

Práctica 3

Balanceo de carga en un sitio web



José Manuel Soto Hidalgo



José Manuel Soto Hidalgo

Dpto. Ingeniería de Computadores, Automática y Robótica
Universidad de Granada

jmsoto@ugr.es

Objetivos

- Diseñar una distribución de balanceo de carga para la granja web
- Configurar diferentes balanceadores de carga con diferentes opciones de balanceo
- Habilitar estadísticas de balanceo
- Evaluar los balanceadores mediante el uso de un benchmark
- Realizar conclusiones comparativas de distintos balanceadores

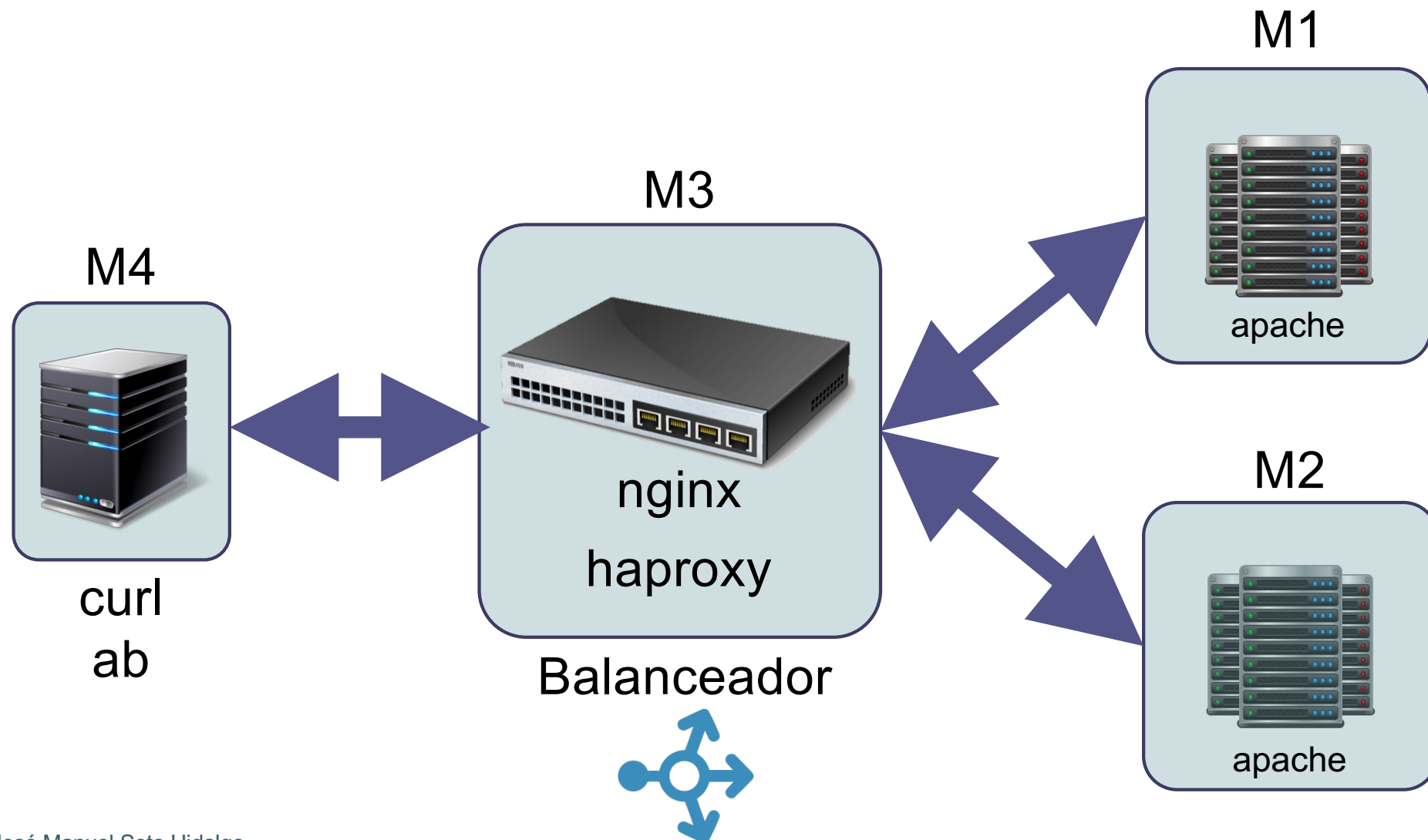


Duración: 3 sesiones

Balanceo de carga

- Distribuir peticiones a servicios en distintos servidores
 - **Soluciones hardware:** Routers y Switches con soft. de balanceo o hardware específico.
 - **Soluciones software:** Métodos instalados en el backend de los servidores
 - HaProxy: <http://haproxy.1wt.eu/>
 - Pound: <http://www.apsis.ch/pound/>
 - Varnish: <http://varnish-cache.org>
 - NginX: <http://nginx.org/>
 - Lighty: <http://www.lighttpd.net/>
 - Apache: <http://httpd.apache.org/>

Esquema general de la práctica



Balanceo de carga con NGINX

- Nginx es un servidor web ligero de alto rendimiento
- Usado por:
 - WordPress, Hulu, GitHub, Ohloh, SourceForge, TorrentReactor
- Instalar nginx en una nueva máquina M3 (con Ubuntu Server llamada m3-usuarioUGR y con usuario usuarioUGR y contraseña Swap12324)
 - `sudo apt-get update && sudo apt-get dist-upgrade && sudo apt-get autoremove`
 - `sudo apt-get install nginx`
