



UNIVERSIDAD DE GRANADA

Servidores Web de Altas Prestaciones

Práctica 3 – Balanceo de Carga de un Sitio Web

Arturo Alonso Carbonero

ÍNDICE

1. Introducción

2. Objetivos básicos

2.1. Nginx

2.2. HAProxy

2.3. Test comparativo

3. Objetivos extra

3.1. Nginx

3.2. HAProxy

3.3. Gobetween

3.4. Zevenet

3.5. Test comparativo

4. Referencias

1. Introducción

Para poder llevar a cabo la práctica ha sido necesaria la creación de una tercera máquina, que actuará como balanceadora de carga de la granja, para repartir las peticiones entre las otras dos. Para poder proceder con el balanceo, es necesario que dicha máquina disponga de conexión a internet y a al resto de máquinas. A continuación se muestra la configuración de red de dicha máquina, cuyo nombre es **m3-alonsoarturo**, siguiendo el estándar de las prácticas anteriores.

```
alonsoarturo@m3-alonsoarturo:~$ ip -c a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:8d:b6:12 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 84438sec preferred_lft 84438sec
    inet6 fe80::a00:27ff:fe8d:b612/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:d9:91:c9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.250.7/24 brd 192.168.250.255 scope global dynamic enp0s8
        valid_lft 440sec preferred_lft 440sec
    inet6 fe80::a00:27ff:fed9:91c9/64 scope link
        valid_lft forever preferred_lft forever
```

Redes de la máquina m3-alonsoarturo

Además, se han establecido como ficheros principales para mostrar en las otras dos máquinas, modificando el fichero de sitios disponibles de Apache (tal y como en la primera práctica), dos ficheros con extensión *.html* que indican a qué máquina pertenecen, de forma que podamos distinguir si los distintos balanceadores están realizando correctamente su trabajo.

```
alonsoarturo@m1-alonsoarturo:~$ cat /var/www/html/swap.html
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
alonsoarturo@m1-alonsoarturo:~$ ip -c a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:3b:07:15 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 85486sec preferred_lft 85486sec
    inet6 fe80::a00:27ff:fe3b:715/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:8d:09:73 brd ff:ff:ff:ff:ff:ff
    inet 192.168.250.4/24 brd 192.168.250.255 scope global dynamic enp0s8
        valid_lft 586sec preferred_lft 586sec
    inet6 fe80::a00:27ff:fe8d:973/64 scope link
        valid_lft forever preferred_lft forever
```

Fichero e IP de la máquina m1-alonsoarturo

```

alonsoarturo@m2-alonsoarturo:~$ cat /var/www/html/swap.html
<html>
  <body>
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
alonsoarturo@m2-alonsoarturo:~$ ip -c a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    link/ether 08:00:27:b1:f0:55 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 85459sec preferred_lft 85459sec
    inet6 fe80::a00:27ff:feb1:f055/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    link/ether 08:00:27:f1:33:1d brd ff:ff:ff:ff:ff:ff
    inet 192.168.250.3/24 brd 192.168.250.255 scope global dynamic enp0s8
        valid_lft 560sec preferred_lft 560sec
    inet 192.168.250.6/24 brd 192.168.250.255 scope global secondary enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fef1:331d/64 scope link
        valid_lft forever preferred_lft forever

```

Fichero e IP de la máquina m2-alonsoarturo

2. Objetivos básicos

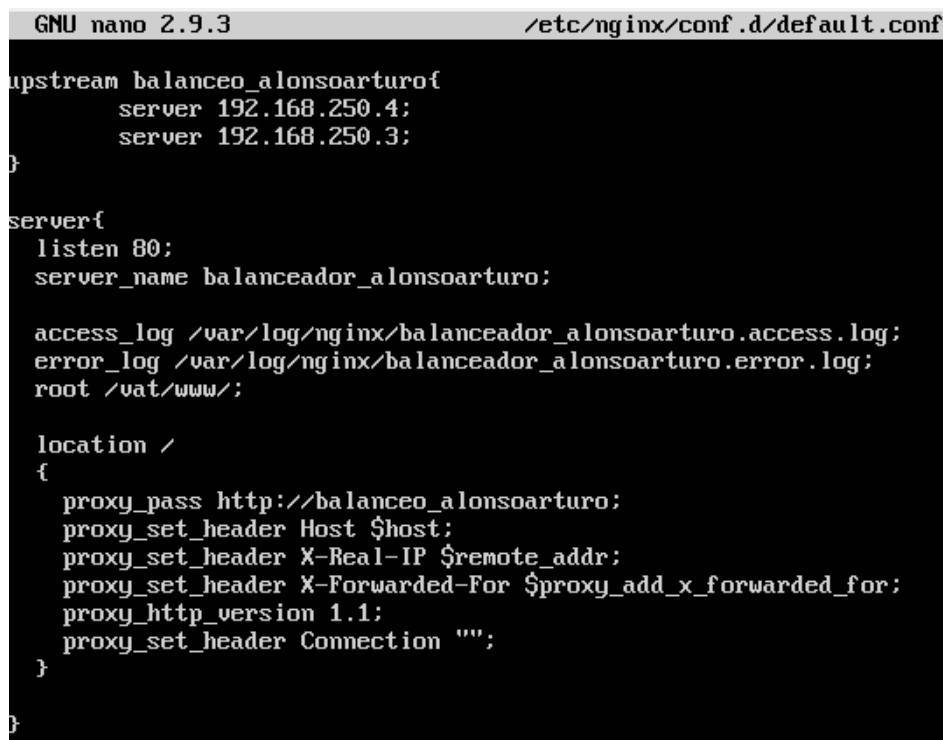
2.1. Nginx

Para poder trabajar con Nginx es necesario instalar el servicio en la máquina, para lo que basta con ejecutar el comando **sudo apt-get install nginx**. Para comprobar si se ha instalado correctamente, hacemos uso del comando **systemctl status nginx**. Si todo funciona correctamente, podemos pasar a la fase de configuración.

