

Benchmarking y ajustes del sistema

Julio A. Fresneda García – 49215154F

Phoronix

Phoronix es una plataforma en la cual podemos encontrar una gran cantidad de tests para hacer benchmarks. Podemos obtener tanto tests como suites (conjuntos de tests). Photonix dispone de phromatic, la cual es una interfaz gráfica.

Instalación y configuración de Phoronix

Para instalar phoronix, ejecutamos el siguiente comando:

\$ sudo apt-get install phoronix-test-suite

Vamos a necesitar algunas dependencias, como php, php-sqlite3, php-zip y firefox, las cuales debemos instalar también.

Lo primero que hay que hacer es movernos a:

/usr/bin/phoronix-test-suite

Y ejecutar el archivo "phoronix-test-suite" para instalarlo.

Después, tenemos que configurarlo. Para eso abrimos el archivo:

~/phoronix-test-suite/user-config.xml

Lo único que hay que modificar es asignarle un puerto en RemoteAccessPort. En este caso he puesto 20000.

Configuración de Phoromatic

Vamos a lanzar ahora phoromatic. El comando es:

phoronix-test-suite start-phoromatic-server

Para acceder a phoromatic, abrimos Firefox. Como Ubuntu Server no tiene interfaz gráfica, hay que abrirlo desde Ubuntu Desktop (U otro sistema con interfaz gráfica) conectado al servidor por ssh.

Para esto, usamos el comando:

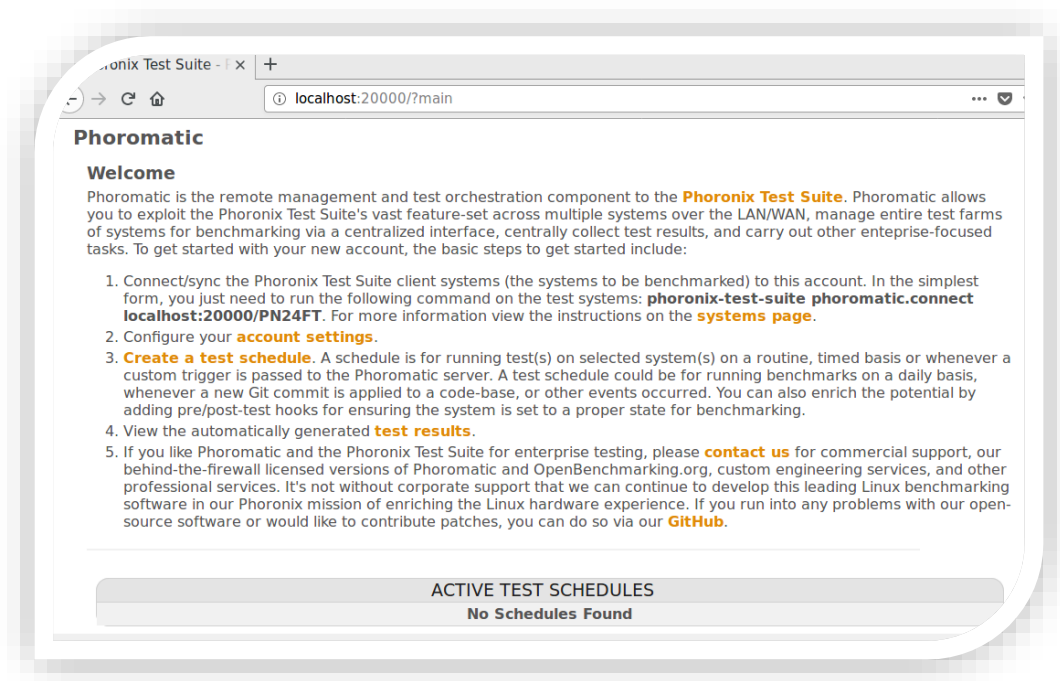
ssh -X http://192.168.57.102

Abrimos Firefox, y nos metemos a localhost:20000.