- Lanzar nginx
 - `sudo systemctl start nginx` **O** `sudo service nginx start`

Balanceo de carga con NGINX

- Deshabilitar servidor web para que éste actúe como balanceador – En archivo **/etc/nginx/nginx.conf**
 - `#include /etc/nginx/sites-enabled/*;`
- Configurar “upstream” con el conjunto de máquinas a repartir el tráfico (en este caso M1 y M2)
 - En archivo **/etc/nginx/conf.d/default.conf**

```
upstream balanceo_usuarioUGR {
    server ip_maquinaM1;
    server ip_maquinaM2;
}
```

Balanceo de carga con NGINX


En archivo `/etc/nginx/conf.d/default.conf`

```
server{
    listen 80;
    server_name balanceador_usuarioUGR;

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

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

upstream



Balanceo de carga con NGINX

Repartir carga en función de pesos

```
upstream balanceo_usuarioUGR {  
    server ip_maquinaM1 weight=1;  
    server ip_maquinaM2 weight=2;  
}
```

“cada tres peticiones que lleguen al balanceador, la máquina M2 atenderá dos y la máquina M1 atenderá una”

Balanceo de carga con NGINX

Repartir carga de misma IP a mismo servidor

```
upstream balanceo_usuarioUGR {
    ip_hash;
    server ip_maquinaM1;
    server ip_maquinaM2;
}
```

“todas las peticiones que vengan de la misma IP se dirijan a la misma máquina servidora final”

Todos los usuarios detrás de proxy o NAT son dirigidos a mismo back-end → balanceo no equilibrado



Balanceo de carga con NGINX

Repartir carga de misma IP a mismo servidor

