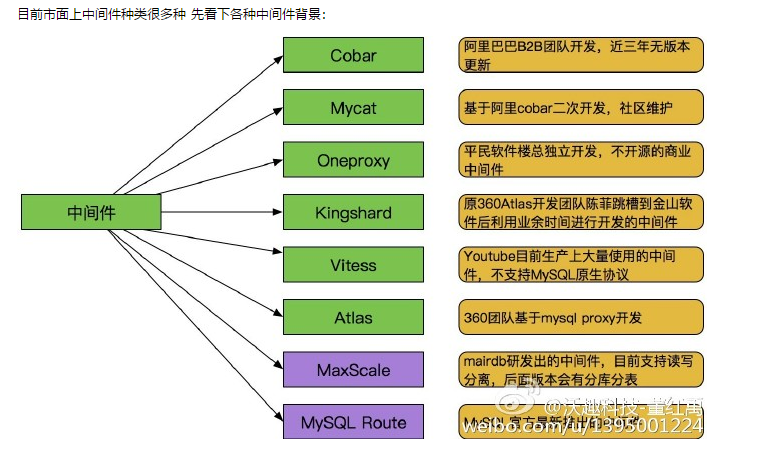
MySQL中间件



环境：rhel7.5操作系统，后台MySQL一主一从，测试各个中间件

[root@master ~]# yum -y install mariadb-server

[root@slave ~]# yum -y install mariadb-server

#配置master主库

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

[root@master ~]# cat /etc/my.cnf | grep -v ^$ | grep -v ^#

[mysqld]

server\_id=1

log-bin=master

character\_set\_server="utf8"

innodb\_file\_per\_table=1

datadir=/var/lib/mysql

socket=/var/lib/mysql/mysql.sock

symbolic-links=0

[mysqld\_safe]

log-error=/var/log/mariadb/mariadb.log

pid-file=/var/run/mariadb/mariadb.pid

!includedir /etc/my.cnf.d

[root@master ~]# systemctl start mariadb

[root@master ~]# mysql

MariaDB [(none)]> grant all on \*.\* to root@'%' identified by '123456' with grant option;

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant all on \*.\* to root@'localhost' identified by '123456' with grant option;

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant replication slave on \*.\* to repluser@'%' identified by '123456';

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> reset master;

Query OK, 0 rows affected (0.07 sec)

MariaDB [(none)]> show master status \G

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

File: master.000001

Position: 245

Binlog\_Do\_DB:

Binlog\_Ignore\_DB:

1 row in set (0.00 sec)

#配置slave从库

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

[root@slave ~]# cat /etc/my.cnf | grep -v ^$ | grep -v ^#

[mysqld]

server\_id=2

log-bin=slave

character\_set\_server="utf8"

innodb\_file\_per\_table=1

datadir=/var/lib/mysql

socket=/var/lib/mysql/mysql.sock

symbolic-links=0

[mysqld\_safe]

log-error=/var/log/mariadb/mariadb.log

pid-file=/var/run/mariadb/mariadb.pid

!includedir /etc/my.cnf.d

[root@slave ~]# systemctl start mariadb

[root@slave ~]# mysql

MariaDB [(none)]> grant all on \*.\* to root@'%' identified by '123456' with grant option;

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant all on \*.\* to root@'localhost' identified by '123456' with grant option;

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> reset slave;

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> change master to

-> master\_host="192.168.4.1",

-> master\_user="repluser",

-> master\_password='123456',

-> master\_log\_file="master.000001",

-> master\_log\_pos=245;

Query OK, 0 rows affected (0.10 sec)

MariaDB [(none)]> start slave;

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> show slave status \G

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

#完成后台主从同步

#测试主从同步

[root@master ~]# mysql -uroot -p123456 -e "create database mydb default character set utf8;"

[root@master ~]# mysql -uroot -p123456 -e "show databases;"

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

| Database |

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

| information\_schema |

| mydb |

| mysql |

| performance\_schema |

| test |

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

[root@slave ~]# mysql -uroot -p123456 -e "show databases;"

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

| Database |

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

| information\_schema |

| mydb |

| mysql |

| performance\_schema |

| test |

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

##测试中间件

1、MySQL-Proxy

#MySQL-Proxy是有lua语言开发的，所以需要安装lua环境

[root@proxy ~]# yum -y install lua

[root@proxy ~]# ls

mysql-proxy-0.8.5-linux-glibc2.3-x86-64bit.tar.gz

[root@proxy ~]# tar -xf mysql-proxy-0.8.5-linux-glibc2.3-x86-64bit.tar.gz

[root@proxy ~]# mv mysql-proxy-0.8.5-linux-glibc2.3-x86-64bit /usr/local/mysql\_proxy

