

SERVIDORES WEB DE ALTAS PRESTACIONES

PRÁCTICA 3: BALANCEO DE CARGA EN UN SITIO WEB

CURSO 2020-2021

ELENA ORTIZ MORENO

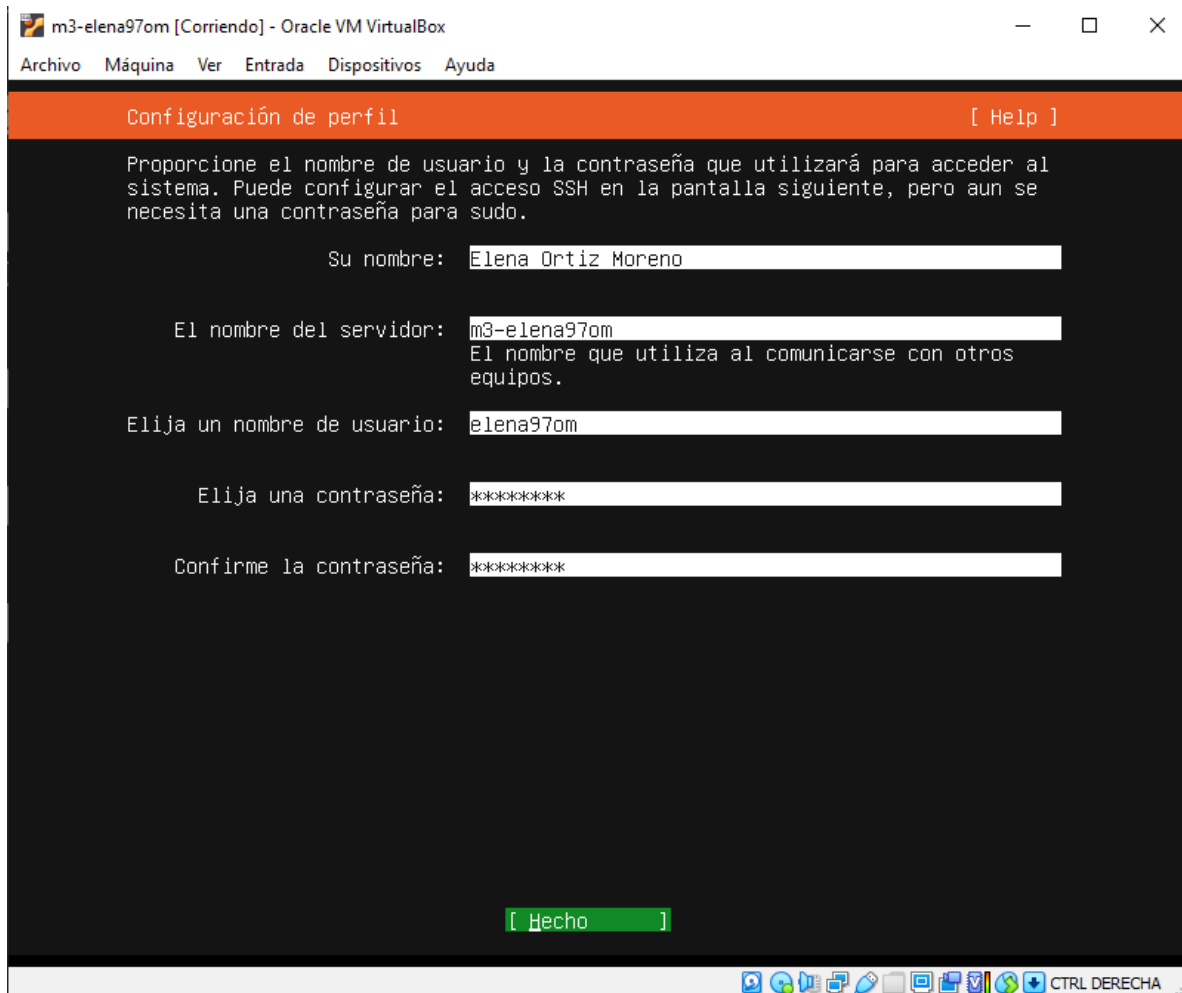
CORREO: elena97om@correo.ugr.es

ÍNDICE:

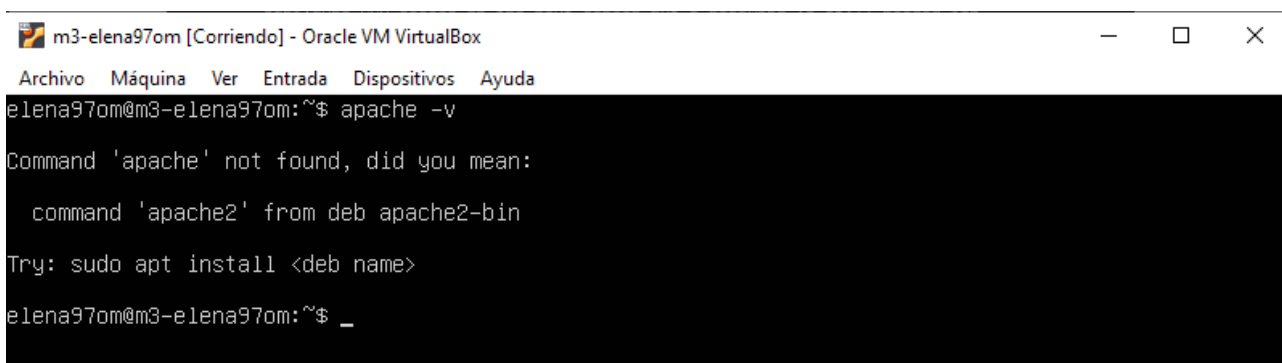
1. Configuración M3	3
2. Instalación y configuración nginx	4-6
3. Instalación y configuración haproxy	7-8
4. Nginx con ponderación	9
5. Haproxy con ponderación	10
6. Apache Benchmark	11-14
7. Análisis comparativo	15-16

1.Configuración M3:

Primero creo una nueva máquina M3 en la que no debe haber ningún software que se apropie del puerto 80, que será utilizado para el balanceo.



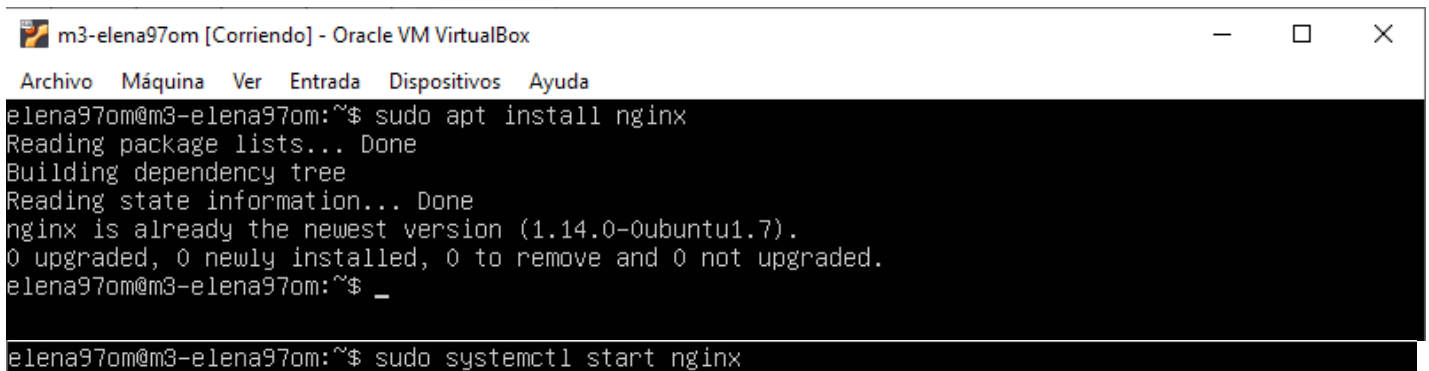
Compruebo que Apache no esté instalado:



2.Instalación y configuración nginx

Con los siguientes comandos preparo la instalación y una vez instalado lo lanzo:

```
sudo apt-get update && sudo apt-get dist-upgrade && sudo apt-get autoremove  
sudo apt-get install nginx  
  
sudo systemctl start nginx
```



The screenshot shows a terminal window titled 'm3-elena97om [Corriendo] - Oracle VM VirtualBox'. The terminal output shows the following commands and their results:

```
m3-elena97om@m3-elena97om:~$ sudo apt install nginx  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
nginx is already the newest version (1.14.0-0ubuntu1.7).  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
m3-elena97om@m3-elena97om:~$ _  
  
m3-elena97om@m3-elena97om:~$ sudo systemctl start nginx
```

Una vez lanzado, pasamos a configurarlo:

En el archivo **/etc/nginx/nginx.conf** deshabilito el servidor web para que actúe como balanceador comentando la siguiente línea:

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

Una vez hecho esto pasamos a configurar upstream con M1 y M2. Para ello modificamos el archivo **/etc/nginx/conf.d/default.conf**.

