mysql\_07 读写分离

1. **读写分离**

1.1读写分离概述

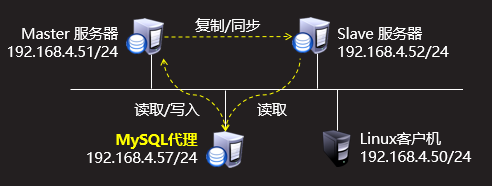
1.2部署maxscale服务

用4台虚拟机，如图-1所示。其中192.168.4.51和192.168.4.52，分别提供读、写服务，

均衡流量，通过主从复制保持数据一致性，由MySQL代理192.168.4.57面向客户端提供服务，

收到SQL写请求时，交给主服务器处理，收到SQL读请求时，交给从服务器处理。在客户机

192.168.4.50测试配置。

1.2.0 搭建MySQL一主一从同步结构

[1] 配置租服务器192.168.4.51

]#vim /etc/my.cnf

[mysqld]

serverid=51 #指定服务器ID号

log-bin=master51 #启用binlog日志,并指定文件名前缀

:wq

[root@host51]#systemc tl restart mysqld #重启数据库服务

[2]主服务器授权用户,并查看binlog日志

]#mysql -uroot -p123456

mysql> grant all on \*.\* to ‘repluser’@”%” identified by ‘123456’ #用户授权

mysql> show master status; #查看授权用户的日志名和偏移量

[3] 配置从服务器192.168.4.52

]# vim /etc/my.cnf

[mysqld]

server\_id=52 //指定服务器ID号，不要与Master的相同

:wq

]# systemctl restart mysql

[4]配置从服务器192.168.4.52，指定主服务器信息，日志文件、偏移位置（参考MASTER上

的状态输出）

]# mysql -uroot -p123456

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;

Query OK, 0 rows affected, 2 warnings (0.01 sec)

mysql> start slave;

Query OK, 0 rows affected (0.01 sec)

mysql> show slave status\G;

[5] 测试配置，在主服务器本机创建数据库 aa库

]# mysql –uroot –p123456

mysql> create database aa;

Query OK, 1 row affected (0.00 sec)

mysql> show databases;

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

| Database |

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

| information\_schema |

| aa |

| mysql |

| performance\_schema |

| sys |

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

5 rows in set (0.00 sec)

[6] 从服务器上查看，有aa库

mysql> show databases;

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

| Database |

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

| information\_schema |

| aa |

| mysql |

| performance\_schema |

| sys |

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

5 rows in set (0.00 sec)

1.2.1安装软件(代理服务器)

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

配置文件 /etc/maxscale.cnf

命令 maxadmin maxbinlogcheck maxpasswd

maxavrocheck maxkeys maxscale

日志 /var/log/maxscale

服务程序 maxscale

1.2.2修改主配置文件

备份 cp /etc/maxscale.cnf /opt/maxscale.cng.max #备份配置文件

]#vim /etc/maxscale.cnf

[maxscale] #定义线程个数

threads=auto #启动时默认线程数量设成自动

[server1] #定义数据库服务器

type=server

address=192.168.4.51 #master主机IP地址

port=3306

protocol=MySQLBackend

[server2]

type=server

address=19.18.4.52 #slave主机ip地址

port=3306

protocol=MySQLBackend

[MySQL Monitor] #监视mysql服务器

type=monitor

module=mysqlmon

servers=server1,server2 #监视的主机

user=maxscalemon #用户名

passwd=123qqq...A #密码

monitor\_interval=10000

注释掉只读服务

[Read-Write Service] #配置读写分离服务

type=service

router=readwritesplit

servers=server1,server2

user=maxscalerouter

passwd=123qqq...A

max\_slave\_connections=100%

注释只读服务监听端口号

[Read-Write Listener] #读写服务监听端口

type=listener

service=Read-Write Service

protocol=MySQLClient

port=4006

[MaxAdmin Listener] #管理服务监听端口号

type=listener

service=MaxAdmin Service

protocol=maxscaled

socket=default

port=4016

:wq

sed -i '/#/d' /etc/maxscale.cnf #删除被注释的行

1.2.3 配置数据库服务器 server1 server2

根据maxscale.cnf文件配置，在主/从服务器上添加对应的授权用户，因为2台数据库服务器是主从同步结构，只在主数据库服务器添加用户即可，从服务器会自动同步

maxscalemon:

mysql> grant replication slave,replication client on \*.\* to maxscalemon@"%"

identified by "123qqq...A";

maxscalerouter

mysql> grant select on \*.\* to maxscalerouter@"%" identified by "123qqq...A";

