

Ingeniería de Servidores

Práctica 4: Benchmarks

Cuestiones de la Práctica 4

ADRIÁN PORTILLO SÁNCHEZ
Universidad de Granada
23 de diciembre de 2015

Índice

Cuestión 1	4
Cuestión Opcional 1.	6
Cuestión 2.	7
Cuestión 3.	7
Cuestión 4.	11
Cuestión opcional 4.	14
Cuestión 5.	16

Índice de figuras

1.	Instalando Phoronix Suite en Ubuntu Server.	4
2.	Instalando Phoronix Suite en CentOS.	5
3.	Lista de Benchmarks de Phoronix.	5
4.	Resultados benchmark apache.	6
5.	Resultados del Benchmark de Apache en CentOS.	8
6.	Resultados del Benchmark de Apache en Ubuntu Server.	9
7.	Resultados del Benchmark de IIS en Windows Server.	10
8.	Tabla comparativa de resultados del Benchmark Apache.	10
9.	Creando un grupo de hilos.	11
10.	Cambiando configuración por defecto de petición.	11
11.	Añadimos la petición HTTP.	12
12.	Resultado de la configuración realizada.	12
13.	Gráfica de resultados de la petición.	13
14.	Estado final de los discos de sistema.	14
15.	Benchmark de Escritura de memoria.	15
16.	Benchmark de lectura, escritura de la memoria y caches.	15
17.	Gráfica de Tiempo de escritura.	17
18.	Gráfica de Velocidad de escritura.	18
19.	Gráfica de Tiempo de lectura.	19
20.	Gráfica de Velocidad de lectura.	19
21.	Gráfica de Tiempo de escritura.	20
22.	Gráfica de Velocidad de escritura.	21
23.	Gráfica de Tiempo de lectura.	22
24.	Gráfica de Velocidad de lectura.	22
25.	Diagrama de Flujo del programa que ejecutará la prueba de rendimiento.	23

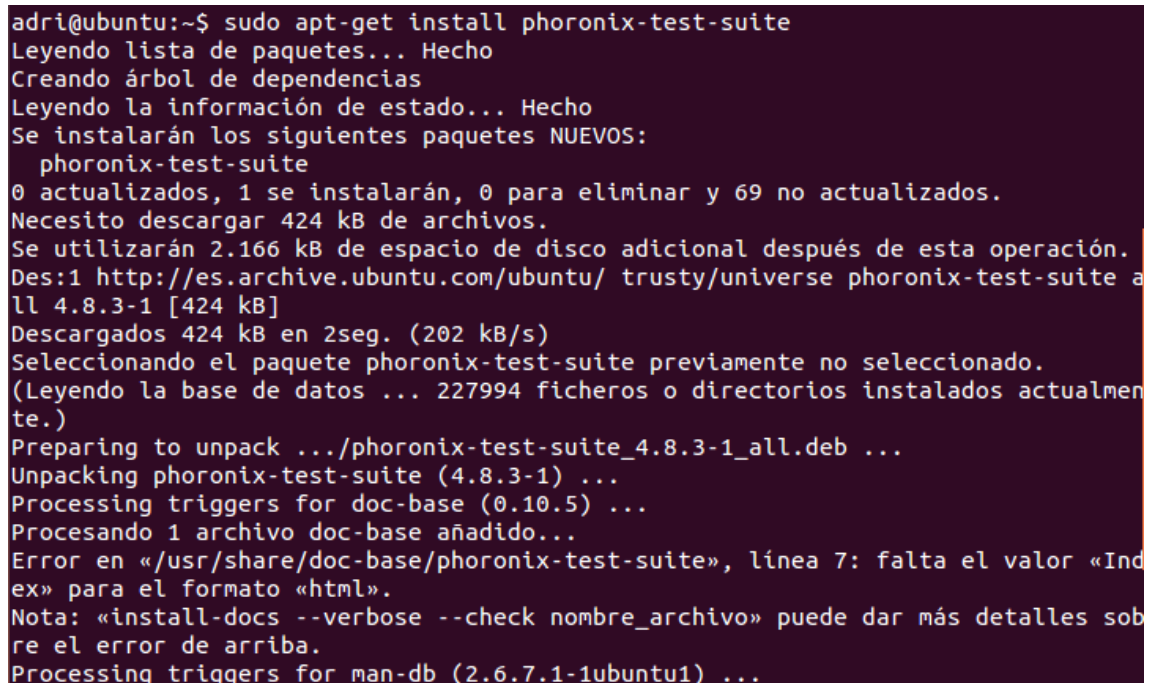
Cuestión 1

Instale Phoronix Suite, seleccione un benchmark de entre los disponibles, descárguelo y ejecútelo. Describa el propósito del benchmark y su interés en el mismo. Explique razonadamente el significado de los resultados obtenidos.

Lo primero que deberemos hacer es instalar la aplicación “Phoronix Test Suite” [1], tanto en Ubuntu Server como en CentOS podremos instalarla directamente con el gestor de paquetes que usemos.

En el caso de Ubuntu Server deberemos introducir

```
# sudo apt-get install phoronix-test-suite
```



```
adri@ubuntu:~$ sudo apt-get install phoronix-test-suite
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  phoronix-test-suite
0 actualizados, 1 se instalarán, 0 para eliminar y 69 no actualizados.
Necesito descargar 424 kB de archivos.
Se utilizarán 2.166 kB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu/ trusty/universe phoronix-test-suite a
ll 4.8.3-1 [424 kB]
Descargados 424 kB en 2seg. (202 kB/s)
Seleccionando el paquete phoronix-test-suite previamente no seleccionado.
(Leyendo la base de datos ... 227994 ficheros o directorios instalados actualmen
te.)
Preparing to unpack .../phoronix-test-suite_4.8.3-1_all.deb ...
Unpacking phoronix-test-suite (4.8.3-1) ...
Processing triggers for doc-base (0.10.5) ...
Procesando 1 archivo doc-base añadido...
Error en «/usr/share/doc-base/phoronix-test-suite», línea 7: falta el valor «Ind
ex» para el formato «html».
Nota: «install-docs --verbose --check nombre_archivo» puede dar más detalles sob
re el error de arriba.
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
```

Figura 1: Instalando Phoronix Suite en Ubuntu Server.

En el caso de CentOS en primer lugar descargamos Phoronix Suite de la web oficial <http://www.phoronix-test-suite.com/?k=downloads> y seleccionaremos el paquete genérico que posteriormente descomprimiremos y ejecutaremos desde terminal el archivo **'installsh'**. Una vez se ha realizado esto podremos ejecutar Phoronix si tenemos las dependencias de php necesarias según la web solo es necesario el paquete *'php5-cli'*, sin embargo puede ser que se tengan que instalar las dependencias una a una dependiendo de la que falte como es mi caso, faltaba DOM y fue necesaria la ejecución de:

```
# sudo yum install php-xml
```

```
[adri@localhost phoronix-test-suite]$ ./install-sh
ERROR: /usr is not writable. Run this installer as root or specify a different d
irectory prefix as the first argument sent to this script.
[adri@localhost phoronix-test-suite]$ sudo ./install-sh

Phoronix Test Suite Installation Completed

Executable File: /usr/bin/phoronix-test-suite
Documentation: /usr/share/doc/phoronix-test-suite/
Phoronix Test Suite Files: /usr/share/phoronix-test-suite/

[adri@localhost phoronix-test-suite]$ phoronix-test-suite

The following PHP extensions are REQUIRED by the Phoronix Test Suite:

DOM          The PHP Document Object Model (DOM) is required for XML operations.

The following PHP extensions are OPTIONAL but recommended:

GD           The PHP GD library is recommended for improved graph rendering.
POSIX        PHP POSIX support is highly recommended.
```

Figura 2: Instalando Phoronix Suite en CentOS.