Primero añado las IP de M1 y M2, que son las máquinas donde se repartirá el tráfico.

```
upstream balanceo_elena97om{  
    server 192.168.56.102;  
    server 192.168.56.103;  
}
```

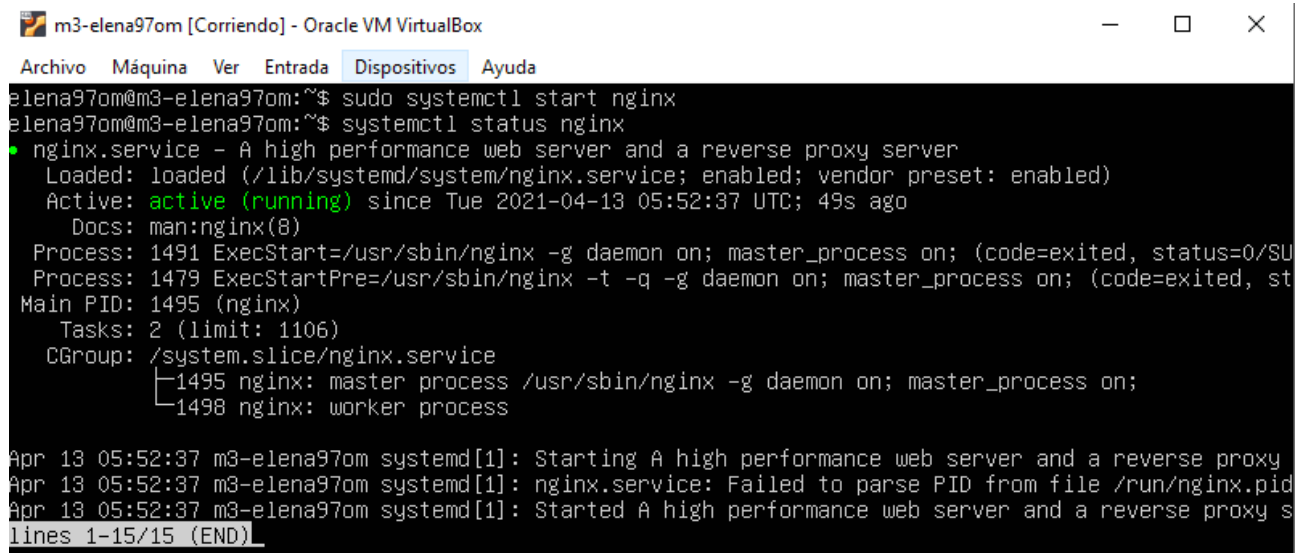
Y a continuación usaremos balanceo mediante el algoritmo de Round-Robin, es decir, sin prioridad:

```
server{
    listen 80;
    server_name balanceador_elena97om;

    access_log /var/log/nginx/balanceador_elena97om.access.log;
    error_log /var/log/nginx/balanceador_elena97om.error.log;
    root /var/www/;

    location /
    {
        proxy_pass http://balanceo_elena97om;
        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 "";
    }
}
```

Una vez hecho esto volvemos a lanzar el servicio con: `sudo systemctl start nginx`



```
m3-elena97om [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
elena97om@m3-elena97om:~$ sudo systemctl start nginx
elena97om@m3-elena97om:~$ 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 Tue 2021-04-13 05:52:37 UTC; 49s ago
     Docs: man:nginx(8)
  Process: 1491 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Process: 1479 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 1495 (nginx)
    Tasks: 2 (limit: 1106)
   CGroup: /system.slice/nginx.service
           └─1495 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─1498 nginx: worker process

Apr 13 05:52:37 m3-elena97om systemd[1]: Starting A high performance web server and a reverse proxy.
Apr 13 05:52:37 m3-elena97om systemd[1]: nginx.service: Failed to parse PID from file /run/nginx.pid
Apr 13 05:52:37 m3-elena97om systemd[1]: Started A high performance web server and a reverse proxy service.
lines 1-15/15 (END)
```

