

| | | | | | | |
|--------------|----|----|----|----|----|----|
| < 2018年12月 > | | | | | | |
| 日 | 一 | 二 | 三 | 四 | 五 | 六 |
| 25 | 26 | 27 | 28 | 29 | 30 | 1 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | 1 | 2 | 3 | 4 | 5 |

公告

昵称：MacoLee
 园龄：2年7个月
 粉丝：36
 关注：8
[+加关注](#)

搜索

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

随笔分类

[Git\(1\)](#)
[python\(23\)](#)
[python之路\(1\)](#)
[Web前端\(5\)](#)
[算法\(1\)](#)

HAProxy 安装配置详解

简介

HAProxy提供高可用性、负载均衡以及基于TCP和HTTP应用的代理，支持虚拟主机，它是免费、快速并且可靠的一种解决方案。
 HAProxy特别适用于那些负载特大的web站点，这些站点通常又需要会话保持或七层处理。

HAProxy运行在当前的硬件上，完全可以支持数以万计的并发连接。并且它的运行模式使得它可以很简单安全的整合进您当前的架构中， 同时可以保护你的web服务器不被暴露到网络上。

HAProxy实现了一种事件驱动, 单一进程模型，此模型支持非常大的并发连接数。多进程或多线程模型受内存限制 、系统调度器限制以及无处不在的锁限制，很少能处理数千并发连接。事件驱动模型因为在有更好的资源和时间管理的用户空间(User-Space) 实现所有这些任务，所以没有这些问题。此模型的弊端是，在多核系统上，这些程序通常扩展性较差。这就是为什么他们必须进行优化以 使每个CPU时间片(Cycle)做更多的工作。

安装



```

#下载
wget http://fossies.org/linux/misc/haproxy-1.6.9.tar.gz

#解压
tar -zxvf haproxy-1.6.9.tar.gz
cd haproxy-1.6.9

#安装
make TARGET=linux2628 ARCH=x86_64 PREFIX=/usr/local/haproxy
make install PREFIX=/usr/local/haproxy

#参数说明
    
```

运维(38)

随笔档案

2018年8月 (1)

2017年4月 (1)

2017年3月 (1)

2017年2月 (1)

2017年1月 (1)

2016年12月 (2)

2016年11月 (3)

2016年10月 (1)

2016年9月 (15)

2016年8月 (14)

2016年7月 (20)

2016年6月 (14)

积分与排名

积分 - 85799

排名 - 5054

最新评论

1. Re:Pycharm如何导入Django项目

@dalyday首先，你要在pycharm中打开你的django项目，然后再操作...

--MacoLee

2. Re:Pycharm如何导入Django项目

第二步箭头1处不显示项目名称，麻烦能指点下

--dalyday

3. Re:Ansible之playbook

天道酬勤，-i的用法

--Michael2397

4. Re:Python之Fabric模块

@runs_once #查看本地系统信息，当有多台主机时只运行一次这句没看懂 查看本地系统信息，怎么会有多台主机呢？？？

--Lemon_乐

5. Re:HAProxy安装配置详解

赞

--_BLUE

```
TARGET=linux26 #内核版本，使用uname -r查看内核，如：2.6.18-371.el5，此时该参数就为linux26；kernel 大于2.6.28的用：
TARGET=linux2628
ARCH=x86_64 #系统位数
PREFIX=/usr/local/haproxy #/usr/local/haproxy为haproxy安装路径
```

配置(自己创建)