Una vez instaladas las dependencias ejecutamos phoronix con el comando *'phoronix-test-suite'* y completamos algunas opciones de configuración[3]. Antes de poder realizar los test, debemos conocer qué benchmarks están disponibles, por lo que necesitamos una lista con todos los test disponibles. Para listar los benchmarks disponibles introducimos en el terminal:

```
# phoronix-test-suite list-tests
```

```
[adri@localhost phoronix-test-suite]$ phoronix-test-suite list-tests

Phoronix Test Suite v6.0.1
Available Tests

pts/aio-stress          - AIO-Stress          Disk
pts/apache              - Apache Benchmark    System
pts/apitest             - APITest             Graphics
pts/apitrace            - APITrace            Graphics
pts/askap               - ASKAP tConvolv CUDA Graphics
pts/battery-power-usage - Battery Power Usage System
pts/bioshock-infinite   - BioShock Infinite   Graphics
pts/blake2              - BLAKE2              Processor
pts/blogbench           - BlogBench           Disk
pts/bork                - Bork File Encrypter Processor
pts/botan               - Botan               Processor
pts/build-apache        - Timed Apache Compilation Processor
pts/build-firefox       - Timed Firefox Compilation Processor
pts/build-imagemagick   - Timed ImageMagick Compilation Processor
pts/build-linux-kernel - Timed Linux Kernel Compilation Processor
pts/build-mplayer       - Timed MPlayer Compilation Processor
pts/build-php           - Timed PHP Compilation Processor
pts/build-webkitgtk     - Timed WebKitGTK Compilation Processor
pts/bullet              - Bullet Physics Engine Processor
```

Figura 3: Lista de Benchmarks de Phoronix.


Cuestión Opcional 1.

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

Viendo la lista de benchmarks decidimos el benchmark que vamos a utilizar[4], en nuestro caso usaremos *'pts/apache'*, por lo que es necesario aplicar el siguiente comando: *'phoronix-test-suite install pts/apache'* y comenzara a instalar el benchmark, por último para usarlo solo es necesario el comando *'phoronix-test-suite benchmark pts/apache'*

En los resultados obtenemos que se han realizado 4637.51 peticiones por segundo (figura 4), además se muestra el error normal de 72.93 y la desviación típica 3.15 %; Se usan peticiones por segundo debido a que el benchmark de apache consiste en realizar peticiones al servidor HTTP GET (concurrentemente también) para ver cuanto es el tiempo de respuesta, esto permite realizar una comparación de servidores apache, en un futuro incluso se podría realizar un benchmark con distintos protocolos web como HTTP2 y ver que protocolo funciona mejor con apache.

Results Overview

results	
 Phoronix Test Suite	result
Apache Benchmark	4637.51
Standard Error	72.93
Standard Deviation	3.15%
PHORONIX-TEST-SUITE.COM	

Test Results

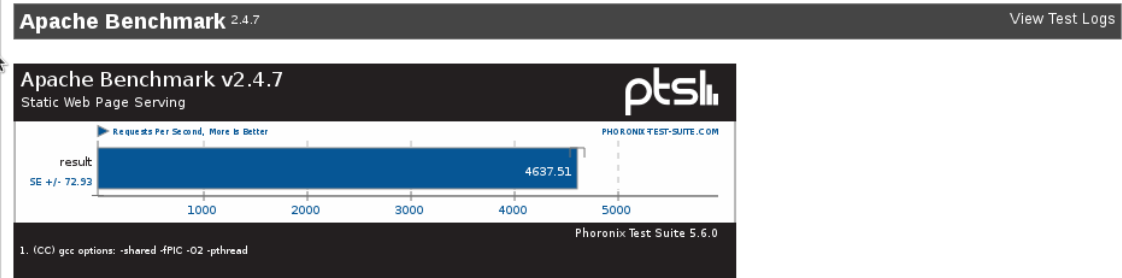


Figura 4: Resultados benchmark apache.

Cuestión 2.

De los parámetros que le podemos pasar al comando ¿Qué significa -c 5? ¿y -n 100? Monitore la ejecución de ab contra alguna máquina (cualquiera) ¿cuántos procesos o hebras crea ab en el cliente?

El parámetro '-c' indica la concurrencia[5], pasar como parámetro '-c 30', indicará que vamos ejecutar concurrentemente 30 solicitudes a la vez.

-c concurrency
Number of multiple requests to perform at a time. Default is one request at a time.
Número de solicitudes a realizar al mismo tiempo. Por defecto se realiza una a cada vez.

El parámetro '-n' indica el número de solicitudes, usar '-n 1000' significará que le vamos a hacer 1000 solicitudes al servidor.

-n requests
Number of requests to perform for the benchmarking session. The default is to just perform a single request which usually leads to non-representative benchmarking results.
Número de solicitudes a realizar para la sesión de benchmarking. Por defecto se realiza una sola solicitud lo que suele llevar a resultados no representativos.

Cuestión 3.

Ejecute ab contra a las tres máquinas virtuales (desde el SO anfitrión a las máquinas virtuales de la red local) una a una (arrancadas por separado) y muestre y comente las estadísticas. ¿Cuál es la que proporciona mejores resultados? Fíjese en el número de bytes transferidos, ¿es igual para cada máquina?

Para hacer las pruebas a todos los servidores de las máquinas virtuales en igual de condiciones, ejecutaremos ab desde nuestro sistema operativo anfitrión[5], como en nuestro caso es Windows 7, vamos a descargarnos XAMPP, que ya contiene los ejecutables de Apache y podremos usarlos sin tener que hacer ninguna instalación, lo descargaremos desde <http://www.apachefriends.org/download.php?xampp-portable-lite-win32-1.8.1-VC9.zip> y podremos ejecutar directamente Apache Benchmark, que se encuentra dentro de la carpeta 'apache/bin'.

Hechas las pruebas a los servidores de las 3 máquinas virtuales los resultados son los siguientes:

- Benchmark del Apache en CentOS.

```
PS C:\xampp\apache\bin> ./ab -k -c 30 -n 1000 -g centos.txt http://192.168.147.136/
This is ApacheBench, Version 2.3 <$Revision: 1706008 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.147.137 (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/2.4.7
Server Hostname:      192.168.147.137
Server Port:          80

Document Path:        /
Document Length:      11510 bytes

Concurrency Level:    30
Time taken for tests:  4.094 seconds
Complete requests:    1000
Failed requests:       0
Keep-Alive requests:  996
Total transferred:    11818839 bytes
HTML transferred:     11510000 bytes
Requests per second:  244.27 [#/sec] (mean)
Time per request:     122.815 [ms] (mean)
Time per request:     4.094 [ms] (mean, across all concurrent requests)
Transfer rate:        2819.33 [Kbytes/sec] received

Connection Times (ms)
              min    mean[+/-sd] median    max
Connect:        0      0   0.5      0      16
Processing:      0  48 375.3      0  4078
Waiting:        0  48 375.3      0  4078
Total:          0  48 375.3      0  4078

Percentage of the requests served within a certain time (ms)
 50%    0
 66%    0
 75%    0
 80%    0
 90%    0
 95%   16
 98%   94
 99%  3062
100% 4078 (longest request)
```

Figura 5: Resultados del Benchmark de Apache en CentOS.