En primer lugar, accedemos al fichero de configuración de Nginx (**/etc/nginx/nginx.conf**) y comentamos la línea de la imagen siguiente, ya que la configuración básica se corresponde con la de servidor web.

```
include /etc/nginx/conf.d/*.conf;  
# include /etc/nginx/sites-enabled/*;
```

A continuación, definimos qué grupo de máquinas formaran el clúster en la función *upstream*. Además, indicamos en la función *server* que utilice ese grupo de máquinas, así como otra serie de directivas que debe tener en cuenta el servicio a la hora de procesar las peticiones, como borrar la cabecera *Connection* o escuchar en el puerto 80. Todo ello se incluye en el fichero **/etc/nginx/conf.d/default.conf**. Cabe destacar que la máquina ha sido creada desde 0, por lo que no tiene Apache instalado y el puerto 80 está libre, pero en el caso de que algún otro servicio esté utilizando dicho puerto, debemos liberarlo.



```
GNU nano 2.9.3 /etc/nginx/conf.d/default.conf  
  
upstream balanceo_alonsoarturo{  
    server 192.168.250.4;  
    server 192.168.250.3;  
}  
  
server{  
    listen 80;  
    server_name balanceador_alonsoarturo;  
  
    access_log /var/log/nginx/balanceador_alonsoarturo.access.log;  
    error_log /var/log/nginx/balanceador_alonsoarturo.error.log;  
    root /vat/www/;  
  
    location /  
    {  
        proxy_pass http://balanceo_alonsoarturo;  
        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 "";  
    }  
}
```

Configuración de upstream y server

Tras esto, reiniciamos el servicio con **sudo systemctl restart nginx** y comprobamos que funciona correctamente mediante el mismo comando mencionado tras el proceso de instalación.

```
alonsoarturo@m3-alonsoarturo:~$ systemctl status nginx
• nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2023-04-23 19:53:16 CEST; 1min 9s ago
     Docs: man:nginx(8)
  Process: 1656 ExecStop=/sbin/start-stop-daemon --quiet --stop --retry QUIT/5 --pidfile /run/nginx.
  Process: 1670 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SU
  Process: 1659 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, st
 Main PID: 1671 (nginx)
    Tasks: 2 (limit: 2314)
   CGroup: /system.slice/nginx.service
           └─1671 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─1673 nginx: worker process

abr 23 19:53:16 m3-alonsoarturo systemd[1]: Stopped A high performance web server and a reverse proxy
abr 23 19:53:16 m3-alonsoarturo systemd[1]: Starting A high performance web server and a reverse proxy
abr 23 19:53:16 m3-alonsoarturo systemd[1]: nginx.service: Failed to parse PID from file /run/nginx.
abr 23 19:53:16 m3-alonsoarturo systemd[1]: Started A high performance web server and a reverse proxy
```

Nginx activo

La configuración básica de Nginx implementa el algoritmo de balanceo *Round-Robin*, es decir, reparte la carga entre las máquinas por turnos. Para comprobar el funcionamiento del servicio, he sometido a la granja a peticiones mediante el comando **curl 192.168.250.7** (IP de la máquina balanceadora) desde el anfitrión. A continuación se muestran un ejemplo de uso.

```
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl http://192.168.250.7
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> p
  ara SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl http://192.168.250.7
<html>
  <body>
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> p
  ara SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
```

Resultado de Round-Robin con Nginx

De ahora en adelante, para realizar las pruebas, se hará uso de un script (**multiCurl.sh**) que ejecuta el comando **curl** 5 veces seguidas. A continuación se muestra el contenido de dicho script.

```
#!/bin/bash

for i in {1..5}
do
  curl 192.168.250.7
done
```

ip_hash

Si el usuario mantiene algún tipo de estado durante la navegación, es interesante que todas las peticiones vayan a la misma máquina. Para ello, hay que añadir en la función *upstream* la directiva *ip_hash*.

```
upstream balanceo_alonsoarturo{
    ip_hash;
    server 192.168.250.4;
    server 192.168.250.3;
}
```

```
alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Users/Usuario/Desktop/UGR/2º Cuatri/SWAP/Práctica 3$ bash multiCurl.sh
<html>
  <body>
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
```

Resultado con ip_hash

fail_timeout

Mediante la directiva *fail_timeout*, podemos indicar el tiempo máximo de espera a cierta máquina. En este caso, como la máquina con dirección IP **192.168.250.8** no existe, lanzará todas las peticiones a la otra máquina tras esperar el tiempo indicado. Para comprobarlo, se ha ejecutado el script de prueba con el comando **time**.

```
upstream balanceo_alonsoarturo{
    server 192.168.250.4;
    server 192.168.250.8 fail_timeout=5s;
}
```

```
alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Users/Usuario/Desktop/UGR/2° Cuatri/SWAP/Práctica 3$ time bash multiCurl.sh
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
real    0m3.381s
user    0m0.016s
sys     0m0.234s
```