【/usr/local/haproxy/haproxy.cfg】

```
#####全局配置#####
global
    log 127.0.0.1 local0 #[日志输出配置，所有日志都记录在本机，通过local0输出]
    log 127.0.0.1 local1 notice #定义haproxy 日志级别[error warringinfo debug]
    daemon #以后台形式运行harpoxy
    nbproc 1 #设置进程数量
    maxconn 4096 #默认最大连接数,需考虑ulimit-n限制
    #user haproxy #运行haproxy的用户
    #group haproxy #运行haproxy的用户所在的组
    #pidfile /var/run/haproxy.pid #haproxy 进程PID文件
    #ulimit-n 819200 #ulimit 的数量限制
    #chroot /usr/share/haproxy #chroot运行路径
    #debug #haproxy 调试级别，建议只在开启单进程的时候调试
    #quiet

#####默认配置#####
defaults
    log global
    mode http #默认的模式mode { tcp|http|health }, tcp是4层，http是7层，health只会返回OK
    option httplog #日志类别,采用httplog
    option dontlognull #不记录健康检查日志信息
    retries 2 #两次连接失败就认为是服务器不可用，也可以通过后面设置
    #option forwardfor #如果后端服务器需要获得客户端真实ip需要配置的参数，可以从Http Header中获得客户端ip
    option httpclose #每次请求完毕后主动关闭http通道,haproxy不支持keep-alive,只能模拟这种模式的实现
    #option redispatch #当serverId对应的服务器挂掉后，强制定向到其他健康的服务器，以后将不支持
    option abortonclose #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
```

| |
|--|
| 2. HAProxy安装配置详解(28058) |
| 3. Nmap命令的29个实用范例(20562) |
| 4. LVS安装使用详解(18316) |
| 5. linux下查看进程运行的时间(18286) |
| 6. linux系统中rsync+inotify实现服务器之间文件实时同步(10440) |
| 7. Pycharm如何导入Django项目(8912) |
| 8. Django数据库设计中字段为空的方式(8327) |
| 9. Saltstack系列3: Saltstack常用模块及API(7643) |
| 10. Keepalived安装使用详解(6001) |

| |
|---------------------------|
| 评论排行榜 |
| 1. Pycharm如何导入Django项目(2) |
| 2. HAProxy安装配置详解(1) |
| 3. Python之Fabric模块(1) |
| 4. Ansible之playbook(1) |

| |
|---------------------------------------|
| 推荐排行榜 |
| 1. HAProxy安装配置详解(5) |
| 2. LVS安装使用详解(4) |
| 3. Nmap命令的29个实用范例(3) |
| 4. linux下查看进程运行的时间(2) |
| 5. linux下进程、端口号相互查看方法(2) |
| 6. Python之Rpyc模块(1) |
| 7. Saltstack系列3: Saltstack常用模块及API(1) |
| 8. CentOS下puppet安装(1) |

```
maxconn 4096 #默认的最大连接数
timeout connect 5000ms #连接超时
timeout client 30000ms #客户端超时
timeout server 30000ms #服务器超时
#timeout check 2000 #心跳检测超时
#timeout http-keep-alive10s #默认持久连接超时时间
#timeout http-request 10s #默认http请求超时时间
#timeout queue 1m #默认队列超时时间
balance roundrobin #设置默认负载均衡方式，轮询方式
#balance source #设置默认负载均衡方式，类似于nginx的ip_hash
#balnace leastconn #设置默认负载均衡方式，最小连接数

#####统计页面配置#####
listen stats
    bind 0.0.0.0:1080 #设置Frontend和Backend的组合体，监控组的名称，按需要自定义名称
    mode http #http的7层模式
    option httplog #采用http日志格式
    #log 127.0.0.1 local0 err #错误日志记录
    maxconn 10 #默认的最大连接数
    stats refresh 30s #统计页面自动刷新时间
    stats uri /stats #统计页面url
    stats realm XingCloud\ Haproxy #统计页面密码框上提示文本
    stats auth admin:admin #设置监控页面的用户和密码:admin,可以设置多个用户名
    stats auth Frank:Frank #设置监控页面的用户和密码: Frank
    stats hide-version #隐藏统计页面上HAProxy的版本信息
    stats admin if TRUE #设置手工启动/禁用，后端服务器(haproxy-1.4.9以后版本)

#####设置haproxy 错误页面#####
#errorfile 403 /home/haproxy/haproxy/errorfiles/403.http
#errorfile 500 /home/haproxy/haproxy/errorfiles/500.http
#errorfile 502 /home/haproxy/haproxy/errorfiles/502.http
#errorfile 503 /home/haproxy/haproxy/errorfiles/503.http
#errorfile 504 /home/haproxy/haproxy/errorfiles/504.http

#####frontend前端配置#####
frontend main
    bind *:80 #这里建议使用bind *:80的方式，要不然做集群高可用的时候有问题，vip切换到其他机器就不能访问了。
    acl web hdr(host) -i www.abc.com #acl后面是规则名称，-i为忽略大小写，后面跟的是要访问的域名，如果访问www.abc.com这个域名，就触发web规则，。
    acl img hdr(host) -i img.abc.com #如果访问img.abc.com这个域名，就触发img规则。
    use_backend webserver if web #如果上面定义的web规则被触发，即访问www.abc.com，就将请求分发到webserver这个作用域。
```