Lo primero que hay que hacer es configurarlo:

Vamos a crear una nueva cuenta con user juliofgx17, pass practicas,ISE

Después seguimos los pasos que la propia web nos pone:



Una vez completado, ya tendríamos Phoromatic listo:

The screenshot shows a web browser window with the address bar displaying 'localhost:20000/?systems/7V800'. The page title is 'ubuntu' and the subtitle is 'Ubuntu Server'. Below this, there is a link 'Edit Task & Enable/Disable System'. The 'System State' section lists the following information:

- Status: Idling, Waiting For Task
- State: Active
- Phoronix Test Suite Client: 5.2.1
- Last IP: 127.0.0.1
- Last Communication: 19 December 01:04
- Initial Creation: 19 December 00:38
- System ID: 7V800

The 'System Components' section is divided into two columns:

Component	Value	Component	Value
Processor	Intel Core i5-6200U @ 2.40GHz (1 Core)	OS	Ubuntu 16.04
Motherboard	Oracle VirtualBox v1.2	Kernel	4.4.0-87-generic (x86_64)
Chipset	Intel 440FX- 82441FX PMC	File-System	ext4
Memory	1024MB	Screen Resolution	800x600
Disk	11GB VBOX HDD	System Layer	Oracle VirtualBox
Graphics	InnoTek VirtualBox		
Audio	Intel 82801AA AC 97 Audio		

On the right side of the interface, there is a sidebar with the text 'Active Systems ubuntu' and a status summary: '19:48 CET - 19 DECEMBER 2017', '1 SYSTEM', '0 SCHEDULES', '0 RESULTS', and 'LOG-OUT'.

Benchmarking

Vamos a instalar y probar distintos tests: C-Ray, Compress-7zip y AIO-Stress.

Estos tests se han probado en un PC con las siguientes características:

The screenshot shows the Cray3 benchmarking tool interface. At the top, there is a logo for 'cray3'. Below it, the text 'OpenBenchmarking.org' and 'Phoronix Test Suite 7.4.0' are visible. The main content area displays a list of system components and their specifications:

- Processor: Intel Core i5-6200U @ 2.40GHz (1 Core)
- Motherboard: Oracle VirtualBox v1.2
- Chipset: Intel 440FX- 82441FX PMC
- Memory: 1024MB
- Disk: 11GB VBOX HDD
- Graphics: LLVMpipe
- Audio: Intel 82801AA AC 97 Audio
- Network: Intel 82540EM Gigabit
- OS: Ubuntu 16.04
- Kernel: 4.4.0-87-generic (x86_64)
- OpenGL: 3.3 Mesa 17.0.7 Gallium 0.4
- Compiler: GCC 5.4.0 20160609
- File-System: ext4
- Screen Resolution: 800x600
- System Layer: Oracle VirtualBox

At the bottom, there is a section for 'Cray3 Benchmarks' with the following command line: '--build=x86_64-linux-gnu --disable-browser-plugin --disable-ytable-verify --disable-werror'.

C-Ray

C-Ray es un test de CPU. El comando es para lanzarlo es:

phoronix-test-suite benchmark c-ray

Este test consiste en un raytracer (algoritmo para síntesis de imágenes tridimensionales) simple diseñado para probar el rendimiento de CPU con operaciones de coma flotante.

Esta prueba tiene varios subprocesos (16 subprocesos por núcleo), disparará 8 rayos por píxel para suavizar, y generará una imagen de 1600 x 1200.

Como vemos en las siguientes imágenes, el test ha consistido en ejecutar el algoritmo antes descrito tres veces, y calcular la media.

En este caso, mi pc tarda en generar la imagen, de media, 319.56 segundos. Los resultados son éstos:

```
C-Ray 1.11:
pts/c-ray-1.1.0
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 16 Minutes
  Started Run 1 @ 18:39:18
  Started Run 2 @ 18:44:40
  Started Run 3 @ 18:50:09 [Std. Dev: 2.27%]


Test Results:
  319.7
  326.733
  312.253

Average: 319.56 Seconds

Do you want to view the results in your web browser (Y/n): Y
Would you like to upload the results to OpenBenchmarking.org (Y/n): Y
Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y/n): Y

Results Uploaded To: https://openbenchmarking.org/result/1712194-KH-CRAY3182607
Do you want to launch OpenBenchmarking.org (Y/n): Y

julio@ubuntu:~$
```

cray3	
	cray
c-ray: Total Time	319.56
Standard Error	4.18
Standard Deviation	2.27%
OpenBenchmarking.org	

Compress-7zip

También es un test de CPU.