Ahora, desde M4 hago curl a M3, que deberá devolver los html de M1 y M2 de forma alterna:

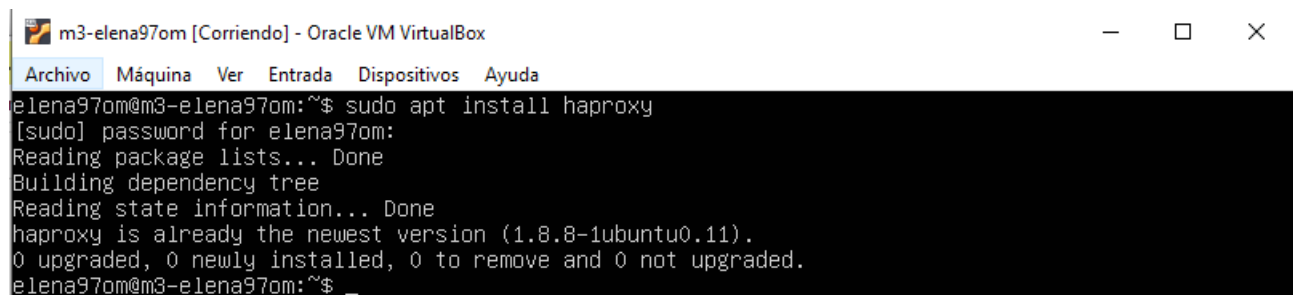
```
elena@DESKTOP-ADLCI6E: ~  
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html  
<HTML>  
<BODY>  
!!!M2!!!  
Web de ejemplo de elena97om para SWAP  
Email: elena97om@correo.ugr.es  
</BODY>  
</HTML>  
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html  
<HTML>  
<BODY>  
!!!M1!!!  
Web de ejemplo de elena97om para SWAP  
Email: elena97om@correo.ugr.es  
</BODY>  
</HTML>  
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html  
<HTML>  
<BODY>  
!!!M2!!!  
Web de ejemplo de elena97om para SWAP  
Email: elena97om@correo.ugr.es  
</BODY>  
</HTML>  
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html  
<HTML>  
<BODY>  
!!!M1!!!  
Web de ejemplo de elena97om para SWAP  
Email: elena97om@correo.ugr.es  
</BODY>  
</HTML>  
elena@DESKTOP-ADLCI6E:~$
```

3.Instalación y configuración haproxy

Es un balanceador de carga y también un proxy, por lo que podrá repartir cualquier tipo de tráfico.

Para instalarlo ejecuto en M3 el siguiente comando:

```
sudo apt-get install haproxy
```

A screenshot of a terminal window titled 'm3-elena97om [Corriendo] - Oracle VM VirtualBox'. The terminal shows the command 'sudo apt install haproxy' being executed. The output indicates that haproxy is already the newest version (1.8.8-1ubuntu0.11) and no upgrades are needed. The prompt returns to the user.

```
m3-elena97om [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
elena97om@m3-elena97om:~$ sudo apt install haproxy
[sudo] password for elena97om:
Reading package lists... Done
Building dependency tree
Reading state information... Done
haproxy is already the newest version (1.8.8-1ubuntu0.11).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
elena97om@m3-elena97om:~$ _
```

Para la configuración básica y para habilitar las estadísticas añado lo siguiente al archivo `/etc/haproxy/haproxy.cfg`:

```
frontend http-in
    bind *:80
    default_backend balanceo_elena97om

backend balanceo_elena97om
    server m1 192.168.56.102:80 maxconn 32
    server m2 192.168.56.103:80 maxconn 32

global
    stats socket /var/lib/haproxy/stats

listen stats
    bind *:9999
    mode http
    stats enable
    stats uri /stats
    stats realm HAProxy\ Statistics
    stats auth elena97om:elena97om
```

Paramos nginx con `sudo service nginx stop` para poder lanzar haproxy posteriormente con `sudo service haproxy restart`

