# Linux HAProxy负载均衡

## 简介

HAProxy是免费、高效、可靠的高可用及负载均衡解决方案。该软件非常适合处理高负载站点的七层数据请求。HAProxy的工作模式使其可以非常容易且安全的集成到我们现有的站点架构中。使用类似的代理软件还可以对外屏蔽内部的真实Web服务器，防止内部服务器遭受外部攻击。

终端用户通过访问HAProxy代理服务器获得站点页面，而代理服务器在收到客户端请求后会根据自身规则将数据请求转发给后端真实服务器。为了让同一客户端访问服务器时可以保持会话（同一客户端第二次访问网站时可以被转发至相同的后端真实服务器），HAProxy有三种解决方案：客户端IP、Cookie以及Session。

第一种方式中，HAProxy将客户端IP进行hash计算并保存，以此确保相同的IP访问代理服务器时可以转发固定的真实服务器。第二种方式中，HAProxy依靠真实服务器发送给客户端的Cookie信息进行会话保持。第三种方式中，HAProxy将保存真实服务器的Session与服务器标识，实现会话保持功能。

HAProxy软件包可以在官网下载。

官方网站：<https://github.com/haproxy/haproxy>

镜像下载：<http://pkgs.fedoraproject.org/repo/pkgs/haproxy/>

这里使用yum方式安装：

**[root@haproxy ~]# yum -y install haproxy**

## 配置解析

HAProxy安装后没有默认的配置文件，需要手动创建。创建/etc/haproxy/haproxy.cfg配置文件，当启动HAProxy服务时需要使用-f选项指定配置文件路径。HAProxy配置文件主要包含全局设置段与代理段，global代表全局段，defaults、listen、frontend、backend为代理段。fronted用来匹配客户端请求的域名或URI等，backend定义后端服务器集群，listen是frontend与backend的集合，有时仅使用listen即可替代frontend与backend。

