

SACC 第八届中国系统架构师大会
2016 SYSTEM ARCHITECT CONFERENCE CHINA 2016

架构 创新之路

负载均衡利器 HAProxy 功能剖析及部署案例

赵伟 @阿里云

Godbach@ChinaUnix

nylzaowei@gmail.com

议程

- 版本介绍
- 重要功能
- 配置实例
- 部署案例
- LVS or HAProxy or Nginx ???
- 参考

版本介绍

- 稳定可靠，高性能的 TCP/HTTP 负载均衡
- IPv4 & IPv6 Dual Stack
- 最新 stable 版本：1.6.9
- 最新 dev 版本：1.7-dev4
- 官网：<https://www.haproxy.org>

The logo for HA Proxy, featuring the letters 'HA' in blue and red above the word 'PROXY' in multi-colored letters (P: blue, R: green, O: yellow, X: red, Y: blue), all set against a background of a network diagram with nodes and connecting lines.

议程

- 版本介绍
- **重要功能**
- 配置实例
- 部署案例
- LVS or HAProxy or Nginx ???
- 参考

- 负载均衡算法
- 持久化 Persistence
- 内容路由 Content Routing
- 内容重写 Content Rewriting
- Health Check - Real server 健康检查, 邮件告警
- SSL Offload - 支持 TCPS/HTTPS
- HTTP 压缩 - 支持 gzip/deflate/raw-deflate
- HTTP Basic Authentication - 基本的 HTTP 认证
- Transparent Proxy, PROXY 协议, Lua 脚本.....

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。

负载均衡算法 -- Hash 类算法

- **source**: 根据源 IP hash
- **uri**: 根据 URI hash
- **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

持久化 Persistence

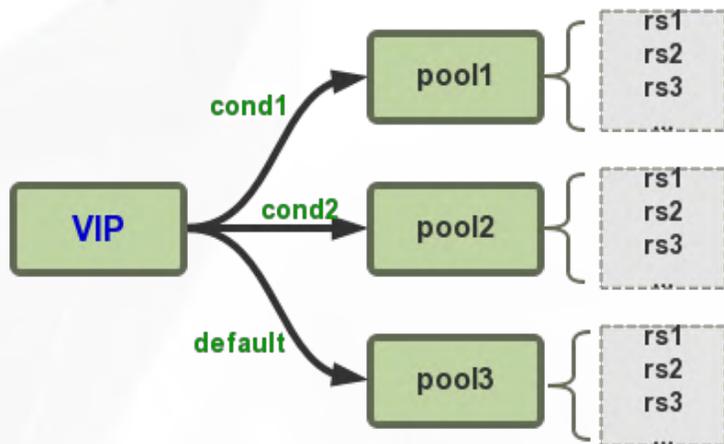
- source ip:
- cookie: insert/rewrite/prefix...
- SSL session ID
- appsession:
- ...

内容路由 Content Routing (1)

