haproxy 基本配置

经过了【haproxy 安装文档】之后，下面就是配置一下配置文件了。
网上一个很全面的haproxy说明文档 → http://www.verydemo.com/demo_c152_i76079.html
只要你想的到的疑问，这个文档里基本都可以找的到！

基本的配置文件说明

首先说明一下，配置文件主要包含4块。
①global：一些全局配置，系统配置
②defaults：“defaults”段用于为所有其它配置段提供默认参数，这配置默认配置参数可由下一个“defaults”所重新设定
③forntend：“frontend”段用于定义一系列监听的套接字，这些套接字可接受客户端请求并与之建立连接。
④backend：“backend”段用于定义一系列“后端”服务器，代理将会将对应客户端的请求转发至这些服务器。
⑤listen：“listen”段通过关联“前端”和“后端”定义了一个完整的代理，通常只对TCP流量有用。

```
# this config needs haproxy-1.1.28 or haproxy-1.2.1

global
#haproxy是通过系统的syslog/rsyslog服务进行日志输出的；
#日志配置后面有详细说明
 log 127.0.0.1 local0
#每个进程可用的最大连接数
 maxconn 40000
#改变工作主目录，指定为安装目录即可
 chroot /usr/local/haproxy
#进程的用户和组，99=nobody，直接使用99即可
 uid 99
 gid 99
#后台运行模式
 daemon
#进程的数量
 nbproc 1
#spreadchecks <0..50, in percent>：在haproxy后端有着众多服务器的场景中，在精确的时间间隔后统一对众服务器进行健康状况检查可能会带来意外问题；此选项用于将其检查的时间间隔长度上增加或减小一定的随机时长；
 spread-checks 20
#pid文件位置
 pidfile /usr/local/haproxy/haproxy.pid

defaults
#7层转发，http模式；mode { tcp|http|health }
 mode http
#日志配置后面有详细说明
 log 127.0.0.1 local0
```

```
#采用HTTP日志格式
 option httplog
#保证HAProxy不记录上级负载均衡发送过来的用于检测状态没有数据的心跳包
 option dontlognull
#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
#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
#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
#记录健康检查日志
 option log-health-checks
#允许对单个socket进行统计
 option socket-stats
#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
#设定每个前端以及后端的最大连接数
 maxconn 3000
#各种连接超时设置，默认单位是毫秒ms，常用的还有秒s，分钟m，小时h，天d
 contimeout 5000ms
 clitimeout 50000ms
 srvtimeout 50000ms

listen haproxy-status
#日志配置
 log 127.0.0.1 local0
#监听端口，*和0.0.0.0均代表本机所有IP；也可以修改成本机的某个IP
 bind *:60000
#开启监控，以下关于统计的相关配置，其实可以配置在listen以及任意backend，不论通过哪个端口来看这个统计页面，其实是一模一样的
 stats enable
#统计页面url
 stats uri /haproxy-status
#统计页面密码框上提示文本，空格需要用转义符，并且文本不需要加引号
 stats realm Kuanlian\ Haporxy\ Status\ Page
#统计页面用户名和密码设置，可以设置多个帐号密码，写成多行即可；明文传输
 stats auth root:passw0rd
#统计页面自动刷新时间，看自己需要是否开启
# stats refresh 10s
#隐藏统计页面上HAProxy的版本信息
 stats hide-version
#统计页面上可以对后端服务器进行操作；将TRUE改为LOCALHOST，即只有本机可以打开统计页面，更加安全
 stats admin if LOCALHOST
# stats admin if TRUE

frontend nginx
#日志配置
 log 127.0.0.1 local0
```

```
#监听端口
 bind *:80
#指定后端的服务器集群
 default_backend nginx-pool

backend nginx-pool
#轮询策略，下面有说明
 balance roundrobin
#进行http检测
 option httpchk GET /index.html HTTP/1.0
#后端具体server的配置；weight只在roundrobin相关下才起作用；check该server进行健康检测；inter检测间隔，单位毫秒；rise/fall检测结果需要满足多少次才能判断该server  up/down
 server nginx1 192.168.223.102:80 weight 5 cookie nginx1 check inter 2000 rise 3 fall 3
 server nginx2 192.168.223.103:80 weight 5 cookie nginx2 check inter 2000 rise 3 fall 3 backup

 #同上
frontend tomcat
 bind *:8080
 default_backend tomcat-pool

backend tomcat-pool
 balance leastconn
#tomcat用7层检测似乎会有问题，用4层是ok的
#option httpchk GET /test.html HTTP/1.0
 server tomcat1 192.168.223.102:8080 weight 5 cookie tomcat1 check inter 2000 rise 3 fall 3
 server tomcat2 192.168.223.103:8080 weight 5 cookie tomcat2 check inter 2000 rise 3 fall 3
```

