**10\_RDMBS202数据读写分离mysql多实例**

**一 数据读写分离**

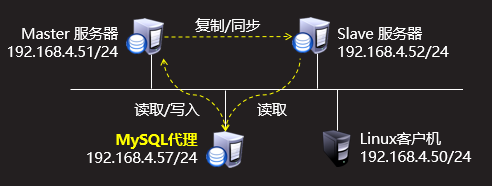
1.1 原因 主从复制的应用局限性

1.2 如何分离mysql读\写流量

1.3 在客户端分,还是在服务器端区分

(客户端使用程序分离,服务器端使用代理服务器分离)

**1.4 案例拓扑**



**1.5 读写分离原理**

由mysql代理面向客户端提供服务

收到sql**写**请求时,交给**master**服务器处理

收到sql**读**请求时,交给**slave**服务器处理

**1.6 构建思路**

1.6.1 部署mysql一主一从结构,主192.168.4.51,从192.168.4.52

1.6.2 部署mysql代理服务器:装包\修改配置文件\启动服务

1.6.3 测试配置:客户端连接代理服务器访问数据

**二 部署maxscale服务**

**2.1 部署前先配置51为主,52为从(52自动同步51数据),57的mysqld服务关闭**

**2.1.1配置主服务器192.168.4.51并重启mysqld服务**

]# vim /etc/my.cnf

[mysqld]

**server\_id=**51 #指定服务器ID号

**log\_bin=**master51 #启用binlog日志，并指定文件名前缀

mysql51 ~]# systemctl restart mysqld #重启mysqld

**2.1.2 主服务器授权用户，并查看binlog日志信息**

mysql51 ~]# mysql -uroot -p123456

51 mysql> grant **replication salve** on \*.\* to 'repluser'@'%' identified by '123456';

51 mysql> show master status;

**2.1.3 配置从服务器192.168.4.52**

mysql52 ~]# vim /etc/my.cnf

[mysqld]

server\_id=52 #指定服务器ID号，不要与Master的相同

mysql52 ~]# systemctl restart mysqld

**2.1.4配置从服务器192.168.4.52**

mysql52 ~]# mysql -uroot -p123456

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=449;

52 mysql> start slave;

52 mysql> show slave status\G;

**2.1.5 57上完全关闭mysqld**

mysql57 ~]# ss -antulp | grep :3306

mysql57 ~]# systemctl stop mysqld #关闭mysqld服务

mysql57 ~]# systemctl disable mysqld #关闭mysqld服务开机自启

mysql57 ~]# killall -9 mysqld #杀掉所有mysqld进程

**2.2 maxscale代理软件的安装配置(在57上操作,将57配置为代理服务器)**

由mysql的兄弟公司mariadb开发,

下载地址:https://downloads.mariadb.com/files/MaxScale

配置文件:/etc/maxscale.cnf

命令:max开头,可以tab补齐

日志文件:/var/log/maxscale/ 启动服务程序后生成

服务程序:rpm -ql maxscale 查询

**2.3修改配置文件**

mysql57 ~]# rpm -ivh maxscale-2.1.2-1.rhel.7.x86\_64.rpm

mysql57 ~]# cp /etc/maxscale.cnf /etc/maxscale.cnf.bak #备份

mysql57 ~]# vim /etc/maxscale.cnf

**[maxscale]**

threads=**auto** #定义运行的线程个数[auto表示自动]

[server**1**] #定义数据库服务器

type=server

address=**192.168.4.51** #指定master主机ip地址

port=3306

protocol=MySQLBackend

将此5行复制粘贴,定义slave服务器

[server**2**] #定义数据库服务器

type=server

address=**192.168.4.52** #指定slave主机ip地址

port=3306

protocol=MySQLBackend

[MySQL Monitor] #定义监视器

type=monitor

module=mysqlmon

servers=**server1,server2** #要监控的数据库服务器

user=**maxscalemon** #定义监视器用户,此用户要在51 52的mysql里授权

passwd=**123qqq...A** #监视器用户的密码,此密码要在51 52的mysql里授权

monitor\_interval=10000

#[Read-Only Service] #此模块注释掉,不定义只读服务

#type=service

#router=readconnroute

#servers=server1

#user=myuser

#passwd=mypwd

#router\_options=slave

[Read-Write Service] #定义读写分离服务

type=service

router=readwritesplit

servers=**server1,server2** #要进行读写分离的数据库服务器

user=**maxscalerouter** #路由用户,此用户要在51 52的mysql里授权

passwd=**123qqq...A** #路由用户密码,此密码要在51 52的mysql里授权

max\_slave\_connections=100%

[MaxAdmin Service] #定义管理服务

type=service

router=cli

#[Read-Only Listener] #此模块注释掉,不定义只读服务的监听端口

#type=listener

#service=Read-Only Service

#protocol=MySQLClient

#port=4008

[Read-Write Listener] #定义读写监听服务

type=listener

service=Read-Write Service

protocol=MySQLClient

port=4006

[MaxAdmin Listener] #定义maxscale管理服务

type=listener

service=MaxAdmin Service

protocol=maxscaled

socket=default

port=4016 #手动添加,不指定则使用随机默认端口,在启动服务后才能知道默认端口号是多少

**2.4在主\从服务器授权用户**

51 mysql> grant **replication slave,replication client** on \*.\* to maxscalemon@"%" identified by "123qqq...A"; #授权监视器用户