```
upstream balanceo_usuarioUGR {  
    server ip_maquinaM1;  
    server ip_maquinaM2;  
    keepalive 3;  
}
```

“todas las peticiones que vengan de la misma IP se dirijan a la misma máquina servidora final” → número de conexiones keepalive

Nginx plantea varias opciones de conexiones con las directivas keepalive, keepalive_time y keepalive_timeout

Balanceo de carga con NGINX

Más opciones de configuración

`weight = NUMBER`

permite especificar un peso para el servidor (por defecto es 1).

`max_fails = NUMBER`

especifica un número de intentos de comunicación erróneos en "fail_timeout" segundos para considerar al servidor no operativo (por defecto es 1).

`fail_timeout = TIME`

indica el tiempo en el que deben ocurrir "max_fails" intentos fallidos de conexión para considerar al servidor no operativo. Por defecto es 10 segundos.

`down`

marca el servidor como permanentemente offline (para ser usado con ip_hash).

`backup`

reserva este servidor y sólo le pasa tráfico si alguno de los otros servidores no-backup está caído u ocupado. No es compatible con la directiva ip_hash

Balanceo de carga con HAPROXY

- ***haproxy*** es un balanceador de carga y también proxy, de forma que puede balancear cualquier tipo de tráfico
- Instalar ***haproxy*** en la máquina M3
 - `sudo apt-get install haproxy`
- Lanzar ***haproxy***
 - `sudo systemctl start haproxy`
 - 0
 - `sudo service haproxy start`
- **IMPORTANTE:** Al estar escuchando ambos balanceadores (nginx y haproxy) en el mismo puerto (80), los dos a la vez no pueden estar funcionando. Se debería parar uno u otro: `sudo service nginx stop` 0 `sudo service haproxy stop`

Balanceo de carga con HAPROXY

- Archivo **/etc/haproxy/haproxy.cfg**

```
frontend http-in
```

```
    bind *:80
```

```
    default_backend balanceo_usuarioUGR
```

```
backend balanceo_usuarioUGR
```

```
    server m1 ip_maquinaM1:80 maxconn 32
```

```
    server m2 ip_maquinaM2:80 maxconn 32
```

Balanceo de carga con HAPROXY

- Archivo **/etc/haproxy/haproxy.cfg** **Estadísticas**

```
global
    stats socket /var/lib/haproxy/stats

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

Accede a través de http://ip_maquinaM3:9999/stats

Balanceo de carga con HAPROXY

- Archivo `/etc/haproxy/haproxy.cfg` Estadísticas

Statistics Report for pid 23602

> General process information

pid = 23602 (process #1, nbproc = 1, nbthread = 1)
 uptime = 0d 0h01m05s
 system limits: memmax = unlimited; ulimit-n = 4032
 maxsock = 4032; maxconn = 2000; maxpipes = 0
 current conns = 1; current pipes = 0/0; conn rate = 0/sec
 Running tasks: 1/6; 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](#)
- [Disable refresh](#)
- [Refresh now](#)
- [CSV export](#)

External resources:

- [Primary site](#)
- [Updates \(v1.8\)](#)
- [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	Thrtl	
Frontend				0	2	-	1	2	3 000	6			6 220	237 471	0	0	3					OPEN									
Backend	0	0		0	0		0	0	300	0	0	0s	6 220	237 471	0	0		0	0	0	0	1m5s UP		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	644	-	0	15	2 000	15 970			1 309 130	178 856 835	0	0	5					OPEN								

balanceo_jmsoto

	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	Thrtl
m1	0	0	-	0	322		0	5	32	7 983	7 983	10s	654 606	89 433 549		0		0	0	0	no check		1	Y	-					-
m2	0	0	-	0	322		0	10	32	7 982	7 982	10s	654 524	89 422 346		0		0	0	0	no check		1	Y	-					-
Backend	0	0		0	644		0	10	200	15 965	15 965	10s	1 309 130	178 855 895	0	0		0	0	0	0	1m5s UP		2	2	0		0	0s	

Benchmarking sobre granja web

- Varias herramientas para someter a carga HTTP a servidores web
 - Apache Benchmark; siege; httpperf ; OpenSTA; Jmeter; openwebload; the grinder

Desde una máquina distinta a M3 (por ejemplo, anfitrión o una nueva M4)

