



SERVIDORES WEB DE ALTAS PRESTACIONES

Práctica 3 - Balanceo de carga en un sitio web

César Muñoz Reinoso

Curso 2022-2023

1 de abril de 2023

Índice

1. Balanceo de carga con NGINX	2
1.1. Instalación y configuración básica	2
1.2. Configuración avanzada con ponderación, máximo de fallos y conexiones persistentes	4
2. Balanceo de carga con HAPROXY	6
2.1. Instalación y configuración básica	6
2.2. Configuración avanzada con ponderación, máximo de conexiones y cookies persistentes	7
3. Módulo de estadísticas en HAproxy	9
3.1. Configuración básica con autenticación y puerto de acceso	9
3.2. Configuración avanzada con tasa de refresco y timeouts	10
4. Benchmarking sobre granja web	12
4.1. Carga sobre Nginx	12
4.2. Carga sobre HAProxy	14
5. Análisis comparativo de distintos balanceados en base a la carga con AB	17
6. Instalación y configuración de Go-between y Zevenet	18
6.1. Go-between	18
6.2. Zevenet	19

1. Balanceo de carga con NGINX

1.1. Instalación y configuración básica

Creamos una nueva máquina, a la que llamaremos M3 y al igual que en las máquinas virtuales M1 y M2 instalamos Ubuntu Server 22.04.2, con usuario cesarmunoz y contraseña Swap1234. Como hemos aprendido, con netplan le asignamos una ip estática 192.168.56.4 dentro del adaptador usuario-anfitrión usado anteriormente.

Vamos a utilizar NGINX como balanceador de carga, empezamos actualizando los repositorios e instalando el paquete de nginx. Iniciamos el servicio con

```
sudo service nginx start
```

```
cesarmunoz@m3-cesarmunoz:~$ sudo apt-get update && sudo apt-get dist-upgrade &&
sudo apt-get autoremove
[sudo] password for cesarmunoz:
Obj:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Des:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu jammy/main Translation-es [332 kB]
Des:6 http://es.archive.ubuntu.com/ubuntu jammy/restricted Translation-es [964 B]
Des:7 http://es.archive.ubuntu.com/ubuntu jammy/universe Translation-es [1.356 kB]
Des:8 http://es.archive.ubuntu.com/ubuntu jammy/multiverse Translation-es [68,2 kB]
Descargados 2.094 kB en 2s (915 kB/s)
Reading package lists... Done
Reading package lists... Done
```

```
cesarmunoz@m3-cesarmunoz:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes adicionales:
 fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjpeg-turbo8 libjpeg8
 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
 libnginx-mod-stream libnginx-mod-stream-geoip2 libtiff5 libwebp7 libxpm4 nginx-common nginx-core
Paquetes sugeridos:
 libgd-tools fcgiwrap nginx-doc ssl-cert
Se instalarán los siguientes paquetes NUEVOS:
 fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjpeg-turbo8 libjpeg8
 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
 libnginx-mod-stream libnginx-mod-stream-geoip2 libtiff5 libwebp7 libxpm4 nginx nginx-common nginx-core
0 actualizados, 20 nuevos se instalarán, 0 para eliminar y 3 no actualizados
```

Modificamos el archivo de configuración /etc/nginx/conf.d/default.conf, y añadimos información acerca de que máquinas queremos hacer el balanceo de carga.

```
GNU nano 6.2 /etc/nginx/nginx.conf *
user www-data;
worker_processes auto;
pid /run/nginx.pid;
#include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
    # multi_accept on;
}

http {

    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    types_hash_max_size 2048;
    # server_tokens off;

    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ##
    # SSL Settings

    [ Read 83 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_/ Go To Line M-E Redo
```

```
GNU nano 6.2 /etc/nginx/conf.d/default.conf *
upstream balanceo_cesarmunoz {
    server 192.168.56.2;
    server 192.168.56.3;
}

server{
    listen 80;
    server_name balanceador_cesarmunoz;
    access_log /var/log/nginx/balanceador_cesarmunoz.access.log;
    error_log /var/log/nginx/balanceador_cesarmunoz.error.log;
    root /var/www/;
    location /
    {
        proxy_pass http://balanceo_cesarmunoz;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}
```

1.2. Configuración avanzada con ponderación, máximo de fallos y conexiones persistentes

Además del servicio por round-robin, podemos darle diferentes parámetros como `weight = NUMBER`, que atribuye a cada servidor un peso para repartir la carga de la manera que nos convenga.

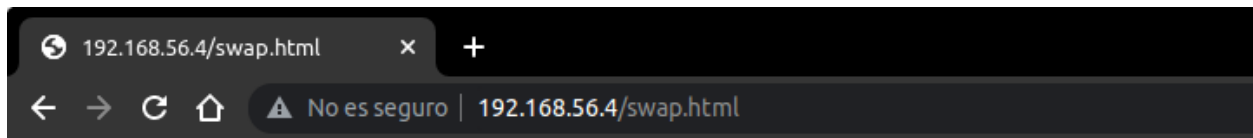
También podemos utilizar `max_fails = NUMBER`, que especifica cuantos intentos fallido máximos puede haber.

Por último podemos usar la directiva `keepalive NUMBER` que especifica cuantas conexiones persistentes deberemos mantener.