下面给出一个监听80端口的HTTP代理案例，代理服务器将转发数据请求到单一后台服务器127.0.0.1:8080。

**global**

**deamon**

**maxconn 256**

**defaults**

**mode http**

**timeout connect 5000ms**

**timeout client 50000ms**

**timeout server 50000ms**

**frontend http-in**

**bind \*:80**

**default\_backend servers**

**backend servers**

**server server1 127.0.0.1:8000 maxconn 32**

以下是HAProxy主配置文件的具体语法格式及描述

全局设置（gloabl）

配置项 描述

chroot <jail dir> 将工作目录切换至<jail dir>并执行chroot。该配置可增强HAProxy

的安全性，但需要使用超级账户启动HAProxy

deamon 配置HAProxy以后进行模式工作

uid <number> 配置进程的账户ID，建议设置为HAProxy专用账户

git <number> 配置进程的组ID，建议设置为HAProxy专用组

log <address> <facility> 配置全局syslog服务器，可以设置两台日志服务器

nbproc <number> 指定后台进程的数量

pdifile <pidifle> 将进程ID号写入<pidfile>文件

ulimit-n <number> 设置每个进程的最大文件描述符数量

maxconn <number> 设置每个进程支持的最大并发数

tune.bufsize <number> 设置buffer大小，默认值为16384，单位为字节（B）

代理设置（defaults，listen，frontend，backend）

mode HAProxy工作模式，可选项为：tcp、http、health

timeout check <timeout> 设置检查超时时间

contimeout <timeout> 设置连接超时时间

balance roundrobin 默认负载均衡工作模式，轮询

bind <address>:<port> 定义一个或多个监听地址或端口

stats auth admin:admin 设置监控界面的用户名称与密码

status refresh <number> 统计页面刷新间隔时间

option httplog 使用http日志

cookie <name> 启动基于cookie的保持连接功能

option forwardfor 允许插入X-Forwarded-For数据包给后端真实服务器，可以让后

端服务器获得客户端的真实IP地址

option abortonclose 服务器负载高时，自动关闭列队里处理时间比较长的连接请求

option allbackups 当后端服务器全部宕机时，是否激活所有备用的服务器，默认仅

启动第一个备用服务器

option dontlognull 不记录空连接日志，主要用于不记录健康检查日志

option redispath 在HTTP模式，如果使用cookie的服务器宕机，客户端还会坚持

连接它，该选项在后端服务器宕机时强制将请求转发给其他健康

主机

monitor-uri <uri> 检查<uri>文件是否存在，依次判断主机健康状态

monitor-fail if site\_dead 当服务器宕机时，返回503错误代码，需要定义ACL

option httpchk <uri> 使用HTTP协议检查服务器的健康状态

retries <value> 服务器连接失败后的重试次数

timeout client <n> 设置客户端最大超时时间为n，默认单位为毫秒（ms）

timeout server <n> 设置服务端最大超时时间为n，默认单位为毫秒（ms）

timeout connect <n> 设置连接最大超时时间为n，默认单位为毫秒（ms）

default\_backend 配置文件中没有use\_backend规则时，设置默认后端服务器组，

服务器组由backend定义

use\_backend 当条件满处时，指定后端服务器组，需要设置ACL

acl <name> <criterion> 定义访问控制列表，配置文件中通过name调用该ACL，常用限

制包括：dst(目标地址)，dst\_port(目标端口)，src(源地址)，hdr(连

接头部信息)，path\_reg(访问路径匹配正则)，url(统一资源定位符)

ACL访问控制表案例如下：

**global**

**daemon**

**maxconn 256**

**defaults**

**mode http**

**timeout connect 5000ms**

**timeout client 50000ms**

**timeout server 50000ms**

**frontend http-in**

**bind \*:80**

**# 定义ACL控制源地址为192.168.0.0/24**

**acl badboys src192.168.0.0/24**

**# 定义ACL控制请求包头信息通过正则匹配，-i表示不区分大小写**

**acl example\_acl hdr\_reg(host) -i ^(www.example.com|web.example.com)$**

**# 当用户访问请求触发名为badboys的ACL规则时，禁止该客户的请求数据**

**block if badboys**

**# 当用户访问请求触发名为example\_acl 的ACL规则时，代理会江请求转发为**

**# servers\_2 后端服务器组**

**use\_backend servers\_2 if example\_acl**

**# 定义没有ACL时默认使用的服务器组**

**default\_backend servers\_1**

**# 定义后端服务器组servers\_1**

**backend servers\_1**

**# 定义服务器组中的具体服务器**

**server server1 127.0.0.1:8000 maxconn 32**

**# 定义后端服务器组servers\_2**

**backend servers\_2**

**server server1 127.0.0.1:8080 maxconn 3**

## HAProxy应用案例

本例将以生产环境为原型，简化网络拓扑后使用HAProxy实现以下所示的高兴能代理服务器架构。

|----------------------------------------------------------|

| |

| |----web1 |

| | |

Client-----Internet-------------HAProxy--------转换器------------web2 |

| | |

| |----web3 |

| |

|----------------------------------------------------------|

网络配置如下：

服务器名称 网络配置

haproxy.example.com eno16777736:10.10.10.10

eno33554984:192.168.0.254

web1.example.com eno16777736:192.168.0.1

web2.example.com eno16777736:192.168.0.2

web3.example.com eno16777736:192.168.0.3

client.example.com eno16777736:10.10.10.100

本例将使用listen定义一个监控端口；使用frontend定义一个前端80端口；通过backend分别定义名为inside\_servers 和 external\_servers的服务器组；使用default\_backend定义默认服务器组为external\_servers；定义ACL规则时，如果内网（192.168.0.0/24）访问Web服务，则由inside\_server服务组提供Web页面。

external\_server 服务器组中包含 web1.example.com 和 web2.example.com两台服务器，inside\_servers服务器组中仅包含web3.example.com一台服务器。

三台服务器基本采用相同的设置，配置如下：

[root@web1 ~]# vim /etc/sysconfig/network-scripts/ifcfg-eno16777736

DEVICE="eno16777736"

BOOTPROTO="static"

ONBOOT="yes"

IPADDR=192.168.0.1

PREFIX=24

GATEWAY=192.168.0.254

TYPE="Ethernet"

[root@web2 ~]# vim /etc/sysconfig/network-scripts/ifcfg-eno16777736

DEVICE="eno16777736"

BOOTPROTO="static"

ONBOOT="yes"

IPADDR=192.168.0.2

PREFIX=24

GATEWAY=192.168.0.254

TYPE="Ethernet"

