

INGENIERÍA DE SERVIDORES (2016-2017)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Memoria Práctica 4

Cristian Vélez Ruiz

23 de diciembre de 2016

Índice

1. Cuestión 1	4
2. Cuestión 2	8
3. Cuestión 3	9
4. Cuestión 4	17
5. Cuestión 5	23

Índice de figuras

1.1. Descargando Heaven	4
1.2. Desempaquetando Heaven	5
1.3. Listando Archivos Heaven	5
1.4. Ejecutando Heaven	6
1.5. Configuración de Heaven	6
1.6. Iniciando benchmark	7
1.7. Resultados de Heaven	7
2.1. Ejecución de ab	8
2.2. Top de ab	8
3.1. Página de XAMPP	9
3.2. Paso 1 instalación XAMPP	9
3.3. Paso 2 instalación XAMPP	10
3.4. Paso 3 instalación XAMPP	10
3.5. Paso 4 instalación XAMPP	10
3.6. Paso 5 instalación XAMPP	11
3.7. Panel Control XAMPP	11
3.8. Benchmark Ubuntu 1	12
3.9. Benchmark Ubuntu 2	12
3.10. Benchmark Ubuntu 3	13
3.11. Benchmark CentOS 1	13
3.12. Benchmark CentOS 2	14
3.13. Benchmark CentOS 3	15
3.14. Benchmark Windows 1	15
3.15. Benchmark Windows 2	16
3.16. Benchmark Windows 3	17
4.1. Descarga jmeter	18
4.2. Descarga jmeter 2	18
4.3. Descomprimiendo jmeter 3	18
4.4. Ejecutando jmeter 4	19
4.5. Ejecutando jmeter 5	19

4.6. Ejecutando jmeter 6	20
4.7. Ejecutando jmeter 7	20
4.8. Ejecutando jmeter 8	21
4.9. Ejecutando jmeter 9	21
4.10. Ejecutando jmeter 10	21
4.11. Ejecutando jmeter 11	22
4.12. Ejecutando jmeter 12	22
4.13. Ejecutando jmeter 13	22
4.14. Ejecutando jmeter 14	22
5.1. Benchmark en el anfitrión	24
5.2. Benchmark en la máquina virtual	24

Índice de tablas

- 1. Seleccione, instale y ejecute uno, comente los resultados. Atención: no es lo mismo un benchmark que una suite, instale un benchmark**

He elegido un benchmark que se centra en la renderización de gráficos en Linux, su nombre es Heaven [4].

Ejecución Heaven Benchmark:

1. Debemos ir a la pagina principal de Heaven y descargar el archivo, pesa unos 200 MiB.

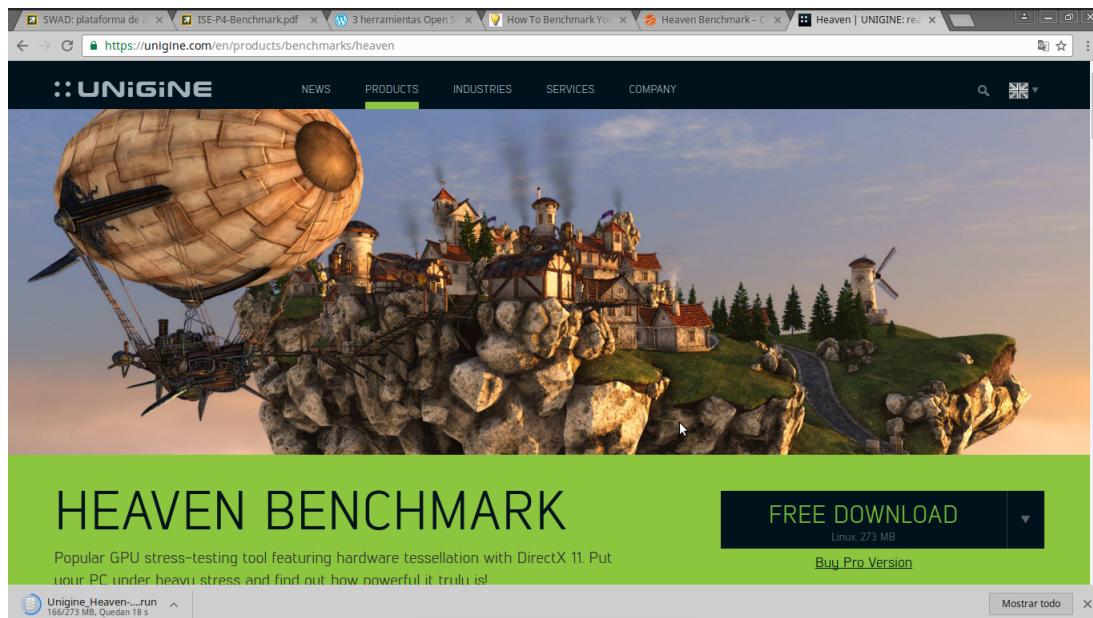


Figura 1.1: Descargando Heaven

2. Una vez tenemos el archivo descargado tenemos que darle permisos de ejecución y ejecutarlo.

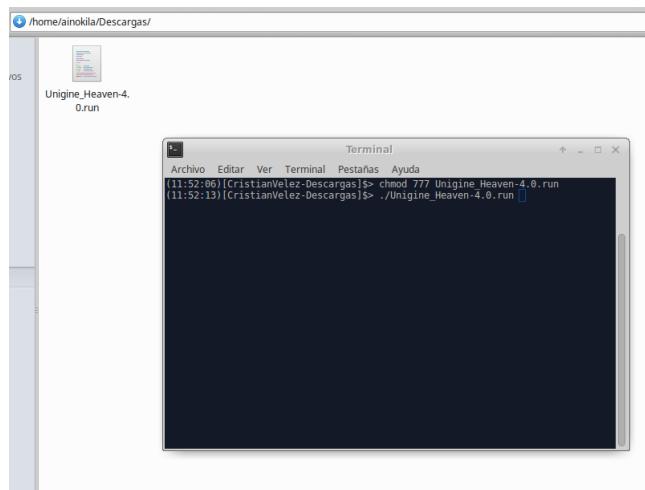


Figura 1.2: Desempaquetando Heaven

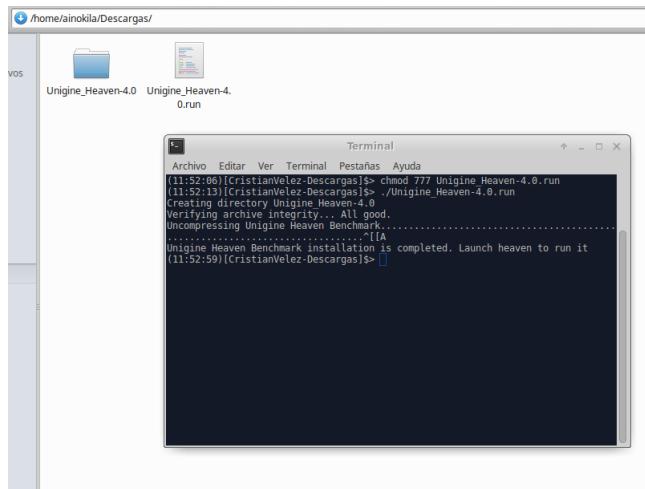


Figura 1.3: Listando Archivos Heaven

3. Ahora nos ha generado un directorio, en el cual debemos acceder y ejecutar `./heaven`.

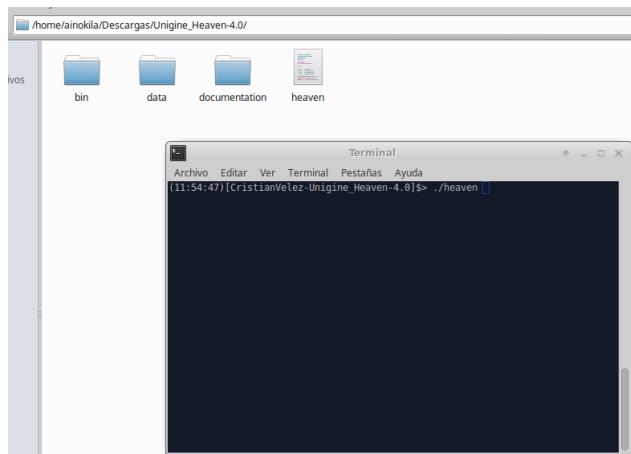


Figura 1.4: Ejecutando Heaven

4. Ahora se nos abrirá la interfaz inicial de Heaven para configurar el benchmark, mi configuración será la siguiente:

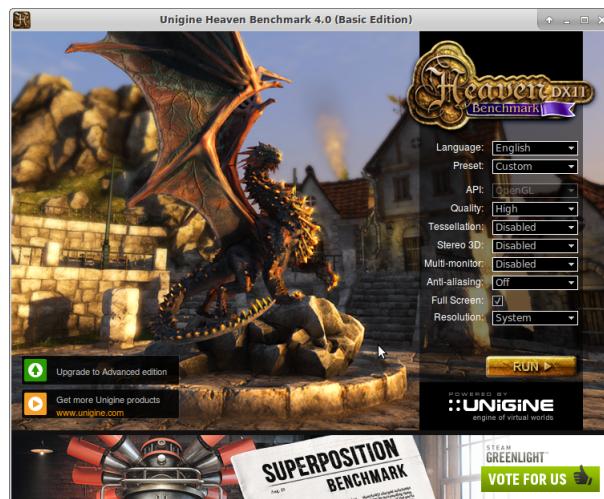


Figura 1.5: Configuración de Heaven

5. Simplemente ahora le damos a *run* y cambiará la interfaz a otra y debemos seleccionar en la parte superior de arriba benchmark.

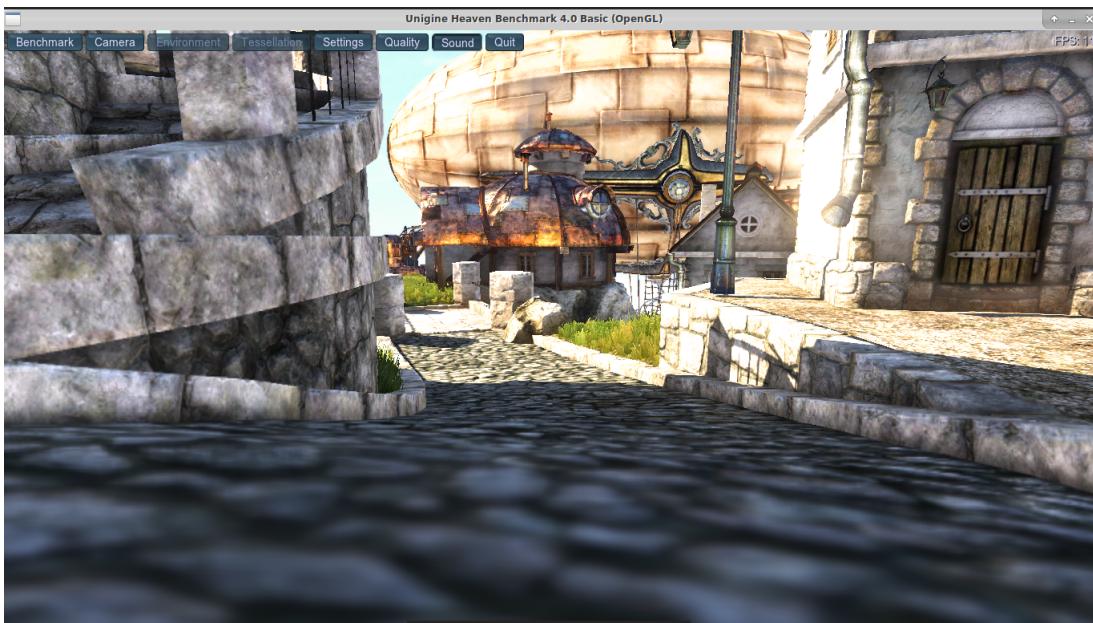


Figura 1.6: Iniciando benchmark

6. Una vez hemos seleccionado benchmark debemos esperar a que se generen 26 escenas distintas de las cuales luego obtendremos los resultados.