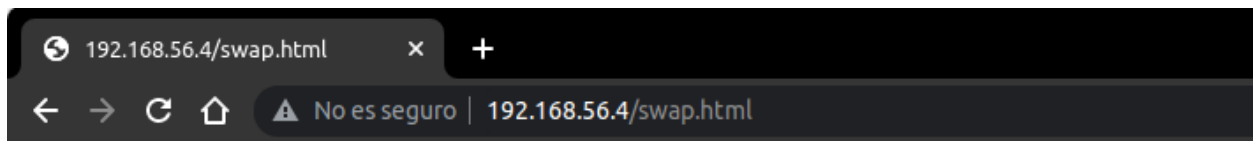
```
upstream balanceo_cesarmunoz {
    server 192.168.56.2 weight=1 max_fails=3;
    server 192.168.56.3 weight=2;
    keepalive 3;
}

server{
    listen 80;
    server_name balanceador_cesarmunoz;
    access_log /var/log/nginx/balanceador_cesarmunoz.access.log;
    error_log /var/log/nginx/balanceador_cesarmunoz.error.log;
    root /var/www/;
    location /
    {
        proxy_pass http://balanceo_cesarmunoz;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}
```

Vemos entonces que cada vez que nos conectemos, el balanceador de carga nos mandará a M1 y M2 dependiendo de la configuración establecida.



MAQUINA M1 Web de ejemplo de cesarmunoz para SWAP Email: cesarmunoz@correo.ugr.es



MAQUINA M2 Web de ejemplo de cesarmunoz para SWAP Email: cesarmunoz@correo.ugr.es

2. Balanceo de carga con HAPROXY

2.1. Instalación y configuración básica

Ahora vamos a utilizar HAPROXY como balanceador de carga, instalamos el paquete de haproxy e iniciamos el servicio con

```
sudo service haproxy start
```

y paramos el servicio nginx si estaba en funcionamiento ya que al estar ambos en el puerto 80, nos daría un error.

```
cesarmunoz@m3-cesarmunoz:~$ sudo apt-get install haproxy
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes adicionales:
  liblua5.3-0
Paquetes sugeridos:
  vim-haproxy haproxy-doc
Se instalarán los siguientes paquetes NUEVOS:
  haproxy liblua5.3-0
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 3 no actualizados.
Se necesita descargar 1.780 kB de archivos.
Se utilizarán 4.193 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://es.archive.ubuntu.com/ubuntu jammy/main amd64 liblua5.3-0 amd64 5.3.6-1build1 [140 kB]
```

Modificamos el archivo de configuración /etc/haproxy/haproxy.cfg, y añadimos información acerca de cual será el frontend y cuales serán los servidores de backend.

```

GNU nano 6.2 /etc/haproxy/haproxy.cfg *
global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    stats socket /var/lib/haproxy/stats
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

    # Default SSL material locations
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private

    # See: https://ssl-config.mozilla.org/#server=haproxy&server-version=2.0.3&config=intermediate
    ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AAA-ECGDSA-SHA256:ECDHE-RSA-AAA-ECGDSA-SHA256:ECDHE-ECDSA-AAA-CHACHA20-POLY1305:ECDHE-RSA-AAA-CHACHA20-POLY1305:ECDHE-ECDSA-AAA-ECGDSA-SHA384:ECDHE-RSA-AAA-ECGDSA-SHA384:ECDHE-ECDSA-AAA-CHACHA20-POLY1305:ECDHE-RSA-AAA-CHACHA20-POLY1305
    ssl-default-bind-ciphersuites TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
    ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets

listen stats
    bind *:9999
    mode http
    stats enable
    stats uri /stats
    stats realm HAProxy\ Statistics
    stats auth cesarmunoz:cesarmunoz

defaults
    log global
    mode http
    option httplog
    option dontlognull
    timeout connect 5000
    timeout client 50000
    timeout server 50000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

frontend http-in
    bind *:80
    default_backend balanceo_cesarmunoz

backend balanceo_cesarmunoz
    server m1 192.168.56.2:80 maxconn 32
    server m2 192.168.56.3:80 maxconn 32

```


podemos tener en el servidor.

Por último podemos usar la directiva cookie COOKIE_NAME que habilita la persistencia de cookies. HAProxy envía una cookie llamada COOKIE_NAME al usuario conectado y lo asocia al backend que primero le ha dado respuesta.

```
GNU nano 6.2 /etc/haproxy/haproxy.cfg
listen stats
    bind *:9999
    mode http
    stats enable
    stats uri /stats
    stats realm HAProxy\ Statistics
    stats auth cesarmunoz:cesarmunoz

defaults
    log global
    mode http
    option httplog
    option dontlognull
    timeout connect 5000
    timeout client 50000
    timeout server 50000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

frontend http-in
    bind *:80
    default_backend balanceo_cesarmunoz

backend balanceo_cesarmunoz
    server m1 192.168.56.2:80 maxconn 32 weight 3
    server m2 192.168.56.3:80 maxconn 32 weight 2
    cookie SERVERUSED insert indirect nocache
```