[root@proxy ~]# cd /usr/local/mysql\_proxy/

[root@proxy mysql\_proxy]# mkdir lua

[root@proxy mysql\_proxy]# mkdir logs

[root@proxy mysql\_proxy]# mkdir etc

[root@proxy mysql\_proxy]# ls

bin etc include lib libexec licenses logs lua share

[root@proxy mysql\_proxy]# cp share/doc/mysql-proxy/rw-splitting.lua ./lua/

[root@proxy mysql\_proxy]# cp share/doc/mysql-proxy/admin-sql.lua ./lua/

#编写配置文件

[root@proxy mysql\_proxy]# vim etc/mysql\_proxy.cnf

[root@proxy mysql\_proxy]# cat etc/mysql\_proxy.cnf

[mysql-proxy]

#运行代理的用户

user = root

#mysql\_proxy连接后端服务器的用户名

admin-username = mysql\_proxy\_user

#mysql\_proxy连接后端服务器的用户的密码

admin\_password = mysql\_proxy\_pass

#mysql-proxy监听的ip和端口，默认4040

proxy-address = 0.0.0.0:3306

#定义后端只读服务器

proxy-read-only-backend-addresses = 192.168.4.2

#定义后端读写服务器

proxy-backend-addresses = 192.168.4.1

#定义读写分离配置文件

proxy-lua-script = /usr/local/mysql\_proxy/lua/rw-splitting.lua

#定义管理脚本路径

proxy-lus-script = /usr/local/mysql\_proxy/lua/admin-sql.lua

#定义日志存放路径和日志名

log-file = /usr/local/mysql\_proxy/logs/mysql\_proxy.log

#定义日志等级

log-level = debug

#mysql\_proxy是否以守护进程的方式运行

daemon = true

#mysql\_proxy崩溃后是否尝试自动重启服务

keepalive = true

#修改读写分离配置文件

[root@proxy mysql\_proxy]# vim lua/rw-splitting.lua

[root@proxy mysql\_proxy]# sed -n '38,46p' lua/rw-splitting.lua

if not proxy.global.config.rwsplit then

proxy.global.config.rwsplit = {

--默认超过4个连接开始读写分离

min\_idle\_connections = 1,

--默认为8，修改为1

max\_idle\_connections = 1,

is\_debug = false

}

#主库授权mysql\_proxy\_user用户

[root@master ~]# mysql -uroot -p123456

MariaDB [(none)]> grant all on \*.\* to 'mysql\_proxy\_user'@'%' identified by 'mysql\_proxy\_pass';

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit

Bye

#启动mysql-proxy服务

[root@proxy mysql\_proxy]# pwd

/usr/local/mysql\_proxy

[root@proxy mysql\_proxy]# ./bin/mysql-proxy --defaults-file=etc/mysql\_proxy.cnf

2019-03-19 02:01:26: (critical) mysql-proxy-cli.c:326: loading config from 'etc/mysql\_proxy.cnf' failed: permissions of etc/mysql\_proxy.cnf aren't secure (0660 or stricter required)

2019-03-19 02:01:26: (message) Initiating shutdown, requested from mysql-proxy-cli.c:328

2019-03-19 02:01:26: (message) shutting down normally, exit code is: 1

#此处报错配置文件权限过大，修改为600

[root@proxy mysql\_proxy]# ll etc/mysql\_proxy.cnf

-rw-r--r-- 1 root root 1137 3月 19 01:51 etc/mysql\_proxy.cnf

[root@proxy mysql\_proxy]# chmod 0600 etc/mysql\_proxy.cnf

[root@proxy mysql\_proxy]# ./bin/mysql-proxy --defaults-file=etc/mysql\_proxy.cnf

[root@proxy mysql\_proxy]# netstat -antpu | grep mysql

tcp 0 0 0.0.0.0:3306 0.0.0.0:\* LISTEN 1780/mysql-proxy

#测试MySQL-Proxy

[root@room8pc205 soft]# mycli -h192.168.4.11 -umysql\_proxy\_user -pmysql\_proxy\_pass

(none)> select @@hostname;