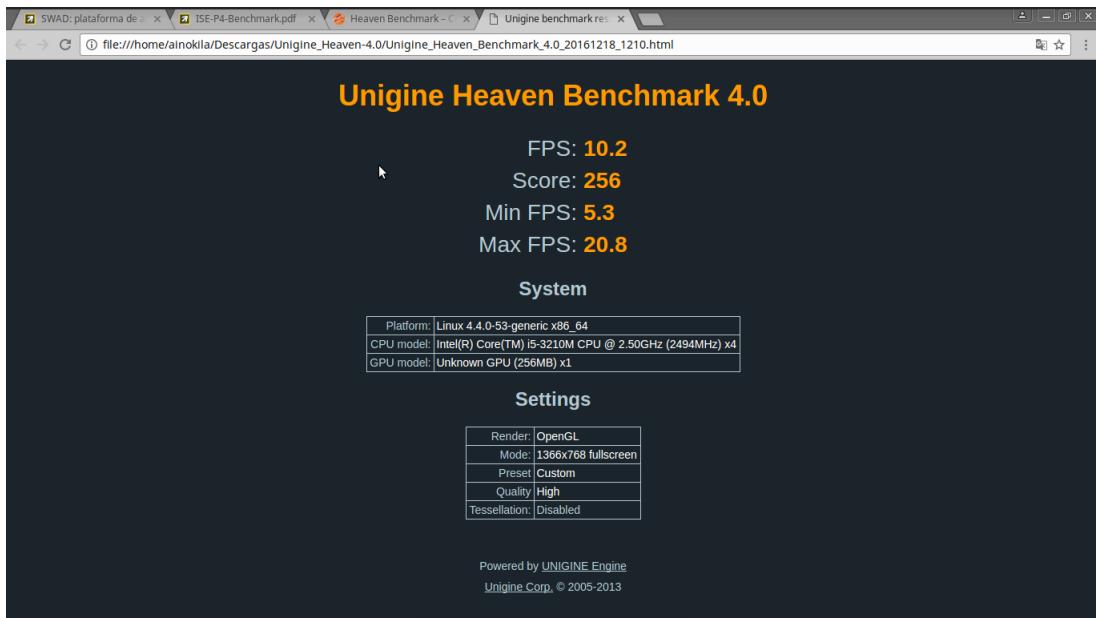


Figura 1.7: Resultados de Heaven

7. Los resultados de ejecutar Heaven en mi máquina portátil han sido que ha obtenido una generación media de fotogramas por segundo de 10.2, una puntuación de 256, también nos informa de cuales ha sido su pico máximo y mínimo de fotogramas por segundo, en este caso 20.8 y 5.3.

Después simplemente nos informa de el tipo de plataforma donde se ha ejecutado y en que CPU-GPU y también la configuración del renderizado.

Este tipo de Benchmark nos sirve para hacer comparaciones de distintas tarjetas gráficas con la facilidad de poder realizar la ejecución en Linux.

2. De los parámetros que le podemos pasar al comando ¿Qué significa -c 5 ? ¿y -n 100? Monitorice la ejecución de ab contra alguna máquina (cualquiera) ¿cuántas “tareas” crea ab en el cliente?

El parámetro **-c 5** indica que vamos a enviar las peticiones en grupos de 5 y con el parámetro **-n 100** indicamos el numero de peticiones totales que vamos a enviar en este caso 100 peticiones.

Voy a ejecutar *ab -n 5000 -c 50 http://192.168.56.102/*, desde la máquina de Ubuntu hacia CentOS.

```
(16:17:37)[CristianVelez--]$: ab -n 1000 -c 5 http://192.168.56.102/
This is ApacheBench, Version 2.3 <Revision: 1528965 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.56.102 (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/2.4.6
```

Figura 2.1: Ejecución de ab

En otra terminal a la misma vez que se ejecuta ab, vamos ejecutar top para ver cuantos procesos crea.

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
3868	almokilla	20	0	60432	5424	3912	R	44,5	0,3	0:08.36	ab
3872	root	20	0	92324	29668	2864	R	3,7	1,4	0:08.11	/usr/share/+
3	root	20	0	0	0	0	R	1,7	0,0	0:01.02	ksortirad@

Figura 2.2: Top de ab

Como se puede apreciar ab crea un proceso en el cliente.

3. Ejecute ab contra a las tres máquinas virtuales (desde el SO anfitrión a las máquina virtuales de la red local) una a una (arrancadas por separado). ¿Cuál es la que proporciona mejores resultados? Muestre y coméntelos. (Use como máquina de referencia Ubuntu Server para la comparativa).

En mi caso para poder ejecutar el benchmark de Apache en Windows 10 debo instalar XAMPP, es una distribución de Apache que también contiene algunas bases de datos para poder instalarlas en Windows.

Instalando XAMPP:

