

HAProxy使用说明

Author : 董旗宇

Email: dqylyln@gmail.com

Blog: <http://blog.sina.com.cn/dqylyln>

Date: 2009-10-16

简介 :

软件负载均衡一般通过两种方式来实现：基于操作系统的软负载实现和基于第三方应用的软负载实现。LVS就是基于Linux操作系统实现的一种软负载，HAProxy就是开源的并且基于第三应用实现的软负载。

HAProxy相比LVS的使用要简单很多，功能方面也很丰富。当前，HAProxy支持两种主要的代理模式："tcp"也即4层（大多用于邮件服务器、内部协议通信服务器等），和7层（HTTP）。在4层模式下，HAProxy仅在客户端和服务端之间转发双向流量。7层模式下，HAProxy会分析协议，并且能通过允许、拒绝、交换、增加、修改或者删除请求(request)或者回应(response)里指定内容来控制协议，这种操作要基于特定规则。

详情可以在HAProxy官方网站(<http://haproxy.1wt.eu>)可以下载配置说明文档(configuration.txt)和架构文件(architecture.txt)作为参考。

HAProxy的安装

在“<http://haproxy.1wt.eu/#down>”下载HAProxy的源码包。这里以1.3.20版本，安装在目录/home/admin/haproxy为例

1. 解压下载的源码包

```
tar -xzf haproxy-1.3.20.tar.gz
```

2. 编译安装

进入解压后的源码目录

```
make TARGET=linux26 PREFIX=/home/admin/haproxy install
```

（注意，TARGET后面根据本机操作系统内核版本来填写，PREFIX是要安装到的目录）

安装完成后会在安装目录下生成doc，sbin，share三个文件夹。解压的源码文件也可以删除掉。

3. 创建配置文件

在安装目录下建立配置文件目录

```
mkdir conf
```

在新创建的目录下创建配置文件

```
vi haproxy.cfg
```

在创建的haproxy.cfg文件中填入
global

```
log 127.0.0.1 local0 info #[err warning info debug]
maxconn 4096
user admin
group admin
```

```

daemon
nbproc 1
pidfile /home/admin/haproxy/logs/haproxy.pid
defaults
maxconn 2000
contimeout 5000
clitimeout 30000
srvtimeout 30000
listen admin_stats
bind 0.0.0.0:1080
mode http
log 127.0.0.1 local0 err
stats uri /admin?stats

```

这个配置是能让HAProxy工作起来的最小配置

4.启动HAProxy

进入安装目录下的sbin目录

启动HAProxy

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

通过浏览器访问“<http://xxx.xxx.xxx.xxx:1080/admin?stats>”(xxx为IP),看到管理页面

HAProxy version 1.3.20, released 2009/08/09

Statistics Report for pid 29835

> General process information

pid = 29835 (process #1, nbproc = 1)
 uptime = 0d 0h00m03s
 system limits : memmax = unlimited ; ulimit-n = 4011
 maxsock = 4011 ; maxconn = 2000 ; maxpipes = 0
 current conns = 1 ; current pipes = 0/0
 Running tasks : 1/1

active UP backup UP
 active UP, going down backup UP, going down
 active DOWN, going up backup DOWN, going up
 active or backup DOWN not checked
 Note: UP with load-balancing disabled is reported as "NOLE".

| | Queue | | | Session rate | | | Sessions | | | | Bytes | | Denied | | Errors | | Warnings | | | | |
|----------|-------|-----|-------|--------------|-----|-------|----------|-----|-------|-------|-------|-------|--------|-----|--------|-----|----------|------|------|-------|--|
| | Cur | Max | Limit | Cur | Max | Limit | Cur | Max | Limit | Total | LbTot | In | Out | Req | Resp | Req | Conn | Resp | Retr | Redis | |
| Frontend | | | | 9 | 9 | - | 1 | 1 | 2 000 | 11 | | 4 395 | 32 932 | 0 | 0 | 0 | | | | | |
| Backend | 0 | 0 | | 4 | 4 | | 0 | 1 | 2 000 | 5 | 0 | 4 395 | 32 932 | 0 | 0 | | 5 | 0 | 0 | 0 | |