Resultado con espera de 5s

Backup

Se puede indicar a Nginx que una de las máquinas sea para *backup*. De esta forma, si la máquina principal falla (en este caso reiniciamos m1 durante la prueba), redirigirá las peticiones a la máquina secundaria hasta recuperar la otra.

```
upstream balanceo_alonsoarturo{  
    server 192.168.250.4;  
    server 192.168.250.3 backup;  
}
```

```
alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Users/Usuario/Desktop/UGR/2° Cuatri/SWAP/Práctica 3$ curl 192.168.250.7  
<html>  
  <body>  
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP  
    Correo: alonsoarturo@correo.ugr.es  
  </body>  
</html>  
alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Users/Usuario/Desktop/UGR/2° Cuatri/SWAP/Práctica 3$ curl 192.168.250.7  
<html>  
  <body>  
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP  
    Correo: alonsoarturo@correo.ugr.es  
  </body>  
</html>  
alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Users/Usuario/Desktop/UGR/2° Cuatri/SWAP/Práctica 3$ curl 192.168.250.7  
<html>  
  <body>  
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP  
    Correo: alonsoarturo@correo.ugr.es  
  </body>  
</html>  
alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Users/Usuario/Desktop/UGR/2° Cuatri/SWAP/Práctica 3$ curl 192.168.250.7  
<html>  
  <body>  
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP  
    Correo: alonsoarturo@correo.ugr.es  
  </body>  
</html>  
alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Users/Usuario/Desktop/UGR/2° Cuatri/SWAP/Práctica 3$ curl 192.168.250.7  
<html>  
  <body>  
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP  
    Correo: alonsoarturo@correo.ugr.es  
  </body>  
</html>
```

Resultado con backup

2.2. HAProxy

Para poder uso del servicio de HAProxy, ejecutamos el comando **sudo apt-get install haproxy** para instalarlo en la máquina y comprobamos su estado mediante el comando **systemctl**, tal y como en el ejemplo de Nginx. Una vez instalado, hay que configurar el fichero **/etc/haproxy/haproxy.cfg**. Entre otras directrices, es fundamental indicar el puerto por el que debe escuchar, las máquinas de la granja y el tipo de balanceo a emplear (*Round-Robin*).

```
frontend http-in
    bind *:80
    mode http
    default_backend balanceo_alonsoarturo

backend balanceo_alonsoarturo
    balance roundrobin
    option forwardfor
    server m1 192.168.250.4:80 maxconn 32
    server m2 192.168.250.3:80 maxconn 32
```

Configuración de HAProxy

```
alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Users/Usuario/Desktop/UGR/2° Cuatri/SWAP/Práctica 3$ bash multiCurl.sh
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
```

Resultado de Round-Robin en HAProxy

```
global
log /dev/log      local0
log /dev/log      local1 notice
chroot /var/lib/haproxy
stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners
stats timeout 30s
stats socket /var/lib/haproxy/stats
user haproxy
group haproxy
daemon

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

# Default ciphers to use on SSL-enabled listening sockets.
# For more information, see ciphers(1SSL). This list is from:
# https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/
# An alternative list with additional directives can be obtained from
# https://mozilla.github.io/server-side-tls/ssl-config-generator/?server=haproxy
ssl-default-bind-ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA$
ssl-default-bind-options no-sslv3

listen stats
bind *:9999
mode http
stats show-node
stats show-legends
stats enable
stats uri /stats
stats realm HAProxy\ Statistics
stats auth alonsoarturo:alonsoarturo
```

Estadísticas de HAProxy

2.3. Test comparativo

Para someter a la granja a altas cantidades de carga, se hará uso del comando **ab** (Apache Benchmark) de la siguiente forma: **ab -n 20000 -c 10 -qd http://192.168.250.7/test.html**. Las opciones añadidas indican lo siguiente:

- **-n 20000** → Número de peticiones.
- **-c 10** → Número de peticiones concurrentes.
- **-q** → Si hay más de 150 peticiones, se muestra el progreso cada 10%.
- **-d** → No mostrar el porcentaje servido.

A continuación se muestra el resultado de ejecutar este comando para cada balanceador. Cabe destacar que ambos utilizan el puerto 80, por lo que es necesario desactivar uno mientras se prueba el otro. Además, se hará uso del comando **top** para verificar que el servicio que está actuando es el deseado.

```
alonsoarturo@DESKTOP-UKJ4G5Q: /mnt/c/Users/Usuario/Desktop/UGR/2º Cuatri/SWAP/Práctica 3$ ab -n 20000 -c 10 -qd http://192.168.250.7/test.html
This is ApacheBench, Version 2.3 <$Revision: 1807734 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.250.7 (be patient).....done

Server Software:      nginx/1.14.0
Server Hostname:      192.168.250.7
Server Port:          80

Document Path:        /test.html
Document Length:      275 bytes

Concurrency Level:    10
Time taken for tests:  37.139 seconds
Complete requests:    20000
Failed requests:       0
Non-2xx responses:    20000
Total transferred:    9080000 bytes
HTML transferred:     5500000 bytes
Requests per second:  538.52 [#/sec] (mean)
Time per request:     18.569 [ms] (mean)
Time per request:     1.857 [ms] (mean, across all concurrent requests)
Transfer rate:        238.76 [Kbytes/sec] received

Connection Times (ms)
  min   mean[+/-sd] median   max
Connect:  0    2   1.8      2    26
Processing: 6   16   6.3     15   286
Waiting:  4   15   6.1     14   286
Total:     8   18   6.5     17   288
```