use_backend imgserver if img #如果上面定义的img规则被触发，即访问img.abc.com，就将请求分发到imgserver这个作用域。
default_backend dynamic #不满足则响应backend的默认页面

#####backend后端配置#####

backend webserver #webserver作用域

mode http

balance roundrobin #balance roundrobin 负载轮询，balance source 保存session值，支持static-rr, leastconn, first, uri等参数

option httpchk /index.html HTTP/1.0 #健康检查，检测文件，如果分发到后台index.html访问不到就不再分发给它

server web1 10.16.0.9:8085 cookie 1 weight 5 check inter 2000 rise 2 fall 3

server web2 10.16.0.10:8085 cookie 2 weight 3 check inter 2000 rise 2 fall 3

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

#rise 2是2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

backend imgserver

mode http

option httpchk /index.php

balance roundrobin

server img01 192.168.137.101:80 check inter 2000 fall 3

server img02 192.168.137.102:80 check inter 2000 fall 3

backend dynamic

balance roundrobin

server test1 192.168.1.23:80 check maxconn 2000

server test2 192.168.1.24:80 check maxconn 2000

listen tcptest

bind 0.0.0.0:5222

mode tcp

option tcplog #采用tcp日志格式

balance source

#log 127.0.0.1 local0 debug

server s1 192.168.100.204:7222 weight 1

server s2 192.168.100.208:7222 weight 1



负载均衡算法

⊕ haproxy负载均衡算法

ACL规则定义

⊕ haproxy acl定义

启动

```
/usr/local/haproxy/sbin/haproxy -f /usr/local/haproxy/haproxy.cfg
```

查看状态

```
http://192.168.1.22:1080/stats
```

#说明:

#1080即haproxy配置文件中监听端口

s#tats 即haproxy配置文件中的监听名称

分类: [运维](#)

标签: [运维](#), [负载均衡](#), [haproxy](#)

好文要顶

关注我

收藏该文





MacoLee

关注 - 8

粉丝 - 36

+加关注

« 上一篇: [Keepalived安装使用详解](#)

» 下一篇: [LVS安装使用详解](#)

posted on 2016-09-08 16:06 [MacoLee](#) 阅读(28056) 评论(1) 编辑 收藏

5

0

评论

#1楼 2016-11-11 16:19 _BLUE

赞

支持(1) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)网站首页。

【推荐】超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库！

【活动】华为云12.12会员节全场1折起 满额送Mate20

【活动】华为云会员节云服务特惠1折起

【推荐】服务器100%基准CPU性能，1核1G首年168元，限时特惠！



腾讯云AMD云服务器

节省IT成本30%
1核1G AMD机型**0.57元/天**起

立即抢购

相关博文：

- [Haproxy安装配置](#)
- [HAProxy负载均衡安装配置](#)
- [keepalived+haproxy 安装配置](#)
- [haproxy1.7编译安装配置](#)
- [Haproxy 安装配置详解-端口映射](#)



最新新闻：

- [应对失业，最该做的事，你可能一直都没做](#)
 - [线下新玩家网易严选首店测评：一半是商品，一半是体验](#)
 - [ES6能否续写蔚来故事？](#)
 - [马斯克公布隧道细节：就像科幻小说中的远距离传送](#)
 - [亚马逊负责国际业务核心高管离职 曾是贝索斯绝对心腹](#)
- » [更多新闻...](#)

Powered by:

博客园

Copyright © MacoLee