51 mysql> grant select on mysql.\* to maxscalerouter@"%" identified by "123qqq...A"; #授权路由用户

51 mysql> select user,host from mysql.user;

52 mysql> select user,host from mysql.user;

查看授权用户的数据是否已同步

**2.5启动服务,查看进程\端口,日志文件,停止服务**

mysql57 ~]# maxscale -f /etc/maxscale.cnf #启动服务

mysql57 ~]# netstat -antulp | grep 4006 #查看端口

mysql57 ~]# netstat -antulp | grep 4016 #查看端口

mysql57 ~]# netstat -antulp | grep maxscale #查看进程

mysql57 ~]# ls /var/log/maxscale/ #列出日志文件

maxscale.log

mysql57 ~]# cat /var/log/maxscale/maxscale.log #查看日志文件

若提示notice,表示正常

mysql57 ~]# killall -9 maxscale #停止服务

mysql57 ~]# killall -9 maxscale

maxscale: no process found

**经常出错的地方:1 配置文件,2 授权用户的时候出错**

**2.6测试配置**

**2.6.1 在代理服务器本机访问管理**

**maxadmin -uadmin -pmariadb -P端口**

[root@mysql57 ~]# maxscale -f /etc/maxscale.cnf

[root@mysql57 ~]# maxadmin -uadmin -pmariadb -P4016

MaxScale> list servers

Server Address Port Connections Status

server1 192.168.4.51 3306 0 Master, Running

server2 192.168.4.52 3306 0 Slave, Running

MaxScale> exit #退出

**2.6.2 客户端连接代理访问数据**

**2.6.2.1主服务器添加客户端访问数据的连接用户**

mysql51 ~]# mysql -uroot -p12356 #管理员登录数据库

51 mysql> create database db7; #创建库

51 mysql> create table db7.a(id int); #创建表

51 mysql> grant select,insert on db7.\* to plj99@"%" identified by "123qqq...A"; #创建连接用户

52 mysql> show grants for plj99@"%"; #52上查看数据是否自动同步

**2.6.2.2 在客户端50上,使用连接用户,登录服务器57上的数据库,并访问数据**

**mysql -h服务器地址 -P端口 -u用户名 -p密码**

mysql50 ~]# mysql -h192.168.4.57 -P4006 -uplj99 -p123qqq...A

50 mysql> show grants;

Grants for plj99@%

GRANT USAGE ON \*.\* TO 'plj99'@'%'

GRANT SELECT, INSERT ON `db7`.\* TO 'plj99'@'%'

50 mysql> select \* from db7.a;

mysql> insert into db7.a values(100);

mysql> select \* from db7.a; 有值100

#写入51值100,52自动同步值100,同步完成后从52把值100读出

**2.6.2.3 测试数据读写分离**

默认写51,读52,因为52是51的从服务器,52自动同步51的数据,51不同步52的数据.若在52写入数据,则写入52的数据在51上没有,但写入的数据能被读

在从服务器52写入数据

52 mysql> insert into db7.a values(52);

52 mysql> select \* from db7.a; 有值100和52

在主服务器查看数据

51 mysql> select \* from db7.a; 有值100

在50上登录57的数据库,读数据

mysql50 ~]# mysql -h192.168.4.57 -P4006 -uplj99 -p123qqq...A

50 mysql> select \* from db7.a; 有值100和52

**2.7读写分离排错**

2.7.1 查看授权用户(监视器用户\路由用户\连接用户的用户名或密码错误)

2.7.2 查看mysql-proxy进程: ps aux | grep “mysql-proxy”