El comando es:

phoronix-test-suite benchmark compress-7zip

Compress-7zip es un test de velocidad de compresión, usando la característica de benchmarks integrada de p7zip (programa 7-Zip).

Como vemos, la media obtenida es de las 3 ejecuciones del algoritmo de compresión es de 2899 MIPS (Millones de instrucciones por segundo).

```
7-Zip Compression 9.20.1:
pts/compress-7zip-1.6.2
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 2 Minutes
Started Run 1 @ 19:00:16
Started Run 2 @ 19:00:56
Started Run 3 @ 19:01:35 [Std. Dev: 0.83%]


Test Results:
2872
2905
2919

Average: 2899 MIPS

Do you want to view the results in your web browser (Y/n): n
Would you like to upload the results to OpenBenchmarking.org (Y/n): Y
Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y/n): Y

Results Uploaded To: https://openbenchmarking.org/result/1712190-KH-7ZIP2935607
Do you want to launch OpenBenchmarking.org (Y/n): Y

julio@ubuntu:~$
```

7zip2	
	7zip
compress-7zip: Compress Speed Test	2899
Standard Error	13.93
Standard Deviation	0.83%
OpenBenchmarking.org	

AIO-Stress

Es un test de disco duro. El comando para lanzarlo es:

phoronix-test-suite benchmark aio-stress

AIO-Stress es un test asíncrono de escritura y lectura de datos creado por SuSE. Este test usa un archivo de texto de 2GB y escribe en bloques de 64kb. Aunque los loops estimados eran de 3, el test ha llevado a cabo el algoritmo 6 veces, obteniendo una media de escritura de 160 MB/s, con un error del 4.77% y una desviación del 7.27%

```
AIO-Stress 0.21:
pts/aio-stress-1.1.1 [Test: Random Write]
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 1 Minute
Started Run 1 @ 19:03:56
Started Run 2 @ 19:04:13
Started Run 3 @ 19:04:30 [Std. Dev: 7.67%]
Started Run 4 @ 19:04:45 [Std. Dev: 8.00%]
Started Run 5 @ 19:05:00 [Std. Dev: 7.93%]
Started Run 6 @ 19:05:15 [Std. Dev: 7.27%]


Test Results:
142.63
149.78
165.52
168.84
171.24
166.77

Average: 160.80 MB/s

Do you want to view the results in your web browser (Y/n): Y
Would you like to upload the results to OpenBenchmarking.org (Y/n): Y
Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y/n): Y

Results Uploaded To: https://openbenchmarking.org/result/1712194-KH-AIOSTRESS74
Do you want to launch OpenBenchmarking.org (Y/n): Y

julio@ubuntu:~$
```

aiostress2	
	aiostress
aio-stress: Test: Random Write	160.80
Standard Error	4.77
Standard Deviation	7.27%
OpenBenchmarking.org	

Apache HTTP Server benchmark

Ya hemos hecho benchmarks a la CPU y al disco duro de nuestro servidor. Ahora vamos a testear a Apache.

Para eso, tenemos dos opciones: ab y JMeter.

ab (Apache Benchmark)

ab sirve para mostrar cuántas solicitudes por segundo puede ejecutar la instalación de Apache. Para lanzarlo, ejecutamos el siguiente comando desde una terminal:

```
ab -c5 -n1000 http://192.168.57.102/
```

-c ajusta la concurrencia (las peticiones que se envían a la vez).

-n ajusta el número de peticiones que se envían.

Ejecutando el comando, obtenemos el siguiente resultado.

```
Server Software:      Apache/2.4.18
Server Hostname:      192.168.57.102
Server Port:          80

Document Path:        /
Document Length:      11321 bytes

Concurrency Level:     5
Time taken for tests:  0.437 seconds
Complete requests:     1000
Failed requests:       0
Total transferred:     11595000 bytes
HTML transferred:     11321000 bytes
Requests per second:   2288.39 [#/sec] (mean)
Time per request:      2.185 [ms] (mean)
Time per request:      0.437 [ms] (mean, across all concurrent requests)
Transfer rate:         25912.02 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median   max
Connect:    0    1   0.6      0    7
Processing:  0    2   1.6      1   19
Waiting:    0    1   1.4      1   17
Total:      1    2   1.7      2   19
WARNING: The median and mean for the initial connection time are not within a normal deviation
These results are probably not that reliable.

Percentage of the requests served within a certain time (ms)
 50%    2
 66%    2
 75%    2
 80%    3
 90%    4
 95%    5
 98%    6
 99%    8
100%   19 (longest request)
```