Comprobamos el correcto funcionamiento de haproxy:

```
elena97om@m3-elena97om:/etc/haproxy$ systemctl status haproxy.service
● haproxy.service - HAProxy Load Balancer
   Loaded: loaded (/lib/systemd/system/haproxy.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2021-04-15 12:02:15 UTC; 8s ago
     Docs: man:haproxy(1)
           file:/usr/share/doc/haproxy/configuration.txt.gz
   Process: 2136 ExecStartPre=/usr/sbin/haproxy -f $CONFIG -c -q $EXTRA_OPTS (code=exited, status=0/SUCCESS)
  Main PID: 2147 (haproxy)
    Tasks: 2 (limit: 1106)
   CGroup: /system.slice/haproxy.service
           └─2147 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid
             └─2148 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid

Apr 15 12:02:15 m3-elena97om systemd[1]: Starting HAProxy Load Balancer...
Apr 15 12:02:15 m3-elena97om haproxy[2147]: Proxy http-in started.
Apr 15 12:02:15 m3-elena97om haproxy[2147]: Proxy http-in started.
Apr 15 12:02:15 m3-elena97om haproxy[2147]: Proxy balanceo_elena97om started.
Apr 15 12:02:15 m3-elena97om haproxy[2147]: Proxy balanceo_elena97om started.
Apr 15 12:02:15 m3-elena97om haproxy[2147]: Proxy stats started.
Apr 15 12:02:15 m3-elena97om haproxy[2147]: Proxy stats started.
Apr 15 12:02:15 m3-elena97om systemd[1]: Started HAProxy Load Balancer.
lines 1-20/20 (END)
```

Y hago curl desde M4 igual que hice con nginx:

```
elena@DESKTOP-ADLCI6E: ~
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html
<HTML>
<BODY>
!!!M2!!!
Web de ejemplo de elena97om para SWAP
Email: elena97om@correo.ugr.es
</BODY>
</HTML>
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html
<HTML>
<BODY>
!!!M1!!!
Web de ejemplo de elena97om para SWAP
Email: elena97om@correo.ugr.es
</BODY>
</HTML>
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html
<HTML>
<BODY>
!!!M2!!!
Web de ejemplo de elena97om para SWAP
Email: elena97om@correo.ugr.es
</BODY>
</HTML>
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html
<HTML>
<BODY>
!!!M1!!!
Web de ejemplo de elena97om para SWAP
Email: elena97om@correo.ugr.es
</BODY>
</HTML>
```


4.Nginx con ponderación

Simplemente añadimos unos pesos o ponderaciones a cada máquina para que reciban la cantidad de peticiones que queramos (que por defecto son 1). Lo que vamos a hacer es asignar 1 a M1 y 2 a M2. Así, si M2 tiene menos carga que M1, recibirá más peticiones

```
upstream balanceo_elena97om{  
    server 192.168.56.102 weight=1;  
    server 192.168.56.103 weight=2;  
}
```

Ahora hago curl para comprobar que efectivamente M2 recibe dos peticiones antes de que vuelva a recibir una M1.

```
elena@DESKTOP-ADLCI6E: ~  
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html  
<HTML>  
<BODY>  
!!!M1!!!  
Web de ejemplo de elena97om para SWAP  
Email: elena97om@correo.ugr.es  
</BODY>  
</HTML>  
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html  
<HTML>  
<BODY>  
!!!M2!!!  
Web de ejemplo de elena97om para SWAP  
Email: elena97om@correo.ugr.es  
</BODY>  
</HTML>  
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html  
<HTML>  
<BODY>  
!!!M2!!!  
Web de ejemplo de elena97om para SWAP  
Email: elena97om@correo.ugr.es  
</BODY>  
</HTML>  
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html  
<HTML>  
<BODY>  
!!!M1!!!  
Web de ejemplo de elena97om para SWAP  
Email: elena97om@correo.ugr.es  
</BODY>  
</HTML>
```

5.Haproxy con ponderación

Igual que antes, simplemente añadido las ponderaciones a cada máquina

```
backend balanceo_elena97om
    server m1 192.168.56.102:80 maxconn 32 weight 2
    server m2 192.168.56.103:80 maxconn 32 weight 1
```