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

| @@hostname |

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

| slave.tedu.cn |

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

1 row in set

Time: 0.035s

(none)>

(none)> use mydb;

You are now connected to database "mydb" as user "mysql\_proxy\_user"

Time: 0.004s

mariadb mysql\_proxy\_user@192.168.4.11:mydb> create table mytb (id int(11));

Query OK, 0 rows affected

Time: 0.040s

mariadb mysql\_proxy\_user@192.168.4.11:mydb> insert into mytb values(1);

Query OK, 1 row affected

Time: 0.018s

mariadb mysql\_proxy\_user@192.168.4.11:mydb> select \* from mytb;

+----+

| id |

+----+

| 1 |

+----+

[root@master ~]# mysql -uroot -p123456 -e "select \* from mydb.mytb;"

+------+

| id |

+------+

| 1 |

+------+

[root@slave ~]# mysql -uroot -p123456 -e "select \* from mydb.mytb;"

+------+

| id |

+------+

| 1 |

+------+

#MySQL-Proxy读写分离测试完成

2、Maxscale

[root@maxscale ~]# yum -y localinstall maxscale-2.1.2-1.rhel.7.x86\_64.rpm

[root@maxscale ~]# vim /etc/maxscale.cnf

[root@maxscale ~]# cat /etc/maxscale.cnf

[maxscale]

#定义运行线程的数量，auto为等于CPU核心数量

threads=auto

#定义数据库服务器

[server1]

type=server

#数据库1的IP地址

address=192.168.4.1

port=3306

#定义为后端数据库

protocol=MySQLBackend

[server2]

type=server

address=192.168.4.2

port=3306

protocol=MySQLBackend

#定义监控数据库服务器

[MySQL Monitor]

type=monitor

module=mysqlmon

#填写上边定义的后台数据库服务器编号，不能写真实ip地址

servers=server1,server2

#监视数据库的连接用户，密码

user=scalemon

passwd=123456

#监视频率，但是毫秒

monitor\_interval=10000

#定义读写分离

[Read-Write Service]

type=service

router=readwritesplit

servers=server1,server2

#验证连接代理服务器是访问数据库的用户是否存在

user=maxscaled

passwd=123456

#所有的slave都支持select

max\_slave\_connections=100%

#定义管理服务

[MaxAdmin Service]

type=service

router=cli

#定义读写服务使用的端口号

[Read-Write Listener]

type=listener

service=Read-Write Service

protocol=MySQLClient

port=4006

#定义管理服务使用的端口号

[MaxAdmin Listener]

type=listener

service=MaxAdmin Service

protocol=maxscaled

socket=default

port=4099

#主库授权maxscale所需用户

[root@master ~]# mysql -uroot -p123456

#授权监控服务用户，监控后台服务器状态

MariaDB [(none)]> grant replication slave,replication client on \*.\* to scalemon@'%' identified by '123456';

Query OK, 0 rows affected (0.01 sec)

#授权检测用户，用于测试连接maxscale的用户是或否存在于真实数据库

MariaDB [(none)]> grant select on mysql.\* to maxscaled@'%' identified by '123456';

Query OK, 0 rows affected (0.01 sec)

#授权测试用户

MariaDB [(none)]> grant all on \*.\* to stu@'%' identified by '123456';

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> select user,host from mysql.user where user in ("scalemon", "maxscaled");

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

| user | host |

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

| maxscaled | % |

| scalemon | % |

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

2 rows in set (0.00 sec)

MariaDB [(none)]> exit

Bye

#从库查看这两个用户是否同步

[root@slave ~]# mysql -uroot -p123456

MariaDB [(none)]> select user,host from mysql.user where user in ("scalemon","maxscaled");

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

| user | host |

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

| maxscaled | % |

| scalemon | % |

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

2 rows in set (0.01 sec)

MariaDB [(none)]> exit

Bye

#启动maxscale服务

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

[root@maxscale ~]# netstat -antpu | grep max

tcp 0 0 192.168.4.12:36550 192.168.4.1:3306 ESTABLISHED 1921/maxscale

tcp 0 0 192.168.4.12:59172 192.168.4.2:3306 ESTABLISHED 1921/maxscale

tcp6 0 0 :::4099 :::\* LISTEN 1921/maxscale

tcp6 0 0 :::4006 :::\* LISTEN 1921/maxscale

#从管理服务端口登录maxscale控制台，查看情况

[root@maxscale ~]# maxadmin -P4099 -uadmin -pmariadb

MaxScale> list servers

Servers.

