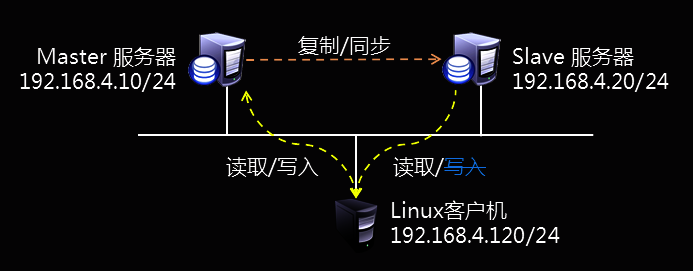
# 部署mysql主从同步结构

## 主从同步概述

### MYSQL主从同步

客户端访问主库，从库复制、同步主库的所有操作

单项复制时，建议将从库设置为只读



### 主从同步原理——从库 从主机上同步数据的工作过程

### Master，记录数据更改操作

— 启用binlog日志

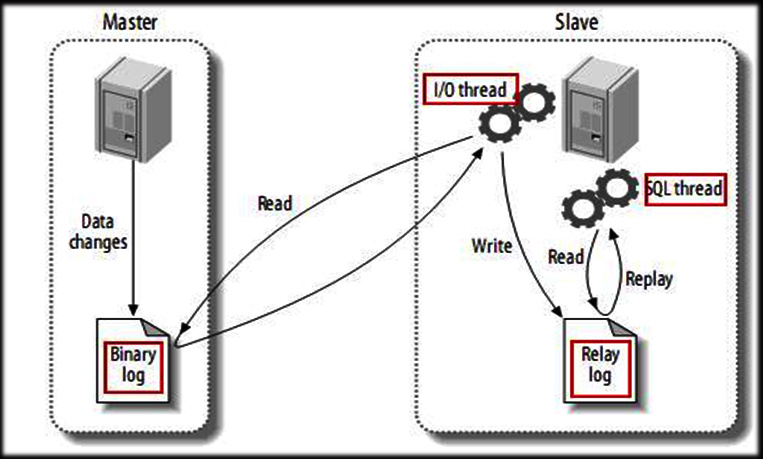
— 设置binlog日志格式

— 设置server\_id

### 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



## 构建主从同步

### 基本构建思路

### 1. 确保数据相同

– 从库必须要有主库上的数据。

### 2. 配置主服务器

– 启用 binlog 日志及设置格式,设置 server\_id, 授权用户

### 3. 配置从服务器

– 设置 server\_id ,指定主数据库服务器信息

### 4. 测试配置

– 客户端连接主库,写入的数据,在连接从库的时候也

### 确保数据一致

**Master服务器**

– 应包括希望同步的所有库

– 对采用 MyISAM 的库,可离线备份

主库配置选项

|  |  |
| --- | --- |
| 选项 | 用途 |
| binlog\_do\_db=name | 设置Master对那些库记日志 |
| binlog\_ignore\_db=name | 设置Master对哪些库不记日志 |

**Slave 服务器**

– 离线导入由 Master 提供的备份

– 清空同名库(若有的话

从库配置选项

|  |  |
| --- | --- |
| 选项 | 用途 |
| log\_slave\_updates | 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（省略时复制所有库） |
| replicate\_ignore\_db=test | 不复制那些库，其他库将被忽略，ignort-db与do-db只需选用其中一种 |

## 主从配置常用参数

### 写在主库配置文件里参数（对所有从库都有效）

binlog\_do\_db=库名列表 //允许同步的库

binlog\_ignore\_db=库名列表 //不允许同步的库

### 写在从库配置文件里参数（只针对从库本机有效）

replicate\_do\_db=库名列表 //指定只同步的库

replicate\_ignore\_db=库名列表 //指定不同步的库

## 主从同步结构模式

### 基本应用：

### — 单项复制：一主一从

### 扩展应用：

### — 一主多从

### — 链式复制：主从从

### — 主主（互为主从）

## mysql主从同步配置步骤

### 配置主库：

启用binlog日志

用户授权

查看当前正在使用的binlog日志

### 配置从库

指定server\_id

指定主库信息

启动slave程序

查看slave程序的状态信息

## 【一主一从**同步结构**配置】

### 确保数据相同

主库：192.168.4.51

### 启用 binlog 日志及设置格式,设置 server\_id

[root@mysql51 ~]# vim /etc/my.cnf

[mysqld]