Y hago curl desde M4 para comprobar que las tiene en cuenta.

```
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html
<HTML>
<BODY>
!!!M1!!!
Web de ejemplo de elena97om para SWAP
Email: elena97om@correo.ugr.es
</BODY>
</HTML>
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html
<HTML>
<BODY>
!!!M2!!!
Web de ejemplo de elena97om para SWAP
Email: elena97om@correo.ugr.es
</BODY>
</HTML>
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html
<HTML>
<BODY>
!!!M1!!!
Web de ejemplo de elena97om para SWAP
Email: elena97om@correo.ugr.es
</BODY>
</HTML>
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html
<HTML>
<BODY>
!!!M1!!!
Web de ejemplo de elena97om para SWAP
Email: elena97om@correo.ugr.es
</BODY>
</HTML>
elena@DESKTOP-ADLCI6E:~$ curl http://192.168.56.104/ejemplo.html
<HTML>
<BODY>
!!!M2!!!
Web de ejemplo de elena97om para SWAP
Email: elena97om@correo.ugr.es
</BODY>
</HTML>
elena@DESKTOP-ADLCI6E:~$
```

6.Someter a una alta carga el servidor balanceado

Voy a comprobar el rendimiento del servidor web con la utilidad siguiente de apache desde M4:

```
ab -n 1000 -c 10 http://ip_maquinaM3/index.html
```

Primero con nginx configurado con Round-Robin en marcha, y obtengo la siguiente información:

```
eIena@DESKTOP-ADLCI6E:~$ ab -n 1000 -c 10 http://192.168.56.104/ejemplo.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 192.168.56.104 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests


Server Software:      nginx/1.14.0
Server Hostname:      192.168.56.104
Server Port:          80

Document Path:        /ejemplo.html
Document Length:      111 bytes

Concurrency Level:    10
Time taken for tests:  1.889 seconds
Complete requests:    1000
Failed requests:       0
Total transferred:    380000 bytes
HTML transferred:     111000 bytes
Requests per second:  529.42 [#/sec] (mean)
Time per request:     18.888 [ms] (mean)
Time per request:     1.889 [ms] (mean, across all concurrent requests)
Transfer rate:        196.47 [Kbytes/sec] received


Connection Times (ms)
              min    mean[+/-sd] median   max
Connect:        0     5   4.0      5     29
Processing:      3    13   5.6     12     36
Waiting:        0    10   5.6      9     33
Total:          4    18   6.4     17     47


Percentage of the requests served within a certain time (ms)
 50%    17
 66%    19
 75%    21
 80%    23
 90%    27
 95%    31
 98%    35
 99%    37
100%    47 (longest request)
```

Y después con haproxy configurado con Round-Robin en marcha, obteniendo lo siguiente:

```
elena@DESKTOP-ADLCI6E:~$ ab -n 1000 -c 10 http://192.168.56.104/ejemplo.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 192.168.56.104 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests


Server Software:      Apache/2.4.29
Server Hostname:      192.168.56.104
Server Port:          80

Document Path:        /ejemplo.html
Document Length:       111 bytes

Concurrency Level:     10
Time taken for tests:   1.588 seconds
Complete requests:      1000
Failed requests:         0
Total transferred:      381000 bytes
HTML transferred:       111000 bytes
Requests per second:    629.57 [#/sec] (mean)
Time per request:       15.884 [ms] (mean)
Time per request:       1.588 [ms] (mean, across all concurrent requests)
Transfer rate:          234.24 [Kbytes/sec] received


Connection Times (ms)
              min      mean[+/-sd]  median    max
Connect:        0         4    4.9         3     71
Processing:      2        12    6.7        11     77
Waiting:         0        10    5.5         9     49
Total:           3        15    8.8        14    104


Percentage of the requests served within a certain time (ms)
 50%      14
 66%      16
 75%      17
 80%      18
 90%      21
 95%      26
 98%      34
 99%      78
100%     104 (longest request)
```