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

Server | Address | Port | Connections | Status

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

server1 | 192.168.4.1 | 3306 | 0 | Master, Running

server2 | 192.168.4.2 | 3306 | 0 | Slave, Running

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

MaxScale> list services

Services.

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

Service Name | Router Module | #Users | Total Sessions | Backend databases

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

Read-Write Service | readwritesplit | 1 | 1 | server1, server2

MaxAdmin Service | cli | 3 | 3 |

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

MaxScale> exit

#测试读写分离

[root@room8pc205 soft]# mysql -h192.168.4.12 -P4006 -ustu -p123456 -e "select @@hostname"

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

| @@hostname |

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

| slave.tedu.cn |

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

[root@room8pc205 soft]# mysql -h192.168.4.12 -P4006 -ustu -p123456 -e "insert into mydb.mytb values(2);"

[root@room8pc205 soft]# mysql -h192.168.4.12 -P4006 -ustu -p123456 -e "select \* from mydb.mytb;"

+------+

| id |

+------+

| 1 |

| 2 |

+------+

3、mycat

[root@mycat ~]# yum -y install java-1.8.0-openjdk-devel

[root@mycat ~]# tar -xf Mycat-server-1.6-RELEASE-20161028204710-linux.tar.gz

[root@mycat ~]# mv mycat/ /usr/local/

[root@mycat ~]# cd /usr/local/mycat/

[root@mycat mycat]# ls

bin catlet conf lib logs version.txt

#修改mycat配置文件

[root@mycat mycat]# cp conf/server.xml conf/server.xml.bak

[root@mycat mycat]# vim conf/server.xml

[root@mycat mycat]# sed -n '80,89p' conf/server.xml

<user name="root">

<property name="password">123456</property>

<property name="schemas">vmydb</property>

</user>

<user name="user">

<property name="password">user</property>

<property name="schemas">vmydb</property>

<property name="readOnly">true</property>

</user>

#此处定义的两个user是连接mycat的用户，root具有读写权限，user仅有读权限

#这两个用户可以不存在真实的数据库中

#定义逻辑库名为vmydb

[root@mycat mycat]# cp conf/schema.xml conf/schema.xml.bak

[root@mycat mycat]# vim conf/schema.xml

[root@mycat mycat]# cat conf/schema.xml

<?xml version="1.0"?>

<!DOCTYPE mycat:schema SYSTEM "schema.dtd">

<mycat:schema xmlns:mycat="http://io.mycat/">

<schema name="vmydb" checkSQLschema="false" sqlMaxLimit="100" dataNode="dn1"></schema>

<dataNode name="dn1" dataHost="localhost1" database="mydb" />

<dataHost name="localhost1" maxCon="1000" minCon="10" balance="3"

writeType="0" dbType="mysql" dbDriver="native" switchType="1" slaveThreshold="100">

<heartbeat>select user()</heartbeat>

<writeHost host="master" url="192.168.4.1:3306" user="root" password="123456">

<readHost host="slave" url="192.168.4.2:3306" user="readuser" password="123456" />

</writeHost>

</dataHost>

</mycat:schema>

#schema：逻辑数据库

#dataNode：节点

#dataHost：节点对应的读库写库的地址和连接

#balance指的负载均衡类型，目前的取值有4种：

#balance="0", 不开启读写分离机制，所有读操作都发送到当前可用的writeHost上。

#balance="1"，全部的readHost与stand by writeHost参与select语句的负载均衡

#balance="2"，所有读操作都随机的在writeHost、readhost上分发。

#balance="3"，所有读请求随机的分发到wiriterHost对应的readhost执行，writerHost不负担读压力

#switchType指的是切换的模式，目前的取值也有4种：

#switchType='-1' 表示不自动切换

#switchType='1' 默认值，表示自动切换

#switchType='2' 基于MySQL主从同步的状态决定是否切换,心跳语句为 show slave status

#switchType='3' 基于MySQL galary cluster的切换机制（适合集群）（1.4.1），心跳语句为 show status like 'wsrep%'

#WriteType参数设置：

#writeType=“0”, 所有写操作都发送到可用的writeHost上。

#writeType=“1”，所有写操作都随机的发送到readHost。

#writeType=“2”，所有写操作都随机的在writeHost、readhost分上发。

#主库授权只读用户

[root@master ~]# mysql -uroot -p123456

MariaDB [(none)]> grant select on \*.\* to readuser@'%' identified by '123456';

Query OK, 0 rows affected (0.02 sec)

MariaDB [(none)]> flush privileges;

Query OK, 0 rows affected (0.01 sec)

MariaDB [(none)]> select user, host from mysql.user where user='readuser';

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

| user | host |

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

| readuser | % |

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

1 row in set (0.00 sec)

#启动mycat服务

[root@mycat mycat]# ./bin/mycat start

Starting Mycat-server...

