

Ingeniería de Servidores

-Práctica 4-

Trabajo Realizado por:

José Delgado Dolset

DNI: 26050566-F

3ºC

Cuestión 1. Instale la aplicación. ¿Qué comando permite listar los benchmarks disponibles?

En ubuntu, podemos instalarlo simplemente usando (como root) el comando:

apt-get install phoronix-test-suite

Una vez instalado, tenemos que ejecutarlo, para que nos pregunte si queremos aceptar la licencia y si queremos permitir enviar a los desarrolladores estadísticas anónimas para mejorar el programa. Para ello escribimos:

phoronix-test-suite

Para listar los benchmarks disponibles, podemos utilizar los siguiente comando:

phoronix-test-suite list-available-tests → Para listar los benchmarks

phoronix-test-suite list-available-suites → Para listar las suites

Cuestión 2. Seleccione e instale tres que desee y ejecútelos. Cambie los parámetros de experimentación ¿aprecia cambios en los resultados?

Atención, no es lo mismo un benchmark que una suite, instale un benchmark.

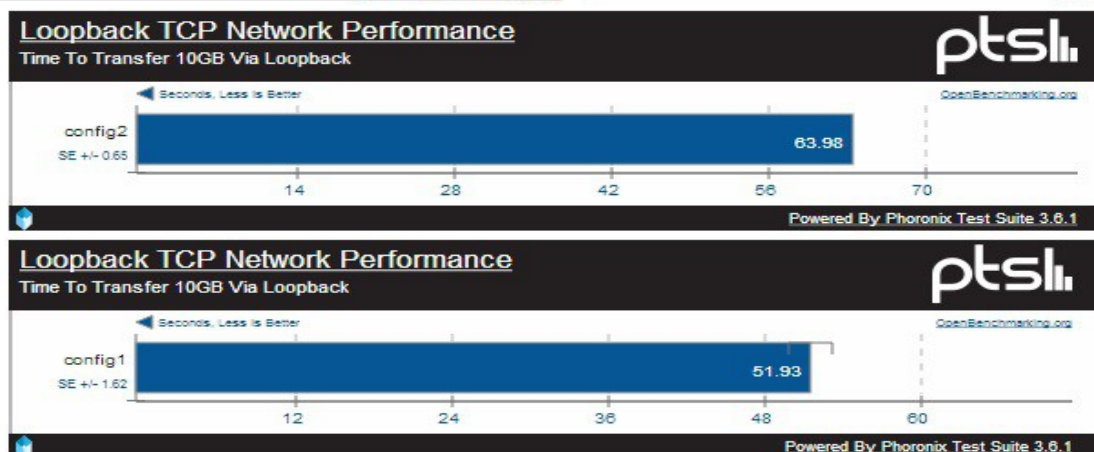
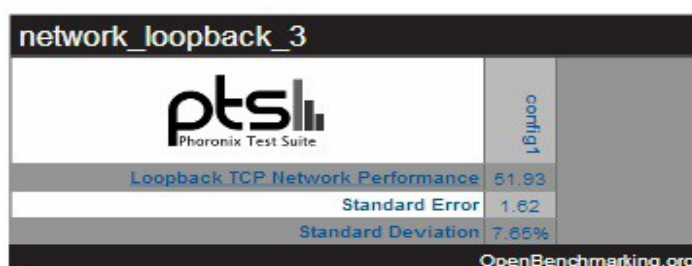
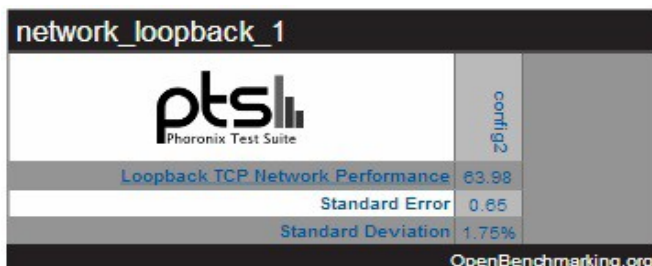
Lo primero es instalar los benchmarks. He elegido:

network_loopback.

Los resultados que tengo (variando la configuración de la tarjeta de red virtual entre bridged y nat) son los siguientes.