日志配置

```
log 127.0.0.1 local0
```

定义了使用系统日志服务syslog来输出日志（在6.0之后，系统已经使用rsyslog来代替了syslog），输出类型为local0，需要与下面对应起来

具体操作

在/etc/rsyslog.conf中，取消下面两行的注释（虚拟机中，注释掉之后就有日志了，不注释掉就没有）

```
$ModLoad imudp.so
$UDPServerRun 514
```

添加一行（中间用tab）

```
# Save haproxy messages to haproxy.log

local0.* /var/log/haproxy/haproxy.log
```

修改一行（将haproxy的日志在messgae中去除）

```
*.info;mail.none;authpriv.none;cron.none;haproxy.none          /var/log/messages
```

在/etc/sysconfig/rsyslog中修改成（这一条不知道有什么作用，但是还是修改一下吧）

```
SYSLOGD_OPTIONS="-c 2 -r"
```

当某个backend的机器全部down的时候，会在所有用户的屏幕上出现提示信息，如下
这个报错是由于/etc/rsyslog.conf中的

```
# Everybody gets emergency messages
*.emerg *
```

起的作用。

---

haproxy轮询算法介绍

HA Proxy算法有如下8种：

一、roundrobin表示简单的轮询，这个不多说，这个是负载均衡基本都具备的;

二、static-rr表示根据权重，建议关注;

三、leastconn表示最少连接者先处理，建议关注;

四、sourc表示根据请求源IP建议关注;

五、uri表示根据请求的URI;

六、url_param表示根据请求的URI参数'balancurl_param'requiranURLparametname

七、hdrname表示根据HTTP请求头来锁定每一次HTTP请求;

八、rdp-cookiname表示根据据cookiname来锁定并哈希每一次TCP请求。

一般主要关注下面几个吧

```
roundrobin:每个server根据权重依次被轮询，
这个算法是动态的，意味着 server的权重可以实时地被调整。对于每个haproxy的backend servers的数目而言被限制在4128个活跃数目之内。
```

```
静态roundrobin(static-rr):跟roundrobin类似，唯一的区别是不可以动态实时server权重和backend 的server数目没有上限。
```

```
最小连接数目负载均衡策略（leastconn）：round-robin适合于各个server负载相同的情况。
最小连接数目算法适合于长时间会话，如LDAP，SQL，TSE，但是并不适合于HTTP短连接的协议。
```

```
源IP hash散列调度：将源ip地址进行hash，再根据hasn求模或者一致性hash定位到hash表中的server上。
```

相同的ip地址的请求被分发到同一个server上。但当server的数量变化时，来自于同一client的请求可能会被分发到不同的server上。这个算法通常用在没有cookie的tcp模式

---

haproxy与操作系统的选择

HAProxy目前主要有两个版本：

- 1.4——提供较好的弹性：衍生于1.2版本，并提供了额外的新特性，其中大多数是期待已久的。
- 1.3——内容交换和超强负载：衍生于1.2版本，并提供了额外的新特性。

一般企业中用的较多的还是haproxy 1.4，但应用还是得看实际情况。

若要获得最高性能，需要在Linux 2.6或打了epoll补丁的Linux 2.4上运行haproxy 1.2.5以上的版本。haproxy 1.1l默认使用的polling系统为select()，其处理的文件数达数千个时性能便会急剧下降。1.2和1.3版本默认的为poll()，在有些操作系统上可会也会有性能方面的问题，但在Solaris上表现相当不错。HAProxy 1.3在Linux 2.6及打了epoll补丁的Linux 2.4上默认使用epoll，在FreeBSD上使用kqueue，这两种机制在任何负载上都能提供恒定的性能表现。在较新版本的Linux 2.6(>=2.6.27.19)上，HAProxy还能够使用splice()系统调用在接口间无复制地转发任何数据，这甚至可以达到10Gbps的性能。

总结：

基于以上事实，在x86或x86_64平台上，要获取最好性能的负载均衡器，建议按顺序考虑以下方案。

-

Linux 2.6.32及之后版本上运行HAProxy 1.4；

-

打了epoll补丁的Linux 2.4上运行HAProxy 1.4；

-

FreeBSD上运行HAProxy 1.4；

-

Solaris 10上运行HAProxy 1.4；

单进程、事件驱动模型显著降低了上下文切换的开销及内存占用。

---

haproxy一些重要参数说明

http://www.verydemo.com/demo_c152_i76079.html 的后半段！！！