- Benchmark del Apache en Ubuntu Server.

```

PS C:\wamp\apache\bin> ./ab -k -c 30 -n 1000 -q ubuntu.txt http://192.168.147.137/
This is ApacheBench, Version 2.3 <Revision: 1706008>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.147.137 (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/2.4.7
Server Hostname:      192.168.147.137
Server Port:          80

Document Path:        /
Document Length:       11510 bytes

Concurrency Level:     30
Time taken for tests:   8.016 seconds
Complete requests:      1000
Failed requests:         9
  (Connect: 0, Receive: 0, Length: 9, Exceptions: 0)
Keep-Alive requests:    985
Total transferred:      11712385 bytes
HTML transferred:       11406410 bytes
Requests per second:    124.76 [#/sec] (mean)
Time per request:       240.466 [ms] (mean)
Time per request:       8.016 [ms] (mean, across all concurrent requests)
Transfer rate:          1426.97 [kbytes/sec] received

Connection Times (ms)
  min   mean[+/-sd] median   max
Connect:    0      0  1.0      0    16
Processing:  0   117 767.4      0  6969
Waiting:    0    78 620.5      0  6969
Total:       0   117 767.5      0  6969

Percentage of the requests served within a certain time (ms)
 50%    0
 66%    0
 75%    0
 80%    0
 90%    0
 95%   16
 98%  4953
 99%  5094
100%  6969 (longest request)

```

Figura 6: Resultados del Benchmark de Apache en Ubuntu Server.

- Benchmark del IIS en Windows Server.

```

PS C:\xampp\apache\bin> ./ab -k -c 30 -n 1000 -q windows.txt http://192.168.147.131/
This is ApacheBench, Version 2.3 <$Revision: 1706008 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.147.131 (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:      Microsoft-IIS/8.5
Server Hostname:      192.168.147.131
Server Port:          80

Document Path:        /
Document Length:      701 bytes

Concurrency Level:    30
Time taken for tests:  0.844 seconds
Complete requests:    1000
Failed requests:       0
Keep-Alive requests:  1000
Total transferred:    948000 bytes
HTML transferred:     701000 bytes
Requests per second:  1185.16 [#/sec] (mean)
Time per request:      25.313 [ms] (mean)
Time per request:      0.844 [ms] (mean, across all concurrent requests)
Transfer rate:         1097.20 [Kbytes/sec] received

Connection Times (ms)
  min   mean[+/-sd] median   max
Connect:    0    0  0.0      0    0
Processing:  0  24 129.3     0   781
Waiting:    0  24 129.3     0   781
Total:       0  24 129.3     0   781

Percentage of the requests served within a certain time (ms)
 50%    0
 66%    0
 75%    0
 80%    0
 90%   15
 95%   16
 98%   757
 99%   757
100%   781 (longest request)

```

Figura 7: Resultados del Benchmark de IIS en Windows Server.

Para facilitar el saber que servidor tiene mejor rendimiento vamos a extraer los datos más significativos en una tabla:

	Tiempo total	Promedio solicitudes	Promedio tiempo	Velocidad transferencia
CentOS	4.094 s	244.27 solicitudes/s	122.815 ms/solicitud	2819.33 KB/s
Ubuntu Server	8.016 s	124.76 solicitudes/s	240.446 ms/solicitud	1426.97 KB/s
Windows Server	0.844 s	1185.16 solicitudes/s	25.313 ms/solicitud	1097.20 KB/s

Figura 8: Tabla comparativa de resultados del Benchmark Apache.

Como podemos ver las diferencias entre los 3 son grandes, tardando CentOS la mitad que Ubuntu Server, pero ambos tardando varios segundos en realizar las 1000 peticiones, mientras Windows Server tardó menos de un segundo en realizar todas las peticiones, siendo el más rápido de todos con diferencia.

Cuestión 4.

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 alguna de sus máquinas virtuales (Puede hacer una página sencilla, usar las páginas de phpmyadmin, instalar un CMS, etc.).

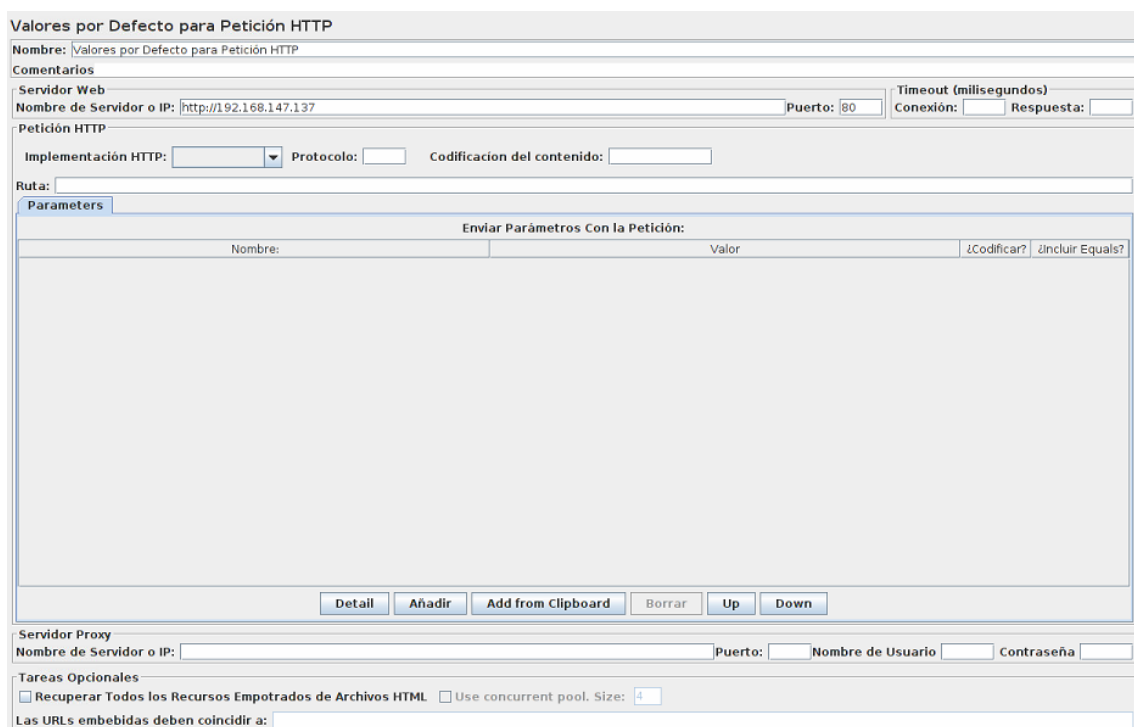
Primero creamos el grupo de hilos:



The screenshot shows the 'Thread Group' configuration window in JMeter. The 'Nombre' field is set to 'prueba phpmyadmin'. Under 'Acción a tomar después de un error de Muestreador', the 'Continuar' radio button is selected. In the 'Propiedades de Hilo' section, 'Número de Hilos' is set to 5, 'Período de Subida (en segundos)' is 1, and 'Contador del bucle' is set to 'Sin fin' with a value of 2. The 'Delay Thread creation until needed' and 'Planificador' checkboxes are unchecked.

Figura 9: Creando un grupo de hilos.

Tras esto configuramos la configuración por defecto de cada petición HTTP:



The screenshot shows the 'Values for Default HTTP Request' configuration window. The 'Nombre' field is 'Valores por Defecto para Petición HTTP'. In the 'Servidor Web' section, 'Nombre de Servidor o IP' is 'http://192.168.147.137' and 'Puerto' is 80. The 'Timeout (milisegundos)' is set to 5000. In the 'Petición HTTP' section, 'Implementación HTTP' is set to 'GET', 'Protocolo' is 'HTTP', and 'Codificación del contenido' is 'UTF-8'. The 'Ruta' field is empty. The 'Parameters' section is expanded, showing a table with columns 'Nombre', 'Valor', '¿Codificar?', and '¿Incluir Equals?'. At the bottom, there are buttons for 'Detail', 'Añadir', 'Add from Clipboard', 'Borrar', 'Up', and 'Down'. The 'Servidor Proxy' section is also visible at the bottom.

Figura 10: Cambiando configuración por defecto de petición.

