

TCP/HTTP 负载均衡利器 — HAProxy 介绍及实践

Godbach

nylzaowe@gmail.com

2015/06/07

大纲

- 简介
- 连接模式 (Connection Mode)
- 功能列表 (Feature List)
- 实例
- Debug & Analysis
- Reference

简介

- TCP/HTTP Load Balancer
- Reliable, High Performance
- 最新 stable 版本: 1.5.12
- 最新 dev 版本: 1.6-dev0
- 官网: www.haproxy.org

连接模式 (1)

- Proxy 典型配置: frontend + backend(s)
- 一个来自 Client 端的 request, HAProxy 要维持两个 TCP 连接:
 - Client -> HAProxy
 - HAProxy -> Real Server
- 支持 HTTP KeepAlive 和 Pipeline
- IPv4/IPv6 Dual stack

连接模式 (2)

HAProxy 在处理 frontend 和 backend 的 HTTP transaction 时，共有以下五种模式：

By default HAProxy operates in keep-alive mode with regards to persistent connections: for each connection it processes each request and response, and leaves the connection idle on both sides between the end of a response and the start of a new request.

HAProxy supports 5 connection modes :

- keep alive : all requests and responses are processed (default)
- tunnel : only the first request and response are processed, everything else is forwarded with no analysis.
- passive close : tunnel with "Connection: close" added in both directions.
- server close : the server-facing connection is closed after the response.
- forced close : the connection is actively closed after end of response.

功能列表

- 负载均衡算法
- Persistence
- Content Switching
- Content Rewriting
- HTTP Compression: 支持 HTTP 压缩, gzip&deflate
- HTTP Basic Authentication: 简单认证
- SSL Offload: 配置 cert, 支持 TCPS/HTTPS
- Transparent Proxy
- Health Check: 对 real server 做健康检查
- Log

功能列表 – 负载均衡算法

● 基本算法

- roundrobin: 动态 rr 算法, 支持动态修改 rs 的 weight
- static-rr: 静态 rr 算法, 参考 roundrobin
- leastconn: 最少连接数
- first: 优先使用 server id 最小的, 超过 maxconn 时选择下一个 server, 适合非 HTTP 的长连接。可结合 cloud 使用。

● Hash 类算法

- source: 源 IP
- uri:
- url_param: URL 中某个指定参数的 value hash
- hdr(<name>): 指定任何一个 header name, 以其 value hash
-

功能列表 – Persistence

- source ip:
- appsession:
- cookie: insert/rewrite/prefix...
- SSL session ID
- ...

功能列表 – Content Switching

- 前面提到，一个 Proxy 由 frontend + backend(s) 组成
- 其中 frontend 一个，backend 可以有多个。通过配置，实现不同的请求交给不同的 backend 处理，即所谓的 Content Switching
- 核心配置：
 `use_backend <backend> [{if | unless} <condition>]`

功能列表 – Content Rewriting

- 支持修改 HTTP Request 以及 Response Header
- 配置项 http-request/http-response

```
add-header <name> <fmt> | set-header <name> <fmt> |  
del-header <name> | set-nice <nice> | set-log-level <level> |  
replace-header <name> <match-regex> <replace-fmt> |  
replace-value <name> <match-regex> <replace-fmt> |
```

实例

- 典型配置
- Hot Configuration
- Mutli-Process
- HA Synchronization
- Statistics

实例 -- 典型配置

```
1 global
2     node hap
3     pidfile /var/log/haproxy/hap.pid
4     stats socket /var/log/haproxy/hap.socket level admin
5     maxconn 4096
6     daemon
7     quiet
8
9 defaults
10     mode http
11     option splice-auto
12     option http-keep-alive
13
14     timeout client 50s
15     timeout server 50s
16     timeout connect 5s
17     timeout http-keep-alive 50s
18     timeout http-request 50s
19
20 frontend fe
21     bind :80
22     use_backend be unless
23
24 backend be
25     balance roundrobin
26     server 1 2.2.2.1:80 id 1 cookie rs1 weight 1 maxconn 0
27     server 2 2.2.2.2:80 id 2 cookie rs2 weight 1 maxconn 0
28
```