server\_id=51

log\_bin=mysql51 //binlog日志文件名称

binlog\_format="mixed" //日志文件格式

[root@mysql51 ~]# systemctl restart mysqld

[root@mysql51 ~]# ls /var/lib/mysql/mysql51\*

**授权用户**

[root@mysql51 ~]# mysql -uroot -p123asd...A

mysql> grant replication slave on \*.\* to repluser@"%" identified by "123asd...A";

mysql> select host,user from mysql.user where user='repluser';

mysql> show master status; //查看使用的日志文件及偏移量

从库：192.168.4.52

[root@mysql52 ~]# vim /etc/my.cnf

[mysqld]

server\_id=52

[root@mysql52 ~]# systemctl restart mysqld

[root@mysql52 ~]# mysql -u root -p123asd...A

mysql> show master status;

mysql> change master to master\_host='192.168.4.51', //主库ip地址

-> master\_user='repluser', //出库授权用户名

-> master\_password='123asd...A', //授权用户密码

-> master\_log\_file='mysql51.000001', //日志文件

-> master\_log\_pos=441; //偏移文件

mysql> start slave; //启动salve进程

mysql> show slave status\G; //查看slave状态

Slave\_IO\_Running: Yes //IO线程状态

Slave\_SQL\_Running: Yes //SQL线程状态

如果 Slave\_IO\_Running为NO，查看下面Last\_IO\_Error: Fatal error:的报错信息

[root@mysql52 mysql]# vim auto.cnf //此配置文件查看sql数据库的uuid

### 从服务器相关文件

|  |  |
| --- | --- |
| 文件名 | 说明 |
| master.info | 主库信息 |
| relay-log.info | 中继日志信息 |
| 主机名-relay-bin.xxxx | 中及日志 |
| 主机名-relay-bin.index | 索引文件 |

### 测试主从同步配置：在及客户端50主机连接主机51对数据库做访问

• 在 Master 上操纵数据

– 新建 newdb 库、 newtbl 表

– 任意插入几条表记录

• 在 Slave 上查看数据更改情况

– 确认新建的 newdb 库、 newtbl 表

– 列出 newtbl 表的所有记录

1. 在主库51上添加授权用户给客户端50主机连接使用

mysql> create database db1;

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

mysql> grant select,insert,update on db1.\* to admin@"%" identified by "123asd...A";

1. 在客户端50主机连接主库服务器51，对记录做访问

[root@client50 ~]# mysql -uadmin -h192.168.4.51 -p123asd...A

mysql> select @@hostname;

mysql> insert into db1.a values(111),(1231),(324);

mysql> select \* from db1.a;

1. 再从库服务器52本机也能查看到相同的数据为成功

[root@mysql52 mysql]# mysql -u root -p123asd...A

mysql> select \* from db1.a;

### **【一主多从 同步结构配置】**

把数据库服务器53 也配置为51的从数据库服务器

主库主机51 已经是主数据库服务器，无需再做配置

配置从库53

与主库数据一致（把主库数据的完全备份拷贝到本机执行恢复）

[root@mysql51 ~]# mysqldump -uroot -p123asd...A db1 > /root/db1.sql

[root@mysql51 ~]# scp db1.sql 192.168.4.53:/root/

[root@mysql53 ~]# mysql -u root -p123asd...A

mysql> create database db1;

mysql> use db1;

mysql> source /root/db1.sql;

mysql> select \* from db1.a;

[root@mysql53 ~]# vim /etc/my.cnf

[mysqld]

server\_id=53

[root@mysql53 ~]# systemctl restart mysqld

### 检测授权用户能否连接数据库

[root@mysql53 ~]# mysql -h192.168.4.51 -urepluser -p123asd...A

### 数据管理员指定主库数据信息

[root@mysql53 ~]# mysql -uroot -p123asd...A

mysql> change master to

-> master\_host='192.168.4.51',

-> master\_user='repluser',

-> master\_password='123asd...A',

-> master\_log\_file='mysql51.000001',

-> master\_log\_pos=1337;

1337:从51主机上执行sql命令mysql> show master status;获取

### 查看slave程序的状态信息

mysql> start slave;

mysql> show slave status\G;

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

### 测试

客户端50连接主机51写入的数据在从库52和53主机上也能查看到数据，即成功

[root@client50 ~]# mysql -uadmin -h192.168.4.51 -p123asd...A

mysql> insert into db1.a values(66066);

51和52主机：登入数据库，执行mysql> select \* from db1.a;查看是否有插入的数据

### 配置从库暂时不同步主机数据：停止从库的slave程序即可

**把从库还原成独立的数据服务器：让主机53不做51的从库**

### 1.停止服务

[root@mysql53 ~]# systemctl stop mysqld

### 2.删除从库文件

[root@mysql53 ~]# rm -rf /var/lib/mysql/master.info

[root@mysql53 ~]# rm -rf /var/lib/mysql/mysql53-relay-bin.\*

[root@mysql53 ~]# rm -rf /var/lib/mysql/relay-log.info

### 3.修改配置

[root@mysql53 ~]# vim /etc/my.cnf

[mysqld]