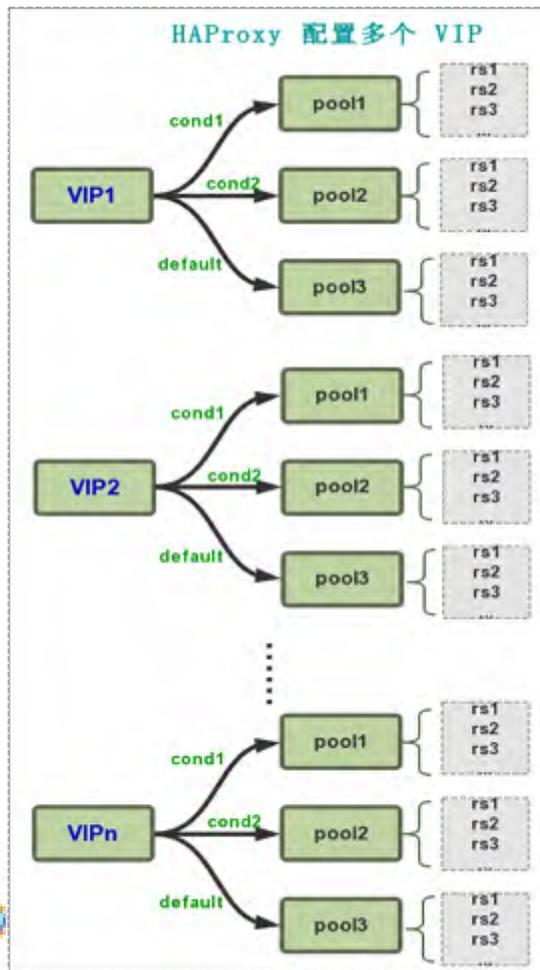
- 一个实例由 **frontend** + **backend(s)** 组成
- frontend 配置 **VIP**, backend 配置 **server pool**, pool 可以多个。不同类型请求由各自的 backend 处理, 即所谓的 Content Routing
- 核心配置: `use_backend <backend> [{if | unless} <condition>]`

内容路由 Content Routing (2)

单个 VIP



HAProxy 配置多个 VIP



内容重写 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> |
```

透明代理 Transparent Proxy

- 以真实的 Client IP 和 Real server 建连
- 配置复杂，除 HAProxy 外，需要配置 iptables 以及策略路由
- 详细配置见后面实例

议程

- 版本介绍
- 重要功能
- **配置实例**
- 部署案例
- LVS or HAProxy or Nginx ???
- 参考

- 典型配置
- 动态配置
- 多进程模式
- 透明代理
- HA 同步数据
- 统计页面

典型配置

- global
- defaults
- frontend (VIP)
- backend
 - RIP1
 - RIP2
 - ...

```
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
```

动态配置

- 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

多进程用法

- 利用多进程获取高性能，多进程间数据共享支持不好，**stick table 问题**
- 支持选项：nbproc/bind-process/process
- 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
```

透明代理 (1)

- linux kernel \geq 2.6.28, 并启用 kernel 转发

```
echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
```

```
echo 1 > /proc/sys/net/ipv4/conf/all/send_redirects
```

```
echo 1 > /proc/sys/net/ipv4/conf/eth0/send_redirects
```

- iptables 报文标记, 并配置策略路由

```
iptables -t mangle -N DIVERT
```

```
iptables -t mangle -A PREROUTING -p tcp -m socket -j DIVERT
```

```
iptables -t mangle -A DIVERT -j MARK --set-mark 111
```

```
iptables -t mangle -A DIVERT -j ACCEPT
```

```
ip rule add fwmark 111 lookup 100
```

```
ip route add local 0.0.0.0/0 dev lo table 100
```

透明代理 (2)

- 允许监听非本机 IP

```
echo 1 > /proc/sys/net/ipv4/ip_nonlocal_bind
```

- HAProxy 编译支持透明代理

```
USE_LINUX_TPROXY=1
```

- HAProxy 启用选项

```
source 0.0.0.0 usesrc clientip
```

HA 同步 -- 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

```
$ haproxy -f h.cfg -L local/remote
```

统计页面

- HAProxy 自带的统计信息 Web 展示，关键配置如下

defaults

stats enable

stats uri /admin?stats

- 更多配置见手册

- stats admin

- stats auth

- stats realm

HAProxy version 1.6.9, released 2016/08/30

Statistics Report for pid 7677

> General process information

pid = 7677 (process #1, nproc = 1)
 uptime = 0d 2h28m15s
 system limits: memmax = unlimited; ulimit-n = 16420
 maxsock = 16420; maxconn = 8192; maxpipes = 0
 current conns = 2; current pipes = 0/0; conn rate = 1/sec
 Running tasks: 1/12, idle = 100 %

Legend for process states:

- active UP
- active UP, going down
- active DOWN, going up
- active or backup DOWN
- active or backup DOWN for maintenance (MAINT)
- active or backup SOFT STOPPED for maintenance
- backup UP
- backup UP, going down
- backup DOWN, going up
- not checked

Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:

- Scope :
- Hide **DOWN** servers
- Refresh now
- CSV export

External resources:

- Primary site
- Updates (v1.5)
- Online manual

va1

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server											
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtme	Thrtle		
Frontend				1	1	884	-	1	11	8 192	3 240		266 846	1 097 559	0	0	3						OPEN									

pool1

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtme	Thrtle
rs1	0	0	-	0	1 884		0	10	4096	3 233	3 233	2h25m	265 106	998 997	0	0	0	0	0	0	0	2h28m UP	L4OK in 0ms	1	Y	-	0	0	0s	-
rs2	0	0	-	0	0		0	0	4096	0	0	?	0	0	0	0	0	0	0	0	0	2h28m DOWN	L4CON in 0ms	1	Y	-	1	1	2h28m	-
Backend	0	0		0	1 884		0	10	8 192	3 233	3 233	2h25m	265 106	998 997	0	0	0	0	0	0	0	2h28m UP		1	1	0	0	0	0s	

va2

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server										
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtme	Thrtle	
Frontend				0	2 115		-	1	10	8 192	4 484		367 566	1 433 040	0	0	16						OPEN								