Estos resultados se interpretan de la siguiente manera:

Server Hostname

La IP del servidor testeado

Server Port

El puerto por el que se ha testeado el servidor

Document Path

El documento del servidor que se ha testeado. En este caso, /

Document Length

Longitud del documento

Concurrency Level

Número de peticiones simultáneas

Time taken for tests

Tiempo que ha tardado el test desde que el primer socket de conexión es creado hasta el momento que se obtiene la última respuesta. Es decir, tiempo que ha tardado en completarse el test

Complete requests

Número de peticiones completadas con éxito

Failed requests

Número de peticiones que han fallado

Total transferred

El número total de bytes recibidos desde el servidor (y enviados desde nuestro PC)

HTML transferred

El número total de bytes del documento html (/ en nuestro caso) recibidos

Requests per second

Peticiones enviadas y recibidas en un segundo

Time per request

La media del tiempo que toma en enviarse una petición (en ms). El primer valor es calculado con la fórmula: $(\text{nivel de concurrencia}) * (\text{tiempo que toma cada petición}) * 1000 / \text{tiempo total}$, mientras que el segundo valor es calculado con la fórmula: $(\text{tiempo que toma cada petición}) * 1000 / \text{tiempo total}$

Transfer rate

Bytes transferidos por segundo, se calcula con la fórmula: $(\text{bytes totales leídos}) / 1024 / (\text{tiempo que toma})$

En cuanto a la concurrencia, si lo hacemos con -c1 (sin concurrencia), el test tarda de media 1.3 segundos, lo cual tiene sentido. Sin embargo, si usamos un número grande, -c50, por ejemplo, tarda de media unos 0.4 segundos, un poco más que con -c5. Esto posiblemente sea porque el servidor no tiene tantas hebras para llevar a cabo la concurrencia y se produce cuello de botella.

Apache JMeter

JMeter es un software que puede medir y generar carga de forma genérica para varios elementos.

Para instalar JMeter, lo descargamos desde su página web, y lo extraemos. Una vez extraído, ejecutamos