Añadimos soporte de Cookies con:

Añadir --> Elemento de Configuración --> Gestor de Cookies HTTP

Añadimos la petición HTTP:

Petición HTTP

Nombre: phpmyadmin

Comentarios

Servidor Web

Nombre de Servidor o IP: Puerto: Timeout (milisegundos): Conexión: Respuesta:

Petición HTTP

Implementación HTTP: Protocolo: Método: GET Codificación del contenido:

Ruta: /phpmyadmin

☐ Redirigir Automáticamente ☒ Seguir Redirecciones ☒ Utilizar KeepAlive ☐ Usar 'multipart/form-data' para HTTP POST ☐ Cabeceras compatibles con navegadores

Parameters Post Body

Enviar Parámetros Con la Petición:

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

Detail Añadir Add from Clipboard Borrar Up Down

Enviar un archivo Con la Petición

Nombre de Archivo: Nombre de Parámetro:

Añadir Navegar... Borrar

Servidor Proxy

Nombre de Servidor o IP: Puerto: Nombre de Usuario: Contraseña:

Tareas Opcionales

☐ Recuperar Todos los Recursos Empotrados de Archivos HTML ☐ Use concurrent pool. Size: 4 ☐ Utilizar como Monitor ☐ ¿Guardar la respuesta como M...

Las URLs embebedas deben coincidir a: Dirección IP fuente:

Figura 11: Añadimos la petición HTTP.

Añadimos una gráfica de escucha con:

Añadir --> Receptor --> Gráfico de Resultados

Así queda el resultado final:

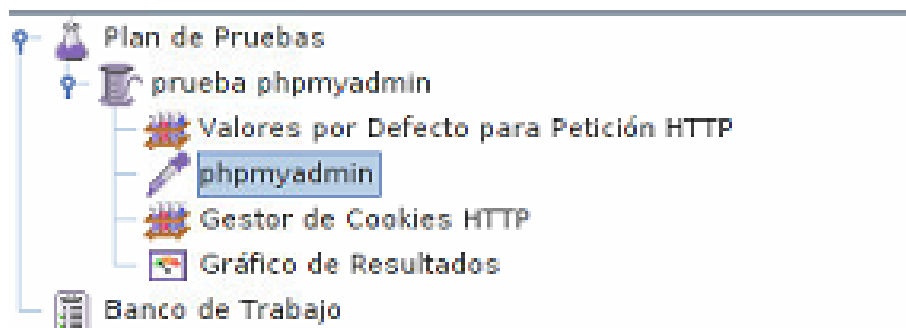


Figura 12: Resultado de la configuración realizada.

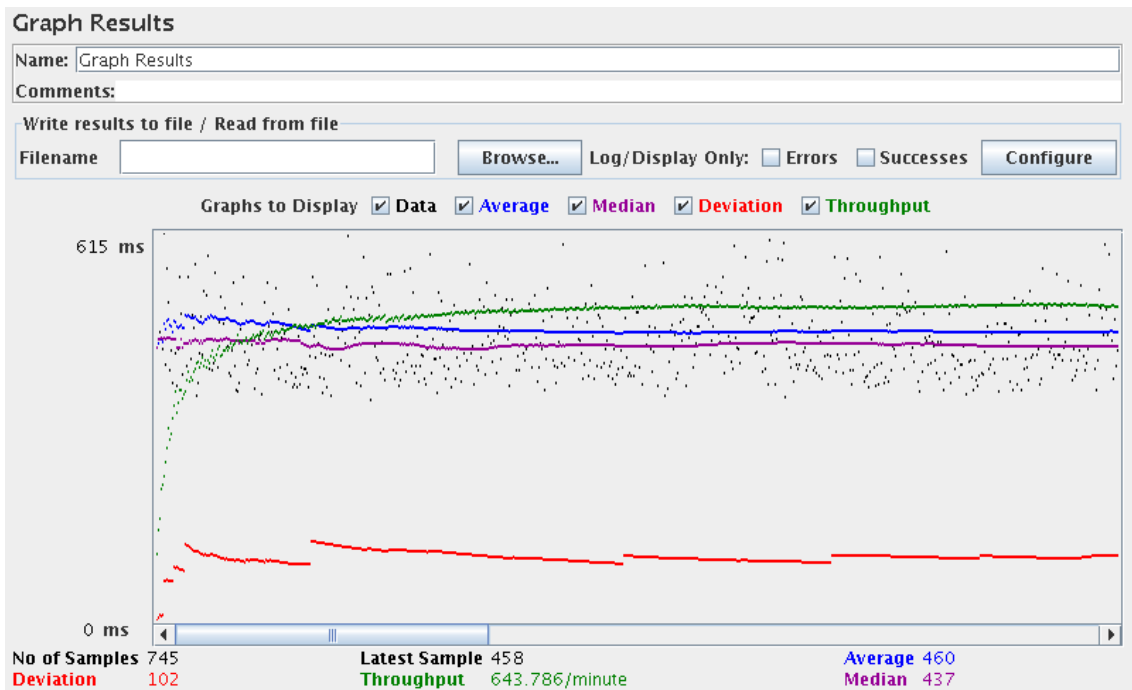


Figura 13: Gráfica de resultados de la petición.

Cuestión opcional 4.

Seleccione un benchmark entre SisoftSandra y Aida. Ejecútelo y muestre capturas de pantalla comentando los resultados.

En el caso de benchmark para windows vamos a instalar AIDA64 que se puede descargar de [6], su instalación es muy sencilla, tan solo ejecutar el archivo '.exe' que se ha descargado y se iniciara la instalación. Una vez instalado obtenemos la vista que podemos apreciar en la figura.

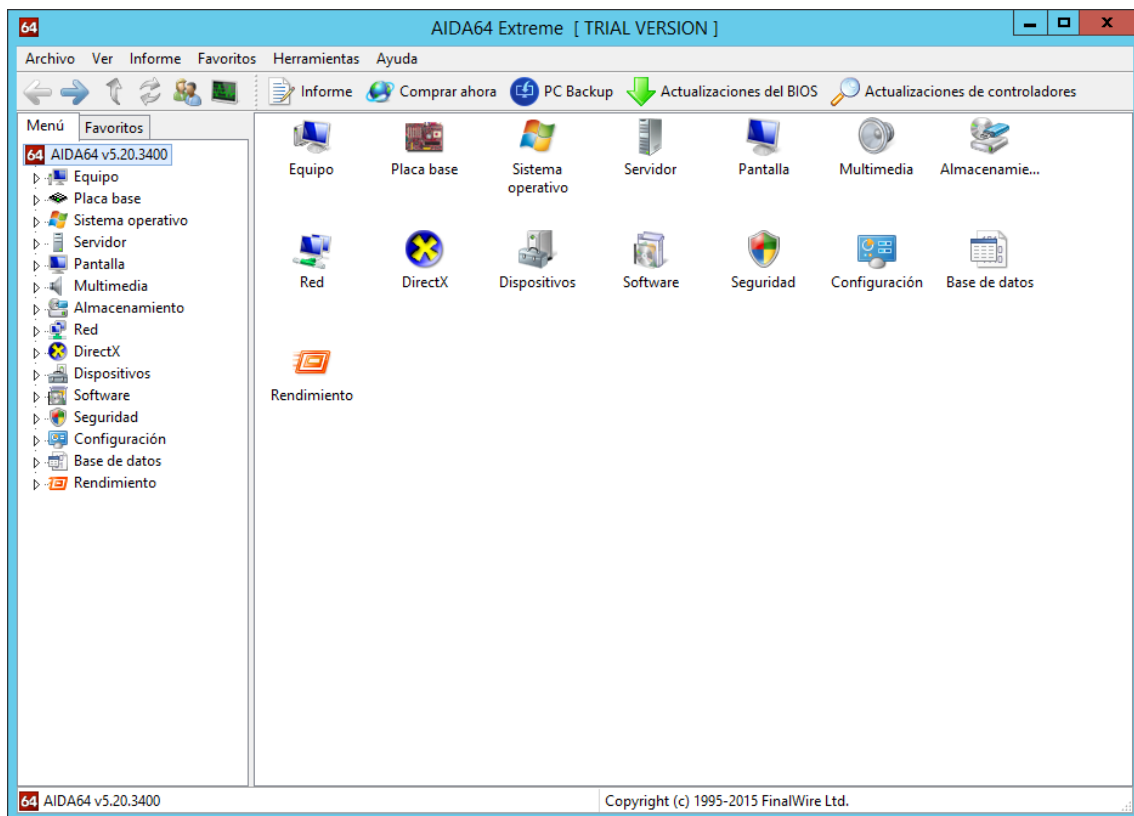


Figura 14: Estado final de los discos de sistema.