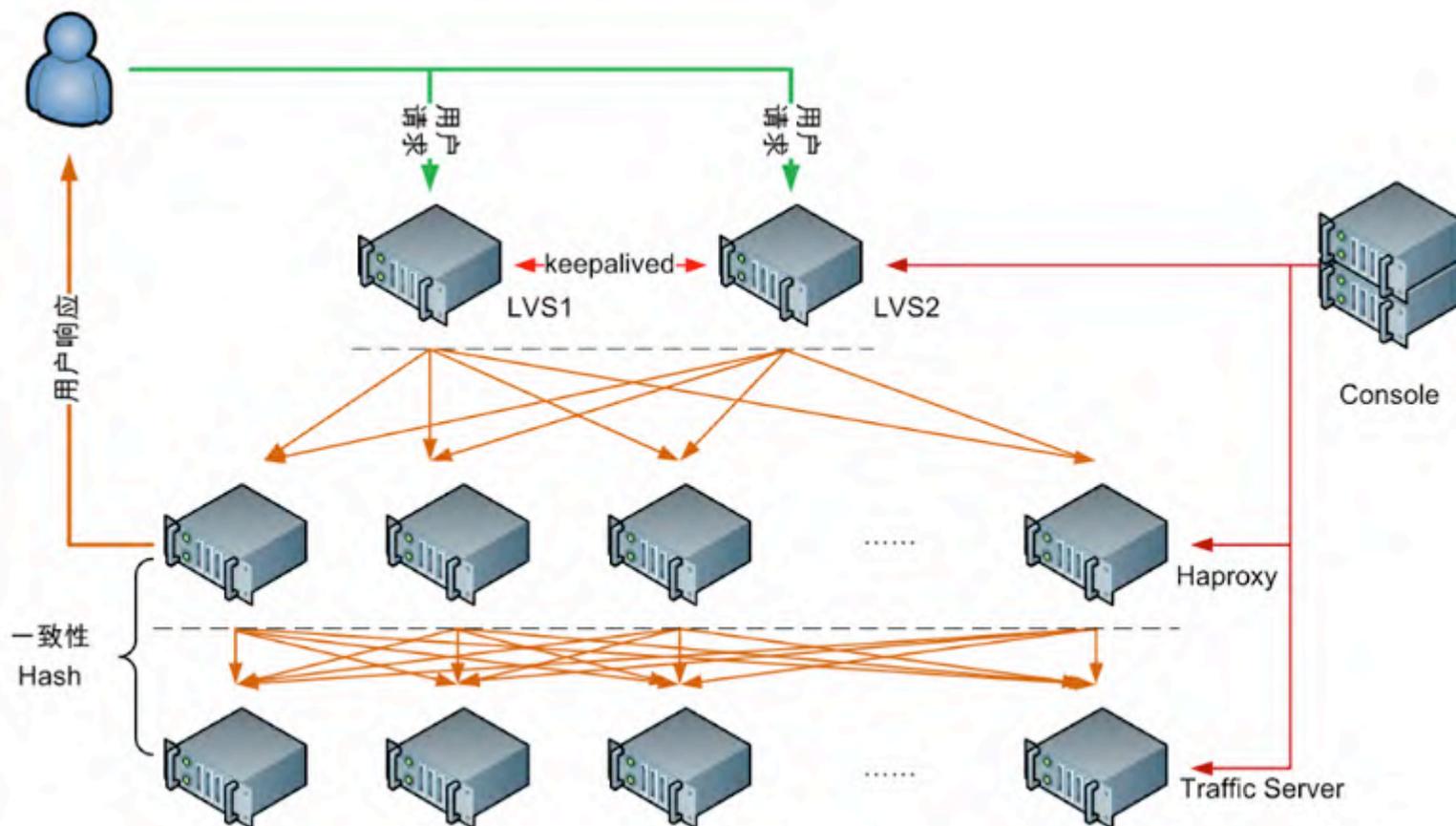
pool2

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtme	Thrtle
rs1	0	0	-	0	2 115		0	10	4096	4 488	4 488	47s	366 822	1 380 804	0	0	0	0	0	0	0	2h28m UP	L4OK in 0ms	1	Y	-	0	0	0s	-
rs2	0	0	-	0	0		0	0	4096	0	0	?	0	0	0	0	0	0	0	0	0	2h28m DOWN	L4CON in 0ms	1	Y	-	1	1	2h28m	-
Backend	0	0		0	2 115		0	10	8 192	4 488	4 488	47s	366 822	1 380 804	0	0	0	0	0	0	0	2h28m UP		1	1	0	0	0	0s	

