　一　mysql主从同步

客户端访问谁，谁便是主库

１，基础

－－主从同步原理

1）master,记录数据更改操作

启用binlog日志

设置binlog日志格式

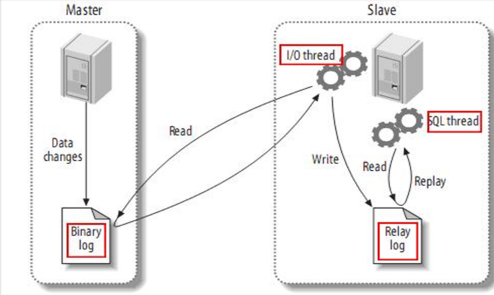
设置server\_id

２）slave运行2个线程

Slave\_io：复制master主机binlog日志文件里的sql到本机的relay-log文件里

Slave\_sql:执行本机relay\_log文件里的sql语句，重现master的数据操作

主从同步原理图：



－－构建主从同步

１）基本思路

确保数据相同：从库必须要有主库上的数据

配置主服务器：启用binlog日志，授权用户，查看当前正在使用的日志

配置从服务器：设置server\_id（唯一），指定主库信息

测试配置：客户端连接主库写入数据，在从库上也能查询到

２）确保数据一致

Masterr服务器

备份所有库

Slave服务器

清空同名库（如果有的话）

离线导入由master提供的备份

Server\_id n #1~255

提示：Relay-log是中继日志文件

２，主从mysql实战

－－基本配置

１）配置主服务

[root@mysql-51 ~]# vim /etc/my.cnf

[mysqld]

log\_bin=master51

server\_id=51　　　#mysql主从里，id必须唯一，范围：1~255

binlog\_format="mixed"

[root@mysql-51 ~]# systemctl restart mysqld #重启刷新文件配置

mysql> show master status\G; #查看主库master状态，查看当前使用的binlog日志文件

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

File: master51.000001

Position: 441

Binlog\_Do\_DB:

Binlog\_Ignore\_DB:

Executed\_Gtid\_Set:

1 row in set (0.00 sec)

mysql> grant replication slave on \*.\* to repluser@'%' identified by '123456'; #授权给从服务器，单个数据库授权无效，必须设置\*.\*

２）配置从服务器

mysql> show slave status;　　　#查看从库信息

mysql> show master status; #查看binlog日志，查看主库信息

[root@mysql-52 ~]# vim /etc/my.cnf

[mysqld]

server\_id=52　　　　　#从库id，要唯一

[root@mysql-52 ~]# systemctl restart mysqld

mysql> change master to

-> master\_host='192.168.4.51',

-> master\_user='repluser',

-> master\_password='123456',

-> master\_log\_file='master51.000001',　　#与主库binlog日志名相同

-> master\_log\_pos=441;　　　　#偏移量，与主库相同

提示：任意参数指定不对，io线程读取不了信息

mysql> start slave;　　　　#启动slave(io,sql线程)

mysql> show slave stattus\G;

Slave\_IO\_State: Waiting for master to send event

Master\_Host: 192.168.4.51

Master\_User: repluser

Master\_Port: 3306

Connect\_Retry: 60

Master\_Log\_File: master51.000001

Read\_Master\_Log\_Pos: 441

Relay\_Log\_File: mysql-52-relay-bin.000002

Relay\_Log\_Pos: 319

Relay\_Master\_Log\_File: master51.000001

Slave\_IO\_Running: Yes #看到yes就成功了

Slave\_SQL\_Running: Yes　　　#ok了

........

提示：sql线程在执行中继日志的sql命令时，一旦出错，sql线程立即停止

方案：停掉slave，在从库新建相关表,重启slave

3）测试配置

在master上操作数据

mysql> create database db1;　　　#新建测试数据库

mysql> create table db1.t1(id int );　　#新建测试表

mysql> grant select,insert on db1.\* to rose@'%' identified by '123456';　#给客户端授予相应权限

在client上插入数据

[root@mysql-50 ~]# mysql -h192.168.4.51 -urose -p123456#连接master主库

mysql> show grants;　　　#查看自己的权限

mysql> use db1;　　　　　#切换库

mysql> show tables;　　　#查看表

mysql> desc t1;　　　　　#查看表结构

mysql> insert into t1 values(1);　　　#插入验证数据

mysql> insert into t1 values(２);

在slave上查看

mysql> show databases;　　#查看数据库

+--------------------+

| Database |

+--------------------+

| information\_schema |

| db1 |　　　　　　　　　#多了数据库db1

mysql> select \* from db1.t1; #查看数据，验证成功

+------+

| id |

+------+

| 1 |

| 2 |

+------+

2 rows in set (0.01 sec)