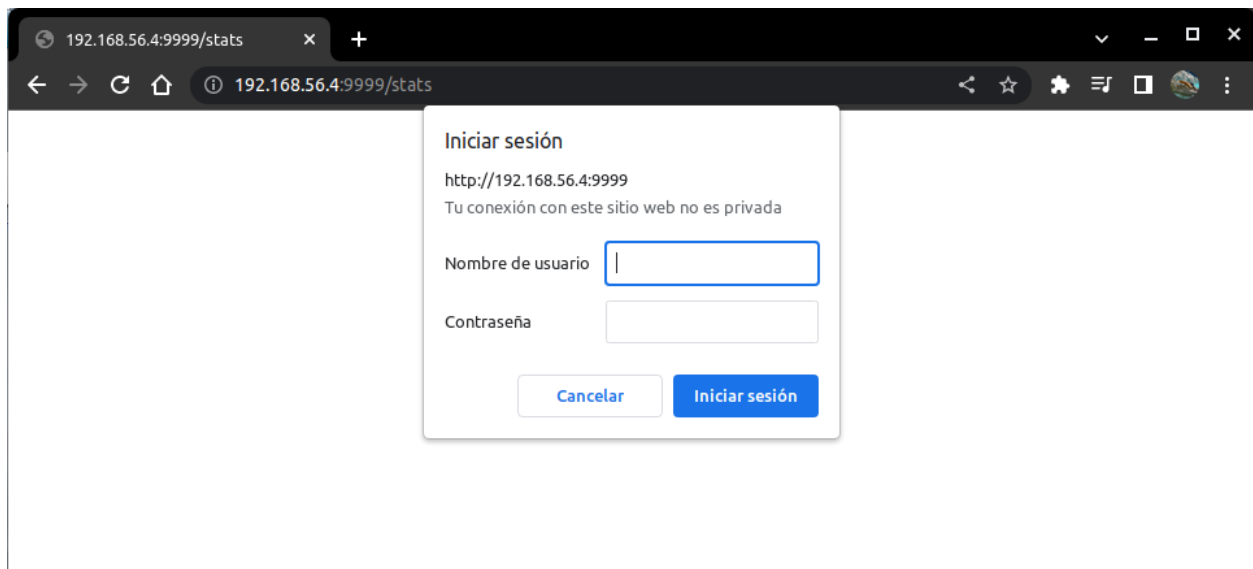
3. Módulo de estadísticas en HAProxy

3.1. Configuración básica con autenticación y puerto de acceso

En el anterior archivo de configuración, hemos añadido el código necesario para poder ver las estadísticas del balanceador de carga HAProxy.

```
listen stats
  bind *:9999
  mode http
  stats enable
  stats uri /stats
  stats realm HAProxy\ Statistics
  stats auth cesarmunoz:cesarmunoz
```

Como podemos ver, habilitamos el puerto 9999 para acceder a las estadísticas por http en la dirección /stats. Por seguridad establecemos una autenticación con el usuario y contraseña cesarmunoz.



Statistics Report for HAProxy x +

No es seguro | 192.168.56.4:9999/stats

HAProxy version 2.4.18-0ubuntu1.2, released 2023/02/13

Statistics Report for pid 1488

> General process information

pid = 1488 (process #1, nbproc = 1, nbthread = 1)
 uptime = 0d 0h 01m 59s
 system limits: memmax = unlimited; ulimit-n = 524288
 maxsock = 524288; maxconn = 262125; maxpipes = 0
 current conns = 1; current pipes = 0/0; conn rate = 0/sec; bit rate = 0.000 kbps
 Running tasks: 0/11; idle = 100 %

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
 JSON export (schema)

External resources:
 Primary site
 Updates (v2.4)
 Online manual

stats

	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	Dwntme	Thrtle		
Frontend	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OPEN		0/0	0	0	0	0	0	0	0
Backend	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1m59s UP		0/0	0	0	0	0	0	0	0	

http-in

	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	Dwntme	Thrtle		
Frontend	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OPEN									

balanceo.cesarmunoz

	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	Dwntme	Thrtle	
m1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	no check		1/1	Y	-	-	-	-	-	-
m2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	no check		1/1	Y	-	-	-	-	-	-
Backend	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1m59s UP		2/2	2	0	0	0	0	0	0

3.2. Configuración avanzada con tasa de refresco y timeouts

Configuramos ahora el archivo /etc/haproxy/haproxy.cfg con stats refresh 30s para tener una tasa de refresco de 30s y así no tener que hacerlo manualmente. También podemos configurar para cada lado del servidor un tiempo máximo de inactividad, a partir del cual se desconectará.

Para ello establecemos timeout client, timeout server y timeout connect a 100s. Así se configura el tiempo para los clientes, servidor y tiempo de intento de conexión con el servidor. Con timeout queue 100s establecemos el tiempo máximo de espera para una conexión con un nuevo slot.

```
listen stats
    bind *:9999
    mode http

    timeout client      100s
    timeout server      100s
    timeout connect     100s
    timeout queue       100s

    stats enable
    stats uri /stats
    stats realm HAProxy\ Statistics
    stats auth cesarmunoz:cesarmunoz
    stats refresh 30s
```

Como añadido, con `sudo haproxy -c -f /etc/haproxy/haproxy.cfg` podemos comprobar si el archivo de configuración es correcto y si por el contrario, tiene errores, donde se encuentran.

4. Benchmarking sobre granja web

4.1. Carga sobre Nginx

Para realizar benchmarking con los balanceadores de carga instalados utilizaremos Apache Benchmark y estableceremos la misma carga a ambos servidores en el archivo de configuración `/etc/nginx/conf.d/default.conf` de la máquina. Luego en un la máquina anfitrión, fuera de la granja web ejecutamos el benchmark:

```
ab -k -n 100000 -c 100 http://192.168.56.4/swap.html
```

donde `-k` permite enviar varias peticiones a la misma `http`, con `-n NUMBER`, enviamos un número de peticiones y con `-c NUMBER` enviamos concurrentemente las peticiones.