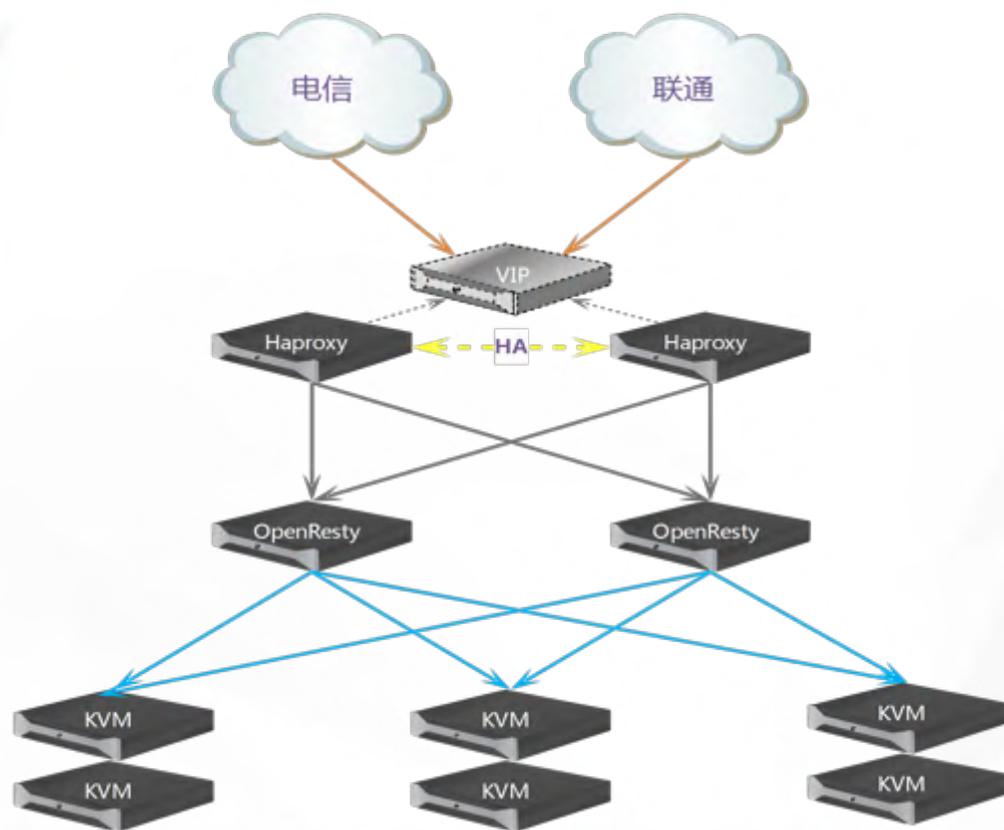
议程

- 版本介绍
- 重要功能
- 配置实例
- **部署案例**
- LVS or Haproxy or Nginx ???
- 参考

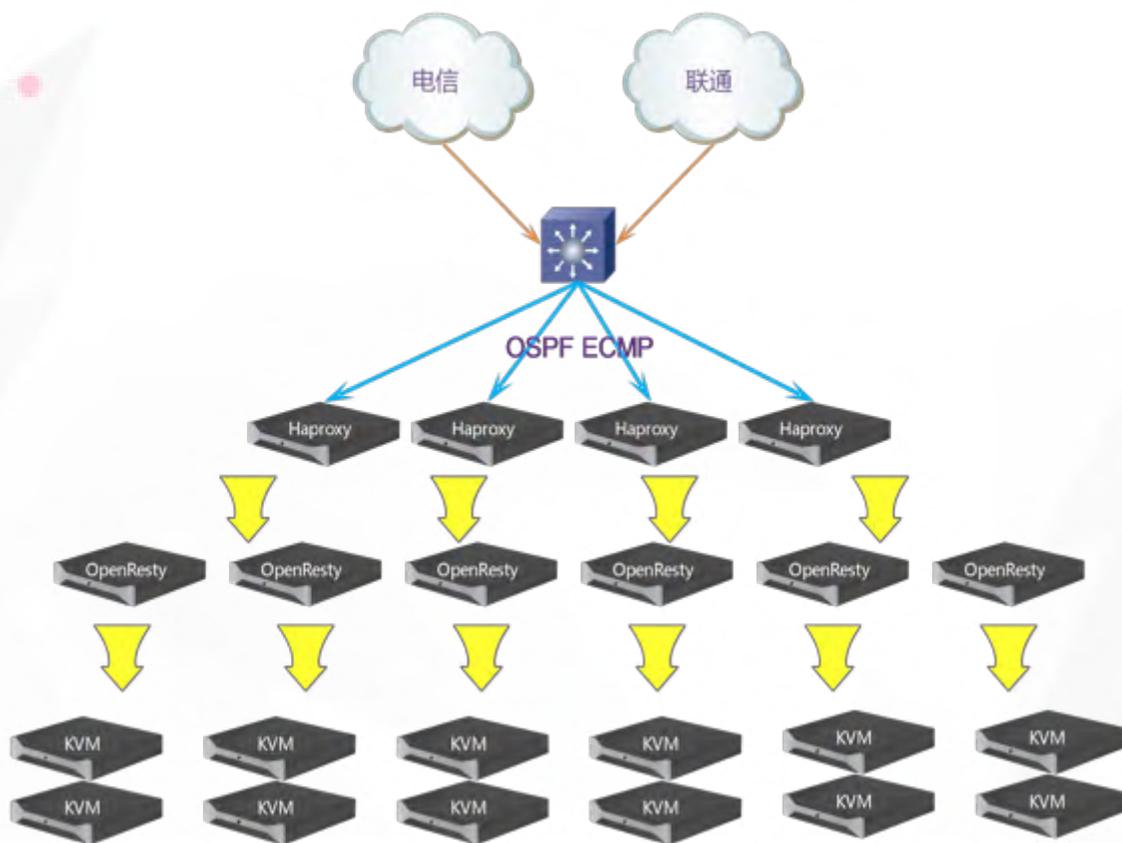
阿里云 CDN 早期部署方案