1. 主从库配置选项

从库相关文件

Master.info 　　　 　主库信息

Relay-log.info 　　　 中继日志信息

主机名-relay-bin.xxxx 　　　 中继日志

主机名-relay-bin.index 　　　索引文件

提示：删除掉这几个文件，重启就还原了

例子：

[root@mysql-52 mysql]# rm -rf mysql-52-relay-bin.\*

[root@mysql-52 mysql]# rm -rf master.info

[root@mysql-52 mysql]# rm -rf relay-log.info

[root@mysql-52 mysql]# systemctl restart mysqld

[root@mysql-52 mysql]# mysql -uroot -p123456 -e "show slave status"　#无数据输出，说明已经不是从库

主库配置选项（对所有从库有效）

Binlog\_do\_db=name 设置master对哪些库记日志

Binlog\_ignore\_db=name　　设置master对哪些库不记日志

例子：

[root@mysql-51 ~]# vim /etc/my.cnf

[mysqld]

binlog\_do\_db=db1, db2, db3 记录哪些库

#binlog\_ignore\_db=db9　　　不记录哪些库

mysql> show master status\G

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

File: master51.000003

Position: 154

Binlog\_Do\_DB: db1,db2,db3

Binlog\_Ignore\_DB:

Executed\_Gtid\_Set:

1 row in set (0.00 sec)

从库配置选项（只对slave服务器有效）

Log\_slave\_update 　　　记录从库更新，允许链式复制（a-b-c）

Relay\_log=dbsvr2-relay-bin 指定中继日志文件名

Replicate\_do\_db=mysql 仅复制指定库，其他库将被忽略，此选项可设置多条

Replicate\_ignore\_d=test　　　不复制哪些库，其他库将被忽略

例子：

[root@mysql-52 mysql]# vim /etc/my.cnf

[mysqld]

replicate\_do\_db=db1,db2

#replicate\_ignore\_do=db3

[root@mysql-52 mysql]# systemctl restart mysqld

mysql> show slave status\G;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Slave\_IO\_State: Waiting for master to send event

Master\_Host: 192.168.4.51

Master\_User: repluser

Master\_Port: 3306

Connect\_Retry: 60

Master\_Log\_File: master51.000003

Read\_Master\_Log\_Pos: 154

Relay\_Log\_File: mysql-52-relay-bin.000007

Relay\_Log\_Pos: 319

Relay\_Master\_Log\_File: master51.000003

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

Replicate\_Do\_DB: db1,db2　　　#允许仅复制指定库

Replicate\_Ignore\_DB:

Replicate\_Do\_Table:

Replicate\_Ignore\_Table:

3，主从同步模式

－－结构模式

基本应用

单向复制：主－－＞从

扩展应用

链式复制:主－－＞从－－＞从　　　＃主从从

互为主从：主＜－－＞主　　　#不能同时被访问，不能单独使用

一主多从：从＜－－主－－＞从

１）一主多从

[root@mysql-51 ~]# mysqldump -uroot -p123456 db1 > db1.sql

[root@mysql-51 ~]# scp db1.sql 192.168.4.53:/root

mysql> create database db1;　　　　#53主机

mysql> source /root/db1.sql;　　　　#可以直接执行sql语句

[root@mysql-53 ~]# mysql -uroot -p123456 db1 < db1.sql

[root@mysql-53 ~]# vim /etc/my.cnf

[mysqld]

server\_id=53

[root@mysql-53 ~]# systemctl restart mysqld

mysql> change master to

-> master\_host="192.168.4.51",

-> master\_user="repluser",

-> master\_password="123456",

-> master\_log\_file="master51.000005",

-> master\_log\_pos=574;

mysql> start slave;

mysql> show slave status\G;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Slave\_IO\_State: Waiting for master to send event

Master\_Host: 192.168.4.51

Master\_User: repluser

Master\_Port: 3306

Connect\_Retry: 60

Master\_Log\_File: master51.000005

Read\_Master\_Log\_Pos: 574

Relay\_Log\_File: mysql-53-relay-bin.000002

Relay\_Log\_Pos: 319

Relay\_Master\_Log\_File: master51.000005

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

２）主从从

#延续上面的实验

[root@mysql-52 mysql]# vim /etc/my.cnf　　　#设置52即为从又为主库

[mysqld]

log\_bin=master52

binlog\_format="mixed"

server\_id=52

Log\_slave\_updates #记录从库更新，允许链式复制

[root@mysql-52 mysql]# systemctl restart mysqld

mysql> show master status\G;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

File: master52.000001

Position: 154

Binlog\_Do\_DB:

Binlog\_Ignore\_DB:

Executed\_Gtid\_Set:

1 row in set (0.00 sec)

mysql> grant replication slave on \*.\* to repluuser@'%'

-> identified by '123456';