Config 1 = Bridged

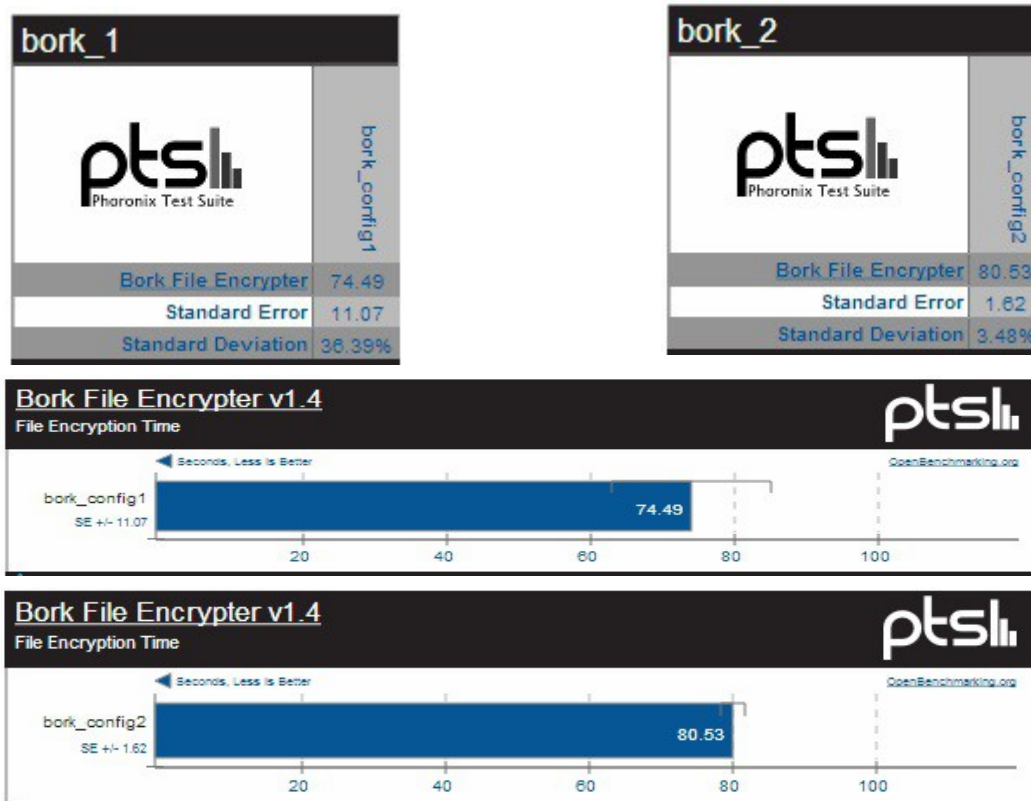
Config 2 = Nat



Las diferencias, como vemos, no son muy significativas, aunque se pueden notar.

Bork.

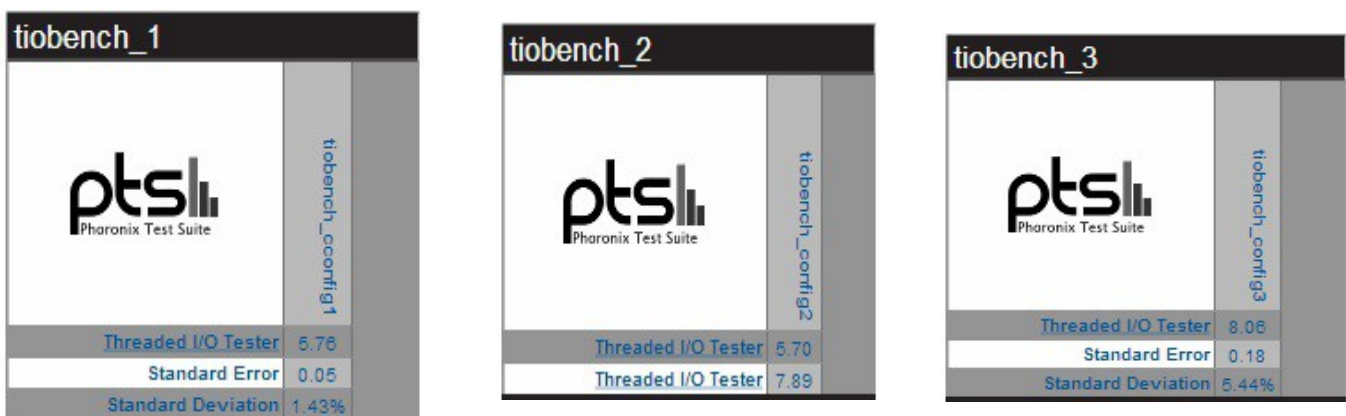
El test bork ejecutado dos veces nos da estos datos:



TIOBENCH

Ejecutamos el benchmark TIOBENCH tres veces, con distintas opciones:

- 1ª Write – Size Per Thread: 64 MB – Thread Count: 4
- 2ª Write – Size Per Thread: 128 MB – Thread Count: 4
- 3ª Write – Size Per Thread: 64 MB – Thread Count: 8





Observamos que la ejecución de 8 threads da muchos mejores resultados, mientras que el tamaño de cada thread no parece muy significativo en el resultado final.

Cuestión 3. De los parámetros que le podemos pasar al comando ¿Qué significa `-c 30` ? ¿y `-n 1000`?

La opción `-c` *concurrency* indica el número máximo de peticiones que se podrán ejecutar simultáneamente. Con `-c 30` lo ponemos a 30.

La opción `-n` *peticiones* establece el número de peticiones de página web que se harán al servidor. Con `-n 1000` establecemos que se hagan mil peticiones.

Cuestión 4. Ejecute `ab` contra a las máquinas virtuales una a una (arrancadas por separado) y muestre las estadísticas. ¿Cuál es la que proporciona mejores resultados? (Use como mínimo dos páginas webs distintas, una con imágenes y otra sin ellas).