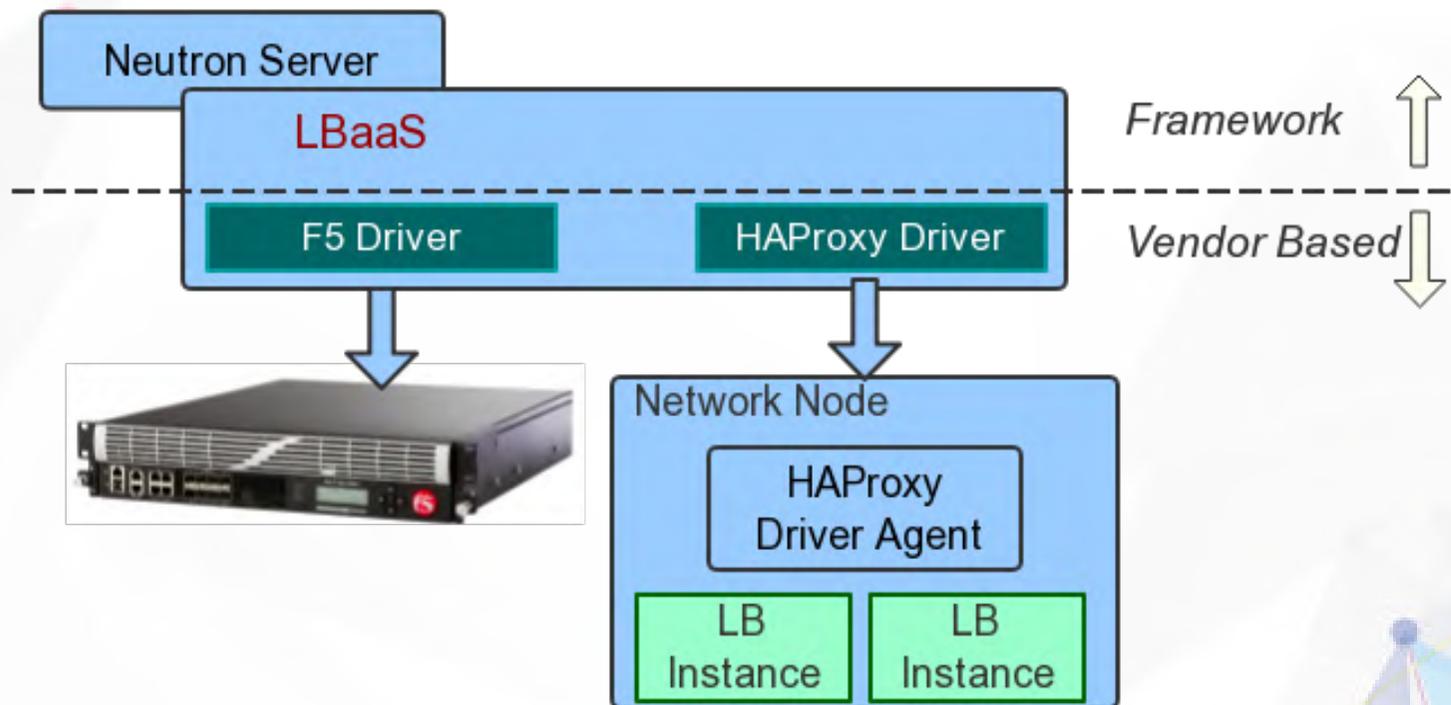
天天拍车 -- 当前部署方案



天天拍车 -- 未来计划方案



OpenStack Neutron LBaaS



议程

- 版本介绍
- 重要功能
- 配置实例
- 部署案例
- **LVS or HAProxy or Nginx ???**
- 参考

HAProxy or LVS ?