Test de carga para balanceo Round-Robin con Nginx

```
Tareas: 84 total, 3 ejecutar, 42 hibernar, 0 detener, 0 zombie
%Cpu(s): 8,6 usuario, 25,8 sist, 0,0 adecuado, 26,9 inact, 0,0 en espera, 0,0 hardw int, 38,7 s
KiB Mem : 2040924 total, 1676432 libre, 101600 usado, 262892 búfer/caché
KiB Intercambio: 239696 total, 239696 libre, 0 usado. 1795480 dispon Mem
```

PID	USUARIO	PR	NI	UIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
607	www-data	20	0	143820	6964	5056	R	57,1	0,3	0:44.76	nginx
7	root	20	0	0	0	0	S	2,3	0,0	0:02.02	ksoftirqd/0
8	root	20	0	0	0	0	R	0,3	0,0	0:00.59	rcu_sched
966	alonsoa+	20	0	46732	4148	3516	R	0,3	0,2	0:00.09	top
1	root	20	0	77520	8588	6612	S	0,0	0,4	0:02.53	systemd

Verificación con top

```

alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Users/Usuario/Desktop/UGR/2º Cuatri/SWAP/Práctica 3$ ab -n 20000 -c 10 -qd http://192.168.250.7/test.html
This is ApacheBench, Version 2.3 <$Revision: 1807734 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.250.7 (be patient).....done

Server Software:      Apache/2.4.29
Server Hostname:      192.168.250.7
Server Port:          80

Document Path:        /test.html
Document Length:      275 bytes

Concurrency Level:    10
Time taken for tests:  34.495 seconds
Complete requests:    20000
Failed requests:       0
Non-2xx responses:    20000
Total transferred:    9100000 bytes
HTML transferred:     5500000 bytes
Requests per second:  579.80 [#/sec] (mean)
Time per request:     17.247 [ms] (mean)
Time per request:     1.725 [ms] (mean, across all concurrent requests)
Transfer rate:        257.63 [Kbytes/sec] received

Connection Times (ms)
      min      mean[+/-sd] median    max
Connect:    0       2   1.7      1      26
Processing:  5      15   5.4     14     108
Waiting:    5      14   5.3     13     105
Total:      6      17   5.7     16     110

```

Test de carga para balanceo Round-Robin con HAProxy

```

Tareas: 83 total, 3 ejecutar, 41 hibernar, 0 detener, 0 zombie
%Cpu(s): 14,7 usuario, 27,2 sist, 0,0 adecuado, 20,6 inact, 0,0 en espera, 0,0 hardw int, 37,5 s
KiB Mem : 2040924 total, 1676484 libre, 98940 usado, 265500 búfer/caché
KiB Intercambio: 239696 total, 239696 libre, 0 usado. 1798072 dispon Mem

```

PID	USUARIO	PR	NI	VRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
1006	haproxy	20	0	54544	2732	1380	R	54,3	0,1	0:03.68	haproxy
533	syslog	20	0	263044	4420	3648	S	13,8	0,2	0:00.90	rsyslogd
7	root	20	0	0	0	0	R	2,0	0,0	0:02.47	ksoftirqd/0
8	root	20	0	0	0	0	I	0,7	0,0	0:00.71	rcu_sched
1009	alonsoa+	20	0	46732	4124	3512	R	0,7	0,2	0:00.06	top
1	root	20	0	77520	8632	6612	S	0,0	0,4	0:02.55	systemd

Verificación con top

El número de peticiones por segundo que ha procesado Nginx es de 538.52, mientras que HAProxy ha procesado 579.8, lo que indica que HAProxy ha funcionado mejor para este test concreto. Aun así, ambos han tardado un tiempo razonable en resolver todas las peticiones (37.139s y 34.495s respectivamente) y han completado todas ellas sin fallo.

3. Objetivos extra

3.1. Nginx

Para aplicar el balanceo por ponderación, basta con indicar el peso que se desea asignar a cada máquina en la función *upstream*. En este caso, la máquina m1 tendrá el doble de peso que la máquina m2, ya que suponemos el doble de potencia. Para ello, se indica al lado de la IP de cada máquina el valor del parámetro *weight* deseado. El resultado muestra como por cada petición a la máquina m2, se han llevado a cabo dos peticiones a la máquina m1.

```
upstream balanceo_alonsoarturo{  
    server 192.168.250.4 weight=2;  
    server 192.168.250.3 weight=1;  
}
```

Asignación de pesos

```
alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Users/Usuario/Desktop/UGR/2º Cuatri/SWAP/Práctica 3$ bash multiCurl.sh  
<html>  
  <body>  
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP  
    Correo: alonsoarturo@correo.ugr.es  
  </body>  
</html>  
<html>  
  <body>  
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP  
    Correo: alonsoarturo@correo.ugr.es  
  </body>  
</html>  
<html>  
  <body>  
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP  
    Correo: alonsoarturo@correo.ugr.es  
  </body>  
</html>  
<html>  
  <body>  
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP  
    Correo: alonsoarturo@correo.ugr.es  
  </body>  
</html>  
<html>  
  <body>  
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP  
    Correo: alonsoarturo@correo.ugr.es  
  </body>  
</html>
```