Primero lo hacemos con ubuntu.

```
ppemg@UbuntuServer:~$ ab -c 30 -n 1000 localhost/sin_imagenes
This is ApacheBench, Version 2.3 <$Revision: 655654 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (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/2.2.22
Server Hostname:      localhost
Server Port:          80

Document Path:        /sin_imagenes
Document Length:      40417 bytes

Concurrency Level:    30
Time taken for tests:  0.176 seconds
Complete requests:    1000
Failed requests:       0
Write errors:          0
Total transferred:    40771000 bytes
HTML transferred:     40417000 bytes
Requests per second:  5690.03 [#/sec] (mean)
Time per request:      5.272 [ms] (mean)
Time per request:      0.176 [ms] (mean, across all concurrent requests)
Transfer rate:         226550.99 [Kbytes/sec] received


Connection Times (ms)
              min    mean[+/-sd] median    max
Connect:        0      1   1.3         0     13
Processing:      2      4   2.8         4     21
Waiting:         2      4   2.6         3     20
Total:           2      5   3.8         4     30


Percentage of the requests served within a certain time (ms)
 50%    4
 66%    4
 75%    5
 80%    5
 90%    6
 95%    7
 98%   20
 99%   29
100%   30 (longest request)
ppemg@UbuntuServer:~$
```



```
ppemg@UbuntuServer:~$ ab -c 30 -n 1000 localhost/con_imagenes
This is ApacheBench, Version 2.3 <$Revision: 655654 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (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/2.2.22
Server Hostname:      localhost
Server Port:          80


Document Path:        /con_imagenes
Document Length:       40504 bytes


Concurrency Level:     30
Time taken for tests:   0.157 seconds
Complete requests:     1000
Failed requests:        0
Write errors:           0
Total transferred:     40858000 bytes
HTML transferred:      40504000 bytes
Requests per second:   6365.98 [#/sec] (mean)
Time per request:       4.713 [ms] (mean)
Time per request:       0.157 [ms] (mean, across all concurrent requests)
Transfer rate:          254005.10 [Kbytes/sec] received


Connection Times (ms)
              min    mean[+/-sd] median    max
Connect:        0      1   1.1      0      8
Processing:      2      4   1.8      3     15
Waiting:         1      3   1.6      3     13
Total:           2      5   2.8      4     23


Percentage of the requests served within a certain time (ms)
 50%    4
 66%    4
 75%    4
 80%    5
 90%    6
 95%    9
 98%   17
 99%   20
100%   23 (longest request)
ppemg@UbuntuServer:~$
```

Y ahora lo hacemos con CentOS:

```
Server Software:      Apache/2.2.15
Server Hostname:      localhost
Server Port:          80

Document Path:        /sin_imagenes.html
Document Length:      294 bytes

Concurrency Level:    30
Time taken for tests:  0.256 seconds
Complete requests:    1000
Failed requests:       0
Write errors:          0
Non-2xx responses:    1002
Total transferred:    474948 bytes
HTML transferred:     294588 bytes
Requests per second:  3905.61 [#/sec] (mean)
Time per request:     7.681 [ms] (mean)
Time per request:     0.256 [ms] (mean, across all concurrent requests)
Transfer rate:        1811.49 [Kbytes/sec] received

Connection Times (ms)
      min    mean[+/-sd] median    max
Connect:    0      1   0.6      1      4
Processing:  1      6   1.7      6     14
Waiting:    1      6   1.7      6     13
Total:      2      7   1.9      7     14

Percentage of the requests served within a certain time (ms)
 50%      7
 66%      8
 75%      8
 80%      8
 90%     10
 95%     11
 98%     13
 99%     14
100%     14 (longest request)
[ppemg@CentOS2 ~]$ █
```

```
Server Software:      Apache/2.2.15
Server Hostname:      localhost
Server Port:          80

Document Path:        /con_imagenes.html
Document Length:      40504 bytes

Concurrency Level:     30
Time taken for tests:  0.246 seconds
Complete requests:     1000
Failed requests:        0
Write errors:           0
Total transferred:     40890504 bytes
HTML transferred:      40616600 bytes
Requests per second:   4071.30 [#/sec] (mean)
Time per request:      7.369 [ms] (mean)
Time per request:      0.246 [ms] (mean, across all concurrent requests)
Transfer rate:         162575.55 [Kbytes/sec] received
```

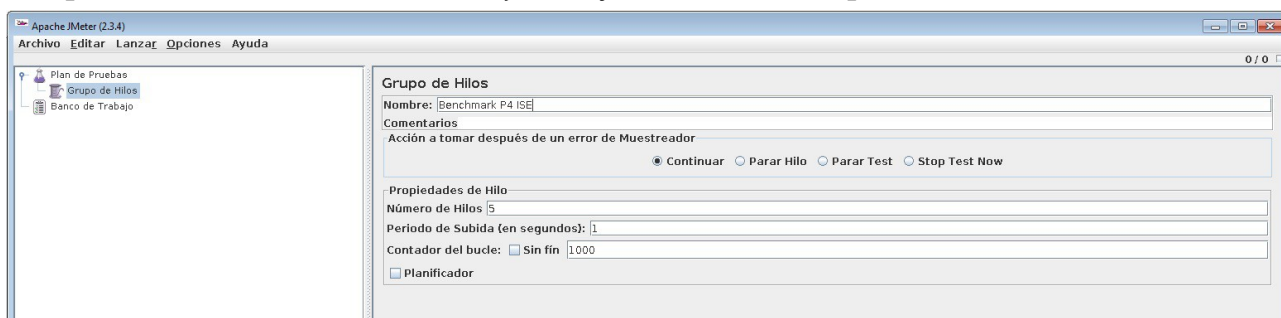
```
Connection Times (ms)
              min  mean[+/-sd] median  max
Connect:        0    1   0.7      1    5
Processing:      4    6   1.1      6    9
Waiting:         2    4   1.1      5    7
Total:          5    7   1.1      7   12
```

Percentage of the requests served within a certain time (ms)