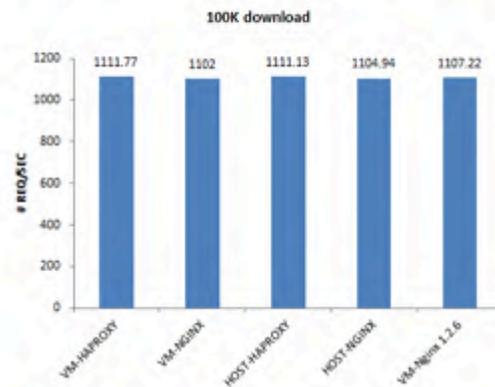
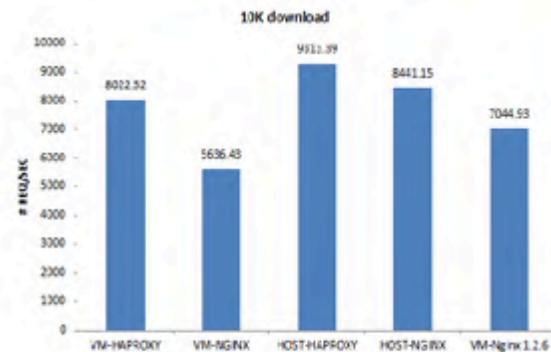
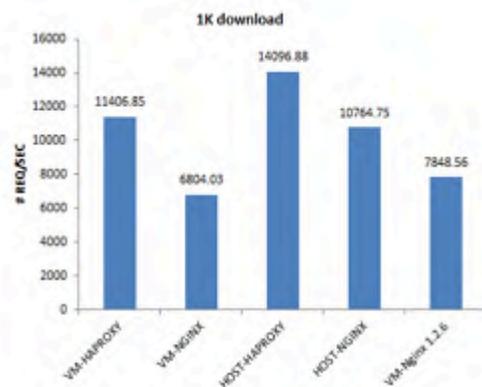
- 性能&业务量
 - LVS 四层负载均衡，性能高
- 功能
 - HAProxy 支持对内容检测，自带健康检查，部分动态配置
- 易运维
 - LVS 配置部分 kernel 参数，还要注意 conntrack，要注意一些坑

HAProxy or Nginx ?

- 功能！功能！功能！
 - 列举需要功能清单和优先级，逐一比较
- 易运维
- 性能
 - 结合业务场景，实际压测

HAProxy vs Nginx

- 性能比较：小 object 下，HAProxy 优势明显



HAProxy 性能优化缩影

- 参考链接<http://blog.chinaunix.net/uid-10167808-id-4004066.html>

- 运算符'|'和'||'性能

- '|' 核心指令 3 条
- '||' 核心指令 4 ~ 6 条



```
/* t.c */
int bitwise_or(int a, int b)
{
    return a | b;
}

int logical_or(int a, int b)
{
    return a || b;
}
```

参考

- [HAProxy 官网](#)
- [HAProxy 透明代理设置](#)
- [Neutron 是如何实现负载均衡器虚拟化的](#)
- [Preliminary benchmark for ELB](#)
- [ChinaUnix 论坛【集群和高可用】](#)

阿里云 CDN 求贤若渴

◇ 职位

✓ 产品

✓ 视频/调度/后台

✓ 网络/系统/安全

◇ 职位详情扫码

◇ 简历发送:

nylzhaowei@gmail.com



THANK

SequeMedia
SEQUENCE

IT 168
IT 168

ChinaUnix
CHINA UNIX

IT PLANET
IT PLANET