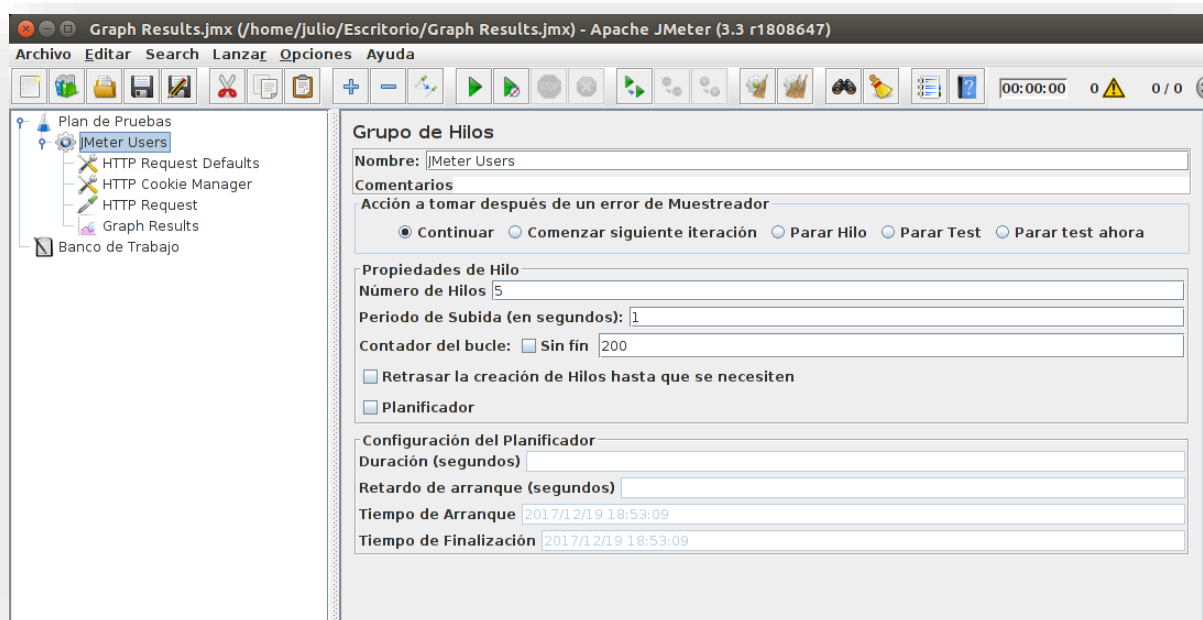
./jmeter

Creando un Test

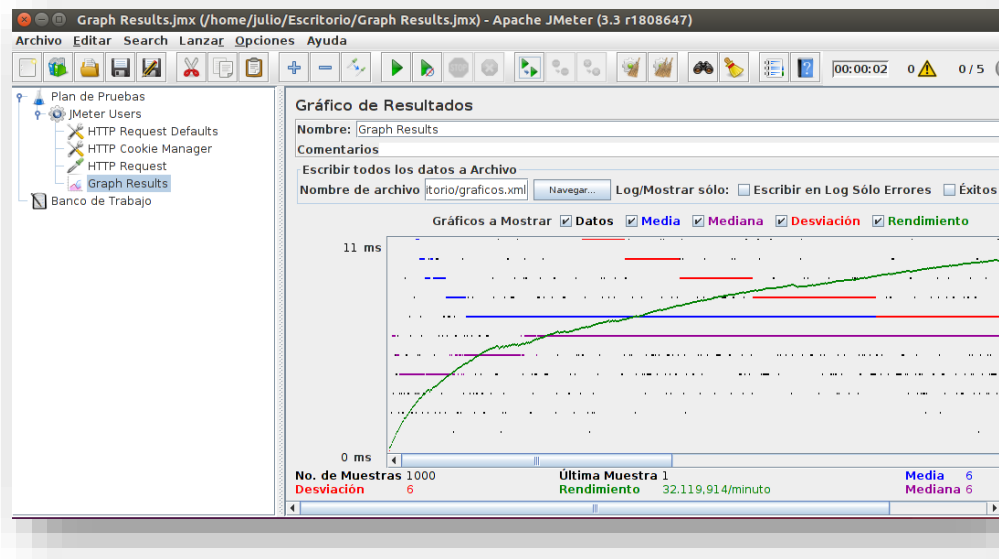
Lo primero es crear un test básico. Para eso, debemos crear los siguientes elementos:

- Thread Group, el cual contendrá el resto de elementos, y donde configuraremos la concurrencia y el número de peticiones.
- HTTP Request Defaults, donde configuraremos cosas como la IP y el puerto
- HTTP Cookie Manager, para manejar cookies
- HTTP Request, es decir, la petición HTTP
- Graph Results, el cual es un Listener, y nos sirve para ver los resultados gráficamente.

En nuestra configuración vamos a usar 5 hilos concurrentes, y 200 bucles, lo que hacen un total de 1000 peticiones. En el elemento "HTTP Request Defaults" pondremos la IP del servidor, y en "HTTP" Request, el path, / en este caso.



Una vez ejecutado el test, podemos ver gráficamente la media, mediana, desviación típica y rendimiento.



Gracias a la interfaz gráfica podemos ver que funciona correctamente y configurarlo con facilidad, pero una vez creado el test, lo ideal es ejecutarlo en modo non-gui (comandos).

Ejecutando el test

Para ejecutar el test usamos el siguiente comando:

```
./jmeter -n -t TestJMeter.jmx -l log.jtl
```

Donde `-n` nos dice que no usamos interfaz, en `-t <archivo>` pondremos el archivo que contiene el test a ejecutar, y en `-l` el archivo donde se guardarán los logs. Podemos ver el resultado:

```
julio@julio-VirtualBox:~/Escritorio/apache-jmeter-3.3/bin$ ./jmeter -n -t Graph\ Results.jmx -l log.jtl
Creating summariser <summary>
Created the tree successfully using Graph Results.jmx
Starting the test @ Tue Dec 19 19:42:06 CET 2017 (1513708926641)
Waiting for possible Shutdown/StopTestNow/Heapdump message on port 4445
summary = 1000 in 00:00:02 = 538,2/s Avg: 5 Min: 0 Max: 93 Err: 0 (0,00%)
Tidying up ... @ Tue Dec 19 19:42:09 CET 2017 (1513708929357)
... end of run
julio@julio-VirtualBox:~/Escritorio/apache-jmeter-3.3/bin$
```

Como vemos, se envían y reciben 538,2 peticiones por segundo, algo más lento que usando ab.

