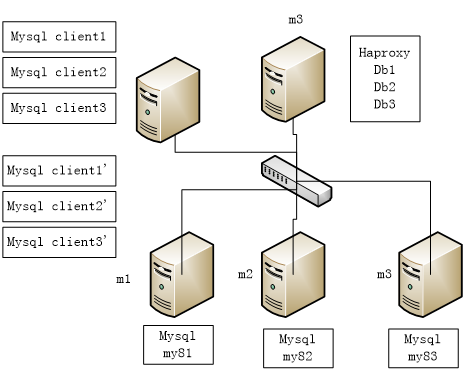
以下为初步版，

本文主要描述haproxy的安装及简单配置过程。并且应用这个简单的配置做mysql的负载均衡，同时haproxy的配置文件发生变化时我们如何重新启动haproxy程序，并且确定haproxy程序重新启动后对原来已经进行的连接回话的影响情况。

**0. 准备工作**

为了进行本文的研究，我们需要3台[vmware](http://www.iigrowing.cn/tag/vmware)的虚拟机，在每一台虚拟机上，安装centos程序，安装mysql程序， 每一台机器都安装mysql程序。如下图：



m1:192.168.65.81   m2:192.168.65.82    m3:192.168.65.83

**1. Haproxy程序的安装**

下载并解压缩

wget http://haproxy.1wt.eu/download/1.4/src/haproxy-1.4.9.tar.gz

tar xzvf haproxy-1.4.9.tar.gz

进入解压缩目录，编译并安装程序

make TARGET=linux26 PREFIX=/work/apps/haproxy install

创建配置文件

cd /work/apps/haproxy

mkdir conf

vi conf/haproxy.cfg

编辑文件内容如下：

global

    log 127.0.0.1 local0 info #[err warning info debug]

    maxconn 4096

    user haproxy

    group haproxy

    daemon

    nbproc 1

    pidfile /work/apps/haproxy/logs/haproxy.pid

defaults

    maxconn 2000

    contimeout 5000

    clitimeout 30000

    srvtimeout 30000

frontend db\_read

    bind :3307

    default\_backend cluster\_db\_read

backend cluster\_db\_read

    mode tcp

    balance roundrobin

    server db01\_slave 192.168.65.81:3306

    server db02\_slave 192.168.65.82:3306

    server db03\_slave 192.168.65.83:3306

listen admin\_stats

    bind 0.0.0.0:1080

    mode http

    log 127.0.0.1 local0 err

    stats uri /admin?stats

创建日志目录

mkdir logs

启动haproxy

cd /work/apps/haproxy/sbin

./haproxy -f ../conf/haproxy.conf

启动后， 通过netstat -ntlp 查看3307端口是否已经启动了。

**2. 启动mysql程序，创建数据库**

登陆m1服务器，执行下面命令

mysql -h localhost -u root -p

用下面命令创建数据库

create database my81

创建用户并授权:grant all privileges on \*.\* to mq@’%’ identified by ‘mq’ WITH GRANT OPTION;FLUSH PRIVILEGES;

登陆m2服务器， 执行如下命令

mysql -h localhost -u root -p

用下面命令创建数据库

create database my82

创建用户并授权:grant all privileges on \*.\* to mq@’%’ identified by ‘mq’ WITH GRANT OPTION;FLUSH PRIVILEGES;

登陆m3服务器， 执行如下命令

mysql -h localhost -u root -p

用下面命令创建数据库

create database my83

创建用户并授权:grant all privileges on \*.\* to mq@’%’ identified by ‘mq’ WITH GRANT OPTION;FLUSH PRIVILEGES;

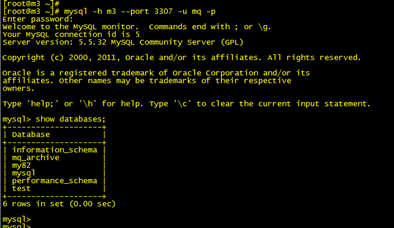
这样，我们在3个服务器上有了不同的数据库， 我们仅仅需要 show databases命令，我们就可以知道我们连接的是哪个数据库了。

**3. 通过haproxy连接mysql数据库**

执行如下命令， 连接数据库

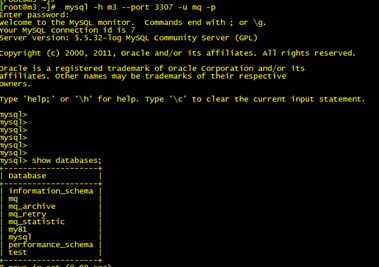
mysql -h 192.168.65.83 –port 3307 -u mq -p