AIDA64 proporciona varios benchmark de lectura, escritura de memoria y benchmark de CPU, además tiene opciones para crear informes completos con todos estos benchmark realizados por lo que podemos pulsar en la estaña de rendimiento con click derecho presionar en informe rápido y elegir un tipo de salida de forma que se realizarán todos los benchmark, estos benchmark nos permiten comparar el rendimiento de nuestros componentes con otros componentes del mercado y hacernos una idea de lo que tenemos en nuestra máquina por ejemplo en el benchmark de velocidad de lectura se miden las velocidades de escritura en memoria de nuestra máquina para poder compararla con el resto de componentes (figura x). Además AIDA64 también nos proporciona herramientas como pruebas de rendimiento para cache y memoria (figura x+1) que calcula las velocidades de escritura, lectura de nuestra memoria y caches. (Como es una versión Trial hay datos que no se muestran).

Informe - AIDA64							
Archivo Guardar en archivo Enviar por correo electrónico Enviar a FinalWire Vista previa de impresión Imprimir Cerrar							
21404 MB/s	4x A10-5800K	3800 MHz	Asus F2A55-M	A55 Int.	Dual DDR3-1866	9-11-10-27 CR2	21404 MB/s
21284 MB/s	4x A10-6800K	4100 MHz	Gigabyte GA-F2A85X-UP4	A85X Int.	Dual DDR3-2133	9-11-10-27 CR2	21284 MB/s
21210 MB/s	4x Core i7-965 Extreme HT	3200 MHz	Asus P6T Deluxe	X58	Triple DDR3-1333	9-9-9-24 CR1	21210 MB/s
21110 MB/s	6x Core i7-990X Extreme HT	3466 MHz	Intel DX58SO2	X58	Triple DDR3-1333	9-9-9-24 CR1	21110 MB/s
21006 MB/s	4x A10-7850K	3700 MHz	Gigabyte GA-F2A88XM-D3H	A88X Int.	Dual DDR3-2133	9-11-10-31 CR2	21006 MB/s
19763 MB/s	12x Opteron 2431	2400 MHz	Supermicro H8DI3+-F	SR5690	Unganged Dual DDR2-800R	6-6-6-18 CR1	19763 MB/s
19552 MB/s	4x Core i7-2600 HT	3400 MHz	Asus P8P67	P67	Dual DDR3-1333	9-9-9-24 CR1	19552 MB/s
18834 MB/s	8x Opteron 2378	2400 MHz	Tyan Thunder n3600R	nForcePro-3600	Unganged Dual DDR2-800R	6-6-6-18 CR1	18834 MB/s
17639 MB/s	4x Xeon X3430	2400 MHz	Supermicro X8SIL-F	i3420	Dual DDR3-1333	9-9-9-24 CR1	17639 MB/s
16638 MB/s	4x A8-3850	2900 MHz	Gigabyte GA-A75M-UD2H	A75 Int.	Dual DDR3-1333	9-9-9-24 CR1	16638 MB/s
14730 MB/s	6x Phenom II X6 Black 1100T	3300 MHz	Gigabyte GA-890GPA-UD3H v2	AMD890GX Int.	Unganged Dual DDR3-1333	9-9-9-24 CR2	14730 MB/s
13453 MB/s	Celeron J1900	2000 MHz	Gigabyte GA-J1900N-D3V	BayTrailD Int.	Dual DDR3-1333	9-9-9-24 CR1	13453 MB/s
13118 MB/s	8x Opteron 2344 HE	1700 MHz	Supermicro H8DME-2	nForcePro-3600	Unganged Dual DDR2-667R	5-5-5-15 CR1	13118 MB/s
12592 MB/s	Core i7	2575 MHz	[TRIAL VERSION]	i440FX			12592 MB/s
11534 MB/s	4x Opteron 2210 HE	1800 MHz	Tyan Thunder h2000M	BCM5785	Dual DDR2-600R	5-5-5-15 CR1	11534 MB/s
11178 MB/s	2x Core i5-650 HT	3200 MHz	Supermicro C7SIM-Q	Q57 Int.	Dual DDR3-1333	9-9-9-24 CR1	11178 MB/s
10112 MB/s	4x Phenom II X4 Black 940	3000 MHz	Asus M3N78-EM	GeForce8300 Int.	Ganged Dual DDR2-800	5-5-5-18 CR2	10112 MB/s
9714 MB/s	2x Athlon64 X2 Black 6400+	3200 MHz	MSI K9N SLI Platinum	nForce570SLI	Dual DDR2-800	4-4-4-11 CR1	9714 MB/s

Figura 15: Benchmark de Escritura de memoria.

AIDA64 Cache & Memory Benchmark

	Read	Write	Copy	Latency
Memory	16324 MB/s	TRIAL VERSION	TRIAL VERSION	75.2 ns
L1 Cache	TRIAL VERSION	390209.7 GB/s	390209.7 GB/s	1.3 ns
L2 Cache	249734.2 GB/s	31838 MB/s	249734.2 GB/s	TRIAL VERSION
L3 Cache				
L4 Cache				
CPU Type	Intel Core i7 (Ivy Bridge-MB)			
CPU Stepping	E1/L1/N0/P0			
CPU Clock	2601.4 MHz (original: [TRIAL VERSION] MHz)			
CPU FSB				
CPU Multiplier	North Bridge Clock			
Memory Bus	DRAM:FSB Ratio			
Memory Type				
Chipset	Intel Natoma 82440FX			
Motherboard	[TRIAL VERSION]			

AIDA64 v5.20.3400 / BenchDLL 4.1.633-x64 (c) 1995-2015 FinalWire Ltd.

Save

Start Benchmark

Close

Figura 16: Benchmark de lectura, escritura de la memoria y caches.

Cuestión 5.

Programe un benchmark usando el lenguaje que desee. El benchmark debe incluir:

1. **Objetivo del benchmark.**
2. **Métricas (unidades, variables, puntuaciones, etc.).**
3. **Instrucciones para su uso.**
4. **Ejemplo de uso analizando los resultados.**

Tenga en cuenta que puede comparar varios gestores de BD, lenguajes de programación web (tiempos de ejecución, gestión de memoria, ...), duración de la batería, servidor DNS, etc., Alternativamente, puede descargar alguno de algún repositorio en github y modificarlo según sus necesidades.

El recurso sobre el que se va a plantear la hipótesis después de pensarlo, se hará usando memorias extraíbles USB por motivos más que obvios como pueden ser la falta de tiempo, y la facilidad de implementación y trabajo físico con el ordenador ya que al no disponer de HDD sata u otros no puedo realizar otro tipo de análisis, aunque el análisis implementado sirve también para HDD sata u extraíbles. La idea es hacer una comparativa entre tres elementos, un disco duro extraíble que usa puerto USB TrekStor, una memoria USB Alcor 2.0 y una memoria USB Kingston 3.0, para estudiarlos nos interesa obtener datos acerca de la escritura y la lectura de archivos sean grandes o pequeños por eso se han planteado dos experimentos acordes a esta idea:

- El primer experimento consistirá en medir el tiempo y la velocidad de operaciones de escritura, lectura y una mezcla de dichas operaciones de un archivo con un volumen de 2.0GB.
- El segundo experimento consistirá en medir el tiempo y la velocidad de operaciones de escritura, lectura y una mezcla de dichas operaciones de muchos archivos (en mi caso 100) cada uno con un volumen de 50 MB lo que viene a corresponder a 5.00 GB aproximadamente.