[root@mycat mycat]# jps

2547 Jps

2517 WrapperSimpleApp

[root@mycat mycat]# netstat -antpu | grep java

tcp 0 0 127.0.0.1:32000 0.0.0.0:\* LISTEN 2517/java

tcp6 0 0 :::1984 :::\* LISTEN 2517/java

tcp6 0 0 :::8066 :::\* LISTEN 2517/java

tcp6 0 0 :::38180 :::\* LISTEN 2517/java

tcp6 0 0 :::9066 :::\* LISTEN 2517/java

tcp6 0 0 :::39340 :::\* LISTEN 2517/java

tcp6 0 0 192.168.4.13:56522 192.168.4.2:3306 ESTABLISHED 2517/java

tcp6 0 0 192.168.4.13:54756 192.168.4.1:3306 ESTABLISHED 2517/java

tcp6 0 0 192.168.4.13:54752 192.168.4.1:3306 ESTABLISHED 2517/java

tcp6 0 0 192.168.4.13:54762 192.168.4.1:3306 ESTABLISHED 2517/java

tcp6 0 0 192.168.4.13:54746 192.168.4.1:3306 ESTABLISHED 2517/java

tcp6 0 0 192.168.4.13:56528 192.168.4.2:3306 ESTABLISHED 2517/java

tcp6 0 0 127.0.0.1:31000 127.0.0.1:32000 ESTABLISHED 2517/java

tcp6 0 0 192.168.4.13:54754 192.168.4.1:3306 ESTABLISHED 2517/java

tcp6 0 0 192.168.4.13:54750 192.168.4.1:3306 ESTABLISHED 2517/java

tcp6 0 0 192.168.4.13:56520 192.168.4.2:3306 ESTABLISHED 2517/java

tcp6 0 0 192.168.4.13:54758 192.168.4.1:3306 ESTABLISHED 2517/java

tcp6 0 0 192.168.4.13:56524 192.168.4.2:3306 ESTABLISHED 2517/java

tcp6 0 0 192.168.4.13:54744 192.168.4.1:3306 ESTABLISHED 2517/java

tcp6 0 0 192.168.4.13:54760 192.168.4.1:3306 ESTABLISHED 2517/java

tcp6 0 0 192.168.4.13:54748 192.168.4.1:3306 ESTABLISHED 2517/java

tcp6 0 0 192.168.4.13:56526 192.168.4.2:3306 ESTABLISHED 2517/java

#测试读写分离

[root@room8pc205 soft]# mysql -h192.168.4.13 -P8066 -uroot -p123456 -e "select @@hostname"

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

| @@hostname |

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

| slave.tedu.cn |

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

[root@room8pc205 soft]# mysql -h192.168.4.13 -P8066 -uuser -puser -e "select @@hostname"

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

| @@hostname |

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

| slave.tedu.cn |

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

[root@room8pc205 soft]# mysql -h192.168.4.13 -P8066 -uroot -p123456 -e "show databases;"

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

| DATABASE |

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

| vmydb |

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

[root@room8pc205 soft]# mysql -h192.168.4.13 -P8066 -uroot -p123456

MySQL [(none)]> show databases;

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

| DATABASE |

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

| vmydb |

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

1 row in set (0.00 sec)

MySQL [(none)]> use vmydb

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

MySQL [vmydb]> insert into mytb values(3);

Query OK, 1 row affected (0.04 sec)

MySQL [vmydb]> select \* from mytb;

+------+

| id |

+------+

| 1 |

| 2 |

| 3 |

+------+

3 rows in set (0.01 sec)

MySQL [vmydb]> exit

Bye

4、MySQL-Route

[root@router ~]# ls

mysql-router-2.0.4-linux-glibc2.12-x86-64bit.tar.gz

[root@router ~]# tar -xf mysql-router-2.0.4-linux-glibc2.12-x86-64bit.tar.gz

[root@router ~]# mv mysql-router-2.0.4-linux-glibc2.12-x86-64bit /usr/local/mysql\_router