Resultado de balanceo por ponderación con Nginx

Test de carga

```
alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Users/Usuario/Desktop/UGR/2º Cuatri/SWAP/Práctica 3$ ab -n 20000 -c 10 -qd http://192.168.250.7/test.html
This is ApacheBench, Version 2.3 <$Revision: 1807734 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.250.7 (be patient).....done


Server Software:      nginx/1.14.0
Server Hostname:      192.168.250.7
Server Port:          80

Document Path:        /test.html
Document Length:      275 bytes

Concurrency Level:     10
Time taken for tests:  33.940 seconds
Complete requests:     20000
Failed requests:        0
Non-2xx responses:     20000
Total transferred:     9080000 bytes
HTML transferred:      5500000 bytes
Requests per second:   589.27 [#/sec] (mean)
Time per request:      16.970 [ms] (mean)
Time per request:      1.697 [ms] (mean, across all concurrent requests)
Transfer rate:         261.26 [Kbytes/sec] received


Connection Times (ms)
      min    mean[+/-sd] median   max
Connect:    0      2   3.8      1    463
Processing:  6     15  10.5     14   480
Waiting:    5     14  10.5     13   478
Total:      7     17  11.1     15   490
```

Test de carga para balanceo por ponderación con Nginx

3.2. HAProxy

En cuanto a la configuración de HAProxy para que actúe como balanceador por ponderación, es similar a la de Nginx. Basta con añadir el peso (*weight*) a la derecha de cada máquina con el valor deseado.

```
backend balanceo_alonsoarturo
    balance roundrobin
    option forwardfor
    server m1 192.168.250.4:80 maxconn 32 weight 2
    server m2 192.168.250.3:80 maxconn 32 weight 1
```

Asignación de pesos

```
alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Users/Usuario/Desktop/UGR/2º Cuatri/SWAP/Práctica 3$ bash multiCurl.sh
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
```

Resultado de balanceo por ponderación con HAProxy

Test de carga

```
alonsoarturo@DESKTOP-UJKJ4G50:/mnt/c/Users/Usuario/Desktop/UGR/2º Cuatri/SWAP/Práctica 3$ ab -n 20000 -c 10 -qd http://192.168.250.7/test.html
This is ApacheBench, Version 2.3 <$Revision: 1807734 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/
```

```
Benchmarking 192.168.250.7 (be_patient).....done
```

```
Server Software:      Apache/2.4.29
Server Hostname:      192.168.250.7
Server Port:          80
```

```
Document Path:      /test.html
Document Length:    275 bytes
```

```

Concurrency Level:      10
Time taken for tests:   32.885 seconds
Complete requests:      20000
Failed requests:        0
Non-2xx responses:      20000
Total transferred:      9100000 bytes
HTML transferred:       5500000 bytes
Requests per second:    608.18 [#/sec] (mean)
Time per request:       16.442 [ms] (mean)
Time per request:       1.644 [ms] (mean, across all concurrent requests)
Transfer rate:          270.24 [Kbytes/sec] received

```

Connection Times (ms)

	min	mean[+/-sd]	median	max
Connect:	0	1 1.5	1	29
Processing:	6	15 4.9	14	102
Waiting:	6	14 4.8	14	100
Total:	7	16 5.1	15	103

Test de carga para balanceo Round-Robin con HAProxy

HAProxy version 1.8.8-1ubuntu0.13, released 2023/02/13

Statistics Report for pid 1319 on m3-alonsoarturo

> General process information

```
pid = 1319 (process #1, nbproc = 1, nbthread = 1)
uptime = 0d 0h03m23s
system limits: memmax = unlimited; ulimit-n = 4042
maxsock = 4042; maxconn = 2000; maxpipes = 0
current conns = 2; current pipes = 0/0; conn rate = 2/sec
Running tasks: 1/7; idle = 100 %
```

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

Note: "NOI RV/DRAIN" = UP with load-balancing disabled

Display option:

- Scope :
- [Hide 'DOWN' servers](#)
- [Refresh now](#)
- [CSV export](#)

External resources:

- [Primary site](#)
- [Updates \(v1.8\)](#)
- [Online manual](#)

	state																														
	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	Downtime	Thrtle	
Frontend				2	2	-	2	2	2	2000	2				0	0	0	0	0			OPEN									
Backend	0	0	0				2	2	200	0	0	0s	0	0	0	0	0	0	0	0	0	3m23s UP		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	Bok	Chk	Dwn	Dwntme	Thrflte	
Frontend				0	657	-	0	11	2,000	20,007			1,800,385	9,102,318	0	0	2					OPEN									

balanceo_alonsoarturo																															
	Queue						Sessions						Bytes		Denied		Errors		Warnings		Server										
	Curr	Max	Limit	Curr	Max	Limit	Curr	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	439		0	7	32	13 337	13 337	2m4s	1 200 278	8 098 067		0		0	0	0	0	no check		2	Y	-					-
m2	0	0	-	0	220		0	4	32	8 668	8 668	2m4s	800 107	3 033 873		0		0	0	0	0	no check		1	Y	-					-
Backend	0	0		0	658		0	10	200	20 005	20 005	2m4s	1 800 385	9 101 840	0	0	0	0	0	0	0	3m23s UP		3	2	0		0		0s	

