

mysql 配置 mysql-proxy 读写分离

mysql 配置读写分离

在这里你再也不用担心高手省略了大段代码了，因为我也是新手。

下面开整

先安装 lua

Lua 是一个小巧的脚本语言。Lua 由标准 C 编写而成，代码简洁优美，几乎在所有操作系统和平台上都可以编译，运行。

一个完整的 Lua 解释器不过 200k，在目前所有脚本引擎中，Lua 的速度是最快的。这一切都决定了 Lua 是作为嵌入式脚本的最佳选择。

先安装可能的依赖包

1. [root@bogon ~]# yum -y install gcc* gcc-c++* autoconf* automake* zlib* libxml* ncurses-devel* libmcrypt* libtool* flex* pkgconfig*
2. libevent* glib* readline-devel
3. [root@bogon ~]# mkdir /opt/install/
4. [root@bogon ~]# cd /opt/install/
5. [root@bogon install]# wget http://www.lua.org/ftp/lua-5.1.4.tar.gz
6. [root@bogon install]# tar zxvf lua-5.2.3.tar.gz
7. [root@bogon install]# cd lua-5.2.3
8. [root@bogon install]# make linux

此时如果出现

lua.c:67:31: fatal error: readline/readline.h: No such file or directory

说明缺少 libreadline-dev 依赖包

1. [root@bogon install]# make install
2. [root@bogon install]# export LUA_CFLAGS="-I/usr/local/include" LUA_LIBS="-L/usr/local/lib -llua -ldl" LDFLAGS="-lm"

这行代码我也不知道啥作用，好像是的是指定环境变量。

下面开始安装 mysql-proxy

安装之前先看一下自己机器是否是 64 位系统

查看 centos 内核的版本

1. [root@bogon install]# cat /etc/issue
查看 linux 版本
1. [root@bogon install]# cat /proc/version
2. [root@bogon install]# lsb_release -a

如果以上命令都没有出现 64 字眼说明是 32 位系统，这里主要是确定你将要下载的 mysql-proxy 版本应该是哪个。

下载地址：http://mysql.cdpa.nsysu.edu.tw/Downloads/MySQL-Proxy/

1. [root@bogon install]# cd /opt/install
2. [root@bogon install]# wget http://mysql.cdpa.nsysu.edu.tw/Downloads/MySQL-Proxy/mysql-proxy-0.8.4-linux-rhel5-x86-32bit.tar.gz
3. [root@bogon install]# tar zxvf mysql-proxy-0.8.4-linux-rhel5-x86-32bit.tar.gz
4. [root@bogon install]# mkdir /opt/mysql-proxy/
5. [root@bogon install]# cp /opt/install/mysql-proxy-0.8.4-linux-rhel5-x86-32bit/* /opt/mysql-proxy/
6. [root@bogon install]# cd /opt/mysql-proxy/
7. [root@bogon install]# export PATH=\$PATH:/opt/mysql-proxy/bin/

编辑 mysql-proxy 配置文件

1. [root@bogon install]# cp doc/mysql-proxy/rw-splitting.lua ./
2. [root@bogon install]# vi rw-splitting.lua

找到：

1. if not proxy.global.config.rwsplit then
2. proxy.global.config.rwsplit = {
3. min_idle_connections = 4,
4. max_idle_connections = 8,
5. is_debug = false
6. }
7. end

这里的 4、8 是指定链接数，大家可以在做试验时调整为 1。

ok，保存，退出。

1. [root@bogon install]# mysql-proxy --proxy-read-only-backend-addresses=192.168.23.132:3306 --proxy-backend-addresses=192.168.2.131:3306 --proxy-lua-script=/opt/mysql-proxy/share/rw-splitting.lua&

这里的 192.168.23.131 与 192.168.23.132 是主从数据库。参见 <http://blog.csdn.net/e421083458/article/details/19574049>

以上命令后缀追加&表示为后端任务。

1. [root@bogon install]# bg

将正在执行命令隐藏到后端

1. [root@bogon install]# jobs

查看正在后端运行的命令

1. [richard@bogon ~]\$ fg 1

将后端命令显示到前台

打开 4040 端口防火墙端口，因为 mysql-proxy 监听端口是 4040 再也不是 3306 了。正好做以区分。

1. [root@bogon install]# /sbin/iptables -I INPUT -p tcp --dport 4040 -j ACCEPT
2. [root@bogon install]# /etc/rc.d/init.d/iptables save
3. [root@bogon install]# service iptables save

查看监听情况：

1. [root@bogon install]# netstat -tln | grep mysql-proxy
2. tcp 0 0 *:yo-main *: LISTEN 13364/mysql-proxy

yo-main 很奇怪吧，对我也奇怪，不管它了，下面我们测试一下功能吧。

刚刚我们提到的 192.168.23.131 与 192.168.23.132 两台服务器，131 是主，132 是从，131 是读写，132 是只读。mysql-proxy 的 IP 是 192.168.23.130