[root@router ~]# cd /usr/local/mysql\_router/

[root@router mysql\_router]# mkdir etc

[root@router mysql\_router]# mkdir logs

[root@router mysql\_router]# mkdir data

[root@router mysql\_router]# ls

bin data etc include lib logs run share

[root@router mysql\_router]# cp share/doc/mysqlrouter/sample\_mysqlrouter.ini etc/mysqlrouter.cnf

[root@router mysql\_router]# cp share/doc/mysqlrouter/sample\_mysqlrouter.init /etc/init.d/mysqlrouterd

[root@router mysql\_router]# useradd -s /sbin/nologin mysql

[root@router mysql\_router]# id mysql

uid=1000(mysql) gid=1000(mysql) 组=1000(mysql)

[root@router mysql\_router]# chown -R mysql.mysql /usr/local/mysql\_router/

[root@router mysql\_router]# pwd

/usr/local/mysql\_router

[root@router mysql\_router]# vim etc/mysqlrouter.cnf

[root@router mysql\_router]# cat etc/mysqlrouter.cnf

#默认配置

[default]

#定义mysql-router家目录

home\_dir = /usr/local/mysql\_router

#日志路径

logging\_folder = {home\_dir}/logs/

#定义插件路径

plugin\_folder = {home\_dir}/lib/mysqlrouter

#定义配置文件路径

config\_folder = {home\_dir}/etc/mysqlrouter.cnf

#定义运行参数路径

runtime\_folder = {home\_dir}/run/mysqlroute.pid

#定义数据目录

data\_folder = {home\_dir}/data/

#定义日志相关

[logger]

#定义日志级别

level = INFO

#定义高可用（主库，能执行写操作）

[routing:basic\_failover]

#监听地址

bind\_address = 0.0.0.0

#写操作监听端口

bind\_port = 7000

#工作模式

mode = read-write

#能执行写操作的服务器列表（主库）

destinations = 192.168.4.1:3306

#定义负载均衡（从库，能执行读操作）

[routing:load\_balance]

#监听地址

bind\_address = 0.0.0.0

#读操作监听端口

bind\_port = 6000

#工作模式

mode = read-only

#能执行读操作的服务器列表（从库）

destinations = 192.168.4.2:3306

#保活检测

[keepalive]

interval = 60

[root@router mysql\_router]# ./bin/mysqlrouter -c etc/mysqlrouter.cnf &

[1] 2549

[root@router mysql\_router]# Logging to /usr/local/mysql\_router/logs/mysqlrouter.log

#验证读写分离

[root@router mysql\_router]# netstat -antpu | grep mysql

tcp 0 0 0.0.0.0:6000 0.0.0.0:\* LISTEN 2549/./bin/mysqlrou

tcp 0 0 0.0.0.0:7000 0.0.0.0:\* LISTEN 2549/./bin/mysqlrou

[root@router mysql\_router]# ls logs/

mysqlrouter.log

[root@room8pc205 soft]# mysql -h192.168.4.14 -P7000 -uroot -p123456 -e "select @@hostname"

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

| @@hostname |

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

| master.tedu.cn |

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

[root@room8pc205 soft]# mysql -h192.168.4.14 -P6000 -uroot -p123456 -e "select @@hostname"

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

| @@hostname |

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

| slave.tedu.cn |

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

[root@room8pc205 soft]# mysql -h192.168.4.14 -P7000 -uroot -p123456 -e "insert into mydb.mytb values(4)"

[root@room8pc205 soft]# mysql -h192.168.4.14 -P6000 -uroot -p123456 -e "select \* from mydb.mytb"

+------+

| id |

+------+

| 1 |

| 2 |

| 3 |

| 4 |

|  |
| --- |
|  |

#性能测试

[root@room8pc205 tsql]# cat sesql.py

#!/usr/bin/env python3

import sys

import subprocess

import threading

def test\_se(host, port, user, password):

rc = subprocess.call(

'mysql -h%s -P%s -u%s -p%s -e "show databases;" &> /dev/null' % (host,port,user,password),

shell=True

)

if rc:

print('no')

else:

print('yes')

if \_\_name\_\_ == '\_\_main\_\_':

host = sys.argv[1]

port = sys.argv[2]

user = sys.argv[3]

password = sys.argv[4]

times = int(sys.argv[5])

for i in range(times):

t = threading.Thread(target=test\_se(host,port,user,password))

t.start()