Estadísticas de HAProxy tras el test de carga con ab

3.3. Gobetween

Para instalar el servicio de Gobetween hay que ejecutar **sudo snap install gobetween --edge**. Para comprobar que se ha instalado correctamente tenemos dos opciones. Podemos hacer uso de **snap** ejecutando **sudo snap services gobetween** o emplear el comando **systemctl**. A continuación se muestran ambas opciones, así como el uso del comando **netstat** para comprobar que se está escuchando en el puerto deseado.

```
alonsoarturo@m3-alonsoarturo:~$ snap services gobetween
Servicio      Encendido Actual Notas
gobetween.gobetween activado activo -

alonsoarturo@m3-alonsoarturo:~$ sudo systemctl status snap.gobetween.gobetween.service
• snap.gobetween.gobetween.service - Service for snap application gobetween.gobetween
   Loaded: loaded (/etc/systemd/system/snap.gobetween.gobetween.service; enabled; vendor preset: ena
   Active: active (running) since Mon 2023-05-01 20:25:57 CEST; 3min 24s ago
   Main PID: 4035 (bash)
   Tasks: 8 (limit: 2314)
   CGroup: /system.slice/snap.gobetween.gobetween.service
           └─4035 /bin/bash /var/snap/gobetween/57/gobetween.sh
             4069 /snap/gobetween/57/bin/gobetween -c /var/snap/gobetween/common/gobetween.toml

may 01 20:25:57 m3-alonsoarturo systemd[1]: Started Service for snap application gobetween.gobetween
may 01 20:25:57 m3-alonsoarturo gobetween.gobetween[4035]: gobetween v0.8.0+snapshot
may 01 20:25:57 m3-alonsoarturo gobetween.gobetween[4035]: 2023-05-01 20:25:57 [INFO] (manager): In
may 01 20:25:57 m3-alonsoarturo gobetween.gobetween[4035]: 2023-05-01 20:25:57 [INFO] (server): Cre
may 01 20:25:57 m3-alonsoarturo gobetween.gobetween[4035]: 2023-05-01 20:25:57 [INFO] (scheduler):
may 01 20:25:57 m3-alonsoarturo gobetween.gobetween[4035]: 2023-05-01 20:25:57 [INFO] (manager): In
may 01 20:25:57 m3-alonsoarturo gobetween.gobetween[4035]: 2023-05-01 20:25:57 [INFO] (metrics): Me
may 01 20:25:57 m3-alonsoarturo gobetween.gobetween[4035]: 2023-05-01 20:25:57 [INFO] (api): Starti
may 01 20:25:57 m3-alonsoarturo gobetween.gobetween[4035]: 2023-05-01 20:25:57 [INFO] (ani): Starti

alonsoarturo@m3-alonsoarturo:~$ netstat -tulpn
(No todos los procesos pueden ser identificados, no hay información de propiedad del proceso
no se mostrarán, necesita ser superusuario para verlos todos.)
Conexiones activas de Internet (solo servidores)
Proto Recib Envuad Dirección local Dirección remota Estado PID/Program name
tcp 0 0 0 192.168.250.7:80 0.0.0.0:* ESCUCHAR -
tcp 0 0 0 127.0.0.53:53 0.0.0.0:* ESCUCHAR -
tcp6 0 0 0 :::8888 :::* ESCUCHAR -
udp 0 0 0 127.0.0.53:53 0.0.0.0:* -
udp 0 0 0 192.168.250.7:68 0.0.0.0:* -
udp 0 0 0 10.0.2.15:68 0.0.0.0:* -
```

Gobetween activo

Para configurar el servicio, hay que editar el fichero de configuración del mismo y añadir el bloque de la siguiente imagen en el apartado `[servers]`. Indicamos la IP, el puerto, el modo de balanceo y la lista de IPs a manejar, tal y como en el resto de balanceadores. El fichero de configuración es `/var/snap/gobetween/common/gobetween.toml`.

```
# ----- Balanceo Round-Robin - P3 alonsoarturo ----- #

[servers.balanceo_alonsoarturo]
bind = "192.168.250.7:80"
protocol = "tcp"
balance = "roundrobin"

[servers.balanceo_alonsoarturo.discovery]
interval = "1s"
kind = "static"
static_list = [
    "192.168.250.4:80",
    "192.168.250.3:80"
]
```

Configuración de Gobetween para Round-Robin

```
alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Users/Usuario/Desktop/UGR/2º Cuatri/SWAP/Práctica 3$ bash multiCurl.sh
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
```

Resultado de balanceo Round-Robin con Gobetween

Para realizar balanceo por ponderación con Gobetween basta con añadir el peso deseado a cada una de las máquinas tal y como se muestra en la siguiente imagen.

```
# ----- Balanceo Round-Robin - P3 alonsoarturo ----- #

[servers.balanceo_alonsoarturo]
bind = "192.168.250.7:80"
protocol = "tcp"
balance = "weight"

[servers.balanceo_alonsoarturo.discovery]
interval = "1s"
kind = "static"
static_list = [
    "192.168.250.4:80 weight=2",
    "192.168.250.3:80 weight=1"
]
```

Configuración de Gobetween para Round-Robin

```
alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Users/Usuario/Desktop/UGR/2º Cuatri/SWAP/Práctica 3$ bash multiCurl.sh
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
<html>
  <body>
    Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
    Correo: alonsoarturo@correo.ugr.es
  </body>
</html>
```