5.修改haproxy启动文件权限

(如果你是用ROOT用户启动HAProxy，这步可以省略，这个步骤是为了能让非ROOT用户也能通过HAProxy监听80端口，因为LIUNIX的80端口一定要使用ROOT账户启动)

使用root账户进入sbin目录

更改haproxy的所有者为root

chown root:root haproxy

给haproxy增加S权限

chmod u+s haproxy

6.配置日志

创建日志目录

使用非root账户这里是admin账户创建日志文件

mkdir /home/admin/haproxy/logs

修改操作系统的日志配置

HAProxy可以收集本机及其他后端服务器日志，但是需要在HAProxy和操作系统上作一些配置。

使用root账户首先修改/etc/sysconfig/syslog文件，将SYSLOGD_OPTIONS="-m 0" 修改为SYSLOGD_OPTIONS="-m 0 -r -x"，支持收集远程服务器日志。

然后修改/etc/syslog.conf，增加如下语句：

local0.* /home/admin/haproxy/logs/haproxy.log // haproxy.log地址代表了需要存储日志的地址，其中local0这个级别要和haproxy.cfg配置的log级别一样

执行service syslog restart，重新启动系统日志器

更改日志文件用户

chown admin:admin /home/admin/haproxy/logs/haproxy.log

7.创建启动/停止脚本（如果你认为现在的启动脚本已经够方便了，这步可以省略）

为了方便以后启动或停止，所以编写一个启动脚本

在sbin目录下创建hactl.sh

vi hactl.sh

填入一下内容

```
#!/bin/sh#
```

```
cd `dirname $0`/..
```

```
BASE_DIR=""`pwd`"
```

```
ARGV="$@"
```

```
start()
```

```
{
```

```
    echo "START HAPoxy SERVERS"
```

```
    $BASE_DIR/sbin/haproxy -f $BASE_DIR/conf/haproxy.cfg
```

```
}
```

```
stop()
```

```
{
```

```
    #haproxy.pid的路径是haproxy.cfg文件中配置的
```

```
    echo "STOP HAPoxy Listen"
```

```
    kill -TTOU $(cat $BASE_DIR/logs/haproxy.pid)
```

```
    echo "STOP HAPoxy process"
```

```
    kill -USR1 $(cat $BASE_DIR/logs/haproxy.pid)
```

```
}
```

```
case $ARGV in
```

```
    start)
```

```
        start
```

```
        ERROR=$?
```

```
        ;;
```

```
    stop)
```

```
        stop
```

```
        ERROR=$?
```

```
        ;;
```

```
    restart)
```

```
        stop
```

```
        start
```

```
        ERROR=$?
```

```
;;
*)
echo "hactl.sh [start|restart|stop]"
esac
exit $ERROR
```

保存后，给hactl.sh增加可执行权限

```
chmod 755 hactl.sh
```

这样就可以使用./ hactl.sh [start|restart|stop] 的方式启动，关闭，重启了，如果你使用的目录和例子中不一样的话，需要自己对应的修改下以上脚本的目录地址

经过以上1-6步骤以后HAProxy的安装已经结束，之后就是详细的配置下haproxy.cfg文件了

HAProxy的配置

HAProxy配置中分成五部分内容，当然这些组件不是必选的，可以根据需要选择部分作为配置。

global：参数是进程级的，通常和操作系统（OS）相关。这些参数一般只设置一次，如果配置无误，就不需要再次配置进行修改

defaults：配置默认参数的，这些参数可以被利用配置到frontend，backend，listen组件

frontend：接收请求的前端虚拟节点，Frontend可以根据规则直接指定具体使用后端的 backend(可动态选择)。

backend：后端服务集群的配置，是真实的服务器，一个Backend对应一个或者多个实体服务器。

listen：Frontend和Backend的组合物。

下面是HAProxy的一些常用的配置，这个配置是用来说明HAProxy的一些常用功能的配置，具体详细配置请查看安装目录下的doc目录下的文档文件，或者到”