创建用于读写分离的数据库连接用户

在 131 服务器上操作以下命令：

1. [root@bogon ~]# mysql -uroot -p123456
2. mysql> grant all on *.* to 'proxy1'@'192.168.23.130' identified by '123456';

由于我们设置了 131 与 132 的主从复制，所以此时 132 也执行了以上命令，我们可以确定一下。

在 132 服务上操作以下命令：

1. mysql> select user,password,host from mysql.user;
2. +-----+-----+-----+-----+
3. | user | password | host |
4. +-----+-----+-----+-----+
5. | root | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 | localhost |
6. | root | | bogon |
7. | root | | 127.0.0.1 |
8. | proxy1 | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 | 192.168.23.130 |
9. +-----+-----+-----+-----+
10. 4 rows in set (0.00 sec)

没错结果存在。

但是我们要做的试验是测试读写分离，所以要让这两台服务器有误差才行。所以暂停 132 的从服务：

1. mysql> slave stop;

下面我们在 131 服务器上操作以下命令：

1. mysql> insert into first_tb values (007,'first');
2. mysql> insert into first_tb values (110,'second');

最后再到 132 上确认一下数据是否有误差，下面试验开始。

在 131 上打开 4 个终端，依次敲以下的命令：

1. [root@bogon ~]# mysql -uproxy1 -p'123456' -h192.168.23.130 -P4040
2. mysql> select * from first_db.first_tb;

你会发现结果会有不一样，那证明配置试验成功。也可以多此尝试以上两行代码直到得到结果。

试验结束记得将 slave start，方法参照 <http://blog.csdn.net/e421083458/article/details/19574049>

追加内容：设置 mysql-proxy 服务脚本。

1. #!/bin/sh
2. # mysql-proxy This script starts and stops the mysql-proxy daemon
3. # chkconfig: - 78 30
4. # processname: mysql-proxy
5. # description: mysql-proxy is a proxy daemon to mysql
6. # Source function library.
7. . /etc/rc.d/init.d/functions
8. #PROXY_PATH=/usr/local/bin
9. PROXY_PATH=/opt/mysql-proxy/bin
10. prog="mysql-proxy"
11. # Source networking configuration.
12. . /etc/sysconfig/network
13. # Check that networking is up.
14. [\${NETWORKING} = "no"] && exit 0
15. # Set default mysql-proxy configuration.
16. #PROXY_OPTIONS="--daemon"
17. PROXY_OPTIONS="--proxy-read-only-backend-addresses=192.168.23.132:3306 --proxy-backend-addresses=192.168.23.131:3306 --proxy-lua-script=/opt/mysql-proxy/share/rw-splitting.lua"
18. PROXY_PID=/opt/mysql-proxy/run/mysql-proxy.pid
- 19.
20. # Source mysql-proxy configuration.
21. if [-f /etc/sysconfig/mysql-proxy]; then
22. . /etc/sysconfig/mysql-proxy
23. fi
- 24.
25. PATH=\$PATH:/usr/bin:/usr/local/bin:\$PROXY_PATH

```
26.# By default it's all good
27.RETVAL=0
28.# See how we were called.
29.case "$1" in
30. start)
31.     # Start daemon.
32.     echo -n $"Starting $prog: "
33.     $NICELEVEL $PROXY_PATH/mysql-proxy $PROXY_OPTIONS --daemon --pid-
        file=$PROXY_PID --user=root --log-level=debug --log-file=/opt/mysql-proxy/log/mysql-
        proxy.log
34.     RETVAL=$?
35.     echo
36.     if [ $RETVAL = 0 ]; then
37.         touch /var/lock/subsys/mysql-proxy]
38.         echo "ok"
39.     fi
40.     ;;
41. stop)
42.     # Stop daemons.
43.     echo -n $"Stopping $prog: "
44.     killproc $prog
45.     RETVAL=$?
46.     echo
47.     if [ $RETVAL = 0 ]; then
48.         rm -f /var/lock/subsys/mysql-proxy
49.         rm -f $PROXY_PID
50.     fi
51.     ;;
52. restart)
53.     $0 stop
54.     sleep 3
55.     $0 start
56.     ;;
57. condrestart)
58.     [ -e /var/lock/subsys/mysql-proxy ] && $0 restart
59.     ;;
60. status)
61.     status mysql-proxy
62.     RETVAL=$?
63.     ;;
64. *)
65.     echo "Usage: $0 {start|stop|restart|status|condrestart}"
```

```
66.    RETVAL=1
67.    ;;
68.esac
69.exit $RETVAL
```

启动脚本：

```
[root@bogon init.d]# /opt/mysql-proxy/init.d/mysql-proxy start
```