Resultado de balanceo por ponderación con Gobetween

Test de carga

```
alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Users/Usuario/Desktop/UGR/2º Cuatri/SWAP/Práctica 3$ ab -n 20000 -c 10 -qd http://192.168.250.7/test.html
This is ApacheBench, Version 2.3 <$Revision: 1807734 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.250.7 (be patient).....done

Server Software:      Apache/2.4.29
Server Hostname:      192.168.250.7
Server Port:          80

Document Path:        /test.html
Document Length:      275 bytes

Concurrency Level:    10
Time taken for tests:  73.868 seconds
Complete requests:    20000
Failed requests:       0
Non-2xx responses:    20000
Total transferred:    9100000 bytes
HTML transferred:     5500000 bytes
Requests per second:  270.75 [#/sec] (mean)
Time per request:     36.934 [ms] (mean)
Time per request:     3.693 [ms] (mean, across all concurrent requests)
Transfer rate:        120.31 [Kbytes/sec] received

Connection Times (ms)
      min      mean[+/-sd] median   max
Connect:    1      2   1.8      1     31
Processing:  8     35  13.6     32    360
Waiting:    8     33  13.1     31    359
Total:       8     37  13.5     34    361
```

Test de carga para balanceo Round-Robin con Gobetween

```
alonsoarturo@DESKTOP-UKJ4G5Q:/mnt/c/Users/Usuario/Desktop/UGR/2º Cuatri/SWAP/Práctica 3$ ab -n 20000 -c 10 -qd http://192.168.250.7/test.html
This is ApacheBench, Version 2.3 <$Revision: 1807734 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.250.7 (be patient).....done

Server Software:      Apache/2.4.29
Server Hostname:      192.168.250.7
Server Port:          80

Document Path:        /test.html
Document Length:      275 bytes

Concurrency Level:    10
Time taken for tests:  77.787 seconds
Complete requests:    20000
Failed requests:       0
Non-2xx responses:    20000
Total transferred:    9100000 bytes
HTML transferred:     5500000 bytes
Requests per second:  257.11 [#/sec] (mean)
Time per request:     38.894 [ms] (mean)
Time per request:     3.889 [ms] (mean, across all concurrent requests)
Transfer rate:        114.24 [Kbytes/sec] received

Connection Times (ms)
      min      mean[+/-sd] median   max
Connect:    1      2   1.8      1     26
Processing: 10     36  12.7     35    129
Waiting:    10     35  12.5     33    129
Total:       11     38  12.5     37    130
```

Test de carga para balanceo por ponderación con Gobetween

```
Tareas: 89 total, 2 ejecutar, 46 hibernar, 2 detener, 0 zombie
%Cpu(s): 0,8 usuario, 0,7 sist, 0,2 adecuado, 98,0 inact, 0,2 en espera, 0,0 hardw int, 0,1 s
KiB Mem : 2040924 total, 1082608 libre, 155060 usado, 803256 búfer/caché
KiB Intercambio: 239696 total, 239696 libre, 0 usado. 1740608 dispon Mem
```

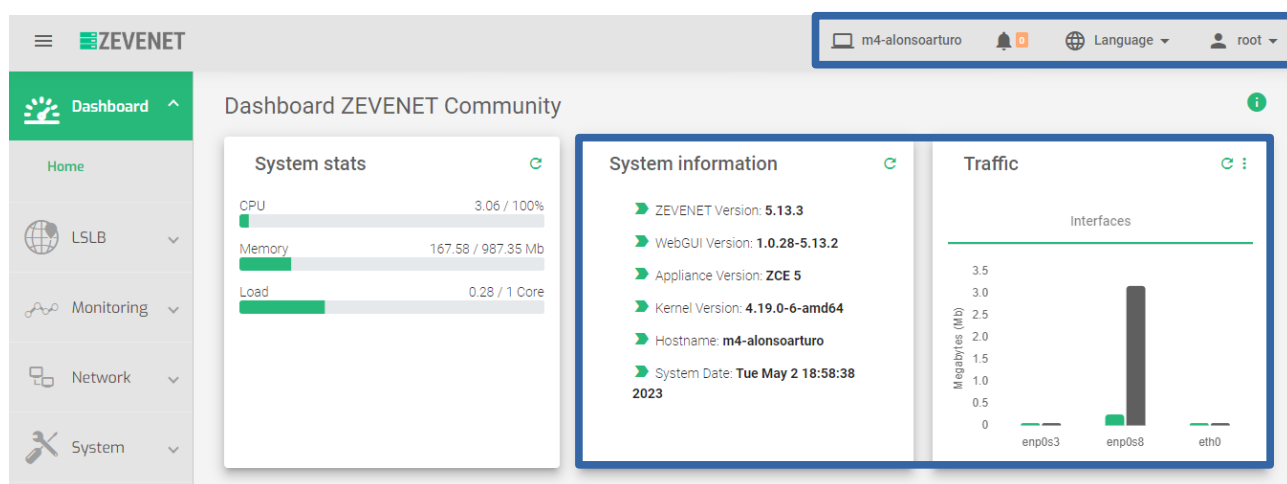
PID	USUARIO	PR	NI	UIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
4069	root	20	0	740640	27664	16144	S	63,6	1,4	0:03.68	gobetween
7	root	20	0	0	0	0	R	9,1	0,0	0:00.60	ksoftirqd/0
4125	alonsoa+	20	0	46736	4084	3476	R	9,1	0,2	0:00.06	top
1	root	20	0	159592	8748	6572	S	0,0	0,4	0:04.35	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd

Verificación con top

3.4. Zevenet

En el caso de Zevenet, el proceso de instalación es diferente al del resto de balanceadores. Para poder hacer uso del mismo, es necesario descargar la imagen ISO de la herramienta (<https://www.zevenet.com/products/community/>) y crear una nueva máquina, que llamaré **m4-alonsoarturo**, con Debian (64 bits) y cargar dicha imagen.