<http://cn.haproxy.org/>”

下载中文配置说明文档

配置文件例子：

```
global
#全局的日志配置 其中日志级别是[err warning info debug]
#local0 是日志设备，必须为如下24种标准syslog设备的一种:
#kern user mail daemon auth syslog lpr news
#uucp cron auth2 ftp ntp audit alert cron2
#local0 local1 local2 local3 local4 local5 local6 local7
#但是之前在/etc/syslog.conf文件中定义的是local0所以
#这里也是用local0
log 127.0.0.1 local0 info #[err warning info debug]
#最大连接数
maxconn 4096
#用户
user admin
```

```
#组
group admin
#使HAProxy进程进入后台运行。这是推荐的运行模式
daemon
#创建4个进程进入daemon模式运行。此参数要求将运行模式设置为"daemon"
nbproc 4
#将所有进程的pid写入文件<pidfile>启动进程的用户必须有权限访问此文件。
pidfile /home/admin/haproxy/logs/haproxy.pid
```

defaults

```
#默认的模式mode { tcp|http|health } , tcp是4层 , http是7层 , health只会返回OK
mode http
#采用http日志格式
option httplog
#三次连接失败就认为是服务器不可用 , 也可以通过后面设置
retries 3
#如果cookie写入了serverId而客户端不会刷新cookie ,
#当serverId对应的服务器挂掉后 , 强制定向到其他健康的服务器
option redispatch
#当服务器负载很高的时候 , 自动结束掉当前队列处理比较久的链接
option abortonclose
#默认的最大连接数
maxconn 4096
#连接超时
timeout 5000
#客户端超时
clitimeout 30000
#服务器超时
srvtimeout 30000
#心跳检测超时
timeout check 2000
```

#注：一些参数值为时间，比如说timeout。时间值通常单位为毫秒(ms)，但是也可以通过加#后缀，来使用其他的单位。

```
#- us : microseconds. 1 microsecond = 1/1000000 second
#- ms : milliseconds. 1 millisecond = 1/1000 second. This is the default.
#- s : seconds. 1s = 1000ms
#- m : minutes. 1m = 60s = 60000ms
#- h : hours. 1h = 60m = 3600s = 3600000ms
#- d : days. 1d = 24h = 1440m = 86400s = 86400000ms
```

#####统计页面配置#####

```
listen admin_stats
#监听端口
bind 0.0.0.0:1080
#http的7层模式
mode http
```

```

#日志设置
log 127.0.0.1 local0 err #[err warning info debug]
#统计页面自动刷新时间
stats refresh 30s
#统计页面url
stats uri /admin?stats
#统计页面密码框上提示文本
stats realm Gemini\ Haproxy
#统计页面用户名和密码设置
stats auth admin:admin
stats auth admin1:admin1
#隐藏统计页面上HAProxy的版本信息
stats hide-version

#####网站检测listen定义#####
listen site_status
    bind 0.0.0.0:1081
    mode http
    log 127.0.0.1 local0 err #[err warning info debug]
    #网站健康检测URL，用来检测HAProxy管理的网站是否可以用，正常返回200，不正
    #常返回500
    monitor-uri /site_status
    #定义网站down时的策略
    #当挂在负载均衡上的指定backend的中有效机器数小于1台时返回true
    acl site_dead nbsrv(denali_server) lt 1
    acl site_dead nbsrv(tm_server) lt
1
    acl site_dead nbsrv(mms_server) lt 1
    #当满足策略的时候返回500
    monitor fail if site_dead
    #如果192.168.0.252或者192.168.0.31这两天机器挂了
    #认为网站挂了，这时候返回500，判断标准是如果mode是
    #http返回200认为是正常的，如果mode是tcp认为端口畅通是好的
    monitor-net 192.168.0.252/31

#####https的配置方法#####
listen login_https_server
    #绑定HTTPS的443端口
    bind 0.0.0.0:443
    #https必须使用tcp模式
    mode tcp
    log global
    balance roundrobin
    option httpchk GET /member/login.jhtml HTTP/1.1\r\nHost:login.daily.taobao.net
    ##回送给server的端口也必须是443
    server vm94f.sqa 192.168.212.94:443 check port 80 inter 6000 rise 3 fall 3