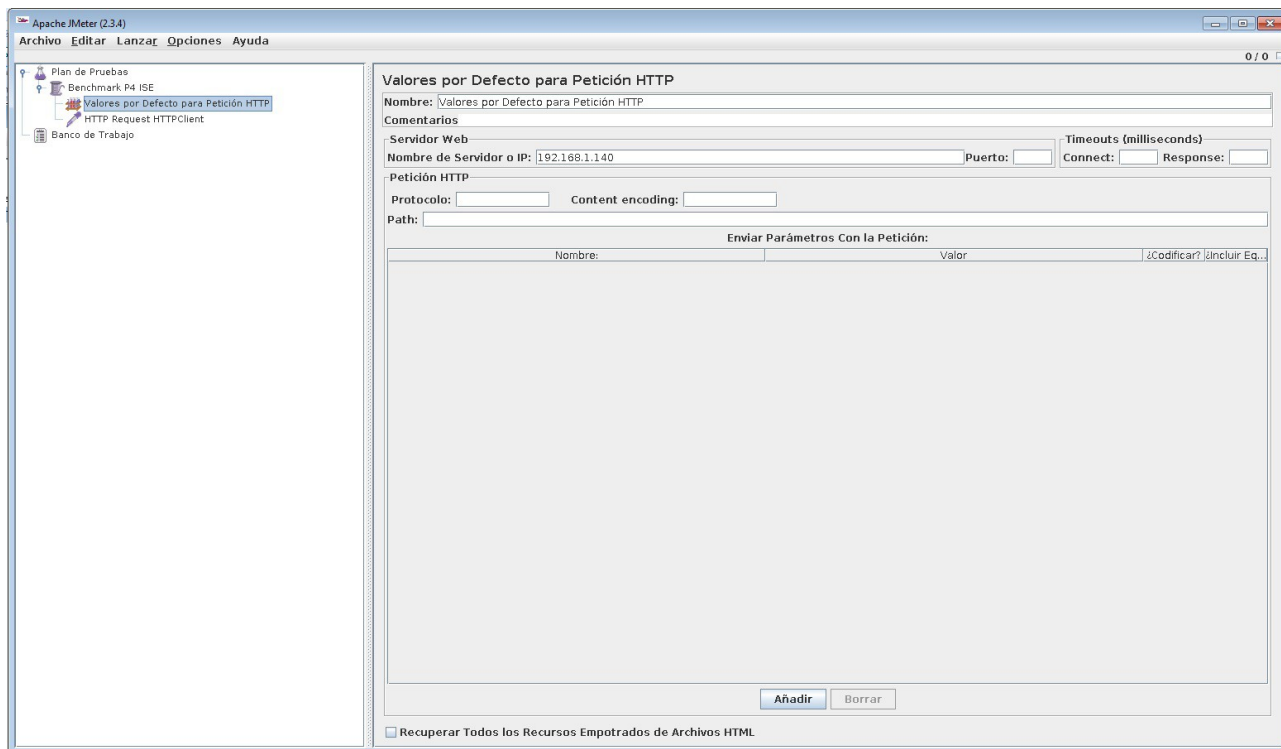
```
 50%      7
 66%      7
 75%      8
 80%      8
 90%      9
 95%      9
 98%     10
 99%     10
100%     12 (longest request)
[ppemg@CentOS2 ~]$ █
```


Cuestión 5. Instale y siga el tutorial en <http://jmeter.apache.org/usermanual/build-web-test-plan.html> realizando capturas de pantalla y comentándolas. En vez de usar la web de jmeter, haga el experimento usando sus máquinas virtuales (Puede hacer una página sencilla, usar las páginas de phpmyadmin, instalar un CMS, etc.).

Después de instalarlo, con los comandos de siempre (*apt-get install jmeter*), lo ejecutamos, hacemos click derecho en “Plan de Pruebas” y seleccionamos *Añadir >> Grupo de Hilos*. Lo modificamos y lo dejamos con estos parámetros:



Ahora creamos las opciones por defecto para las consultas HTTP. Para ello, click derecho, *Añadir >> Elementos de Configuración >> Valores por defecto para Petición HTTP*. Lo dejamos con estas opciones:



Después creamos dos consultas HTTP. Una nos llevará al directorio raíz, otra a una página con imágenes. Para añadir las consultas sería pulsar *Añadir >> Muestreador >> HTTP Request HTTP client*. Añadimos dos y dejamos estos parámetros:

The screenshot shows the Apache JMeter 2.3.4 interface. The left sidebar displays a test plan with a sampler named 'HTTP Request HTTPClient'. The main panel is titled 'HTTP Request HTTPClient' and contains the following configuration fields:

- Nombre:** HTTP Request HTTPClient
- Comentarios:** (empty)
- Servidor Web:**
 - Nombre de Servidor o IP: (empty)
 - Puerto: (empty)
 - Timeouts (milliseconds): Connect: (empty), Response: (empty)
- Petición HTTP:**
 - Protocolo: (empty)
 - Método: GET
 - Content encoding: (empty)
 - Path: /con_imagenes.html
 - ☒ Redirigir Automáticamente
 - ☐ Seguir Redirecciones
 - ☒ Utilizar KeepAlive
 - ☐ Use multipart/form-data for HTTP POST
- Enviar Parámetros Con la Petición:**

Nombre:	Valor	¿Codificar?	¿Incluir Eq...
---------	-------	-------------	----------------

Añadir Borrar
- Enviar un archivo Con la Petición:**

Nombre de Archivo:	Nombre d...	Tipo MIME:
--------------------	-------------	------------

Añadir Navegar... Borrar
- Tareas Opcionales:**
 - ☐ Recuperar Todos los Recursos Empotrados de Archivos HTML
 - ☐ Utilizar como Monitor
 - ☐ Save response as MD5 hash?
- Embedded URLs must match:** (empty)