Con respecto a la carga de trabajo, se han elegido estos tamaños puesto que con lo que vamos a trabajar tiene un espacio limitado, y los dispositivos que usan USB 2.0 no son tan rápidos a la hora de escribir archivos grandes por lo que con esos tamaños será más que suficiente para analizar las memorias, para crear dichos archivos hacemos uso de instrucciones de c++ para escribir archivos binarios con el tamaño que se desee. Con respecto a las medidas tomadas, se analizará tanto el tiempo de escritura total de los archivos como la velocidad en MB/s. La toma de medidas implementada puede ser un poco molesta por el tiempo que se demora ya que son necesarias una cantidad de datos para realizar las medias y por tanto para la realización de ambos experimentos puede demorarse alrededor de 90 minutos para que los datos obtenidos sean más fiables. La hipótesis a la que se quiere llegar realizando estas pruebas es: Los discos duros extraíbles tienen mayor velocidad de escritura y lectura que otros dispositivos USB sean 2.0 o 3.0. Para realizar las pruebas se ha realizado el benchmark que se pide implementar, y posteriormente se ha realizado un análisis teórico que se encuentra en la carpeta "Análisis" del benchmark. Vamos a comentar los resultados que me parecen más relevantes:

Vamos a comenzar con el experimento 1:

- **Escritura en dispositivo:** Como podemos observar en la figura 17 y por los datos aportados por el benchmark, el tiempo de Escritura en las memoria USB, sea 2.0 o 3.0, es mucho más grande que en el disco Extraible, esto puede ser debido a la estructura hardware de cada elemento, ya que las memorias USB son memorias flash y el disco duro usa platos, por lo que escribir en un plato en una determinada sección es posiblemente más rapido que hacerlo en un dispositivo flash.

Tiempos:

USB3.0 DISCOEXTRAIBLE USB2.0

224,827 73,0471 253,679
238,093 63,5761 270,637
240,019 63,2625 299,595
242,304 61,6963 293,076
242,38 67,9739 314,684

Intervalos de confianza(respectivamente):

[228,4385433164, 246,6106566836]
[60,1736212308, 71,6487387692]
[256,3166292447, 316,3517707553]

Velocidades:

USB3.0 DISCOEXTRAIBLE USB2.0

9,10922 28,0367 8,07319
8,6017 32,2133 7,56734
8,53266 32,373 6,83589
8,45218 33,1949 6,98795
8,44954 30,1292 6,50812

Intervalos de confianza(respectivamente):

[8,2866954294, 8,9714245706]
[28,5897085763, 33,7891314237]
[6,4206993818, 7,9682966182]

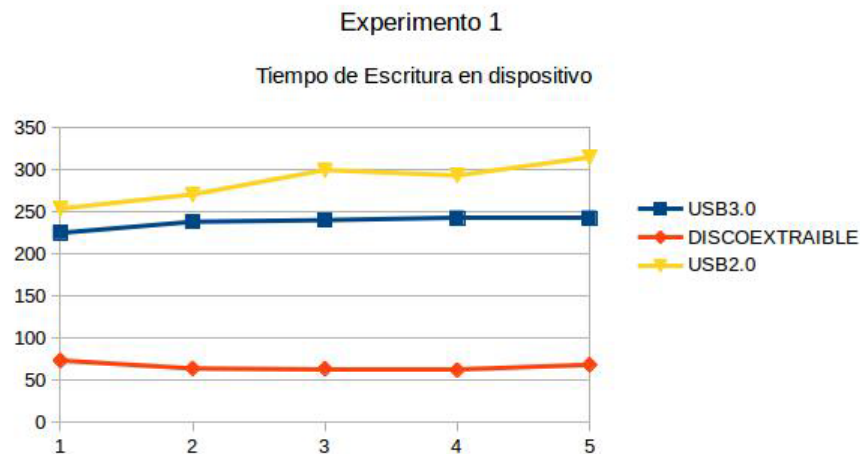


Figura 17: Gráfica de Tiempo de escritura.

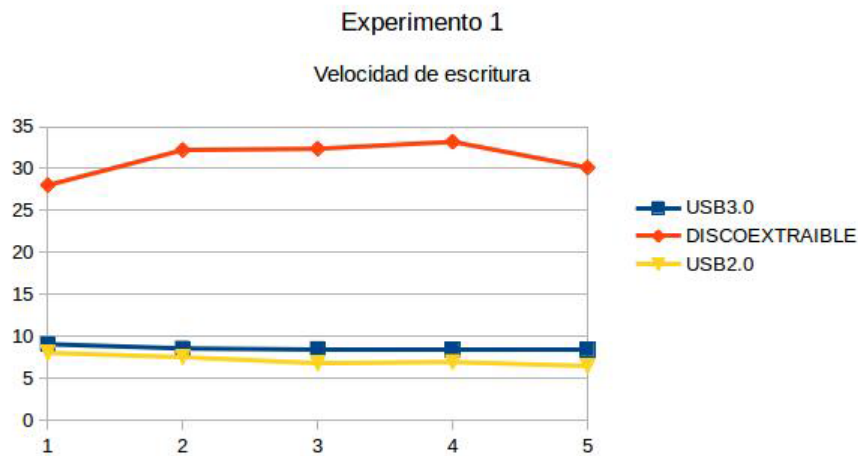


Figura 18: Gráfica de Velocidad de escritura.

- **Lectura en dispositivo:** Sin embargo los datos obtenidos nos dicen que las memorias flash 3.0 son más rápidas a la hora de leer un fichero contenido en la misma, esto puede ser por la velocidad de transferencia de los puertos que están preparados para estas unidades 3.0, y por la tardanza del disco en buscar las secciones. (figura 18)
Tiempos:

USB3.0	DISCOEXTRAIBLE	USB2.0
39,2158	85,5879	134,683
37,97	94,0525	134,744
41,489	96,4047	141,512
40,0996	90,4485	140,747
38,7784	93,6922	142,632

Intervalos de confianza(respectivamente):

[37,8380588981,	41,1830611019]
[86,8431682756,	97,2311517244]
[134,0863965121,	143,6408034879]

Velocidades:

USB3.0	DISCOEXTRAIBLE	USB2.0
52,2238	23,9286	15,206
53,9373	21,7751	15,1991
49,3625	21,2438	14,4723
51,0728	22,6427	14,551
52,8129	21,8588	14,3586

Intervalos de confianza(respectivamente):

[49,7124678733,	54,0512521267]
[20,9942087077,	23,5853912923]
[14,2457254748,	15,2690745252]

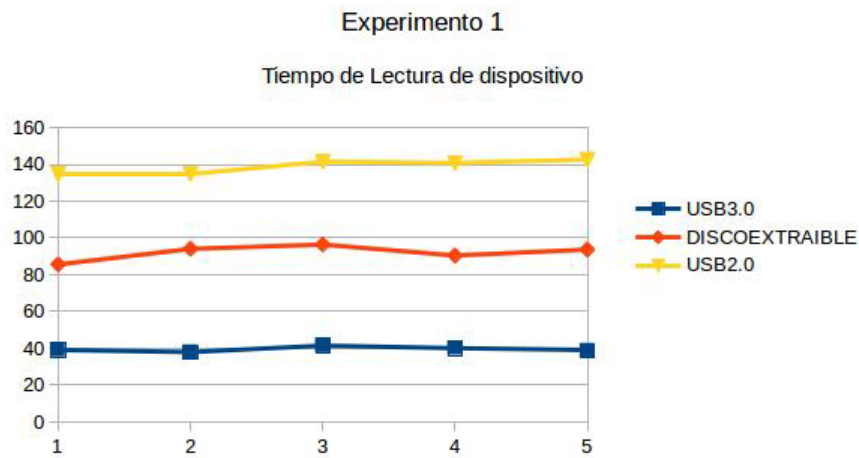


Figura 19: Gráfica de Tiempo de lectura.