Cambiando valores del sistema para optimizar

Vamos a intentar optimizar nuestro servidor. Para esto modificaremos algunas variables de archivos de configuración.

Kernel

He modificado algunos archivos de `/proc/sys/kernel`, como `"num_max_threads"`, o `"perf_cpu_time_max_percent"`. Para ello he usado el comando:

```
sysctl -w kernel.<nombre>
```

Sin embargo, no he notado diferencias destacables, seguramente porque el sistema no estaba demasiado ocupado. Con más carga creo que sí que habría diferencias.

LAMP

HostnameLookups

En Apache tenemos la opción `HostnameLookups`, que viene desactivada por defecto.

Activar esta opción agrega latencia a cada solicitud porque requiere una búsqueda de DNS para completar antes de que la solicitud finalice.

Se recomienda realizar este tipo de posprocesamiento de los archivos de registro en una máquina que no sea la máquina servidor de producción web, para que esta actividad no afecte negativamente al rendimiento del servidor.

Vamos a testear el server con ab con esta opción primero activada, y después desactivada.

Esta variable se cambia (o añade, si no está) en el archivo `/etc/apache2/apache2.conf`

```
# (the actual bytes sent including headers) instead of %b (the size of the
# requested file), because the latter makes it impossible to detect partial
# requests.
#
# Note that the use of %X-Forwarded-For}i instead of %h is not recommended.
# Use mod_remoteip instead.
#
LogFormat "%u:%p %h %l %u %t \"%r\" %>s %D \"%{Referer}i\" \"%{User-Agent}i\""
LogFormat "%h %l %u %t \"%r\" %>s %D \"%{Referer}i\" \"%{User-Agent}i\"" combi
LogFormat "%h %l %u %t \"%r\" %>s %D" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

HostnameLookups on
"apache2.conf" 223L, 7135C escritos
root@ubuntu:/etc/apache2# systemctl restart apache2.service
root@ubuntu:/etc/apache2# _
```

Usando desde otro sistema el comando

```
$ ab -c5 -n1000 http://192.168.57.102/
```

Obtenemos los siguientes resultados.

```
Server Software:      Apache/2.4.18
Server Hostname:      192.168.57.102
Server Port:          80

Document Path:        /
Document Length:      11321 bytes

Concurrency Level:     5
Time taken for tests:  8.111 seconds
Complete requests:     1000
Failed requests:       0
Total transferred:     11595000 bytes
HTML transferred:      11321000 bytes
Requests per second:   123.29 [#/sec] (mean)
Time per request:      40.556 [ms] (mean)
Time per request:      8.111 [ms] (mean, across all concurrent requests)
Transfer rate:         1396.01 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:    0    0   0.2      0    6
Processing: 23   40  16.6     36   148
Waiting:    0    0   0.4      0    6
Total:      23   40  16.6     36   148

Percentage of the requests served within a certain time (ms)
 50%    36
 66%    40
 75%    43
 80%    46
 90%    58
 95%    72
 98%   104
 99%   113
100%   148 (longest request)
julio@julio-VirtualBox:~$
```

Se han enviado 1000 peticiones, con una concurrencia de 5, y el test ha tardado 8.111 segundos.

