
```

```
mysql53 mysql]# rm -rf relay-log.info
```

```
mysql53 mysql]# rm -rf mysql53*
```

```
mysql53 mysql]# systemctl restart mysqld #重启服务
```

```
mysql53 mysql]# mysql -uroot -p123456 #进入数据库,删除多余的库
```

```
53 mysql> show databases;
```

```
53 mysql> drop database bbsdb;
```

```
53 mysql> drop database db5;
```

3.3.1.1 启用 binlog 日志

```
mysql53 ~]# vim /etc/my.cnf
```

```
[mysqld]
```

```
log_bin=master53 #master53 开头文件在/var/lib/mysql 目录下
```

```
server_id=53
```

3.3.1.2 重启数据库服务

```
mysql53 ~]# systemctl restart mysqld
```

3.3.1.3 授权用户

```
mysql53 ~]# mysql -uroot -p123456
```

```
53 mysql> grant replication slave on *.* to repluser@"%"  
identified by "123456";
```

```
53 mysql> show grants for repluser@"%";
```

```
53 mysql> show master status;
```

```
master53.000001      441
```

3.3.1.4 创建备份文件(53 初始后,数据一致.不用做数据同步工作)

3.3.2 配置从服务器 54

3.3.2.1 修改配置文件并重启 mysqld 服务

```
[mysqld]
```

```
server_id=54
```

```
log_bin=master54
```

```
log_slave_updates    #添加此行,很重要
```

```
mysql53 ~]# systemctl restart mysqld
```

3.3.2.2 授权用户

```
54 mysql> grant replication slave on *.* to repluser@"%"  
identified by "123456";
```

```
54 mysql> show grants for repluser@"%";
```

3.3.2.3 指定主库信息

```
54 mysql> change master to
-> master_host="192.168.4.53",
-> master_user="repluser",
-> master_password="123456",
-> master_log_file="master53.000001",
-> master_log_pos=441;
```

```
54 mysql> show slave status\G;
```

3.3.2.4 启动 slave 进程

```
54 mysql> start slave;
```

3.3.2.5 查看 IO SQL 进程

```
54 mysql> show slave status\G;
```

```
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: 192.168.4.53
.....
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
```

3.3.3 配置从服务器 55

3.3.2.1 修改配置文件并重启 mysqld 服务

```
mysql55 ~]# vim /etc/my.cnf
```

```
[mysqld]
```

```
server_id=55
```

```
mysql55 ~]# systemctl restart mysqld
```

3.3.3.2 指定主库信息

```
mysql54 ~]# mysql -uroot -p123456 -e "show master status;"
```

```
master54.000002      441
```

```
mysql55 ~]# mysql -uroot -p123456
```

```
55 mysql> change master to
```

```
-> master_host="192.168.4.54",
```

```
-> master_user="repluser",
```

```
-> master_password="123456",
```

```
-> master_log_file="master54.000002",
```

```
-> master_log_pos=441;
```

```
55 mysql> show slave status\G;
```

3.3.3.3 启动 slave 进程

```
55 mysql> start slave;
```

3.3.3.4 查看 IO SQL 进程

```
55 mysql> show slave status\G;
```

```
***** 1. row *****
```

```
Slave_IO_State: Waiting for master to send event
```

Master_Host: 192.168.4.54

...

Slave_IO_Running: Yes

Slave_SQL_Running: Yes

3.3.4 测试

3.4.1.1 主服务器 53 上查看授权用户及授权用户的权限

```
53 mysql> select user,host from mysql.user;
```

user	host	
repluser	%	
yaya108	%	
mysql.sys	localhost	
root	localhost	

```
53 mysql> show grants for yaya108@"%";
```

3.4.1.1 在 50 上用 yaya108 登录 53 的数据库,并写入数据

```
mysql50 ~]# mysql -uyaya108 -p123456 -h192.168.4.53
```

```
50 mysql> show grants; #yaya108 在 50 上登录 51 数据库后查看自身权限
```

```
50 mysql> create database bbsdb; #创建库.表,并写入数据
```

```
50 mysql> create table bbsdb.a(name char(10));
```

```
50 mysql> insert into bbsdb.a values("tom");
```

3.4.1.2 53 54 55 管理员查看 yaya108 在 50 上登录 51 的数据库写入的数据

```
53 mysql> select * from bbsdb.a;
```

```
54 mysql> select * from bbsdb.a;
```

```
55 mysql> select * from bbsdb.a;
```

四 mysql 主从同步复制模式

4.1 复制模式介绍

4.1.1 异步复制(Asynchronous replication)[默认模式]

主库执行完一次事务后,立即将结果返回给客户端,不关心从库是否已经接收并处理

4.1.2 全同步复制(Fully synchronous replication)

当主库执行完一次事务,且所有从库都执行了该事务后才将结果返回给客户端

4.1.3 半同步复制(Semisynchronous replication)

介于异步复制和全同步复制之间

主库在执行完一次事务后,等待至少一个从库接收到并写到 relay log 中才将结果返回给客户端.

4.2 模式配置

4.2.1 查看是否运行动态加载模块[默认是允许的]

```
mysql> show variables like "have_dynamic_loading";
```

Variable_name	Value
have_dynamic_loading	YES

4.2.2 命令行加载插件并查看

使用数据库管理员 root 用户进行

install plugin 指定插件类型 **SONAME** 指定模块名称

```
mysql> install plugin rpl_semi_sync_master
```

```
-> SONAME "semisync_master.so"    #主服务器上执行,加载 master 模块
```

```
mysql> install plugin rpl_semi_sync_slave
```

```
-> NONAME "semisync_slave.so"    #从服务器上执行,加载 slave 模块
```

```
mysql> select plugin_name,plugin_status
```

```
-> from information_schema.plugins
```

```
-> where plugin_name like "%semi%";    #查看模块是否安装成功
```

plugin_name	plugin_status
-------------	---------------

rpl_semi_sync_master	ACTIVE
----------------------	--------

rpl_semi_sync_slave	ACTIVE
---------------------	--------

4.3 临时启用半同步复制[默认关闭]

```
mysql> set global rpl_semi_sync_master_enabled=1;    #主服务器执行
```

```
mysql> set global rpl_semi_sync_slave_enabled=1;    #从服务器执行
```

```
mysql> show variables like "rpl_semi_sync_%_enabled";
```

 #查看模块是否启用

Variable_name	Value
---------------	-------

rpl_semi_sync_master_enabled	ON
------------------------------	----

rpl_semi_sync_slave_enabled	ON
-----------------------------	----

4.4 永久启用半同步复制

4.4.1 修改主配置文件

```
vim /etc/my.cnf
```

在[mysql]下方添加

```
plugin-load=rpl_semi_sync_master=semisync_master.so
```

```
rpl_semi_sync_master_enabled=1      #主服务器
```

```
plugin-load=rpl_semi_sync_slave=semisync_slave.so
```

```
rpl_semi_sync_slave_enabled=1      #从服务器
```

4.4.2 服务器是master 和 slave 两种角色时,需要同时启用

```
plugin-load="rpl_semi_sync_master=semisync_master.so;
```

```
rpl_semi_sync_slave=semisync_slave.so"
```

```
rpl_semi_sync_master_enabled=1
```

```
rpl_semi_sync_slave_enabled=1
```

```
mysql> show variables like "rpl_semi_sync_%_enabled" #查看
```