# Mysql中间件Atlas实现读写分离

官方文档：<https://github.com/Qihoo360/Atlas/wiki>

主机地址：10.10.182.53

10.10.185.73

VIP:

软件版本：Atlas-2.2.1.el6.x86\_64.rpm

1. 安装Atlas

#rpm –i Atlas-2.2.1.el6.x86\_64.rpm

修改配置文档：

#vim /usr/local/mysql-proxy/conf/mysql-proxy.cnf

安装mysql客户端

#yum install –y mysql-devel

二、配置启动脚本：

#!/bin/sh

#

# mysql-proxy This script starts and stops the mysql-proxy daemon

#

# chkconfig: - 78 30

# processname: mysql-proxy

# description: mysql-proxy is a proxy daemon to mysql

# config: /usr/local/mysql-proxy/conf/mysql-proxy.cnf

# pidfile: /usr/local/mysql-proxy/log/mysql-proxy.pid

#

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

DAEMON="/usr/local/mysql-proxy/bin/mysql-proxy"

CONFIGFILE="/usr/local/mysql-proxy/conf/mysql-proxy.cnf"

PIDFILE="/usr/local/mysql-proxy/log/mysql-proxy.pid"

LOCKFILE="/var/lock/subsys/mysql-proxy"

PROG=`basename $DAEMON`

RETVAL=0

start() {

echo -n $"Starting ${PROG}......"

[ -x $DAEMON ] || exit 5

[ -f $CONFIGFILE ] || exit 6

${DAEMON} --defaults-file=${CONFIGFILE} || echo -n "${PROG} already running"

RETVAL=$?

echo

[[ $RETVAL -eq 0 ]] && touch $LOCKFILE

return $RETVAL

}

stop() {

echo -n $"Stopping ${PROG}......"

if [[ `ps aux | grep bin/mysql-proxy | grep -v grep | wc -l` -gt 0 ]]; then

kill -TERM `ps -A -oppid,pid,cmd | grep bin/mysql-proxy | grep -v grep | awk '{print $2}'`

fi

RETVAL=$?

echo

[[ $RETVAL -eq 0 ]] && rm -f $LOCKFILE $PIDFILE

return $RETVAL

}

restart() {

stop

sleep 1

start

}

case "$1" in

start)

start

;;

stop)

stop

;;

restart)

restart

;;

condrestart)

[[ -e $LOCKFILE ]] && restart

;;

\*)

echo "Usage: $0 {start|stop|restart|condrestart}"

RETVAL=1

;;

Esac

#chmod +x /etc/init.d/mysql-proxy

#service mysql-proxy start

#chkconfig mysql-proxy on

三、配置后端Mysql服务器

给atlas代理服务器授权

GRANT SELECT, INSERT, UPDATE, INDEX, EXECUTE ON \*.\* TO 'jjjr'@'10.10.185.73' IDENTIFIED BY PASSWORD '\*775982FE51E0E9FEC651EA05BFB2372D80682392';

执行命令：mysql -h127.0.0.1 -P1234 -u用户名 -p密码，如果能连上则证明Atlas初步测试正常，可以再尝试发几条SQL语句看看执行结果是否正确。

进入Atlas的管理界面的命令：mysql -h127.0.0.1 -P2345 -uuser -ppwd，进入后执行:select \* from help;查看管理DB的各类命令。

四、配置keepalived

在两台atlas服务器上执行操作

#yum install –y keepalived

#vim /etc/keepalived/keepalived

global\_defs {

notification\_email {

ops@17money.com

}

notification\_email\_from mail@17money.com

smtp\_server 127.0.0.1

smtp\_connect\_timeout 30

router\_id dbproxy1

}

vrrp\_script chk\_mysql\_proxy\_health {

script "/etc/keepalived/scripts/keepalived\_check\_mysql\_proxy.sh"

interval 1

weight -2

}

vrrp\_instance VI\_1 {

state BACKUP

interface eth0

virtual\_router\_id 51

priority 90

advert\_int 1

smtp\_alert

authentication {

auth\_type PASS

auth\_pass 123456

}

virtual\_ipaddress {

172.16.40.6 dev eth0

}

track\_script {

chk\_mysql\_proxy\_health

}

}

状态检测脚本

#vim /etc/keepalived/scripts/keepalived\_check\_mysql\_proxy.sh

#!/bin/sh

PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/bin:/usr/local/sbin

if [[ `pgrep mysql-proxy | wc -l` -eq 0 ]]; then

/sbin/service mysql-proxy start && sleep 5

[[ -z `pgrep mysql-proxy` ]] && /sbin/service keepalived stop

Fi

五、性能测试

利用sysbench测试通过Atlas转发SQL请求和直连DB发送SQL请求这两种情况下， Atlas和Mysql系统的两项数据指标：QPS和每条SQL请求平均处理时间。 通过sysbench发送三类SQL请求：select，update，insert。 每类请求都是单独发送。具体的操作过程如下所示：

执行下面的命令测试sysbench连接Atlas

sysbench --test=oltp \

--num-threads=1 \

--max-requests=80000 \

--oltp-test-mode=nontrx \

--db-driver=mysql \

--mysql-db=my\_db \

--mysql-host=192.168.0.12 \

--mysql-port=1234 \

--mysql-user=buck \

--mysql-password=buck \

--oltp-nontrx-mode=select \

--db-ps-mode=disable \

Prepare|run|cleanup

上述命令是sysbench执行80000次随机select操作，这80000次操作都是非事务的。 通过修改 --oltp-nontrx-mode 选项，可以执行update和insert操作。 通过修改 --num-threads 参数，可以调整并发测试线程的个数。

sysbench创建表的语句是：

CREATE TABLE sbtest (

id int(10) unsigned NOT NULL AUTO\_INCREMENT,

k int(10) unsigned NOT NULL DEFAULT '0',

c char(120) NOT NULL DEFAULT '',

pad char(60) NOT NULL DEFAULT '',

PRIMARY KEY (id),

KEY k (k)

) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=latin1