**2.8 读写分离结构的缺点**

单点故障:51 52 57 三台机器都存在单点故障

**三 多实例**

**3.1 多实例概述**

3.1.1 定义:在一台服务器上运行多个数据库服务

3.1.2 原因:节约运维成本,提高硬件利用率

**3.2 配置多实例**

**3.2.1 解压安装软件,修改目录名,修改PATH路径**

@mysql58 ~]# yum -y install libaio #安装依赖包

@mysql58 ~]# useradd mysql #创建用户

**修改目录名**

mysql58 ~]# tar -zxvf mysql-5.7.20-linux-glibc2.12-x86\_64.tar.gz

mysql58 ~]# mv mysql-5.7.20-linux-glibc2.12-x86\_64

/usr/local/mysql #移动并重命名为mysql

**修改PATH路径**

**可执行程序mysqld\_mutli在路径/usr/local/mysql/bin下**

mysql58 ~]# **PATH=/usr/local/mysql/bin:$PATH** #临时修改

mysql58 ~]# **vim /etc/bashrc** #永久修改

**export PATH=/usr/local/mysql/bin:$PATH** #最下面添加此行

**3.2.2 创建并编辑主配置文件/etc/my.cnf**

**每个实例要有独立的端口\数据库目录\socket文件\pid文件\错误日志文件**

mysql58 ~]# **vim /etc/my.cnf**

**[mysqld\_multi]** #启用多实例

**mysqld = /usr/local/mysql/bin/mysqld\_safe** #指定进程文件路径

**mysqladmin = /usr/local/mysql/bin/mysqladmin** #指定管理命令路径

**user = root** #指定进程用户

**[mysqld1]** #实例进程名称

**port=3307** #端口号

**datadir=/dir1** #数据库目录 ，要手动创建

**socket=/dir1/mysqld1.sock** #指定sock文件的路径和名称

**pid-file=/dir1/mysqld1.pid** #进程pid号文件位置

**log-error=/dir1/mysqld1.err** #错误日志位置

**[mysqld2]**

**port=3308**

**datadir=/dir2**

**socket=/dir2/mysqld2.sock**

**pid-file=/dir2/mysqld2.pid**

**log-error=/dir2/mysqld2.err**

mysql58 ~]# mkdir /dir1 /dir2 #创建配置文件中需要的目录

**3.3 管理多实例**

**3.3.1 启动服务**

**mysqld\_multi start 实例编号**

mysql58 ~]# mysqld\_multi start 1

#启动实例1,注意注意提示信息最后一行的**初始密码**

mysql58 ~]# netstat -antulp | grep :3307 #启动后查看端口

tcp6 0 0 :::3307 :::\* LISTEN 23724/mysqld

mysql58 ~]# ls /dir1 #查看自定义的数据库目录下文件,注意3个文件

auto.cnf ib\_logfile0 mysql **mysqld1.sock** sys

ib\_buffer\_pool ib\_logfile1 **mysqld1.err** mysqld1.sock.lock

ibdata1 ibtmp1 **mysqld1.pid** performance\_schema

**3.3.2 客户端访问**

**本机连接-使用初始密码连接**

**mysql -uroot -p”初始密码” -S sock文件**

mysql58 ~]# mysql -uroot -p'dxA%f/Fav74g' -S /dir1/mysqld1.sock

**修改本机登录密码**

**mysql> alter user user() identified by “新密码”;**

58 mysql> alter user user() identified by "123qqq...A"

58 mysql> exit

mysql58 ~]# mysql -uroot -p123qqq...A -S /dir1/mysqld1.sock

再次登录数据库使用新密码登录

按3.3.1和3.3.2 启动实例2,并查看自定义的端口和数据库目录下文件

**3.3.3 停止服务(停止后.sock和.pid文件消失)**

**mysqld\_multi --user=root --password=”新密码” stop 实例编号**

mysql58 ~]# mysqld\_multi --user=root --password="123qqq...A" stop 1 #停止实例1

mysql58 ~]# ls /dir1 #查看,此时少了.sock和.pid文件

**3.4 多实例排错**

3.4.1 修改配置文件:vim /etc/my.cnf

3.4.2 杀进程:killall -9 mysqld

3.4.3 删除自定义的数据库目录内的所有内容:rm -rf /dir1/\*

**3.5 多实例的使用**

在50上登录58的实例1

**3.5.1修改配置文件**

mysql58 ~]# vim /etc/my.cnf

[mysqld1] #实例1

**server\_id=1**

**log\_bin=db1**

**3.5.2 修改配置文件后,重启实例1**

mysql58 ~]# mysqld\_multi --user=root --password="123qqq...A" stop 1 #关闭实例1

mysql58 ~]# mysqld\_multi start 1 #启动实例1

**3.5.3 58上登录实例1并授权连接用户及创建库.表并写入数据**

mysql58 ~]# mysql -uroot -p123qqq...A -S /dir1/mysqld1.sock

58-1 mysql> select user,host from mysql.user;

58-1 mysql> grant all on \*.\* to admin@"%" identified

by "123qqq...A";

**3.5.4 50上用连接用户登录58上的数据库并查看**

mysql50 ~]# mysql -h192.168.4.58 -uadmin -p123qqq...A -P3307

50 mysql> show grants; #查看自身权限

50 mysql> select \* from bbsdb.a; #查看库.表及数据