实例 -- Hot Configuration

- Unix/TCP Socket Command
- 配置实例:

```
global
    stats socket /var/run/haproxy.sock mode 600 level admin
    stats socket ipv4@192.168.0.1:9999 level admin
    stats timeout 2m
```

- 设置命令示例
\$ echo "show stat" | socat stdio unix-connect:/path/to/hap.socket
- 支持 Command: 查看info、sess, 修改 server 配置, 设置 maxconn, stick table 等等
- REF: <http://cbonte.github.io/haproxy-dconv/configuration-1.5.html#9.2>

实例 -- Mutli-Process

- 利用多进程获取高性能
- nbproc: 指定启动的多进程个数
- bind-process: 当前 fe/be 在哪些进程上启动
- process: bind option, 当前 bind 在那些进程启动
- 配置实例（避免多进程下 **epoll** 惊群）

```
1
2 global
3     nbproc 4
4
5 frontend fe
6     bind 1.1.1.1:80 process 1
7     bind 1.1.1.1:80 process 2
8     bind 1.1.1.1:80 process 3
9     bind 1.1.1.1:80 process 4
```

实例 -- HA Synchronization

- HA 环境下同步 stick table

- 配置 peers section

peers mypeers

peer local 1.1.1.1:10000

peer remote 1.1.1.2:10000

- 引用定义的 peers

stick-table type ip size 20k peers mypeers

- 启动 HAProxy 进程

本地: \$ haproxy -f h.cfg -L local

远程: \$ haproxy -f h.cfg -L remote

实例 -- Statistics

- HAProxy 自带的统计信息 Web 展示, 关键配置如下
backend be
stats enable
stats uri /admin?stats

172.16.34.129:8080/admin?stats

HAProxy version 1.5-dev17, released 2012/12/28

Statistics Report for pid 92017

> General process information

pid = 92017 (process #1, nbproc = 1)
uptime = 0d 0h00m09s
system limits: memmax = unlimited; ulimit-n = 73
maxsock = 73; maxconn = 21; maxpipes = 0
current conns = 1; current pipes = 0/0; conn rate = 1/sec
Running tasks: 1/5; idle = 100 %

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)

backup UP
backup UP, going down
backup DOWN, going up
not checked

Note: UP with load-balancing disabled is reported as "NOLB".

Display option:
[Hide "DOWN" servers](#)
[Refresh now](#)
[CSV export](#)

External resources:
[Primary site](#)
[Updates \(v1.5\)](#)
[Online manual](#)

	Queue			Session rate			Sessions				Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend	1	1	-	1	1	-	1	1	-	1048576	1	0	0	0	0	0	0	0	0	0	OPEN								

vs

	Queue			Session rate			Sessions				Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
1	0	0	-	0	0	-	0	0	-	0	0	0	0	0	0	0	0	0	0	0			1	Y	-				-
2	0	0	-	0	0	-	0	0	-	0	0	0	0	0	0	0	0	0	0	0			1	Y	-				-
Backend	0	0	-	1	1	-	1	1	-	104858	1	0	0	0	0	0	0	0	0	0	9s UP		2	2	0		0	0s	

pool_1

	Queue			Session rate			Sessions				Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
1	0	0	-	0	0	-	0	0	-	0	0	0	0	0	0	0	0	0	0	0			1	Y	-				-
2	0	0	-	0	0	-	0	0	-	0	0	0	0	0	0	0	0	0	0	0			1	Y	-				-
Backend	0	0	-	1	1	-	1	1	-	104858	1	0	0	0	0	0	0	0	0	0	9s UP		2	2	0		0	0s	

Debug&Analysis

- 编译带调试信息版本的 HAProxy
 - Makefile 中添加编译选 -DDEBUG_XXX
 - 启动时 -d option
- 抓包判断出问题的环节
 - 分别在 HAProxy 端和 Real Server 端抓包，并对比判断
 - 条件允许的话，client 端抓包
- 通过 Unix Socket Command，执行 show stat
 - 主要观察 frontend 和 backend 上的相关统计数据

Reference

- HAProxy 官网: <https://www.haproxy.org/>
- HAProxy version 1.5 配置手册:
[http://cbonte.github.io/haproxy-dconv/
configuration-1.5.html](http://cbonte.github.io/haproxy-dconv/configuration-1.5.html)



Q&A