```

```
server v215120.sqa 192.168.215.120:443 check port 80 inter 6000 rise 3 fall 3
```

```
#####frontend配置#####
```

```
frontend http_80_in
```

```
#监听端口
```

```
bind 0.0.0.0:80
```

```
#http的7层模式
```

```
mode http
```

```
#应用全局的日志配置
```

```
log global
```

```
#启用http的log
```

```
option httplog
```

```
#每次请求完毕后主动关闭http通道，HA-Proxy不支持keep-alive模式
```

```
option httpclose
```

```
#如果后端服务器需要获得客户端的真实IP需要配置次参数，将可以从Http Header中
```

```
#获得客户端IP
```

```
option forwardfor
```

```
#####HAProxy的日志记录内容配置#####
```

```
capture request header Host len 40
```

```
capture request header Content-Length len 10
```

```
capture request header Referer len 200
```

```
capture response header Server len 40
```

```
capture response header Content-Length len 10
```

```
capture response header Cache-Control len 8
```

```
#####acl策略定义#####
```

```
#如果请求的域名满足正则表达式返回true -i是忽略大小写
```

```
acl denali_policy hdr_reg(host) -i
```

```
^(www.gemini.taobao.net|my.gemini.taobao.net|auction1.gemini.taobao.net)$
```

```
#如果请求域名满足trade.gemini.taobao.net 返回 true -i是忽略大小写
```

```
acl tm_policy hdr_dom(host) -i trade.gemini.taobao.net
```

```
##在请求url中包含sip_apiname=，则此控制策略返回true,否则为false
```

```
acl invalid_req url_sub -i sip_apiname=
```

```
##在请求url中存在timetask作为部分地址路径，则此控制策略返回true,否则返回false
```

```
acl timetask_req url_dir -i timetask
```

```
#当请求的header中Content-length等于0时返回 true
```

```
acl missing_cl hdr_cnt(Content-length) eq 0
```

```
#####acl策略匹配相应#####
```

```
##当请求中header中Content-length等于0 阻止请求返回403
```

```
block if missing_cl
```

##block表示阻止请求，返回403错误，当前表示如果不满足策略invalid_req，或者满足策略timetask_req，则阻止请求。

```
block if !invalid_req || timetask_req
```

#当满足denali_policy的策略时使用denali_server的backend

```
use_backend denali_server          if denali_policy
```

#当满足tm_policy的策略时使用tm_server的backend

```
use_backend tm_server              if tm_policy
```

#reqisetbe关键字定义，根据定义的关键字选择backend

```
reqisetbe    ^Host:\ img          dynamic
```

```
reqisetbe    ^[\ ]*\ /(img|css)/  dynamic
```

```
reqisetbe    ^[\ ]*\ /admin/stats  stats
```

#以上都不满足的时候使用默认mms_server的backend

```
default_backend mms_server
```

#HAProxy错误页面设置

```
errorfile 400 /home/admin/haproxy/errorfiles/400.http
```

```
errorfile 403 /home/admin/haproxy/errorfiles/403.http
```

```
errorfile 408 /home/admin/haproxy/errorfiles/408.http
```

```
errorfile 500 /home/admin/haproxy/errorfiles/500.http
```

```
errorfile 502 /home/admin/haproxy/errorfiles/502.http
```

```
errorfile 503 /home/admin/haproxy/errorfiles/503.http
```

```
errorfile 504 /home/admin/haproxy/errorfiles/504.http
```