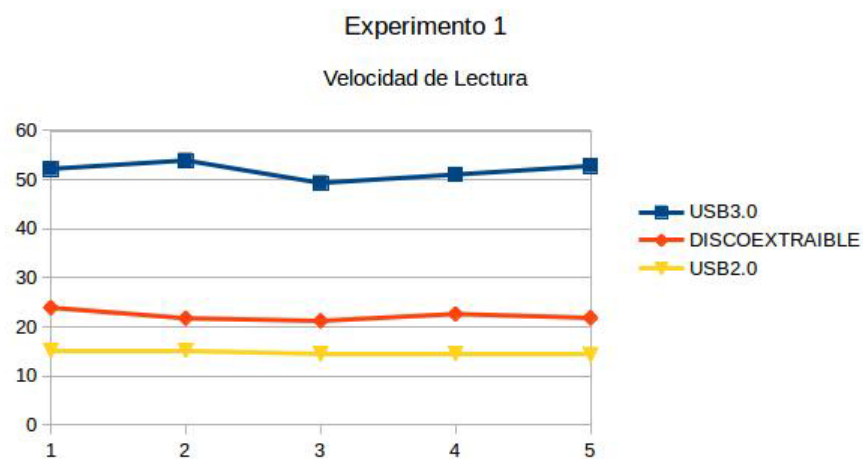


Figura 20: Gráfica de Velocidad de lectura.

- Escritura/Lectura en dispositivo:** La escritura y lectura es una operación alternada entre las dos anteriores sin embargo, es muy similar a la salida que se produce cuando directamente escribimos, parece que el coste en tiempo de lectura de trozos más pequeños mientras se esta escribiendo no tiene un impacto muy grande en tiempo por tanto el disco es el más rápido.

Vamos a comenzar con el experimento 2:

- **Escritura en dispositivo:** Para la escritura de muchos archivos de tamaño reducido se produce lo mismo que en el experimento 1, el disco duro tiene una estructura que permite una velocidad mayor de escritura pese a que la salida sea 2.0.
Tiempos:

USB3.0	DISCOEXTRAIBLE	USB2.0
586,369	157,098	741,888
613,674	164,436	790,578
613,188	160,346	750,954
597,932	154,524	723,095
612,581	170,969	765,9485

Intervalos de confianza(respectivamente):

[589,585895606, 619,911704394]
[153,4360187715, 169,5131812285]
[722,8939242407, 786,0916790393]

Velocidades:

USB3.0	DISCOEXTRAIBLE	USB2.0
8,52705	31,8273	6,73956
8,14765	30,407	6,32449
8,1541	31,1826	6,65819
8,36215	32,3575	5,23515
8,16219	29,2451	5,27145

Intervalos de confianza(respectivamente):

[8,0604695533, 8,4807864467]
[29,4842152052, 32,5235847948]
[5,1268603187, 6,9646756813]

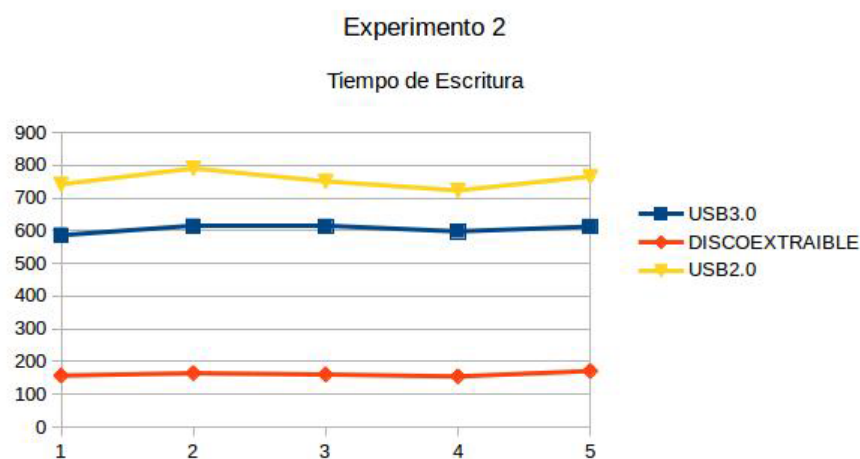


Figura 21: Gráfica de Tiempo de escritura.

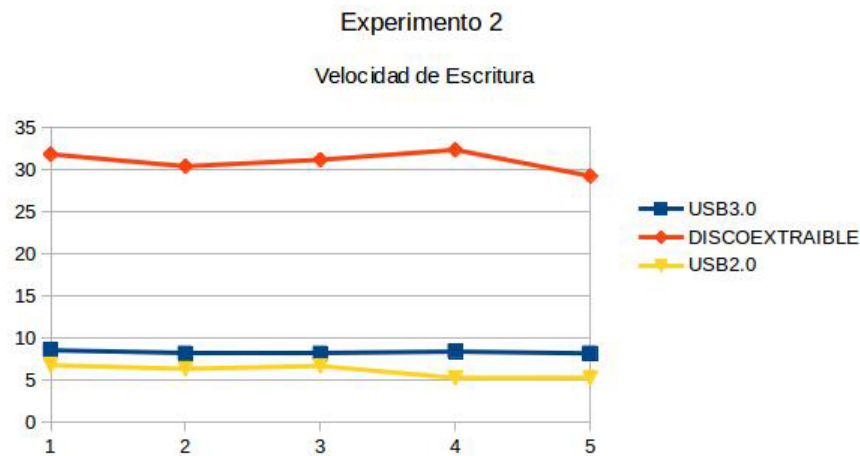


Figura 22: Gráfica de Velocidad de escritura.

- **Lectura en dispositivo:** Ocurre exactamente lo mismo que el experimento 1 el USB 3.0 tiene la mejor velocidad y tiempo de lectura. (figura 23)

Tiempos:

USB3.0 DISCOEXTRAIBLE USB2.0

88,3328 215,326 355,336
 91,1757 216,5 358,154
 88,687 205,875 275,722
 88,0768 215,787 295,82826
 87,8435 205,924 314,9563

Intervalos de confianza(respectivamente):

[87,1445900944, 90,5017299056]
 [205,0810190598, 218,6837809402]
 [274,9104340972, 365,0881899028]

Velocidades:

USB3.0 DISCOEXTRAIBLE USB2.0

56,6041 23,2206 14,0712
 54,8392 23,0947 13,9605
 56,378 24,2866 18,1342
 56,7686 23,171 16,03675
 56,9194 24,2808 17,2674

Intervalos de confianza(respectivamente):

[55,2564035274, 57,3473164726]
 [22,8459201016, 24,3755598984]
 [13,5722208951, 18,2157991049]

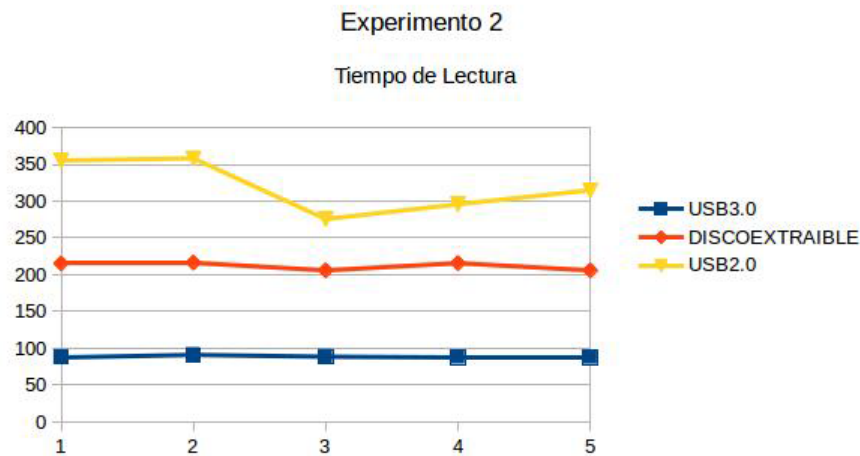


Figura 23: Gráfica de Tiempo de lectura.