[root@web3 ~]# vim /etc/sysconfig/network-scripts/ifcfg-eno16777736

DEVICE="eno16777736"

BOOTPROTO="static"

ONBOOT="yes"

IPADDR=192.168.0.3

PREFIX=24

GATEWAY=192.168.0.254

TYPE="Ethernet"

[root@web[123] ~]# systemctl restart network

[root@web[123] ~]# yum -y install httpd

[root@web[123] ~]# echo "192.168.0.[123]" > /var/www/html/index.html

[root@web[123] ~]# systemctl start httpd

[root@web[123] ~]# firewall-cmd --set-default-zone=trusted

HAProxy代理服务器设置如下：

[root@haproxy ~]# vim /etc/sysconfig/network-scripts/ifcfg-eno16777736

DEVICE="eno16777736"

BOOTPROTO="static"

ONBOOT="yes"

IPADDR=10.10.10.10

PREFIX=8

TYPE="Ethernet"

[root@haproxy ~]# vim /etc/sysconfig/network-scripts/ifcfg-eno33554984

DEVICE="eno16777736"

BOOTPROTO="static"

ONBOOT="yes"

IPADDR=192.168.0.254

PREFIX=24

TYPE="Ethernet"

[root@haproxy ~]# systemctl restart network

[root@haproxy ~]# firewall-cmd --set-default-zone=trusted

[root@haproxy ~]# vim /etc/security/limits.conf # 内核调优，文件中插入如下两行

\* soft nofile 65535

\* hard nofile 65535

[root@haproxy ~]# vim -y install haproxy

[root@haproxy ~]# vim /etc/haproxy/haproxy.cfg # 修改配置文件

**global**

**maxconn 4096**

**log 127.0.0.1 local3 info**

**chroot /var/haproxy**

**uid 99**

**gid 99**

**daemon**

**nbproc 1**

**pidfile /var/run/haproxy.pid**

**ulimit-n 65535**

**stats socket /var/tmp/status**

**defaults**

**log global**

**mode http**

**maxconn 2048**

**option httplog**

**option httpclose**

**option dontlognull**

**option forwardfor**

**option redispatch**

**option abortonclose**

**stats refresh 30**

**retries 3**

**balance roundrobin**

**cookie SRV**

**timeout check 2000ms**

**timeout connect 5000ms**

**timeout server 50000ms**

**timeout client 50000ms**

**listen admin\_status # 定义HAProxy监控界面**

**bind 0.0.0.0:6553**

**mode http**

**log 127.0.0.1 local3 info**

**stats enable**

**stats refresh 5s # 监控统计页面自动刷新时间为5s**

**stats realm