1.2.4 启动服务

maxscale -f /etc/maxscale.cnf #启动服务

killall -9 maxscale.cnf #杀死进程

1.2.5 查看监控信息

ss -nutpl | grep maxscale:

ls /var/log/maxcale 日志文件目录

vim /var/log/maxscale/maxscale.log 查看日志内容

1.2.6 测试配置

maxadmin -uadmin -pmariadb -P4016

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

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

1.2.7测试配置

1.2.7.1 在主服务器51,添加客户端访问数据的连接用户

host51]#mysql -uroot -p123qqq...A

mysql> create database db7

mysql> create table db7.a(id int);

mysql> grant select ,insert on db7.\* plj99”%” idnetifity by “123qqq...A”;

host52]#mysql -uroot -p123qqq...A -e ‘show grants for plj99@”%”;’

1.2.7.2 在客户端连接服务器57访问数据

client50]#mysql -h192.168.4.57 -P4006 -uplj99 -p123qqq...A

mysql> show grants;

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

mysql> select \* from db7.a;

1.2.7.3 测试数据读写分离

在从服务器52本机向表里插入记录

]# mysql -uroot -p123qqq...A

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

mysql> select \* from db7.a;

在主服务器51本机查看表记录

mysql> select \* from db7.a;

在客户端50连接服务器57 访问数据

client50]# mysql -h192.168.4.57 -P4006 -uplj99 -p123qqq...A

1. **Mysql多实例**

准备主机 克隆1台新的虚拟机配置IP地址192.18.4.58/24

软件:mysql-5.7.20-linux-glibc2.12-x86\_64.tar.gz

2.1mysql多实例介绍

在一台物理主机上运行多个数据库服务

为什么要使用多实例

\*节约运维成本

\*提高硬件利用率

2.2配置MySQL多实例,步骤如下

1. 安装软件

yum -y install libaio

useradd mysql

tar -zxvf mysql-5.7.20-linux-glibc2.12-x86\_64.tar.gz

mv mysql-5.7.20-linux-glibc2.12-x86\_64 /usr/local/mysql

PATH=/usr/local/mysql/bin/:$PATH

vim /etc/bashrc

export PATH=/usr/local/mysql/bin/:$PATH

:wq

1. 创建并编辑主配置文件

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,X表示实例编号

port = 3307 //端口号

datadir = /dir1 //数据库目录,需要手动创建

socket = /dir1/mysqld.sock //指定sock文件的路径及名称

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/mysqld.sock

pid-file = /dir2/mysqld2.pid

log-error = /dir2/mysqld2.err

1. 创建数据目录

mkdir /dir1

mkdir /dir2

1. 启动所实例服务

mysqld\_multi start 实例编号 //启动服务

mysqld\_multi start 1

mysqld\_multi --user=root --password=密码 stop 实例编号 //停止服务

1. 客户端访问

mysql -uroot -p ‘初始密码’ -S sock文件

mysql -uroot -p'3v)fkk:)ZP&f' -S /dir1/mysqld1.sock

mysql> alter user root@”localhost” identified by “新密码”;

alter user root@"localhost" identified by "123456";

5.实例使用

[root@host58 ~]# vim /etc/my.cnf

server\_id=1

log\_bin=db1

:wq

[root@host58 ~]# mysqld\_multi --user=root --password=123456 stop 1

[root@host58 ~]# mysqld\_multi start 1

[root@host58 ~]# mysql -uroot -p123456 -S /dir1/mysqld1.sock

mysql> show master status;

mysql> grant select,update,insert on bbsdb.\* to admin@"%" identified by "123456";

mysql> create database bbsdb;

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

[root@host50 ~]# mysql -u admin -p123456 -h192.168.4.58 -P3307

mysql> insert into bbsdb.a values(120);