The screenshot shows the Apache JMeter 2.3.4 interface. The left sidebar displays a test plan with a sampler named 'Cambios'. The main panel is titled 'Petición HTTP' and contains the following configuration fields:

- Nombre:** Cambios
- Comentarios:** (empty)
- Servidor Web:**
 - Nombre de Servidor o IP: (empty)
 - Puerto: (empty)
 - Timeouts (milliseconds): Connect: (empty), Response: (empty)
- Petición HTTP:**
 - Protocolo: (empty)
 - Método: GET
 - Content encoding: (empty)
 - Path: /con_imagenes.html
 - ☒ Redirigir Automáticamente
 - ☐ Seguir Redirecciones
 - ☒ Utilizar KeepAlive
 - ☐ Use multipart/form-data for HTTP POST
- Enviar Parámetros Con la Petición:**

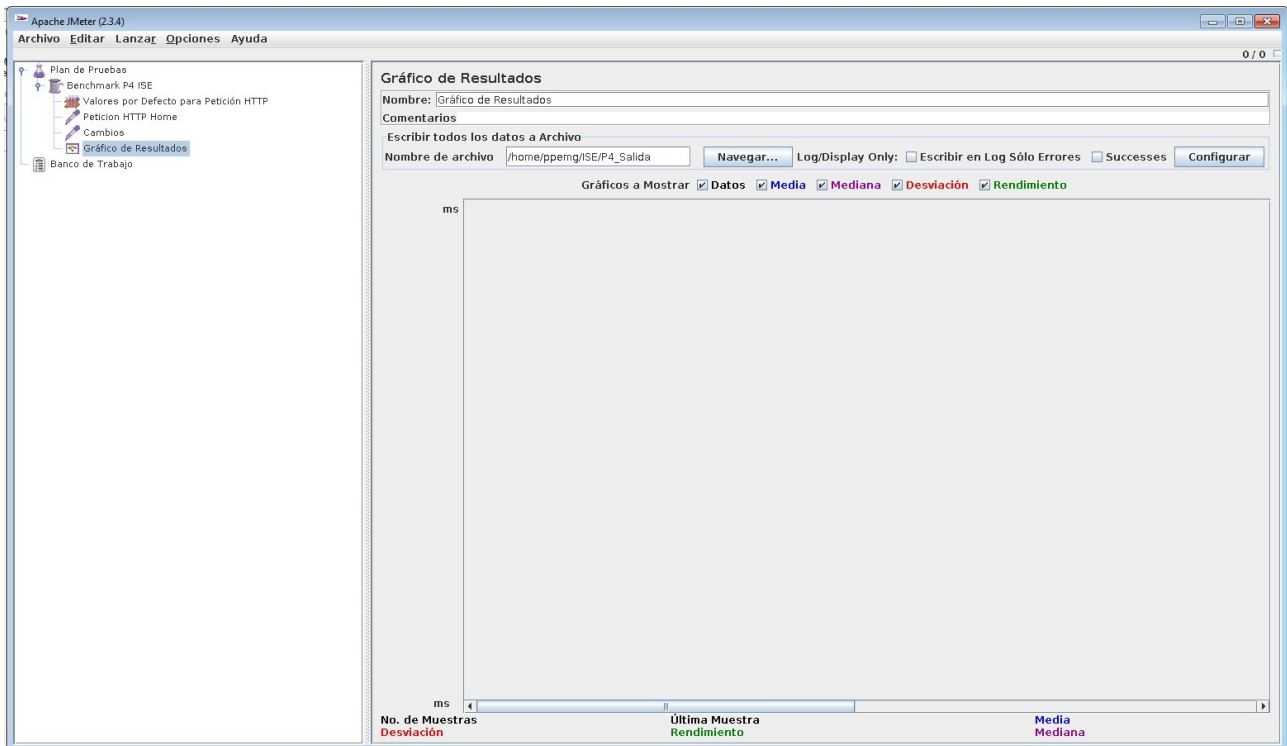
Nombre:	Valor	¿Codificar?	¿Incluir Eq...
---------	-------	-------------	----------------