Haproxy\ Statistics # 登陆监控统计页面提示符**

**stats uri /admin?stats # 监控页面URL路径**

**stats auth admin:admin # 查看HAProxy监控页面的账户和密码**

**stats hide-version # 隐藏HAProxy版本信息**

**frontend web\_service # 定义终端用户的访问前端服务器**

**bind 0.0.0.0:80**

**mode http**

**log global**

**option httplog**

**option httpclose**

**option forwardfor**

**acl inside\_src src 192.168.0.0/24 # 定义ACL**

**# use\_backend 调用ACL定义，如果源地址为192.168.0.0/24，则代理服务器**

**# 将会把请求转发给inside\_servers 服务器组**

**use\_backend inside\_servers if inside\_src**

**default\_backend external\_servers**

**backend external\_servers**

**mode http**

**balance roundrobin # 轮询真实服务器**

**# 检查真实服务器的index.html文件，以此判断服务器的健康状态**

**option httpchk GET /index.html**

**# 定义后端真实服务器，向cookie信息中插入web1信息，check代表允许**

**# 服务器进行健康检查，健康检查的时间间隔为2000ms，连续两次健康检**

**# 查成功则认为服务器是有效开启的，连续三次健康检查失败后，认为服务**

**# 器已经宕机，服务器权重为1**

**server web1 192.168.0.1:80 cookie web1 check inter 2000 rise 2 fall 3 weight 1**

**server web2 192.168.0.2:80 cookie web2 check inter 2000 rise 2 fall 3 weight 1**

**backend inside\_servers**

**mode http**

**balance roundrobin**

**option httpchk GET /index.html**

**server web1 192.168.0.3:80 cookie check inter 1500 rise 3 fall 3 weight 1**

[root@haproxy ~]# vim /etc/rsyslog.conf # 修改日志配置文件，插入以下三行

[root@haproxy ~]# systemctl restart syslog # 重启系统日志服务

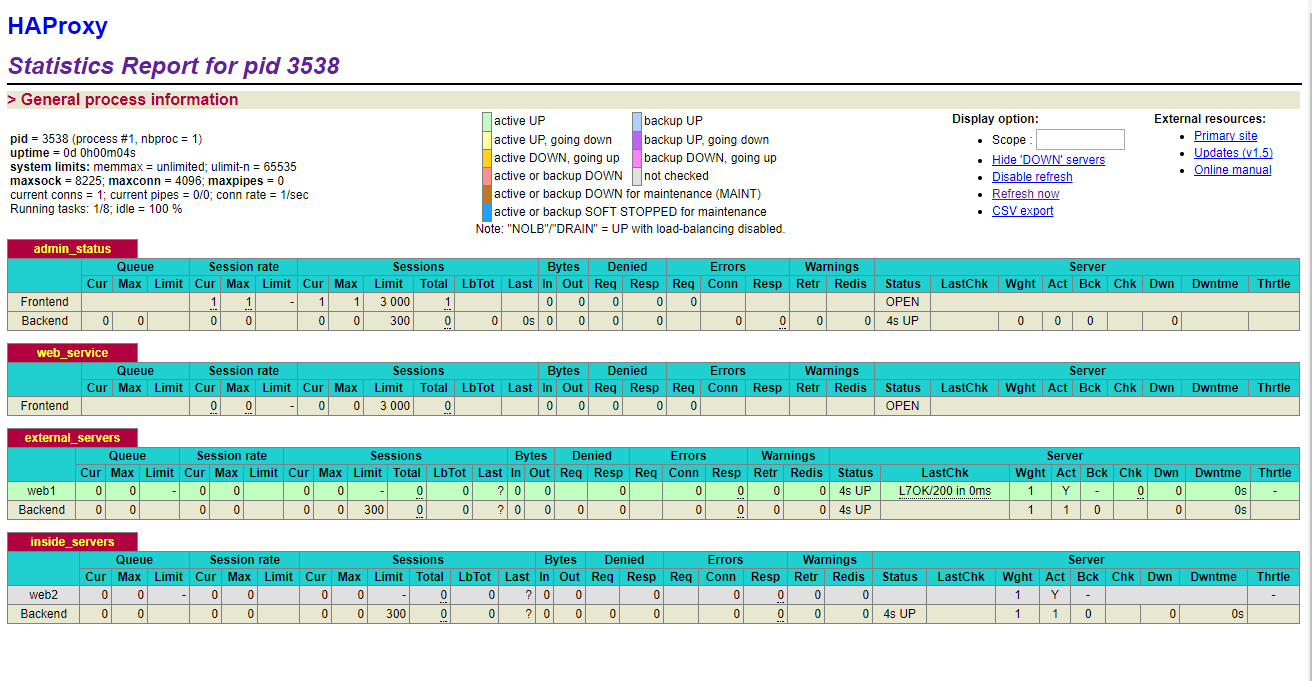
[root@haproxy ~]# haproxy -f /etc/haproxy/haproxy.cfg # 指定配置文件启动HAProxy服务

[root@haproxy ~]# echo "/usr/local/sbin/haproxy -f /etc/haproxy/haproxy.cfg" >> /

> /etc/rc.local # 配置开启自启动

客户端验证：

首先客户端主机配置正确的网络环境，确保客户端与HAProxy代理服务器可以直接连通。使用浏览器访问http://10.10.10.10:6553/admin?stats 查看代理服务器状态统计信息。



配置客户端主机IP地址为10.10.10.100，通过浏览器访问http://10.10.10.10，刷新将分别得到web1.example.com以及web2.example.com两台主机返回的页面信息。如果客户端主机IP的自己配置为192.168.0.0/24网络内的IP，则访问http://192.168.0.254，服务器返回的页面将永远都是web3.example主机的页面信息。

## 更多案例参考

https://www.iyunv.com/forum.php?mod=viewthread&tid=261189