1. Primero debemos ir a la página principal de XAMPP [3] y lo descargamos.



Figura 3.1: Página de XAMPP

2. Ahora lo ejecutamos y damos a next.

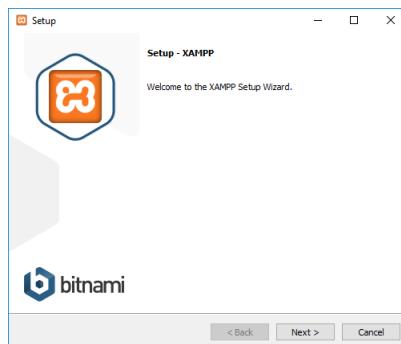


Figura 3.2: Paso 1 instalación XAMPP

3. Seleccionamos los componentes que queremos instalar, en mi caso Apache, PHP y phpMyAdmin.

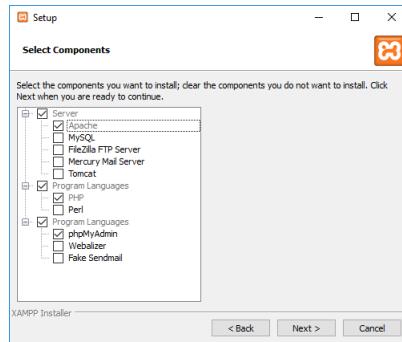


Figura 3.3: Paso 2 instalación XAMPP

4. Seleccionamos la ruta donde queremos que se instale.

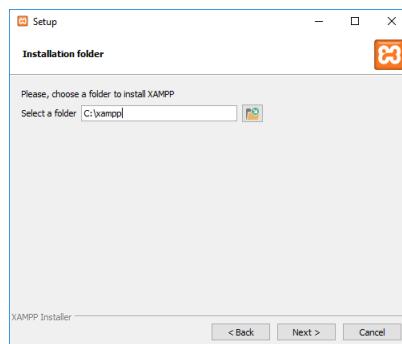


Figura 3.4: Paso 3 instalación XAMPP

5. Ya comienza a instalarse.



Figura 3.5: Paso 4 instalación XAMPP

6. Debemos permitir el acceso a través del firewall de Windows.

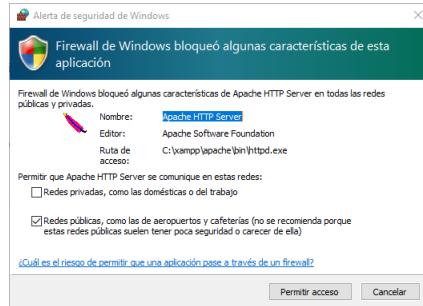


Figura 3.6: Paso 5 instalación XAMPP

7. Una vez terminado de instalarse se nos abrirá la interfaz del panel de Control de XAMPP y debemos darle a start en Apache para ejecutar el servicio.

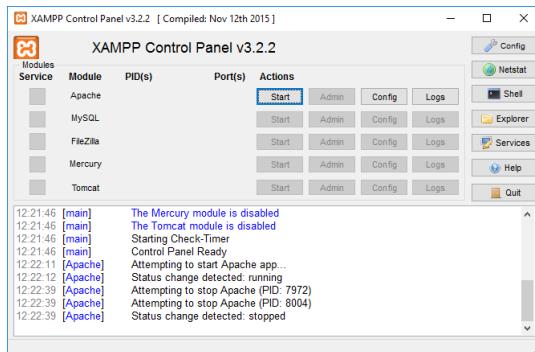


Figura 3.7: Panel Control XAMPP

Una vez tenemos apache instalado ya contamos con su herramienta ab (Apache Benchmark) para poder realizar las pruebas en las máquinas virtuales desde el anfitrión.

Antes de empezar configuraré la página index.html para el benchmark en los tres servidores para que estén en igualdad de condiciones.

El archivo index.html que utilizaré contiene html con un título y h1,h2,h3,h4,h5 y h6 con texto generado en latín.

Al igual que en el ejercicio anterior mandare ejecutaré el benchmark con:

```
ab -n 5000 -c 50 http://host/index.html
```

Generará 5000 peticiones a cada servidor y las mandará en paquetes de 50.

1. **Benchmark Ubuntu:** Ejecutaré `ab -n 5000 -c 50 http://192.168.23.2/index.html` dos veces para verificar que se obtiene un tiempo medio.

```
Cristian@DESKTOP-RPLP927 c:\xampp  
# ab -n 5000 -c 50 http://192.168.23.2/index.html
```

Figura 3.8: Benchmark Ubuntu 1

Ejecución 1:

```
Benchmarking 192.168.23.2 (be patient)  
Completed 500 requests  
Completed 1000 requests  
Completed 1500 requests  
Completed 2000 requests  
Completed 2500 requests  
Completed 3000 requests  
Completed 3500 requests  
Completed 4000 requests  
Completed 4500 requests  
Completed 5000 requests  
Finished 5000 requests  
  
Server Software: Apache/2.4.7  
Server Hostname: 192.168.23.2  
Server Port: 80  
  
Document Path: /index.html  
Document Length: 12084 bytes  
  
Concurrency Level: 50  
Time taken for tests: 2.229 seconds  
Complete requests: 5000  
Failed requests: 0  
Total transferred: 61785000 bytes  
HTML transferred: 60420000 bytes  
Requests per second: 2242.88 [#/sec] (mean)  
Time per request: 22.293 [ms] (mean)  
Time per request: 0.446 [ms] (mean, across all concurrent requests)  
Transfer rate: 27065.64 [Kbytes/sec] received  
  
Connection Times (ms)  
min mean[+/-sd] median max  
Connect: 0 0 1.0 0 21  
Processing: 6 22 7.7 19 60  
Waiting: 6 19 7.3 17 56  
Total: 6 22 7.8 19 60  
  
Percentage of the requests served within a certain time (ms)  
50% 19  
60% 20  
75% 25  
80% 29  
90% 35  
95% 39  
99% 43  
99.9% 46  
100% 60 (longest request)  
Cristian@DESKTOP-RPLP927 c:\xampp
```

Figura 3.9: Benchmark Ubuntu 2

Ejecución 2:

```

Benchmarking 192.168.23.2 (be patient)
Completed 500 requests
Completed 1000 requests
Completed 1500 requests
Completed 2000 requests
Completed 2500 requests
Completed 3000 requests
Completed 3500 requests
Completed 4000 requests
Completed 4500 requests
Completed 5000 requests
Finished 5000 requests

Server Software:      Apache/2.4.7
Server Hostname:     192.168.23.2
Server Port:          80

Document Path:        /index.html
Document Length:      12684 bytes

Concurrency Level:    50
Time taken for tests: 2.564 seconds
Complete requests:   5000
Failed requests:      0
Total transferred:   61785000 bytes
HTML transferred:    60420000 bytes
Requests per second: 1950.35 [#/sec] (mean)
Time per request:   25.636 [ms] (mean)
Time per request:   0.513 [ms] (mean, across all concurrent requests)
Transfer rate:       23535.61 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0  1.1    0    16
Processing:    7   25 10.4   22    79
Waiting:       6   22 10.0   19    78
Total:         7   25 10.5   22    79

Percentage of the requests served within a certain time (ms)
  50%   22
  60%   26
  75%   29
  80%   31
  90%   36
  95%   44
  98%   64
  99%   71
 100%   79 (longest request)

Cristian@DESKTOP-RPLP927 c:\xampp