Añadir Borrar
- Enviar un archivo Con la Petición:**

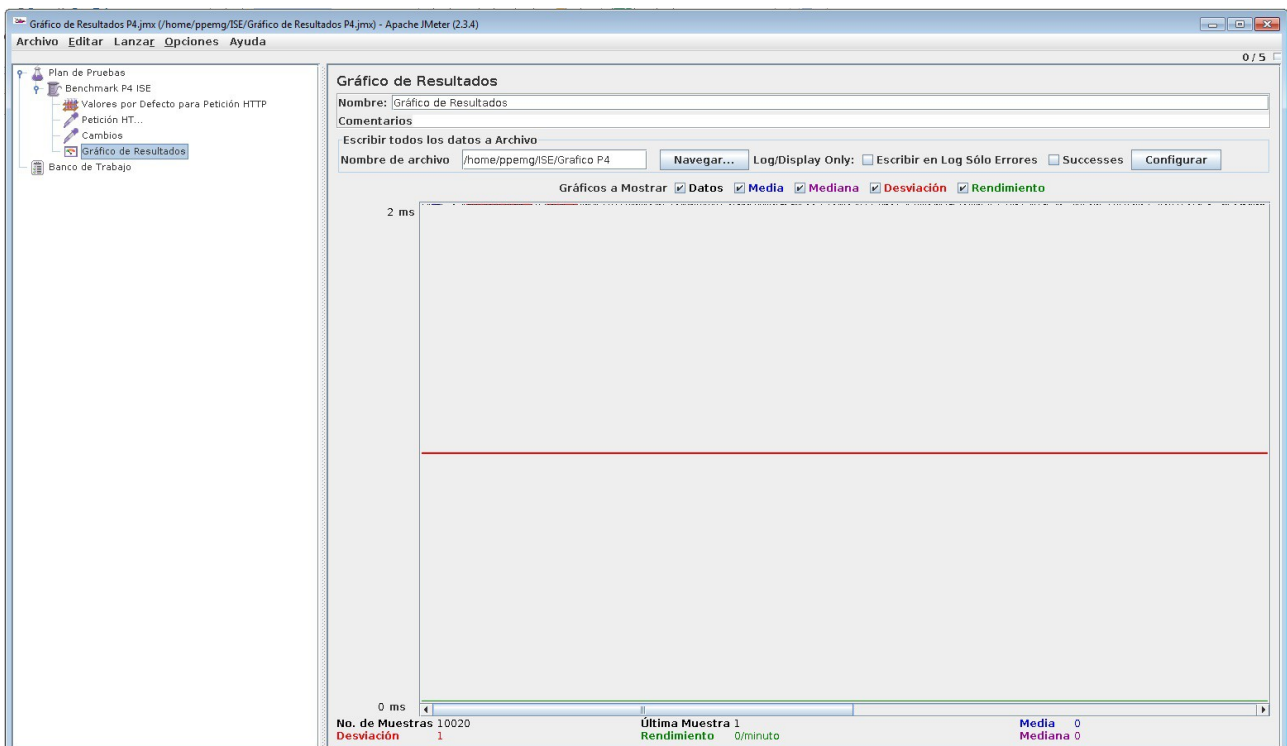
Nombre de Archivo:	Nombre d...	Tipo MIME:
--------------------	-------------	------------

Añadir Navegar... Borrar
- Tareas Opcionales:**
 - ☐ Recuperar Todos los Recursos Empotrados de Archivos HTML
 - ☐ Utilizar como Monitor
 - ☐ Save response as MD5 hash?
- Embedded URLs must match:** (empty)

Por último, solo nos queda añadir el gráfico. *Añadir >> Listener >> Gráfico de Resultados*. Una vez creado, lo guardamos en donde queramos.



Por último, ejecutamos (*Lanzar >> Arrancar*) y se generará un gráfico con los resultados:



Cómo vemos, sale un resultado muy estable, lo cual se debe probablemente a la ejecución del benchmark (y las solicitudes que genera) desde el mismo equipo que el servidor, con lo cual la latencia de la red es mínima, y no se generan pesadas transmisiones.

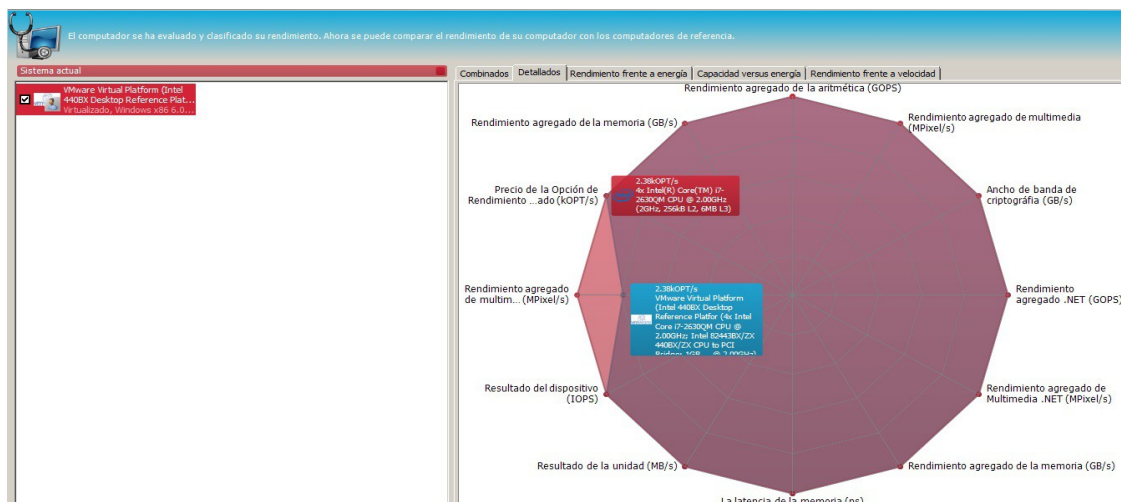
Cuestión 6. Instale la aplicación y realice varios tests a la máquina virtual cambiando los parámetros (número de núcleos y RAM). Muestre capturas de pantalla

Hemos realizado dos tests, con dos configuraciones distintas:

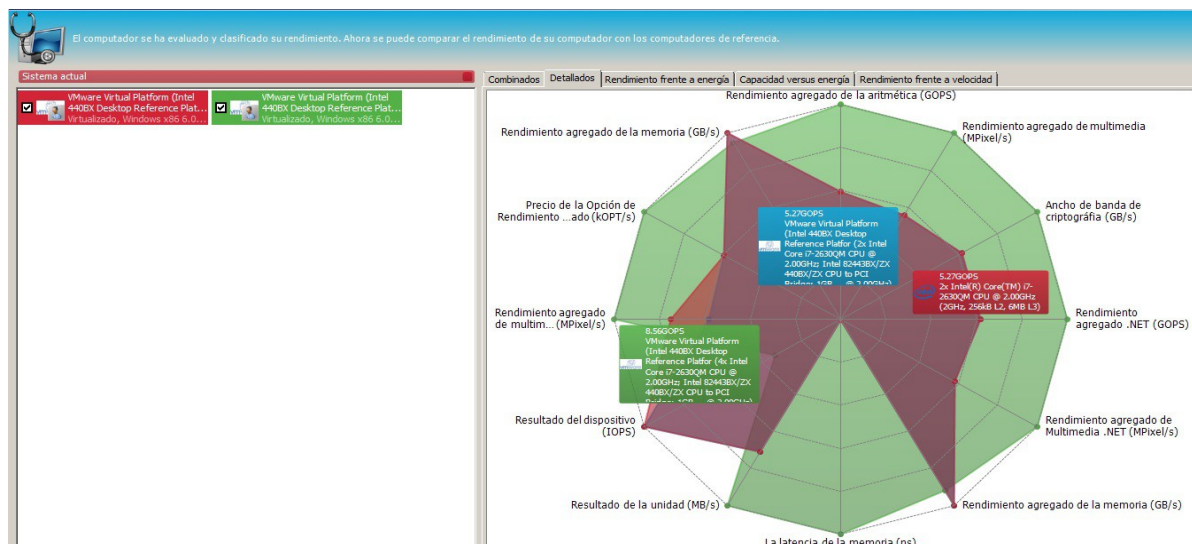
1ª 4 CPU – 1 GB de RAM

2ª 2 CPU – 2 GB de RAM

Realizamos los dos test, y obtenemos dos gráficos:



El gráfico generado tras el primer análisis. Después, cambiamos las opciones de la máquina virtual para establecer la segunda configuración y volvemos a pasar el benchmark. Hecho esto, obtenemos:



Añado el resultado resumido de los test al final, a modo de apéndice, para no alargar en exceso la lectura.

Igual que en la actividad anterior hemos realizado los tests, con dos configuraciones distintas, solo que esta vez hemos realizado varios individuales:

2ª 2 CPU – 2 GB de RAM

[illegible]

AIDA64 Cache & Memory Benchmark

	Read	Write	Copy	Latency
Memory	14249 MB/s	TRIAL VERSION	TRIAL VERSION	75.7 ns
L1 Cache	TRIAL VERSION	438.05 GB/s	320.13 GB/s	1.6 ns
L2 Cache	138.71 GB/s	91013 MB/s	117.98 GB/s	TRIAL VERSION
L3 Cache	15854 MB/s	13759 MB/s	12310 MB/s	TRIAL VERSION
L4 Cache				

CPU Type: 4x Intel Core i7 (Sandy Bridge-MB)
 CPU Clock: 1995.4 MHz (original: [TRIAL VERSION] MHz)
 CPU FSB:
 NB Clock:
 CPU Multiplier:
 CPU Stepping: D2

Memory Bus:
 Memory Type:
 Chipset: Intel 82440BX/ZX
 Motherboard: [TRIAL VERSION]

AIDA64 v4.00.2700 / BenchDLL 4.1.591-x32 (c) 1995-2013 FinalWire Ltd.

Save Start Benchmark Close

AIDA64 CPUID

Processor: Intel [TRIAL VERSION]
 Code Name: Sandy Bridge-MB
 Platform:
 Stepping: D2
 CPUID Vendor: GenuineIntel 32 nm
 CPUID Name: Intel(R) Core(TM) i7-2630QM CPU @ 2.00GHz

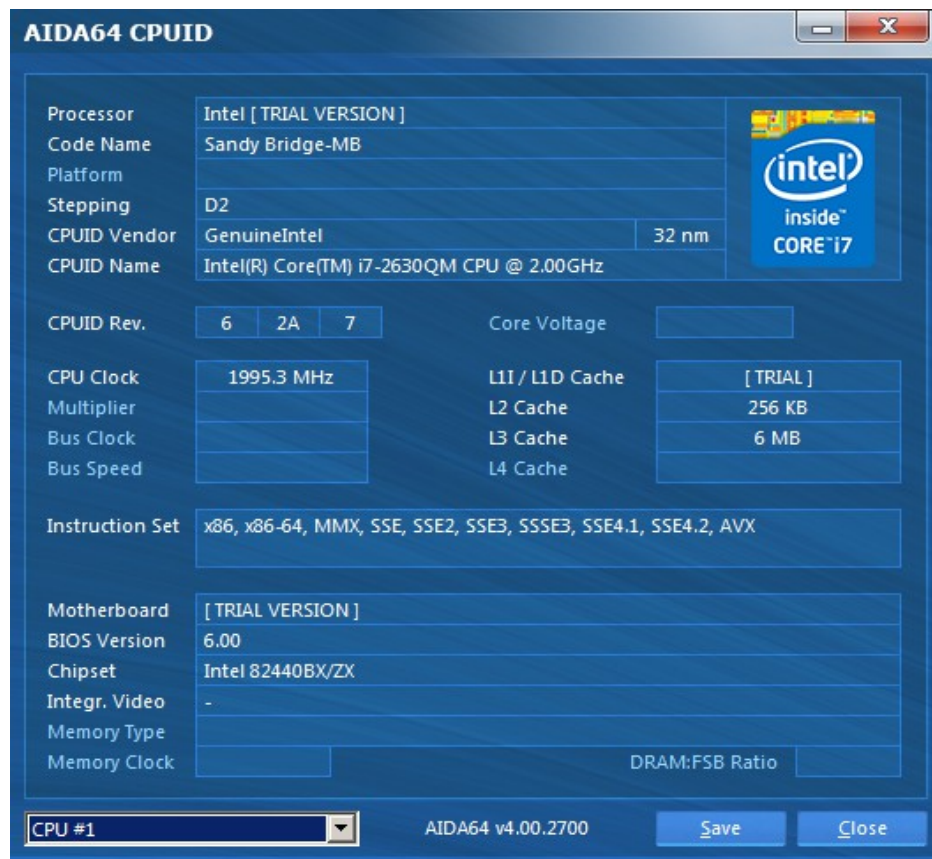
CPUID Rev. 6 2A 7 Core Voltage:
 CPU Clock: 1995.1 MHz L1I / L1D Cache: [TRIAL]
 Multiplier: L2 Cache: 256 KB
 Bus Clock: L3 Cache: 6 MB
 Bus Speed: L4 Cache:
 Instruction Set: x86, x86-64, MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, AVX