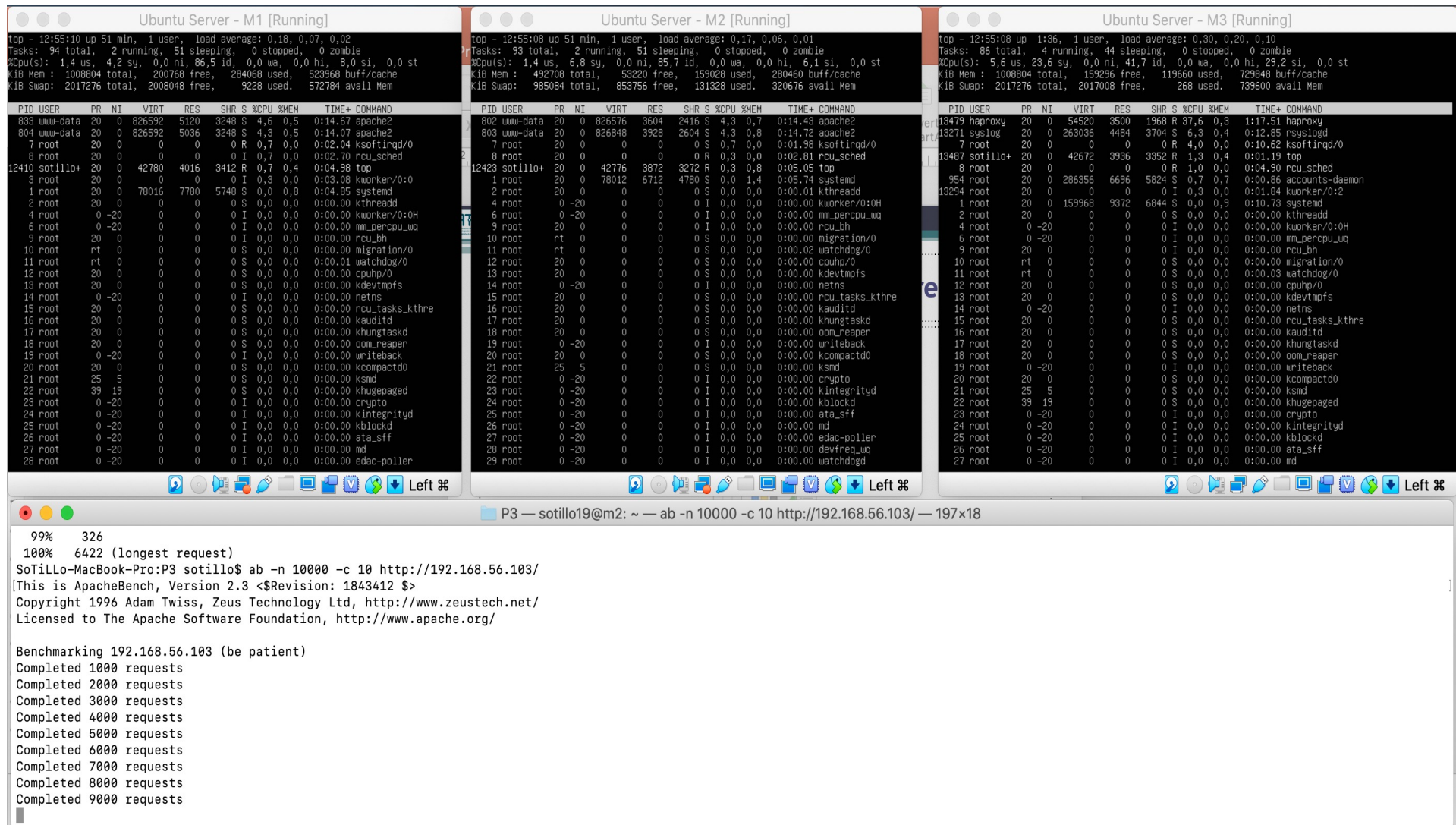
```
ab -n 10000 -c 10 http://ip\_maquinaM3/swap.html
```

NOTA: Si el archivo `swap.html` de las máquinas finales (M1 y M2) difiere en longitud de archivo, debéis utilizar el parámetro `-l` para indicar que las peticiones tendrán longitud variable.

OPCIONES PARA VER QUE EL BALANCEADOR DISTRIBUYE PETICIONES:

1. Modifica el archivo `swap.html` en M1 y M2 añadiendo en el título de la web M1 o M2 para, a modo de ejemplo, distinguir que los balanceadores están sirviendo bien las webs de M1 y M2.
2. Ejecuta el script `getload.php` que proporcionamos en PRADO.