#server\_id=53 //删除此行

**4.重起服务查看主从状态**

[root@mysql53 ~]# systemctl start mysqld

[root@mysql53 ~]# mysql -uroot -p123asd...A -e "show slave status\G"

**【主从从同步结构配置】**

### 配置主库51：

启用binlog日志 用户授权 查看日志信息

### 配置从库52（既做主又做从）

主配置文件里：要启用log\_slave\_updates配置

做主库的配置：启用binlog日志 用户授权 查看日志信息

[root@mysql52 mysql]# vim /etc/my.cnf

[mysqld]

server\_id=52

log\_slave\_updates //记录从库更新，允许链路复制

log\_bin=slave52

binlog\_format="mixed"

[root@mysql52 mysql]# systemctl restart mysqld

[root@mysql52 mysql]# mysql -uroot -p123asd...A

mysql> grant replication slave on \*.\* to repluser2@"%" identified by "123asd...A";

mysql> show master status;

mysql> show slave status\G;

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

做从库的配置：指定server\_id 指定主库信息

启动slave程序 查看slave程序状态信息

把主机53配置为52的从库服务器

在没有配置为从库之前要与主库数据一致

指定server\_id 指定主库信息

启动slave 程序 查看slave程序状态信息

[root@mysql53 ~]# vim /etc/my.cnf

[mysqld]

server\_id=53

[root@mysql53 ~]# systemctl restart mysqld

[root@mysql53 ~]# mysql -urepluser2 -h192.168.4.52 -p123asd...A

[root@mysql53 ~]# mysql -uroot -p123asd...A

mysql> change master to master\_host='192.168.4.52',

-> master\_user='repluser2',

-> master\_password='123asd...A',

-> master\_log\_file='slave52.000001',

-> master\_log\_pos=442;

mysql> start slave;

mysql> show slave status\G;

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

在客户端50上连接主机51写入数据，在52和53主机上都可以查看到即为成功

**【配置主主结构】：**

### 提示：把数据库服务器54 55 配置为主主结构

把数据库服务器51 52 53 恢复为独立的数据库

## mysql主从同步数据复制模式

### 复制模式介绍

### • 异步复制( Asynchronous replication )

– 主库在执行完客户端提交的事务后会立即将结果返给客户端,并不关心从库是否已经接收并处理。

### • 全同步复制( Fully synchronous replication )

– 当主库执行完一个事务,所有的从库都执行了该事务才返回给客户端。

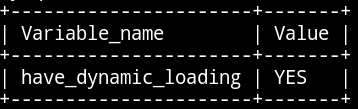
### • 半同步复制( Semisynchronous replication )

– 介于异步复制和全同步复制之间,主库在执行完客户端提交的事务后不是立刻返回给客户端,而是等待至少一个从库接收到并写到 relay log 中才返回给客户端

# 【配置半同步复制模式】

**1. 查看当前的数据库服务器是否支持动态加载模块**

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



1. **使用命令安装模块**

**主库半同步复制的模块**

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

**从库半同步复制的模块**

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

1. **查看模块状态**

mysql> select plugin\_name,plugin\_status

from information\_schema.plugins where plugin\_name like '%semi%';

**4. 启用半同步复制——在安装完插件后，半同步复制默认是关闭的**

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

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

**5. 查看是否启用成功**

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

**6. 把配置写进配置文件使其永久生效**

– 命令配置临时配置,重启服务会失效

– 修改后需要重启服务

– 写在主配置文件 /etc/my.cnf 的 [mysqld] 下方

# vim /etc/my.cnf

plugin-load = "rpl\_semi\_sync\_master=semisync\_master.so;

rpl\_semi\_sync\_slave=semisy nc\_slave.so" //安装模块

rpl-semi-sync-master-enabled = 1 //启用半同步复制

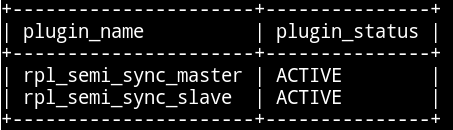
rpl-semi-sync-slave-enabled = 1

# systemctl restart mysqld

查看模块状态

mysql> select plugin\_name,plugin\_status from

information\_schema.plugins where plugin\_name like '%semi%';



查看是否启用成功

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