```

Figura 3.10: Benchmark Ubuntu 3

2. **Benchmark CentOS:** Ejecutaré `ab -n 5000 -c 50 http://192.168.23.3/index.html` dos veces para verificar que se obtiene un tiempo medio.

```

Cristian@DESKTOP-RPLP927 c:\xampp
# ab -n 5000 -c 50 http://192.168.23.3/index.html

```

Figura 3.11: Benchmark CentOS 1

Ejecución 1:

```

Benchmarking 192.168.23.3 (be patient)
Completed 500 requests
Completed 1000 requests
Completed 1500 requests
Completed 2000 requests
Completed 2500 requests
Completed 3000 requests
Completed 3500 requests
Completed 4000 requests
Completed 4500 requests
Completed 5000 requests
Finished 5000 requests

Server Software:      Apache/2.4.6
Server Hostname:     192.168.23.3
Server Port:          80

Document Path:        /index.html
Document Length:     12084 bytes

Concurrency Level:   50
Time taken for tests: 2.261 seconds
Complete requests:  5000
Failed requests:    0
Total transferred:  61800000 bytes
HTML transferred:  60420000 bytes
Requests per second: 2221.36 [#/sec] (mean)
Time per request:   22.610 [ms] (mean)
Time per request:   0.452 [ms] (mean, across all concurrent requests)
Transfer rate:       26691.85 [Kbytes/sec] received

Connection Times (ms)
              min     mean[+/-sd] median     max
Connect:        0       1.1      0       16
Processing:    7      22      7.6     38      58
Waiting:       7      21      7.5     19      58
Total:         7      22      7.6     20      58

Percentage of the requests served within a certain time (ms)
  50%    20
  66%    24
  75%    26
  80%    28
  90%    34
  95%    37
  98%    43
  99%    46
100%   58 (longest request)

Cristian@DESKTOP-RPLP927 c:\xampp
#
```

Figura 3.12: Benchmark CentOS 2

Ejecución 2:

```

Benchmarking 192.168.23.3 (be patient)
Completed 500 requests
Completed 1000 requests
Completed 1500 requests
Completed 2000 requests
Completed 2500 requests
Completed 3000 requests
Completed 3500 requests
Completed 4000 requests
Completed 4500 requests
Completed 5000 requests
Finished 5000 requests

Server Software:      Apache/2.4.6
Server Hostname:     192.168.23.3
Server Port:          80

Document Path:        /index.html
Document Length:     12084 bytes

Concurrency Level:   50
Time taken for tests: 2.302 seconds
Complete requests:   5000
Failed requests:    0
Total transferred:  61800000 bytes
HTML transferred:  60420000 bytes
Requests per second: 2171.89 #[#/sec] (mean)
Time per request:   23.021 [ms] (mean)
Time per request:   0.460 [ms] (mean, across all concurrent requests)
Transfer rate:       26215.37 [kbytes/sec] received

Connection Times (ms)
              min     mean[+/-sd] median     max
Connect:        0      0.6      0     11
Processing:    11     22.3.5     22     45
Waiting:       8     22.3.6     21     39
Total:         11     23.3.6     23     45

Percentage of the requests served within a certain time (ms)
 50%    23
 60%    24
 75%    24
 80%    25
 90%    26
 95%    30
 98%    34
 99%    37
100%   45 (longest request)

Cristian@DESKTOP-RPLP927 c:\xampp
#

```

Figura 3.13: Benchmark CentOS 3

3. Benchmark Windows Server:

Ejecutaré `ab -n 5000 -c 50 http://192.168.23.4/index.html` dos veces para verificar que se obtiene un tiempo medio.

```

Cristian@DESKTOP-RPLP927 c:\xampp
# ab -n 5000 -c 50 http://192.168.23.4/index.html

```

Figura 3.14: Benchmark Windows 1

Ejecución 1:

```

Benchmarking 192.168.23.4 (be patient)
Completed 500 requests
Completed 1000 requests
Completed 1500 requests
Completed 2000 requests
Completed 2500 requests
Completed 3000 requests
Completed 3500 requests
Completed 4000 requests
Completed 4500 requests
Completed 5000 requests
Finished 5000 requests

Server Software:      Microsoft-IIS/7.5
Server Hostname:     192.168.23.4
Server Port:          80

Document Path:        /index.html
Document Length:     1282 bytes

Concurrency Level:   50
Time taken for tests: 5.757 seconds
Complete requests:  5000
Failed requests:    0
Non-2xx responses:  5000
Total transferred:  7190000 bytes
HTML transferred:   6410000 bytes
Requests per second: 868.49 [#/sec] (mean)
Time per request:   57.571 [ms] (mean)
Time per request:   1.151 [ms] (mean, across all concurrent requests)
Transfer rate:       1219.62 [kbytes/sec] received

Connection Times (ms)
              min     mean[+/-sd] median     max
Connect:        0       1.4      1       14
Processing:    1      56     19.8     62     106
Waiting:       1      43     18.4     44     106
Total:         1      57     20.2     63     110

Percentage of the requests served within a certain time (ms)
  50%    63
  60%    67
  75%    70
  80%    73
  90%    76
  95%    83
  99%    99
  99.9% 95
 100%   110 (longest request)

Cristian@DESKTOP-RPLP927 c:\xampp
#
```

Figura 3.15: Benchmark Windows 2

Ejecución 2:

```

Benchmarking 192.168.23.4 (be patient)
Completed 500 requests
Completed 1000 requests
Completed 1500 requests
Completed 2000 requests
Completed 2500 requests
Completed 3000 requests
Completed 3500 requests
Completed 4000 requests
Completed 4500 requests
Completed 5000 requests
Finished 5000 requests

Server Software:      Microsoft-IIS/7.5
Server Hostname:     192.168.23.4
Server Port:          80

Document Path:        /index.html
Document Length:     1282 bytes

Concurrency Level:    50
Time taken for tests: 5.528 seconds
Complete requests:   5000
Failed requests:      0
Non-2xx responses:   5000
Total transferred:   7190000 bytes
HTML transferred:    6410000 bytes
Requests per second: 984.51 (#/sec) (mean)
Time per request:   55.279 [ms] (mean)
Time per request:   1.186 [ms] (mean, across all concurrent requests)
Transfer rate:       1270.28 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    1  1.3     1    17
Processing:    0    54 24.0    63    97
Waiting:       0    40 21.7    44    95
Total:         1    55 24.5    64    98

Percentage of the requests served within a certain time (ms)
  50%    64
  66%    68
  75%    72
  80%    74
  85%    79
  90%    82
  95%    87
  99%    90
100%   98 (longest request)

Cristian@DESKTOP-RPLP927 c:\xampp
#

```

Figura 3.16: Benchmark Windows 3

Conclusión: Obtenemos la siguiente tabla con los tiempos de conexión en milisegundos.