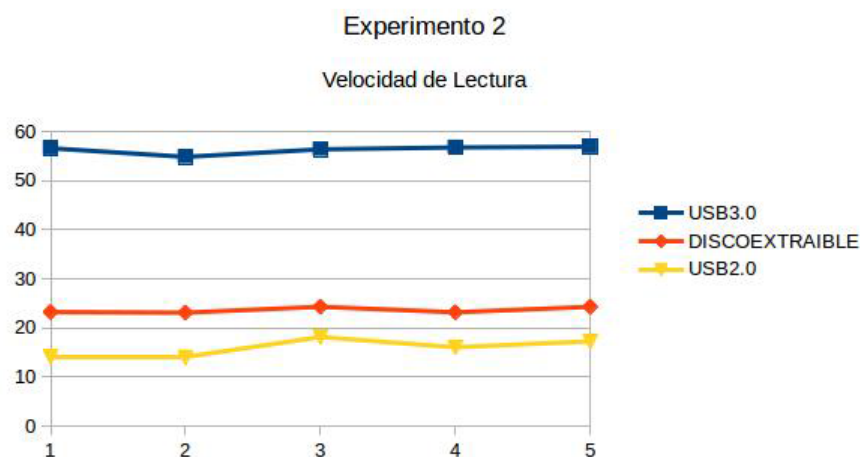


Figura 24: Gráfica de Velocidad de lectura.

- **Escritura/Lectura en dispositivo:** Ocurre exactamente lo mismo que en la escritura debido a que se reduce el tiempo de lectura demasiado por lo que no se produce un impacto muy fuerte en el tiempo total de escritura y lectura.

Conclusión: En mi opinion hemos probado la hipótesis en parte, ya que es cierto que para estos tres dispositivos el disco duro tiene una mayor velocidad de escritura (menor tiempo) que el resto de dispositivos, sin embargo el usb 3.0 lo supera en la velocidad de lectura. Aunque finalmente cuando vamos a usar un disco duro extraible nos interesa normalmente que se produzcan operaciones de lectura y escritura entremezcladas por lo que un disco duro extraible es mejor opción que un usb3.0 por esa misma razón, ya que como se ha indicado antes el impacto de tiempo de la lectura sobre las operaciones de escritura es muy pequeño normalmente y permite mejores tiempo al disco duro extraible.

La programación del Benchmark se ha llevado a cabo haciendo uso del lenguaje c++, implementando una clase que se dedica a hacer el análisis de unidades de almacenamiento. El análisis de los resultados se lleva a cabo de manera manual por motivos de tiempo, por defecto el programa implementado aporta una versión de los datos en formato cvs, para que pueda ser modificado con un editor estadístico sin embargo es muy probable que tengan que cambiarse las preferencias de lenguaje. El funcionamiento del benchmark es como se describe en la figura 25, en primer lugar se iniciará una instancia de la clase benchmark que será la que ejecute los tests, introducimos el número del experimento que queremos realizar en el caso de introducir uno que no exista se volverá a preguntar, es importante mencionar que este diagrama se producirá tantas veces como niveles se quieran hacer, en nuestro caso 3, después de haber seleccionado el experimento se preguntará cual es el nombre del dispositivo conectado al usb y posteriormente se realizarán los tres tests de escritura, lectura y una alternada de ambas operaciones. Estos tests se realizarán 5 veces y por último se calculará la velocidad a partir del tiempo y se copiarán los datos a un archivo cvs.

El benchmark implementado actúa de forma independiente, tan solo se pide al usuario introducir el nombre del dispositivo a analizar cada cierto tiempo.

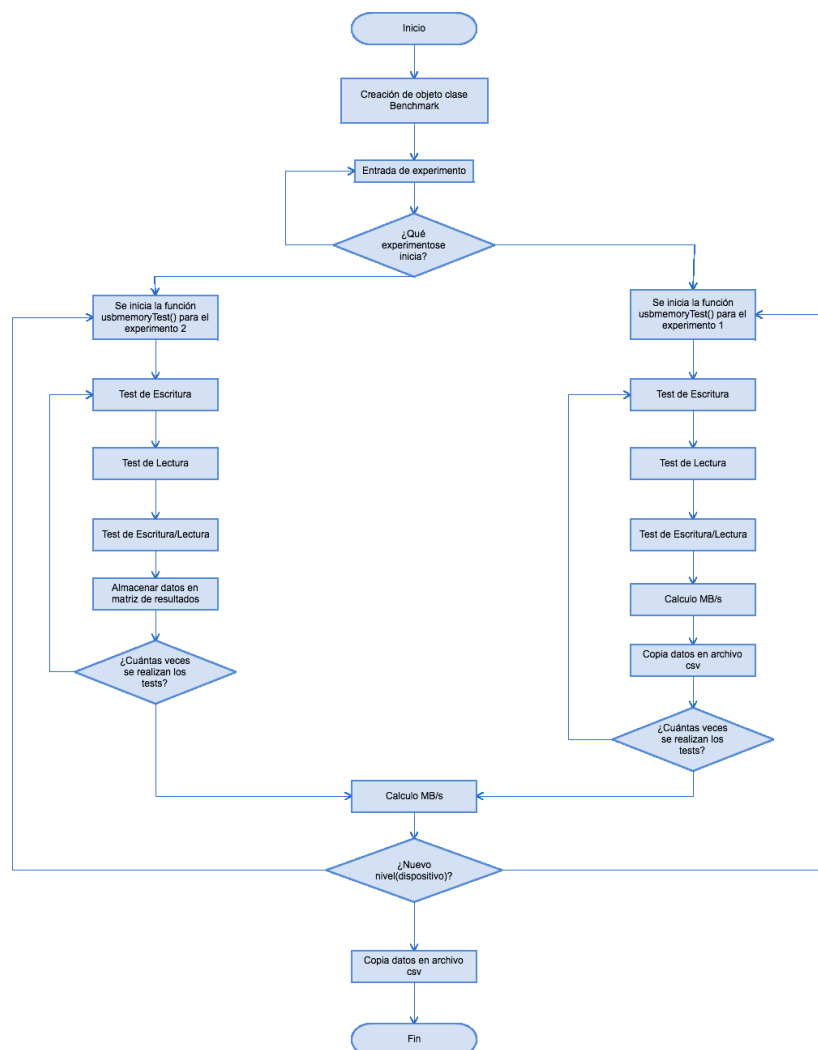


Figura 25: Diagrama de Flujo del programa que ejecutará la prueba de rendimiento.

Esquema de subdirectorios del archivo comprimido:

- benchmark
 - Análisis
 - Exp1.ods
 - Exp2.ods
 - Exp1.csv
 - Exp2.csv
 - bin
 - Bench
 - include
 - benchmark.h
 - object
 - src
 - benchmark.cpp
 - main.cpp
 - data.csv
 - infousb

La salida del programa se realiza en data.csv, los archivos de la carpeta Análisis contienen el análisis teórico de los datos resultantes del benchmark en los archivos Exp1.ods y Exp2.ods, la carpeta bin contiene el archivo binario tal y como se espera, include las cabeceras y src el código del programa.

Referencias

1. <http://www.phoronix-test-suite.com/documentation/phoronix-test-suite.pdf>
2. <http://fedoraproject.org/wiki/EPEL/es>
3. http://www.phoronix.com/scan.php?page=news_item&px=NzA3Mg
4. <http://www.phoronix-test-suite.com/?k=documentation>
5. <https://httpd.apache.org/docs/2.2/programs/ab.html>
6. www.aida64.com/downloads