Benchmarking sobre granja web



Cuestiones a resolver

- En esta práctica el objetivo es configurar las máquinas virtuales de forma que dos hagan de servidores web finales (M1 y M2) mientras que la tercera haga de balanceador de carga por software (M3). Se debe añadir **usuarioUGR** en las distintas configuraciones e ilustrarlo con capturas de pantalla.
- En esta práctica **se llevarán a cabo las tareas básicas:**
 1. Crear y configurar M3 con nginx y haproxy como balanceadores de carga con el algoritmo round-robin
 2. Someter la granja web a una alta carga con la herramienta Apache Benchmark a través de M3, considerando 2 opciones:
 1. nginx con round-robin
 2. haproxy con round-robin
 3. Realizar un análisis comparativo de los resultados considerando *número de peticiones por unidad de tiempo*

Cuestiones a resolver

- En esta práctica el objetivo es configurar las máquinas virtuales de forma que dos hagan de servidores web finales (M1 y M2) mientras que la tercera haga de balanceador de carga por software (M3).
- Como **opciones avanzadas**:
 1. Configurar nginx y haproxy como balanceadores de carga con ponderación, suponiendo que M1 tiene el doble de capacidad que M2.
 2. Habilitar el módulo de estadísticas en Haproxy y analizarlo
 3. Instalar y configurar otros balanceadores de carga (Gobetween, Zevenet, Pound, etc.)
 4. Someter la granja web a una alta carga con la herramienta Apache Benchmark considerando los distintos balanceadores instalados y configurados.
 5. Realizar un análisis comparativo y detallado (gráficas comparativas) de los resultados considerando número de peticiones por unidad de tiempo

Cuestiones a resolver

- Para comprobar el funcionamiento de los balanceadores instalados, debemos hacer peticiones a la dirección IP del balanceador y comprobar que realmente se reparte la carga. Para ello, el index.html de las máquinas finales debe ser diferente para ver cómo las respuestas que recibimos al hacer varias peticiones son diferentes (eso indicará que el balanceador deriva tráfico a las máquinas servidoras finales). *Deshabilita la opción de sincronizar directorios www de la práctica anterior.*
- Además, se comprobará el funcionamiento de los algoritmos de balanceo round-robin y con ponderación (en este caso supondremos que la máquina M1 tiene el doble de capacidad que la máquina M2).

Cuestiones a resolver

- Para someter a alta carga la granja web alta (gran número de peticiones y con alta concurrencia) debemos usar la herramienta **ab** sobre las distintas configuraciones de los balanceadores instalados (con roundrobin y con ponderación). Como resultado, se debe realizar una **comparación del *número de peticiones por unidad de tiempo*** entre los balanceadores, para poder determinar cuál funciona mejor, reflejando unas conclusiones.

Normas de entrega

- La práctica se realizará de manera individual.
- Se entregará un documento *.pdf* con el desarrollo de la práctica según el guion detallando, en su caso, los aspectos básicos y avanzados realizados. Se deja a libre elección la estructura del documento el cual reflejará el correcto desarrollo de la práctica a modo de diario/tutorial. En el documento de texto a entregar se describirá cómo se han realizado las diferentes configuraciones (así como comandos de terminal a ejecutar en cada momento).
- Para la entrega se habilitará una tarea en PRADO donde se entregará el documento desarrollado siguiendo **OBLIGATORIAMENTE** el formato **ApellidosNombreP3.pdf**

Evaluación

- La práctica se evaluará mediante el uso de rúbrica específica (accesible por el estudiante en la tarea de entrega) y una defensa final de prácticas.
- Tiene un peso del 25% del total de prácticas
- La detección de prácticas copiadas implicará el suspenso inmediato de todos los implicados en la copia (tanto del autor del original como de quien las copió). OBLIGATORIO ACEPTAR LICENCIA EULA DE TURNITIN
 - Si la memoria supera un 40% de copia Turnitin —> suspenso
 - del 1-10% -> 0
 - del 11-20% -> -1
 - del 20-30% —> -2
 - del 30-40% —> -3
 - 40% —> suspenso
- Las faltas de ortografía se penalizarán con hasta 1 punto de la nota de la práctica.