然后输入相关密码，最后我们输入：show databases命令，查看相关情况

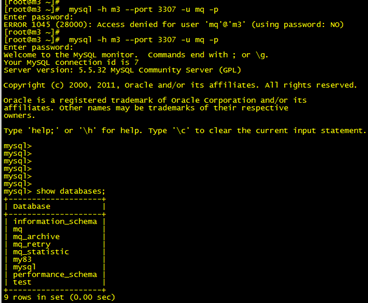


通过数据库我们了解到，我们连接的是具有my82的数据库

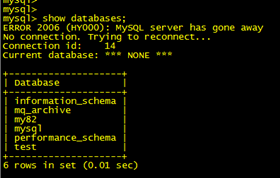
重复上面命令，如下：



在重复上面命令



过10分钟后在重新测试，发现如下



发现数据库连接已经断开了，经过阅读haproxy文档

http://haproxy.1wt.eu/download/1.4/doc/configuration.txt

最后确定是，配置文件中超时时间造成，haproxy的超时。

因此修改相关配置为如下：

defaults

    maxconn 2000

    contimeout 5000000

    clitimeout 3000000

    srvtimeout 3000000

  重新启动haproxy程序，重复上面操作。 过10分钟后，发现mysql的数据库连接还在继续连接没有中断。

结论：通过haproxy，实现了通过一个ip地址，自动连接到多个数据库服务器，简化了相关配置。

**4. 修改配置文件**

修改配置文件为如下：

backend cluster\_db\_read

    mode tcp

    balance roundrobin

    option httpchk

    server db01\_slave 192.168.65.81:3306

    server db02\_slave 192.168.65.82:3306

#    server db03\_slave 192.168.65.83:3306

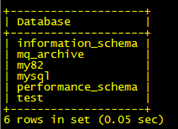
其余的配置文件不变情况

重新启动haproxy程序

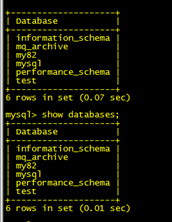
./haproxy -f ../conf/haproxy.cfg  -sf $(cat /work/apps/haproxy/logs/haproxy.pid)

结论：在重新配置完成haproxy后，可以通过上述命令，重新启动haproxy连接， 对原有的工作无影响

**5. 检查重启haproxy后原来连接的有效与否**



通过检查发现原来的连接还在，没有中断tcp连接



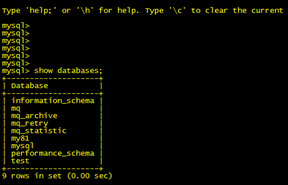
类似检查其他的连接都是连接的，没有中断的现象。

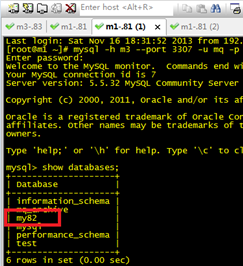
**6. 通过haproxy重新连接数据库**

我们重新开始全新的控制台程序，

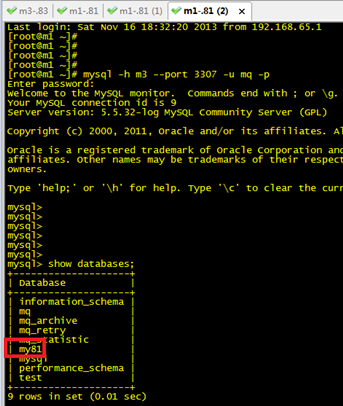
mysql -h 192.168.65.83 –port 3307 -u mq -p

然后输入相关密码，最后我们输入：show databases命令，查看相关情况





重复上面命令， 有如下结果



通过上图，我们可以确定， 我们配置文件中只存在两个服务器了， 因此请求被分配到同一台机器。

**通过以上，我们可以了解到，在合适的配置下， haproxy可以提供mysql的负载均衡，并且支持动态配置修改，并且可以热启动， 启动过程中， 原来的相关连接继续保持连接状态知道超时等**

来源： <<http://www.iigrowing.cn/haproxy-an-zhuang-he-dong-tai-pei-zhi-geng-gai-ce-shi.html>>