Hago lo mismo con nginx configurado con ponderación:

```
elena@DESKTOP-ADLCI6E:~$ ab -n 1000 -c 10 http://192.168.56.104/ejemplo.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 192.168.56.104 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests


Server Software:      nginx/1.14.0
Server Hostname:      192.168.56.104
Server Port:          80

Document Path:        /ejemplo.html
Document Length:      111 bytes

Concurrency Level:    10
Time taken for tests:  1.502 seconds
Complete requests:    1000
Failed requests:       0
Total transferred:    380000 bytes
HTML transferred:     111000 bytes
Requests per second:  665.86 [#/sec] (mean)
Time per request:     15.018 [ms] (mean)
Time per request:     1.502 [ms] (mean, across all concurrent requests)
Transfer rate:        247.10 [Kbytes/sec] received


Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    3   2.3      2    15
Processing:      2   12   5.2     11    45
Waiting:         0   10   5.3      8    42
Total:          3   15   5.1     14    46


Percentage of the requests served within a certain time (ms)
 50%    14
 66%    16
 75%    17
 80%    18
 90%    21
 95%    25
 98%    28
 99%    32
100%   46 (longest request)
```

Y con haproxy configurado con ponderación:

```
elena@DESKTOP-ADLCI6E:~$ ab -n 1000 -c 10 http://192.168.56.104/ejemplo.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 192.168.56.104 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests


Server Software:      Apache/2.4.29
Server Hostname:      192.168.56.104
Server Port:          80

Document Path:        /ejemplo.html