Como vemos con el comando `top`, hay una carga de 100 % en la máquina M3, y un reparto de aproximadamente el 50 % en M1 y M2.

```
cesar@cesar-TM1701:~$ ab -k -n 100000 -c 100 http://192.168.56.4/swap.html
This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.56.4 (be patient)
Completed 10000 requests
Completed 20000 requests
Completed 30000 requests
Completed 40000 requests
Completed 50000 requests
Completed 60000 requests
Completed 70000 requests
Completed 80000 requests
Completed 90000 requests
Completed 100000 requests
Finished 100000 requests


Server Software:      nginx/1.18.0
Server Hostname:      192.168.56.4
Server Port:          80

Document Path:        /swap.html
Document Length:      112 bytes

Concurrency Level:    100
Time taken for tests:  23.033 seconds
Complete requests:    100000
Failed requests:       0
Keep-Alive requests:  99049
Total transferred:    38595245 bytes
HTML transferred:    11200000 bytes
Requests per second:  4341.68 [#/sec] (mean)
Time per request:     23.033 [ms] (mean)
Time per request:     0.230 [ms] (mean, across all concurrent requests)
Transfer rate:        1636.41 [Kbytes/sec] received
```

```

top - 17:39:36 up 7:45, 2 users, load average: 0,61, 0,37, 0,20
Tasks: 108 total, 3 running, 105 sleeping, 0 stopped, 0 zombie
%cpu(s): 30,7 us, 32,0 sy, 0,0 ni, 1,0 id, 0,0 wa, 0,0 hi, 36,3 si, 0,0 st
MiB Mem : 969,5 total, 275,6 free, 180,7 used, 513,2 buff/cache
MiB Swap: 1378,0 total, 1378,0 free, 0,0 used, 646,0 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2812	www-data	20	0	14484	8112	3176	R	96,0	0,8	0:29.86	nginx
13	root	20	0	0	0	0	R	1,7	0,0	0:10.69	ksoftirqd/0
2813	cesarmu+	20	0	10504	3876	3264	R	0,7	0,4	0:00.17	top
14	root	20	0	0	0	0	I	0,3	0,0	0:07.74	rcu_sched
1	root	20	0	100968	11804	8288	S	0,0	1,2	0:16.65	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	netns
8	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H-events_highpri
10	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_percpu_wq
11	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_tasks_rude
12	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_tasks_trace
15	root	rt	0	0	0	0	S	0,0	0,0	0:00.56	migration/0
16	root	-51	0	0	0	0	S	0,0	0,0	0:00.00	idle_inject/0
18	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/0
19	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kdevtmpfs
20	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	inet_frag_wq
21	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kauditd
22	root	20	0	0	0	0	S	0,0	0,0	0:00.02	khungtaskd
23	root	20	0	0	0	0	S	0,0	0,0	0:00.00	oom_reaper
24	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	writeback
25	root	20	0	0	0	0	S	0,0	0,0	0:03.79	kcompactd0
26	root	25	5	0	0	0	S	0,0	0,0	0:00.00	ksmd
27	root	39	19	0	0	0	S	0,0	0,0	0:00.00	khugepaged
73	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kintegrityd
74	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kblockd
75	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	blkcg_punt_bio
76	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	tpm_dev_wq
77	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	ata_sff
78	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	md
79	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	edac-poller
80	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	devfreq_wq
81	root	-51	0	0	0	0	S	0,0	0,0	0:00.00	watchdogd
83	root	0	-20	0	0	0	I	0,0	0,0	0:00.49	kworker/0:1H-kblockd
85	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kswapd0
86	root	20	0	0	0	0	S	0,0	0,0	0:00.00	ecryptfs-kthrea
88	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kthrotld
90	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	acpi_thermal_pm
91	root	20	0	0	0	0	S	0,0	0,0	0:00.00	scsi_eh_0
92	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	scsi_tmf_0
93	root	20	0	0	0	0	S	0,0	0,0	0:00.00	scsi_eh_1
94	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	scsi_tmf_1
96	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	vfiio-irqfd-clea
98	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mld
99	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	ip6v6_addrconf
109	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kstrp