	Ejecución 1	Ejecución 2	Media
Ubuntu	19	22	20.5
CentOS	20	23	21.5
Windows	63	64	63.5

Como podemos apreciar el servidor que mas rápido ha sido es Ubuntu, por muy poca diferencia seguido de CentOS y finalmente Windows Server, con la misma configuración en todas las maquinas.

4. Instale y siga el tutorial en realizando capturas de pantalla y comentándolas. En vez de usar la web de jmeter, haga el experimento usando sus máquinas virtuales ¿coincide con los resultados de ab?

Descarga:

1. Debemos ir a la página de descarga [1] de jmeter [2] y lo descargamos.

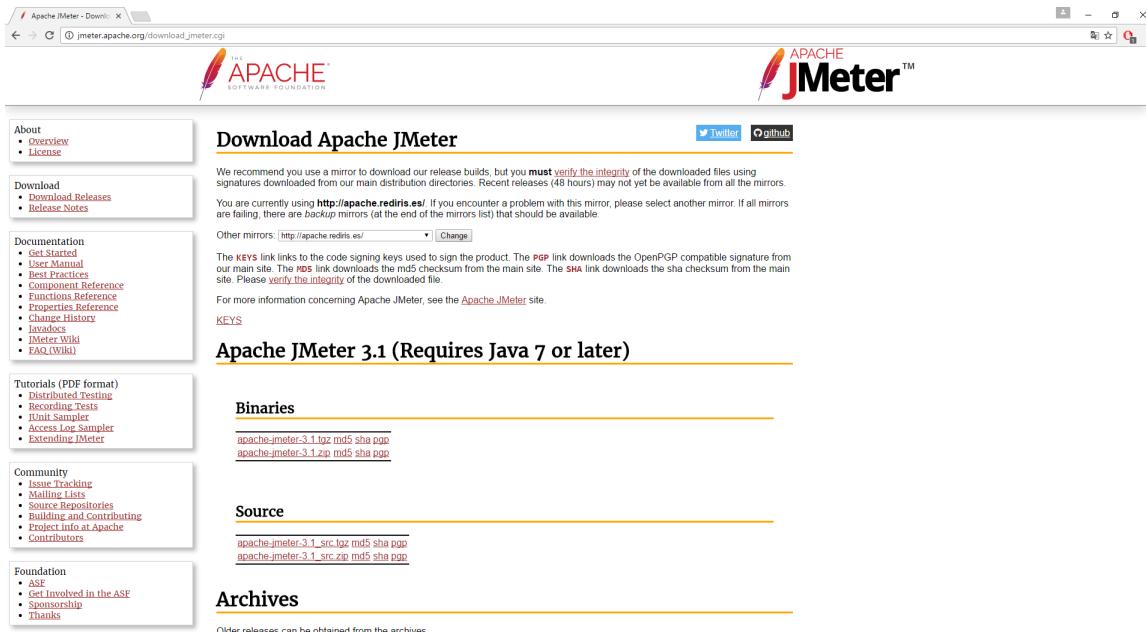


Figura 4.1: Descarga jmeter



Figura 4.2: Descarga jmeter 2

2. Descomprimimos el archivo y ejecutamos el programa a bin/ApacheJMeter

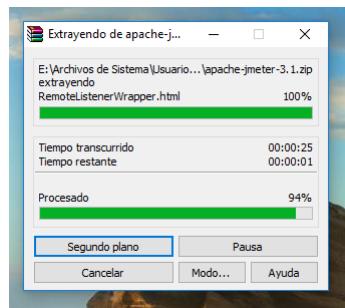


Figura 4.3: Descomprimiendo jmeter 3

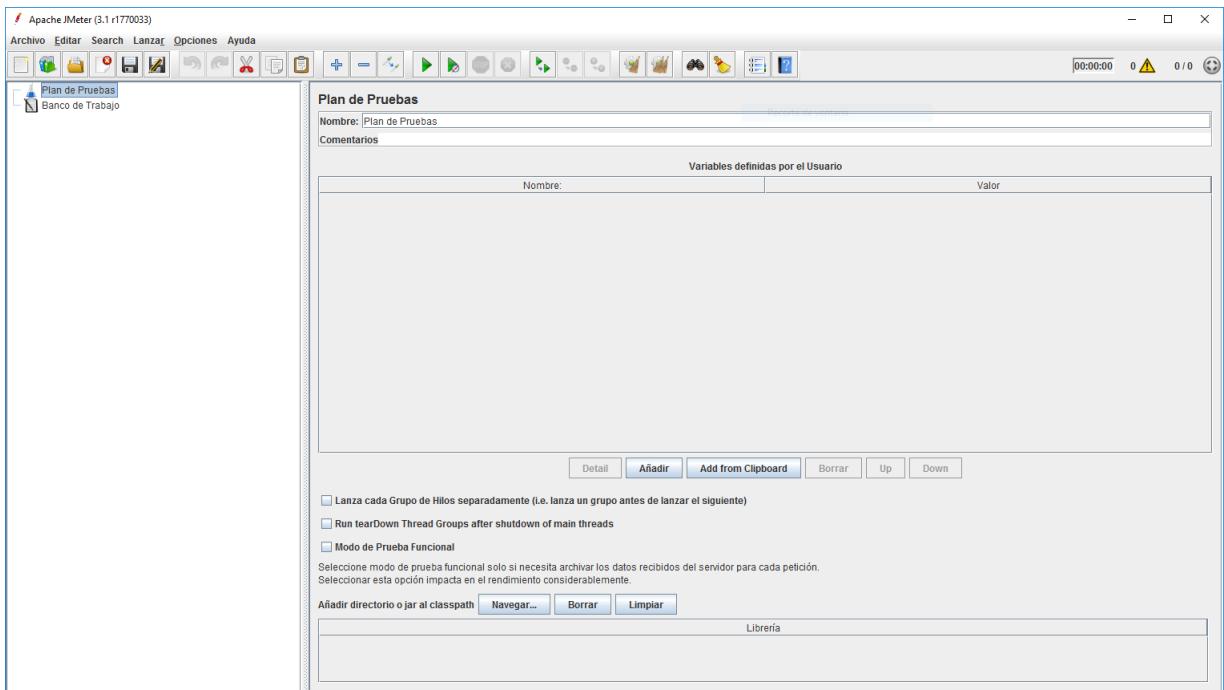


Figura 4.4: Ejecutando jmeter 4

Creación benchmark:

1. Añadimos el nombre al benchmark.



Figura 4.5: Ejecutando jmeter 5

2. Seguidamente damos botón derecho en el panel derecho en el nombre de nuestro benchmark, después a Añadir, y Grupo de Hilos.
Aquí se configuran los usuarios que accederán a cada uno de los servidores, en mi caso 1000 usuarios en un tiempo de 10 segundos.



Figura 4.6: Ejecutando jmeter 6

3. Ahora tenemos que añadir las peticiones http.

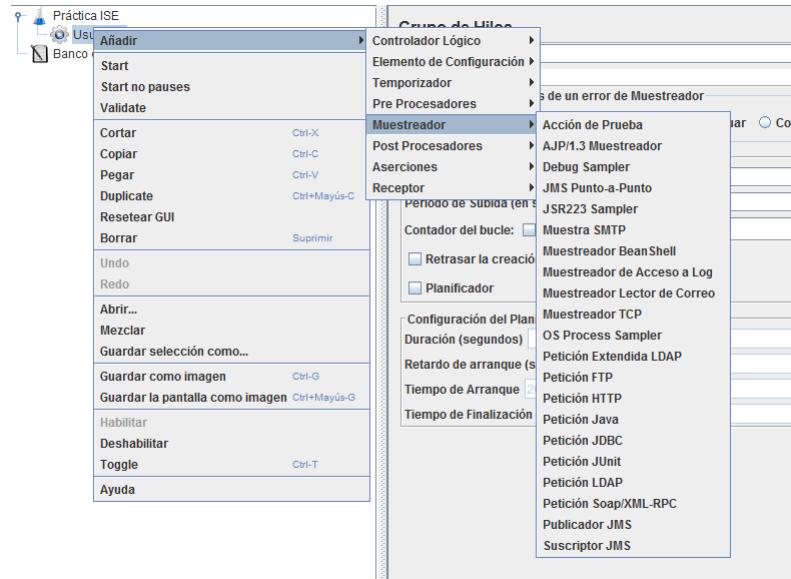


Figura 4.7: Ejecutando jmeter 7

4. Ahora tenemos que configurar las peticiones http, debemos hacer este paso para cada uno de los servidores con su respectiva ip y su index.html.

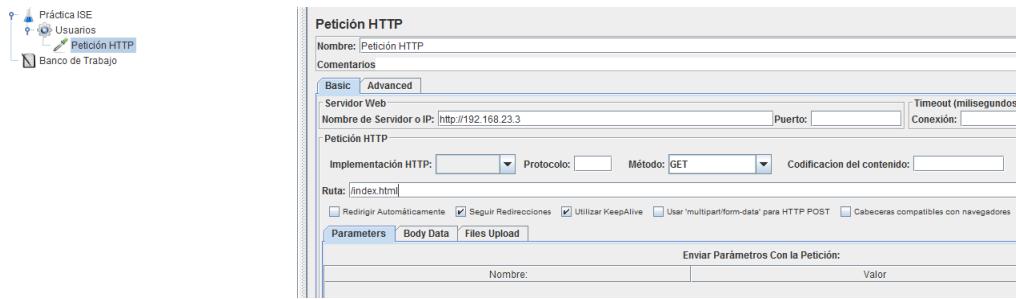


Figura 4.8: Ejecutando jmeter 8

5. Ahora tenemos que añadir como obtendremos los resultados de cada una de las peticiones anteriores que hemos declarado.

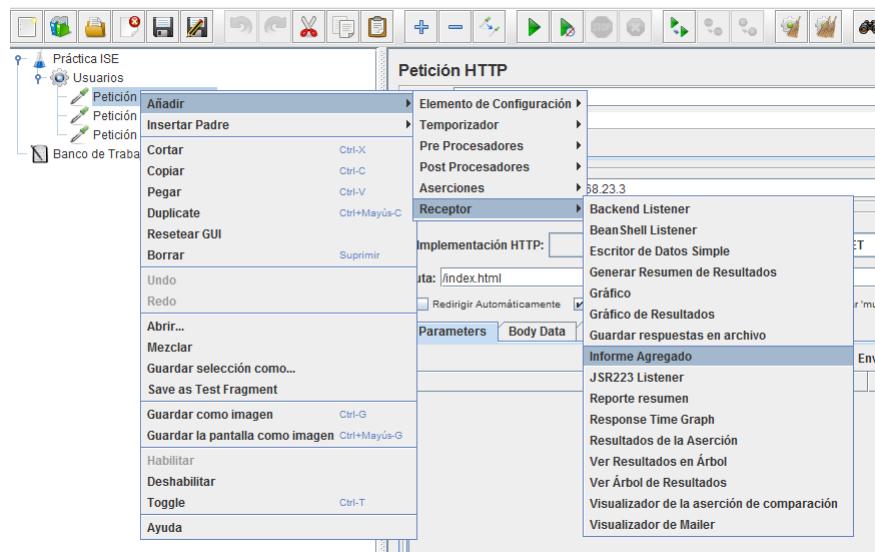


Figura 4.9: Ejecutando jmeter 9

6. Nos debe quedar así.

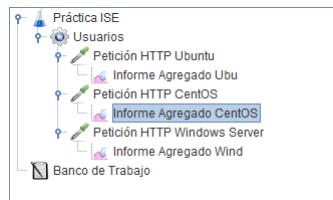


Figura 4.10: Ejecutando jmeter 10

Ejecución:

1. Debemos guardar el proceso para continuar,damos a si.



Figura 4.11: Ejecutando jmeter 11

2. Esperamos a que termine de ejecutarse.
3. Datos obtenidos de Ubuntu.

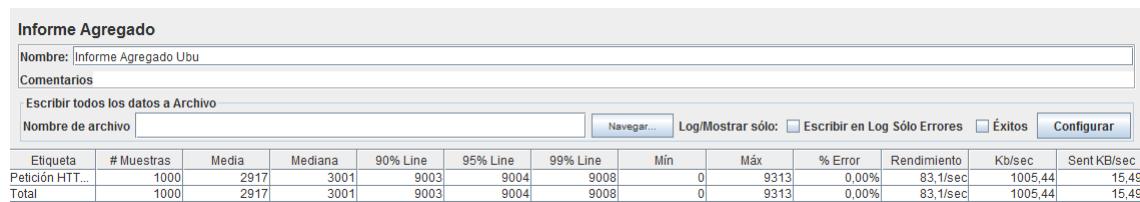


Figura 4.12: Ejecutando jmeter 12

4. Datos obtenidos de CentOS.

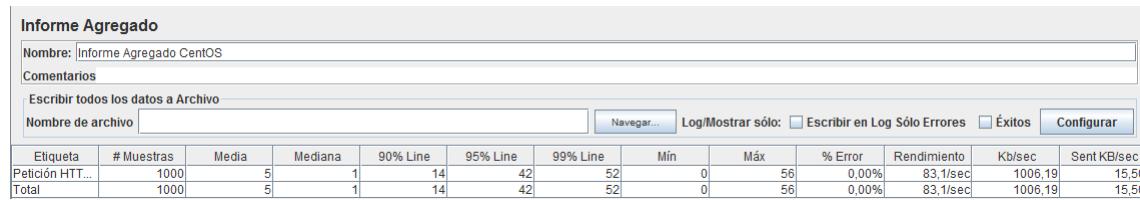


Figura 4.13: Ejecutando jmeter 13

5. Datos obtenidos de Windows Server.

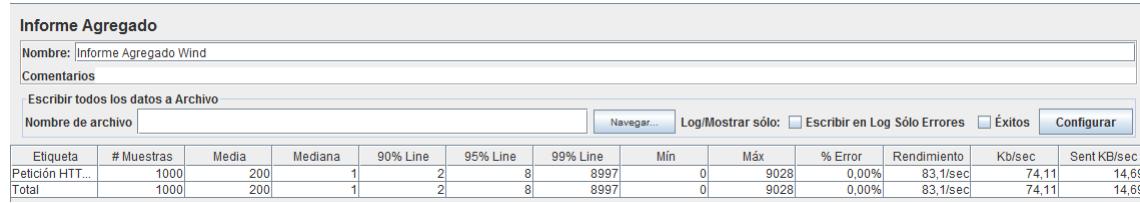


Figura 4.14: Ejecutando jmeter 14

Conclusión: Como podemos observar en las columnas kb/s, el que mas rendimiento ha obtenido es CentOS con 1006 kb/s, seguido de Ubuntu con 1005 kb/s y por último, con mucha diferencia Windows Server con 74 kb/s. Por tanto coinciden con los resultados obtenidos con ab, ya que entre Ubuntu y CentOS había muy poca diferencia.

5. Programe un benchmark usando el lenguaje que deseé. El benchmark debe incluir, Objetivo del benchmark, Métricas (unidades, variables, puntuaciones, etc.), Instrucciones para su uso, Ejemplo de uso analizando los resultados.

Mi objetivo del benchmark es comparar que servidor puede multiplicar la posición de una matriz con otra de una manera mas rápida.

He creado dos matrices de números aleatorios de 3000 filas y 3000 columnas, por tanto el numero de multiplicaciones que realizará será 9.000.000. Al ejecutar el benchmark obtenemos el tiempo en realizar las 9.000.000 multiplicaciones en segundos.

Código:

```
import random
import time

filas = 3000
columnas = 3000

matrizA= []
for i in range(filas):
    matrizA.append([])
    for j in range(columnas):
        matrizA [ i ].append(random.randint(0,100))

matrizB= []
for i in range(filas):
    matrizB.append([])
    for j in range(columnas):
        matrizB [ i ].append(random.randint(0,100))

matrizSol=[]
inicio = time.clock()
for i in range(filas):
    matrizSol.append([])
```

```

for j in range(columnas):
    matrizSol[i].append((matrizA[i][j]*matrizB[i][j]))

final = time.clock()

total = final - inicio

print("El tiempo que ha tardado es", total*10, "s")

```

Ambas maquinas se encuentran en el mismo estado, recién arrancadas y simplemente se ha ejecutado la orden:

python3 benchmark.py

Ejecución en máquina anfitriona:

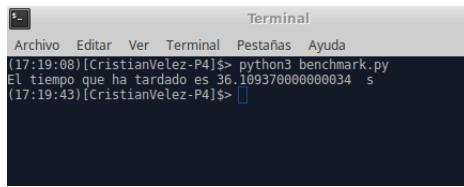


Figura 5.1: Benchmark en el anfitrión

Ejecución en máquina virtualizada:

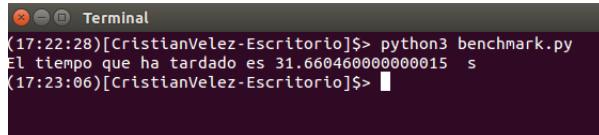


Figura 5.2: Benchmark en la máquina virtual

Conclusión: En mi caso tarda menos en ejecutar el benchmark la maquina virtual tras realizar varias comprobaciones.

Referencias

- [1] Descarga jmeter. http://jmeter.apache.org/download_jmeter.cgi, 2016.
- [2] Página principal de jmeter. <http://jmeter.apache.org>, 2016.
- [3] Página principal y de descarga de xampp. <https://www.apachefriends.org/es/index.html>, 2016.

- [4] Página principal de heaven benchmarl. <https://unigine.com/products/benchmarks/heaven/>, Noviembre de 2016.