配置53的主库是52

mysql> change master to

-> master\_host="192.168.4.52",

-> master\_user="repluser",

-> master\_password="123456",

-> master\_log\_file='master52.000001',

-> master\_log\_pos=442;

mysql> start slave;

注意：

Slave-io线程读取主库中binlog日志放入中继日志(relay-log)，slave-sql线程在执行中继日志的语句时是不写入binlog日志中，需要添加Log\_slave\_updates才能写入binlog日志中

３）互为主从

主机54：

[root@mysql-54 ~]# vim /etc/my.cnf

[mysqld]

log\_bin=master54

server\_id=54

binlog\_format="mixed"

[root@mysql-54 ~]# systemctl restart mysqld

mysql> grant replication slave on \*.\* to masteruser@'%' identified by '123456';

mysql> show master status\G;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

File: master54.000001

Position: 443

．．．．

mysql> change master to \

-> master\_host='192.168.4.55',

-> master\_user='masteruser2',

-> master\_password='123456',

-> master\_log\_file='master55.000001',

-> master\_log\_pos=444;

mysql> start slave;

mysql> show slave status\G;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Slave\_IO\_State: Waiting for master to send event

Master\_Host: 192.168.4.55

Master\_User: masteruser2

Master\_Port: 3306

Connect\_Retry: 60

Master\_Log\_File: master55.000001

Read\_Master\_Log\_Pos: 444

Relay\_Log\_File: mysql-54-relay-bin.000002

Relay\_Log\_Pos: 319

Relay\_Master\_Log\_File: master55.000001

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

主机55：

[root@mysql-55 ~]# vim /etc/my.cnf

[root@mysql-55 ~]# systemctl restart mysqld

mysql> grant replication slave on \*.\* to masteruser2@'%' identified by '123456';

mysql> show master status\G;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

File: master55.000001

Position: 444

．．．．．

mysql> change master to\

-> master\_host="192.168.4.54",

-> master\_user="masteruser",

-> master\_password="123456",

-> master\_log\_file='master54.000001',

-> master\_log\_pos=443;

mysql> start slave;

mysql> show slave status\G;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Slave\_IO\_State: Waiting for master to send event

Master\_Host: 192.168.4.54

Master\_User: masteruser

Master\_Port: 3306

Connect\_Retry: 60

Master\_Log\_File: master54.000001

Read\_Master\_Log\_Pos: 595

Relay\_Log\_File: mysql-55-relay-bin.000002

Relay\_Log\_Pos: 471

Relay\_Master\_Log\_File: master54.000001

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

Replicate\_Do\_DB:

Replicate\_Ignore\_DB:

Replicate\_Do\_Table:

Replicate\_Ignore\_Table:

测试：

在54操作－－＞在55验证

54主机

mysql> create database db1;

mysql> create table db1.t1(id int);

mysql> insert into db1.t1 values(54);

55主机

mysql> select \* from db1.t1;

+------+

| id |

+------+

| 54 |

在55操作数据－－＞在54验证

55主机

mysql> create database db2;

mysql> create table db2.t2(id int);

mysql> insert into db2.t2 values(55);

54主机

mysql> select \* from db2.t2;

+------+

| id |

+------+

| 55 |

－－复制模式

１）基础

异步复制(Asynchronous replication)

　默认工作模式

　　主库执行完一次事务后，立即将结果返回给客户端，并不关心从库是否已经接收并处理

全同步复制( Full 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返回给客户端

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入到relay log中才返回客户端

－－半同步复制

51，52主机设置：

查看是否允许动态加载模块（默认允许）

mysql> show variables like "have\_dynamic\_loading";

+----------------------+-------+

| Variable\_name | Value |

+----------------------+-------+

| have\_dynamic\_loading | YES |

+----------------------+-------+

查看是否安装模块

mysql> select plugin\_name,plugin\_status from information\_schema.plugins where plugin\_name like "%semi%";

Empty set (0.00 sec)

命令行加载插件

mysql> install plugin rpl\_semi\_sync\_master soname 'semisync\_master.so';　mysql> install plugin rpl\_semi\_sync\_slave soname 'semisync\_slave.so';

启用半同步复制

mysql> show variables like 'rpl\_semi\_sync\_%\_enabled%';

mysql> set global rpl\_semi\_sync\_master\_enabled=1;

mysql> set global rpl\_semi\_sync\_slave\_enabled=1;

mysql> show variables like 'rpl\_semi\_sync\_%\_enabled';

+------------------------------+-------+

| Variable\_name | Value |

+------------------------------+-------+

| rpl\_semi\_sync\_master\_enabled | ON |

| rpl\_semi\_sync\_slave\_enabled | ON |

+------------------------------+-------+

永久设置：

[root@mysql-51 ~]# vim /etc/my.cnf

[mysqld]

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

[root@mysql-51 ~]# systemctl restart mysqld