```

top - 17:39:04 up 7:44, 2 users, load average: 0,29, 0,19, 0,10
Tasks: 113 total, 3 running, 110 sleeping, 0 stopped, 0 zombie
%cpu(s): 23,8 us, 25,3 sy, 0,0 ni, 33,7 id, 0,0 wa, 0,0 hi, 17,2 si, 0,0 st
MiB Mem : 969,5 total, 57,7 free, 534,7 used, 377,1 buff/cache
MiB Swap: 1384,0 total, 1382,7 free, 1,3 used, 289,6 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
725	www-data	20	0	754104	6592	3500	S	29,8	0,7	1:13.00	apache2
724	www-data	20	0	754216	6712	3500	S	29,5	0,7	1:12.55	apache2
13	root	20	0	0	0	0	R	1,0	0,0	0:05.83	ksoftirqd/0
789	mysqld	20	0	1332832	393148	35244	S	0,7	39,6	5:49.56	mysqld
14	root	20	0	0	0	0	R	0,3	0,0	0:06.36	rcu_sched
2673	cesarmu+	20	0	10492	3944	3368	R	0,3	0,4	0:01.04	top
1	root	20	0	100920	11684	8292	S	0,0	1,2	0:01.79	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.01	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	netns
8	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H-events_highpri
10	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_percpu_wq
11	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_tasks_rude
12	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_tasks_trace
15	root	rt	0	0	0	0	S	0,0	0,0	0:00.56	migration/0
16	root	-51	0	0	0	0	S	0,0	0,0	0:00.00	idle_inject/0
17	root	20	0	0	0	0	I	0,0	0,0	0:00.04	kworker/0:1-cgroup_destroy
18	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/0
19	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kdevtmpfs
20	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	inet_frag_wq
21	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kauditd
22	root	20	0	0	0	0	S	0,0	0,0	0:00.02	khungtaskd
23	root	20	0	0	0	0	S	0,0	0,0	0:00.00	oom_reaper
24	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	writeback
25	root	20	0	0	0	0	S	0,0	0,0	0:01.08	kcompactd0
26	root	25	5	0	0	0	S	0,0	0,0	0:00.00	ksmd
27	root	39	19	0	0	0	S	0,0	0,0	0:00.00	khugepaged
73	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kintegrityd
74	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kblockd
75	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	blkcg_punt_bio
76	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	tpm_dev_wq
77	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	ata_sff
78	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	md
79	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	edac-poller
80	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	devfreq_wq
81	root	-51	0	0	0	0	S	0,0	0,0	0:00.00	watchdogd
83	root	0	-20	0	0	0	I	0,0	0,0	0:00.53	kworker/0:1H-kblockd
85	root	20	0	0	0	0	S	0,0	0,0	0:00.11	kswapd0
86	root	20	0	0	0	0	S	0,0	0,0	0:00.00	ecryptfs-kthrea
88	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kthrotld
89	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	acpi_thermal_pm
91	root	20	0	0	0	0	S	0,0	0,0	0:00.01	scsi_eh_0
92	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	scsi_tmf_0
93	root	20	0	0	0	0	S	0,0	0,0	0:00.00	scsi_eh_1
94	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	scsi_tmf_1
96	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	vfiio-irqfd-clea

```

top - 17:39:16 up 7:44, 2 users, load average: 0.40, 0.25, 0.12
Tasks: 111 total, 4 running, 107 sleeping, 0 stopped, 0 zombie
%Cpu(s): 23.2 us, 23.9 sy, 0.0 ni, 32.4 id, 0.0 wa, 0.0 hi, 20.4 si, 0.0 st
MiB Mem : 969.5 total, 38.8 free, 312.3 used, 398.4 buff/cache
MiB Swap: 1384.0 total, 1324.5 free, 59.5 used, 310.8 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 724 www-data  20   0 754328 6548 3404  S   29.5   0.7   1:24.00 apache2
 723 www-data  20   0 754216 6560 3400  S   29.1   0.7   1:24.28 apache2
 13 root       20   0      0      0      0  R    1.3   0.0   0:13.04 ksoftirqd/0
 790 mysql     20   0 1332876 319212 15212  S    1.0  32.2   5:38.00 mysqld
 14 root       20   0      0      0      0  R    0.3   0.0   0:07.29 rcu_sched
1212 cesarmu+  20   0 17208  6936 4728  S    0.3   0.7   0:39.99 sshd
  1 root       20   0 109844  9944 6556  S    0.0   1.0   0:01.94 systemd
  2 root       20   0      0      0      0  S    0.0   0.0   0:00.01 kthreadd
  3 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 rcu_gp
  4 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 rcu_par_gp
  5 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 slab_flushwq
  6 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 netns
  8 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 kworker/0:0H-events_highpri
10 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 mm_percpu_wq
11 root       20   0      0      0      0  S    0.0   0.0   0:00.00 rcu_tasks_rude
12 root       20   0      0      0      0  S    0.0   0.0   0:00.00 rcu_tasks_trace
15 root       rt   0      0      0      0  S    0.0   0.0   0:00.56 migration/0
16 root      -51   0      0      0      0  S    0.0   0.0   0:00.00 idle_inject/0
18 root       20   0      0      0      0  S    0.0   0.0   0:00.00 cpuhp/0
19 root       20   0      0      0      0  S    0.0   0.0   0:00.00 kdevtmpfs
20 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 inet_frag_wq
21 root       20   0      0      0      0  S    0.0   0.0   0:00.00 kauditd
22 root       20   0      0      0      0  S    0.0   0.0   0:00.02 khungtaskd
23 root       20   0      0      0      0  S    0.0   0.0   0:00.00 oom_reaper
24 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 writeback
25 root       20   0      0      0      0  S    0.0   0.0   0:01.00 kcompactd0
26 root       25   5      0      0      0  S    0.0   0.0   0:00.00 ksmd
27 root       39  19      0      0      0  S    0.0   0.0   0:00.00 khugepaged
73 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 kintegrityd
74 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 kblockd
75 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 blkcg_punt_bio
76 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 tpm_dev_wq
77 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 ata_sff
78 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 md
79 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 edac-poller
80 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 devfreq_wq
81 root      -51   0      0      0      0  S    0.0   0.0   0:00.00 watchdogd
83 root       0 -20      0      0      0  I    0.0   0.0   0:00.59 kworker/0:1H-kblockd
85 root       20   0      0      0      0  S    0.0   0.0   0:00.36 kswapd0
86 root       20   0      0      0      0  S    0.0   0.0   0:00.00 ecryptfs-kthrea
88 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 kthrotld
89 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 acpi_thermal_pm
91 root       20   0      0      0      0  S    0.0   0.0   0:00.01 scsi_eh_0
92 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 scsi_tmf_0
93 root       20   0      0      0      0  S    0.0   0.0   0:00.00 scsi_eh_1
94 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 scsi_tmf_1
96 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 vfio-lirqfd-clea
98 root       0 -20      0      0      0  I    0.0   0.0   0:00.00 mld

```

4.2. Carga sobre HAProxy

En HAProxy, volveremos a utilizar Apache Benchmark, por lo que estableceremos la misma carga a ambos servidores en el archivo de configuración /etc/haproxy/haproxy.cfg. Luego en un la máquina anfitrión, fuera de la granja web ejecutamos el benchmark:

```
ab -k -n 100000 -c 100 http://192.168.56.4/swap.html
```

donde -k permite enviar varias peticiones a la misma http, con -n NUMBER, enviamos un número de peticiones y con -c NUMBER enviamos concurrentemente las peticiones.