Una vez creada la máquina, procedemos con la instalación por defecto indicando los valores necesarios como la IP, que en este caso será **192.168.250.11**, el nombre de la máquina y la contraseña del usuario (root). Para acceder al panel de control de Zevenet basta con acceder a la IP de la máquina usando https y el puerto 444 (<https://192.168.250.11:444>).



Panel de control de Zevenet

En este caso, me he topado con un error que no he logrado solucionar, y es que Zevenet no detecta las direcciones IP de las interfaces, pero sí detecta estas, por lo que no he podido crear la granja para realizar los tests de carga. Esencialmente, el resultado habría sido parecido al del resto de balanceadores, pero aunque no he podido llevarlo a cabo, me resulta interesante mostrar la interfaz de Zevenet, ya que difiere en gran medida con el resto de herramientas empleadas en la práctica. A continuación muestro el fallo mencionado.

The screenshot shows the 'Create LSLB farm' form in the Zevenet interface. The form has a header 'LSLB > Farms > Create' and a green information icon. The main form area contains several fields: 'Name *' (text input), 'Virtual IP *' (text input with a red error message), 'Virtual port *' (text input), 'Profile *' (dropdown menu showing 'HTTP'), and 'Copy from farm' (dropdown menu showing '--No farm--'). An 'Apply' button is located at the bottom left of the form.

Creación de granja

Network > NIC

NIC list

Bring up
Bring down
Unset

Search

<input type="checkbox"/>	Name	IP	MAC	Netmask	Gateway	Status
<input type="checkbox"/>	enp0s3		08:00:27:71:d1:b5			
<input type="checkbox"/>	enp0s8		08:00:27:5c:1a:91			
<input type="checkbox"/>	eth0		c8:d7:4a:4e:47:50			

Items per page: 10
1 - 3 of 3

Interfaces

Por otro lado, es posible realizar todas las tareas que ofrece Zevenet desde la propia terminal. Para ello, es necesario activar la *ZAPI key* para poder utilizar *zcli* tal y como se muestra en las imágenes siguientes.

System > User settings

ZAPI

User
root

☒ Enable ZAPI permission

ZAPI key
Swap1234

The key cannot be retrieved once is set

Apply
Generate random key
Revert changes

ZAPI key

```
root@m4-alonsoarturo:~# zcli
Type the zapi key for the current load balancer
Load balancer zapi key: Swap1234
Saved as default profile

  zcli

Welcome to the ZEVENET Command Line Interface.

The extendend information can be displayed using the 'help' command.
zcli(localhost): _
```

zcli

3.5. Test comparativo

Para realizar un test comparativo de todos los balanceadores, se muestra a continuación una tabla con la cantidad de peticiones por segundo (p/s) realizadas por cada uno de ellos usando ambos algoritmos de balanceo.

	Nginx	HAProxy	Gobetween	Zevenet
Round-Robin	538.52 p/s	579.8 p/s	270.75 p/s	-
Ponderación	589.27 p/s	608.18 p/s	257.11 p/s	-

Como se puede observar, el balanceador más rápido ha sido HAProxy. Además el balanceo por ponderación ha obtenido mejores resultados que el balanceo usando Round-Robin. Esto no ha sucedido en Gobetween, pero son valores relativamente similares, y es posible que la variación se deba a que el tipo de peticiones que estamos lanzando con ab sea completamente uniforme.

4. Referencias

Nginx

- https://nginx.org/en/docs/http/nginx_http_upstream_module.html#upstream
- <https://upcloud.com/resources/tutorials/configure-load-balancing-nginx>
- <https://rm-rf.es/habilitar-en-nginx-la-pagina-de-status/>
- <https://stackoverflow.com/questions/35868976/nginx-service-failed-because-the-control-process-exited>

HAProxy

- <https://www.haproxy.org/>
- <https://support.ptc.com/help/thingworx/platform/r9/es/index.html#page/ThingWorx/Help/ThingWorxHighAvailability/HAProxyExample.html>
- <https://puerto53.com/linux/configurar-un-balanceador-de-carga-con-haproxy-en-linux/>

Gobetween

- <https://gobetween.io/documentation.html#Installation>
- <https://github.com/yyyar/gobetween/wiki/Installation>
- <https://snapcraft.io/gobetween>

Zevenet

- <https://www.zevenet.com/products/community/>
- <https://www.zevenet.com/knowledge-base/enterprise-edition/enterprise-edition-v4-2-administration-guide/enterprise-edition-v4-2-settings-interfaces/>
- <https://www.zevenet.com/knowledge-base/enterprise-edition/enterprise-edition-v5-0-administration-guide/network-nic/>

Otros

- <https://httpd.apache.org/docs/2.4/programs/ab.html>
- https://linuxhint.com/debian_etc_network_interfaces/
- <https://www.lifewire.com/netstat-command-2618098>
- <https://linuxconfig.org/configuring-virtual-network-interfaces-in-linux>
- <https://techviewleo.com/manage-ubuntu-debian-networking-using-netplan/>