Motherboard: [TRIAL VERSION]
 BIOS Version: 6.00
 Chipset: Intel 82440BX/ZX
 Integr. Video: -
 Memory Type:
 Memory Clock:
 DRAM:FSB Ratio:
 CPU #1

AIDA64 v4.00.2700 Save Close

[illegible]

- 14 -



En el caso de los discos, se nota claramente que se utiliza mucho más la CPU cuando solo tenemos dos procesadores, si bien la diferencia en velocidad no es muy significativa.

En cuanto a la memoria, las diferencias no son muy significativas y, en general, reducen un poco la velocidad de lectura cuando hay más memoria, a cambio de ganar más capacidad.

Cuestión 7: Resultado detallado de los tests (eliminando información repetida):

Configuración 1:

SiSoftware Sandra

CPU : 4x Intel(R) Core(TM) i7-2630QM CPU @ 2.00GHz (2GHz, 256kB L2, 6MB L3)
Velocidad : 1995MHz
Capacidad : 4Unidad(es)
Energía : 223.53W

Memoria : Intel 82443BX/ZX 440BX/ZX CPU to PCI Bridge; 1GB EDO (x2-2-2)
Velocidad : 65537MHz
Capacidad : 1024MB
Energía : 3.00W

Aritmética del procesador
Resultado combinado : 32.23GOPS

Multimedia del procesador
Resultado combinado : 45.73MPixel/s

Criptografía

Resultado combinado : 0.289GB/s

.NET aritmético
Resultado combinado : 8.56GOPS

.NET Multimedia
Resultado combinado : 7.92MPixel/s

Ancho de banda de memoria
Resultado combinado : 12.539GB/s

La latencia de la memoria
Resultado combinado : 6824.2ns

Multimedia del procesador
Resultado combinado : 43.80MPixel/s

Análisis Financiero
Resultado combinado : 2.38kOPT/s

Ancho de banda de memoria
Resultado combinado : 12.818GB/s

Puntuaje general

Resultado combinado : 1.08kPT

Interpretación de los resultados : Los valores de índices más altos son mejores.

Multiplicadores de resultados en sistema decimal (base 10) : 1GPT = 1000MPT, 1MPT = 1000kPT, 1kPT = 1000PT, etc.

Identidad de resultado : VMware Virtual Platform (Intel 440BX Desktop Reference Platfor
(4x Intel Core i7-2630QM CPU @ 2.00GHz; Intel 82443BX/ZX 440BX/ZX CPU to PCI Bridge;
1GB EDO; VMware, VMware Virtual S; 4x Intel Core i7-2630QM CPU @ 2.00GHz)

Finalizado exitosamente : Sí

Configuración 2:

SiSoftware Sandra

CPU : 2x Intel(R) Core(TM) i7-2630QM CPU @ 2.00GHz (2GHz, 256kB L2, 6MB L3)

Velocidad : 1995MHz

Capacidad : 2Unidad(es)

Energía : 111.76W

Memoria : Intel 82443BX/ZX 440BX/ZX CPU to PCI Bridge; 1GB EDO (x2-2-2)

Velocidad : 65537MHz

Capacidad : 1024MB

Energía : 3.00W

Aritmética del procesador

Resultado combinado : 19.08GOPS

Multimedia del procesador

Resultado combinado : 25.64MPixel/s

Criptografía

Resultado combinado : 0.179GB/s

.NET aritmético

Resultado combinado : 5.27GOPS

.NET Multimedia

Resultado combinado : 4.61MPixel/s

Ancho de banda de memoria

Resultado combinado : 13.603GB/s

La latencia de la memoria

Resultado combinado : 29.6ns

Multimedia del procesador

Resultado combinado : 25.75MPixel/s

Análisis Financiero

Resultado combinado : 1.42kOPT/s

Ancho de banda de memoria

Resultado combinado : 13.501GB/s

Puntuaje general

Resultado combinado : 1.35kPT

Interpretación de los resultados : Los valores de índices más altos son mejores.

Multiplicadores de resultados en sistema decimal (base 10) : 1GPT = 1000MPT, 1MPT = 1000kPT, 1kPT = 1000PT, etc.

Identidad de resultado : VMware Virtual Platform (Intel 440BX Desktop Reference Platform (2x Intel Core i7-2630QM CPU @ 2.00GHz; Intel 82443BX/ZX 440BX/ZX CPU to PCI Bridge; 1GB EDO; VMware, VMware Virtual S; 2x Intel Core i7-2630QM CPU @ 2.00GHz)

Finalizado exitosamente : Sí