Como vemos con el comando top, hay una carga de 100 % en la máquina M3, y un reparto de aproximadamente el 50 % en M1 y M2.

```

cesar@cesar-TM1701:~$ ab -k -n 100000 -c 100 http://192.168.56.4/swap.html
This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.56.4 (be patient)
Completed 10000 requests
Completed 20000 requests
Completed 30000 requests
Completed 40000 requests
Completed 50000 requests
Completed 60000 requests
Completed 70000 requests
Completed 80000 requests
Completed 90000 requests
Completed 100000 requests
Finished 100000 requests


Server Software:      Apache/2.4.52
Server Hostname:      192.168.56.4
Server Port:          80

Document Path:        /swap.html
Document Length:      112 bytes

Concurrency Level:    100
Time taken for tests:  21.430 seconds
Complete requests:    100000
Failed requests:      0
Keep-Alive requests:  100000
Total transferred:    51362597 bytes
HTML transferred:    11200000 bytes
Requests per second:  4666.28 [#/sec] (mean)
Time per request:     21.430 [ms] (mean)
Time per request:     0.214 [ms] (mean, across all concurrent requests)
Transfer rate:        2340.55 [Kbytes/sec] received


Connection Times (ms)
              min      mean[+/-sd] median   max
Connect:     0       0   0.1      0      5
Processing:   7      21   4.9     21     161
Waiting:     2      21   4.9     21     161
Total:       7      21   4.9     21     163

```

```

top - 20:37:37 up 10:43, 2 users, load average: 0.44, 0.10, 0.03
Tasks: 111 total, 4 running, 107 sleeping, 0 stopped, 0 zombie
%Cpu(s): 25.2 us, 55.6 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 19.2 si, 0.0 st
MiB Mem : 969,5 total, 66,1 free, 244,4 used, 659,0 buff/cache
MiB Swap: 1378,0 total, 1377,5 free, 0,5 used, 580,9 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3512	haproxy	20	0	179424	41724	3376	R	62,6	4,2	0:08.35	haproxy
652	syslog	20	0	222400	5772	4112	S	35,8	0,6	0:21.52	rsyslogd
13	root	20	0	0	0	0	R	0,7	0,0	0:11.91	ksoftirqd/0
1	root	20	0	100832	11832	8292	S	0,0	1,2	0:18.93	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.01	kthread
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	netns
8	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H-events_highpri
10	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_percpu_wq
11	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_tasks_rude
12	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_tasks_trace
14	root	20	0	0	0	0	R	0,0	0,0	0:09.98	rcu_sched
15	root	rt	0	0	0	0	S	0,0	0,0	0:00.81	migration/0
16	root	-51	0	0	0	0	S	0,0	0,0	0:00.00	idle_inject/0
18	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/0
19	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kdevtmpfs
20	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	inet_frag_wq
21	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kauditd
22	root	20	0	0	0	0	S	0,0	0,0	0:00.02	khungtaskd
23	root	20	0	0	0	0	S	0,0	0,0	0:00.00	oom_reaper
24	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	writeback
25	root	20	0	0	0	0	S	0,0	0,0	0:05.15	kcompactd0
26	root	25	5	0	0	0	S	0,0	0,0	0:00.00	kswapd0
27	root	39	19	0	0	0	S	0,0	0,0	0:00.00	khugepaged
73	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kintegrityd
74	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kblockd
75	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	blkcg_punt_bio
76	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	tpm_dev_wq
77	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	ata_sff
78	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	md
79	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	edac-poller
80	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	devfreq_wq
81	root	-51	0	0	0	0	S	0,0	0,0	0:00.00	watchdogd
83	root	0	-20	0	0	0	I	0,0	0,0	0:00.63	kworker/0:1H-kblockd
85	root	20	0	0	0	0	S	0,0	0,0	0:00.05	kswapd0
86	root	20	0	0	0	0	S	0,0	0,0	0:00.00	ecryptfs-kthrea
88	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kthrotld
90	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	acpi_thermal_pm
91	root	20	0	0	0	0	S	0,0	0,0	0:00.00	scsi_eh_0
92	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	scsi_tmf_0
93	root	20	0	0	0	0	S	0,0	0,0	0:00.00	scsi_eh_1
94	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	scsi_tmf_1
96	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	vfiio-irqfd-clea
98	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mld
99	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	ipv6_addrconf
100	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kstrp