#####backend的设置#####

```
backend mms_server
```

#http的7层模式

```
mode http
```

#负载均衡的方式，roundrobin平均方式

```
balance roundrobin
```

#允许插入serverid到cookie中，serverid后面可以定义

```
cookie SERVERID
```

#心跳检测的URL,HTTP/1.1\r\nHost:XXXX,指定了心跳检测HTTP的版本，XXX为检测时请求

#服务器的request中的域名是什么，这个在应用的检测URL对应的功能有对域名依赖的话需

#要设置

```
option httpchk GET /member/login.jhtml HTTP/1.1\r\nHost:member1.gemini.taobao.net
```

#服务器定义，cookie 1表示serverid为1，check inter 1500 是检测心跳频率

#rise 3是3次正确认为服务器可用，fall 3是3次失败认为服务器不可用，weight代表权重

```
server mms1 10.1.5.134:80 cookie 1 check inter 1500 rise 3 fall 3 weight 1
```

```
server mms2 10.1.6.118:80 cookie 2 check inter 1500 rise 3 fall 3 weight 2
```

```
backend denali_server
```



```

mode http
#负载均衡的方式，source根据客户端IP进行哈希的方式
balance source
#但设置了backup的时候，默认第一个backup会优先，设置option allbackups后
#所有备份服务器权重一样
option allbackups
#心跳检测URL设置
option httpchk GET /mytaobao/home/my_taobao.jhtml HTTP/
1.1\r\nHost:my.gemini.taobao.net
#可以根据机器的性能不同，不使用默认的连接数配置而使用自己的特殊的连接数配置
#如minconn 10 maxconn 20
server denlai1 10.1.5.114:80 minconn 4 maxconn 12 check inter 1500 rise 3 fall 3
server denlai2 10.1.6.104:80 minconn 10 maxconn 20 check inter 1500 rise 3 fall 3
#备份机器配置，正常情况下备机不会使用，当主机全部服务器都down的时候备机启用
server dnali-back1 10.1.7.114:80 check backup inter 1500 rise 3 fall 3
server dnali-back2 10.1.7.114:80 check backup inter 1500 rise 3 fall 3

backend tm_server
mode http
#负载均衡的方式，leastconn根据服务器当前的请求数，取当前请求数最少的服务器
balance leastconn
option httpchk GET /trade/itemlist/prepayCard.htm HTTP/
1.1\r\nHost:trade.gemini.taobao.net
server tm1 10.1.5.115:80 check inter 1500 rise 3 fall 3
server tm2 10.1.6.105:80 check inter 1500 rise 3 fall 3

#####reqisetbe自定义关键字匹配backend部分#####
backend dynamic
mode http
balance source
option httpchk GET /welcome.html HTTP/1.1\r\nHost:www.taobao.net
server denlai1 10.3.5.114:80 check inter 1500 rise 3 fall 3
server denlai2 10.4.6.104:80 check inter 1500 rise 3 fall 3
backend stats
mode http
balance source
option httpchk GET /welcome.html HTTP/1.1\r\nHost:www.taobao.net
server denlai1 10.5.5.114:80 check inter 1500 rise 3 fall 3
server denlai2 10.6.6.104:80 check inter 1500 rise 3 fall 3

```

HA-Proxy的压力测试结果

简单的压力测试采用Apache ab，500并发用户，10w的请求总数。

| | 总耗时(s) | TPS(#/sec) |
|----------|--------|------------|
| HA-2Node | 21.387 | 4675.61 |

HA-5Node

27.371

3653.37

HA-2Node为配置了两个节点作为后段的服务节点，HA-5Node为配置了5个节点作为后端的服务处理节点。上面结果看到2个节点的HA比5个节点的速度来的快。可以确定的是

HAProxy的性能是跟随backend的数量增加而下降，所以当backend节点过多时可以考虑通过拆分到多台HAProxy来提高性能。

相关资源

HAProxy官方网站 <http://haproxy.1wt.eu>

HAProxy 中文网站 <http://cn.haproxy.org/>