Ahora vamos a probar a hacer el mismo test con la opción HostnameLookups desactivada:

```
Server Hostname:      192.168.57.102
Server Port:          80

Document Path:        /
Document Length:      11321 bytes

Concurrency Level:     5
Time taken for tests:  0.447 seconds
Complete requests:     1000
Failed requests:       0
Total transferred:     11595000 bytes
HTML transferred:      11321000 bytes
Requests per second:   2239.05 [#/sec] (mean)
Time per request:      2.233 [ms] (mean)
Time per request:      0.447 [ms] (mean, across all concurrent requests)
Transfer rate:         25353.30 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:    0    1   0.8      0    8
Processing: 0    2   1.2      1    9
Waiting:    0    1   1.1      1    8
Total:      1    2   1.5      2   10
WARNING: The median and mean for the initial connection time are not within a normal deviation
These results are probably not that reliable.

Percentage of the requests served within a certain time (ms)
 50%    2
 66%    2
 75%    2
 80%    3
 90%    4
 95%    5
 98%    7
 99%    8
100%   10 (longest request)
julio@julio-VirtualBox:~$
```

Vemos que el test ha tardado 0.447 segundos, es decir, aproximadamente 16 veces menos que en el anterior test. Por lo tanto, nos conviene mantener la opción desactivada. Este resultado es normal, ya que la búsqueda DNS para cada solicitud consume mucho tiempo.

Como la diferencia es tan evidente, no es necesario hacer varias veces el mismo test y sacar la media, con una sola prueba basta.

FollowSymLinks y SymLinksIfOwnerMatch

Siempre que en las opciones no esté FollowSymLinks activado, o si tenemos SymLinksIfOwnerMatch activado, Apache necesitará emitir llamadas adicionales al sistema para verificar los enlaces simbólicos. (Una llamada adicional por componente de nombre de archivo).

Por ejemplo, si en el archivo de configuración tenemos:

```
<Directory />
```

```
Options SymLinksIfOwnerMatch
```

```
</Directory>
```

y se realiza una solicitud para el URI /index.html, Apache realizará lstat (2) en "/www", "/www/htdocs" y "/www/htdocs/index.html". Los resultados de estos lstats nunca se almacenan en caché, por lo que se producirán en cada solicitud. Para evitar esto, podemos añadir las siguientes líneas al archivo de configuración:

```
<Directory />
```

```
Options -FollowSymLinks + SymLinksIfOwnerMatch
```

```
</Directory>
```

Esto al menos evita las comprobaciones adicionales, ahorrándonos unos milisegundos.

Vamos a probar añadiendo al archivo de configuración estas opciones. Con las opciones por defecto, he ejecutado el test ab 10 veces, y la media aritmética de lo que han tardado es de 0.456 segundos.

Con estas opciones, he vuelto a hacer 10 testeos, y la media aritmética de lo que ha tardado es de 0.402 segundos. La diferencia, como vemos en esta captura, no es demasiado notable, pero nos ahorramos esos milisegundos de las llamadas para verificar los enlaces simbólicos.

```
Document Path:      /
Document Length:    11321 bytes

Concurrency Level:   5
Time taken for tests: 0.388 seconds
Complete requests:   1000
Failed requests:     0
Total transferred:   11595000 bytes
HTML transferred:    11321000 bytes
Requests per second: 2574.08 [#/sec] (mean)
Time per request:    1.942 [ms] (mean)
Time per request:    0.388 [ms] (mean, across all concurrent requests)
Transfer rate:       29146.88 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:    0    1  0.7    0    6
Processing:  0    1  1.0    1   10
Waiting:    0    1  0.9    1    7
Total:      1    2  1.3    1   10
WARNING: The median and mean for the initial connection time are not within a normal deviation
These results are probably not that reliable.
```

Memory-limit

En php hay una variable llamada memory-limit. Se encuentra en:

/etc/php/7.0/apache2/php.ini

La variable tenía 128M, y yo le he puesto 16M, pero al cambiarla, en los tests no he notado diferencia. Seguramente sea porque ni siquiera haya necesitado los 16MB completos.

Opciones de MySQL

Hay muchas opciones para optimizar MySQL, sin embargo, en mi equipo no he notado diferencias, y puede ser por dos razones:

- Nuestro servidor es demasiado simple (sólo tiene el archivo http por defecto)
- Estas opciones están pensadas para optimizar un servidor con Moodle

Las opciones se pueden cambiar en ***/etc/mysql/my.cnf*** y son las siguientes:

- ***query_cache_type = 1***
- ***query_cache_size = 36M***
- ***query_cache_min_res_unit = 2K***
- ***table_cache = 256***