```

top - 06:45:34 up 10:44, 2 users, load average: 0.41, 0.15, 0.05
Tasks: 117 total, 3 running, 114 sleeping, 0 stopped, 0 zombie
%Cpu(s): 19.3 us, 20.7 sy, 0.0 ni, 54.1 id, 0.0 wa, 0.0 hi, 5.9 si, 0.0 st
MiB Mem : 969.5 total, 65.1 free, 531.1 used, 373.3 buff/cache
MiB Swap: 1384.0 total, 1323.5 free, 60.5 used, 300.3 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
724	www-data	20	0	754104	5984	3356	S	22.4	0.6	1:30.87	apache2
725	www-data	20	0	754104	5928	3356	S	21.8	0.6	1:31.42	apache2
13	root	20	0	0	0	0	R	0.3	0.0	0:07.53	ksoftirqd/0
789	mysql	20	0	1332832	320868	14016	S	0.3	32.2	7:49.97	mysqld
1	root	20	0	100920	10664	7272	S	0.0	1.1	0:02.01	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.02	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_rude
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_trace
14	root	20	0	0	0	0	R	0.0	0.0	0:09.19	rcu_sched
15	root	rt	0	0	0	0	S	0.0	0.0	0:00.76	migration/0
16	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
17	root	20	0	0	0	0	I	0.0	0.0	0:00.04	kworker/0:1-cgroup_destroy
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
20	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	inet_frag_wq
21	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kauditd
22	root	20	0	0	0	0	S	0.0	0.0	0:00.03	khungtaskd

```

top - 20:38:32 up 10:44, 2 users, load average: 0.58, 0.15, 0.04
Tasks: 110 total, 3 running, 107 sleeping, 0 stopped, 0 zombie
%Cpu(s): 19.7 us, 20.9 sy, 0.0 ni, 53.6 id, 0.0 wa, 0.0 hi, 6.4 si, 0.0 st
MiB Mem : 969.5 total, 65.2 free, 479.3 used, 425.1 buff/cache
MiB Swap: 1384.0 total, 1324.5 free, 59.5 used, 340.8 avail Mem

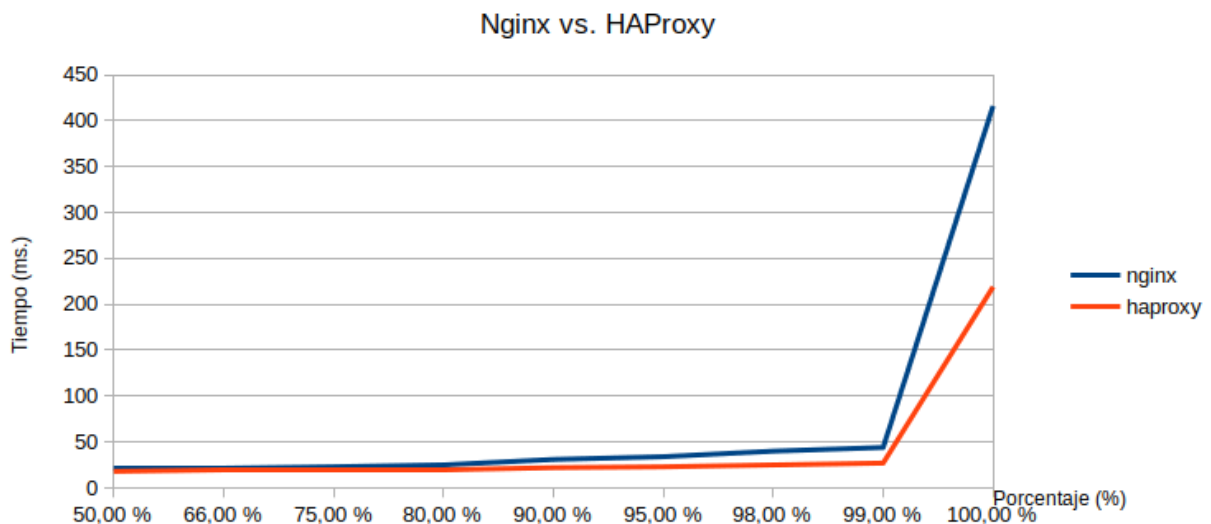
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
723	www-data	20	0	754100	6552	3400	S	22.8	0.7	1:41.07	apache2
724	www-data	20	0	754216	6552	3404	S	21.5	0.7	1:41.21	apache2
13	root	20	0	0	0	0	R	0.7	0.0	0:14.62	ksoftirqd/0
790	mysql	20	0	1332876	319212	15212	S	0.7	32.2	7:33.80	mysqld
3274	cesarmu+	20	0	10492	3944	3364	R	0.3	0.4	0:36.84	top
1	root	20	0	100844	9944	6556	S	0.0	1.0	0:02.07	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_rude
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_trace
14	root	20	0	0	0	0	R	0.0	0.0	0:09.49	rcu_sched
15	root	rt	0	0	0	0	S	0.0	0.0	0:00.72	migration/0
16	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
20	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	inet_frag_wq
21	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kauditd
22	root	20	0	0	0	0	S	0.0	0.0	0:00.03	khungtaskd
23	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper

5. Análisis comparativo de distintos balanceados en base a la carga con AB

Realizamos una comparativa de los balanceadores de carga Ngin y HAProxy con Apache Benchmark. En base a las anteriores ejecuciones del benchmark, con un total de 10000 peticiones al balanceador y con concurrencia de 100, en función del porcentaje de peticiones resueltas, la siguiente tabla:

Porcentaje	Nginx	HAProxy
50,00 %	21	18
66,00 %	22	19
75,00 %	23	20
80,00 %	25	20
90,00 %	31	22
95,00 %	34	23
98,00 %	40	25
99,00 %	44	27
100,00 %	416	219



Vemos pues que hasta el 80 % del benchmark, ambos balanceadores tardan prácticamente lo mismo. A partir del 80 % HAProxy es mejor, resolviendo más peticiones por segundo que Nginx, cuando llega al 99 % el tiempo de ambos se dispara, siendo el de HAProxy el doble de rápido. Para ser exactos, para Nginx tenemos una media de 4341.68 peticiones por segundo, mientras que para HAProxy tenemos una media de 5466.24 peticiones por segundo.

6. Instalación y configuración de Go-between y Zevenet

6.1. Go-between

Probaremos otros balanceadores de carga diferentes a Nginx y HAProxy, el primero es Go-between, lo instalaremos con un paquete snap.