Document Length:      111 bytes

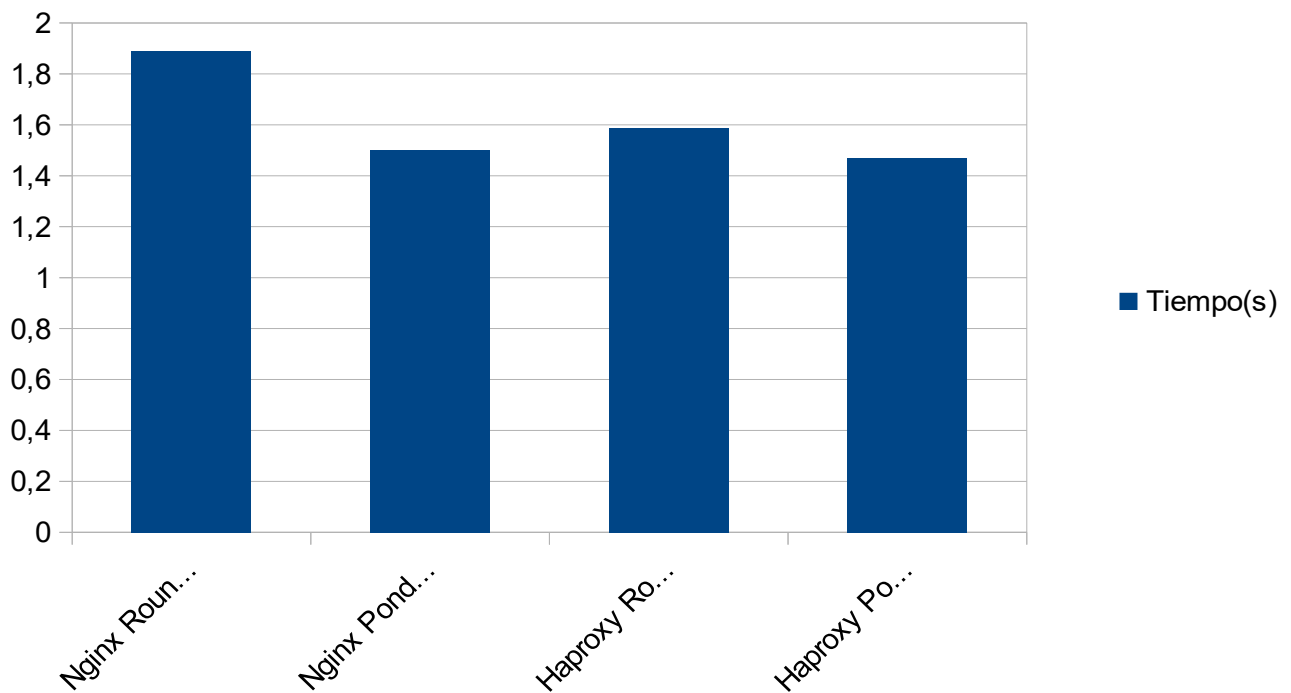
Concurrency Level:    10
Time taken for tests:  1.469 seconds
Complete requests:    1000
Failed requests:       0
Total transferred:    381000 bytes
HTML transferred:     111000 bytes
Requests per second:  680.83 [#/sec] (mean)
Time per request:     14.688 [ms] (mean)
Time per request:     1.469 [ms] (mean, across all concurrent requests)
Transfer rate:        253.32 [Kbytes/sec] received

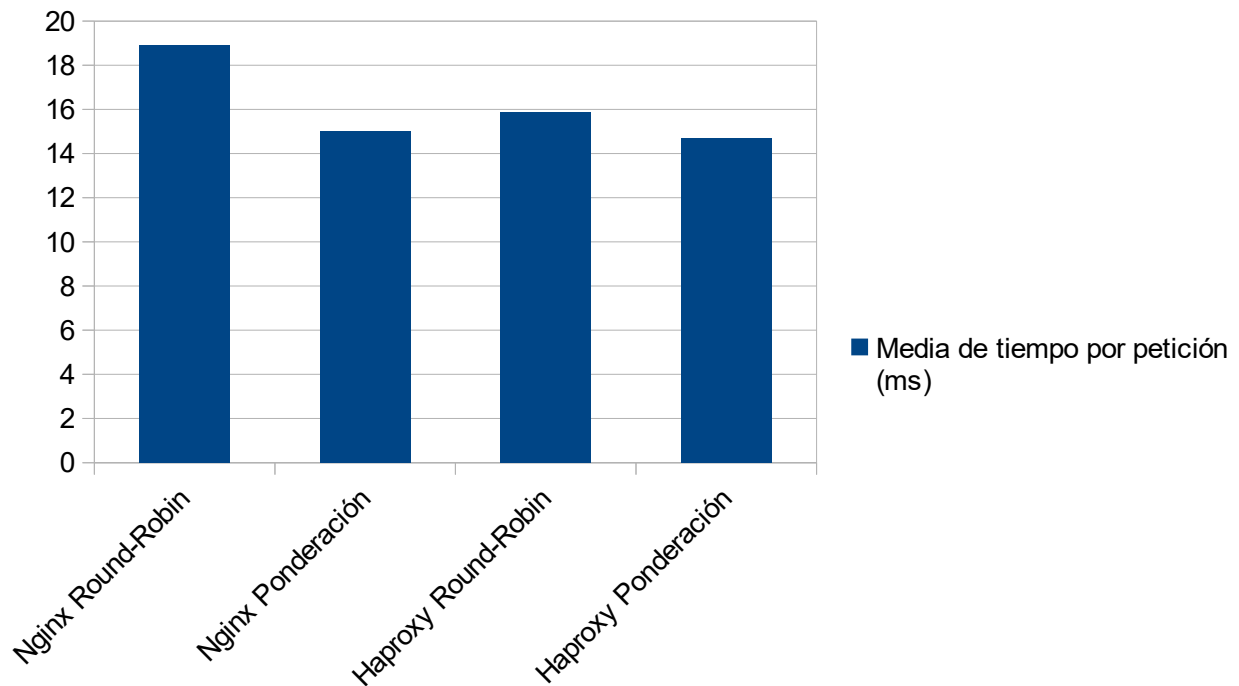

Connection Times (ms)
              min      mean[+/-sd] median   max
Connect:        0         3    2.5         3    15
Processing:      3        11    4.3        10    33
Waiting:         0         9    4.5         8    33
Total:          4        14    4.8        13    39


Percentage of the requests served within a certain time (ms)
 50%      13
 66%      15
 75%      17
 80%      18
 90%      20
 95%      23
 98%      26
 99%      29
100%      39 (longest request)
```

7. Análisis comparativo

Balanceador	Algoritmo-Balanceo	Tiempo(s)	Media de tiempo por petición (ms)
Nginx	Round-Robin	1.889	18.888
Nginx	Ponderación	1.502	15.018
Haproxy	Round-Robin	1.588	15.884
Haproxy	Ponderación	1.469	14.688





Tras estas comprobaciones con apache, podemos ver que en general haproxy tarda menos que nginx en realizar todas las peticiones, aunque no son diferencias muy significativas, ya que prácticamente tardan lo mismo. También podemos ver que tardan menos configurados con ponderación, ya que las peticiones están más repartidas. Con la media de tiempo por petición ocurre algo similar.