```
sudo apt update && sudo apt install snapd && sudo snap install gobetween --edge
```

Luego, debemos parar los servicios de los balanceadores de carga anteriores para dar paso a go-between. Además deberemos configurar el archivo `/var/snap/gobetween/common/gobetween.toml` y en la sección del servidor definimos las ip de las máquinas M1 y M2.

Al igual que en los anteriores balanceadores, podemos establecer parámetros para el peso que tiene cada una, así como la prioridad que tienen, con `max_connections`, podemos elegir el límite de conexiones que podemos tener o con `priority` definir si queremos prioridad en esa máquina o no. Iniciamos el servicio con

```
sudo snap start gobetween
```

```
[servers]

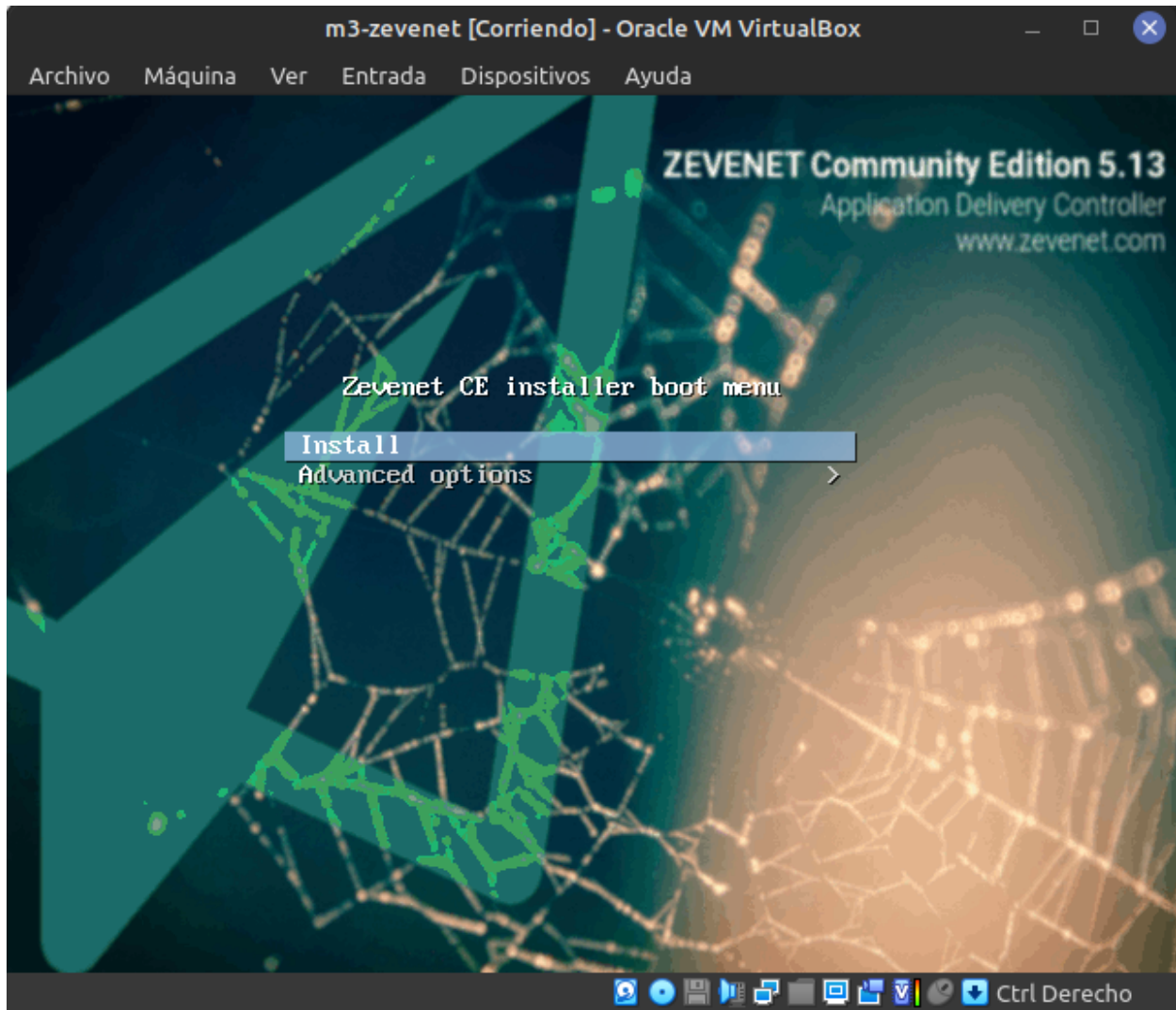
# ----- tcp example ----- #

[servers.sample]
protocol = "tcp"
bind = "*:80"

[servers.sample.discovery]
kind = "static"
static_list = [
    "192.168.56.2:80 weight=2 priority=1",
    "192.168.56.3:80 weight=3 priority=1",
]
```

6.2. Zevenet

Para instalar Zevenet, nos descargamos la imagen iso de la página oficial de Zevenet y la ponemos en un máquina virtual. Durante la instalación ponemos la ip estática 192.168.56.5 con máscara 255.255.255.0 y pasarela 192.168.56.1.



Completamos la instalación instalando el grub en ambas particiones, iniciamos sesión con usuario `debian` y contraseña `1234` y nos debería salir algo así:

```
m3-zevenet [Corriendo] - Oracle VM VirtualBox
```

Archivo Máquina Ver Entrada Dispositivos Ayuda

```
Debian GNU/Linux buster/sid debian tty1  
debian login: debian  
Password:  
Linux debian 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2 (2019-08-28) x86_64  
  
Z E V E N E T  
  
ZEVENET Community Edition  
Software developed by ZEVENET SL  
  
If you want community support contact  
zevenet-ce-users@zevenet.com  
  
or if need professional support open a ticket at  
https://central.zevenet.com/  
  
Get information about our support services visiting  
https://www.zevenet.com/support/  
  
debian@debian:~$ _
```