

1. Micro-benchmarks de Sistemas Informáticos

1.1. Micro-benchmarks en Linux - CentOS y Ubuntu

Cuestión 1. Describa las características principales de la suite y describa los parámetros, carga que usan de prueba y tipo de test de al menos 3 benchmarks que contempla la suite.

Phoronix Test Suite es una suite de pruebas y benchmarking que nos entrega una plataforma, limpia, sencilla, reproducible y aún más importante, de fácil entendimiento. Está desarrollada para funcionar sobre plataformas Linux, Solaris, Mac OS X, Microsoft Windows, y BSD.

En sí, Phoronix Test Suite es un manejador de test (pts-core) que permite instalar un sin número de pruebas desde un banco de test online, y realizar estas pruebas través de un archivo de configuración XML (extremadamente fácilmente configurar), entregándote los resultados de una manera simple y elegante, con gráficos muy fáciles de entender, y además permitiendo compartir tus resultados de manera anónima.

La lista de comandos es bastante extensa, pero algunos serán:

"phoronix-test-suite list-available-tests" y "phoronix-test-suite list-available-suites": Estos dos comandos listan los test y suite disponibles, no sólo muestra el nombre, si no que una pequeña descripción y qué punto es el que prueban. No te asustes si parece que se queda congelado, ya que está bajando las definiciones de las pruebas desde internet, así que puede demorarse bastante.

"phoronix-test-suite install Nombre-del-test-o-suite": Este comando permite instalar un test o una suite indicando su nombre, y si tiene dependencias, el sistema te pedirá tu contraseña para completarlas. Por con ejemplo, con "phoronix-test-suite install pts/encode-flac" instalas la prueba para ver cuánto tiempo tu computador tarda en realizar el encodeo de un archivo WAV en FLAC.

"phoronix-test-suite benchmark Nombre-del-test-o-Nombre-de-la-suite": Este comando realiza la prueba seleccionada, como en el caso anterior, lo que debes escribir es "phoronix-test-suite benchmark pts/encode-flac".

Al realizar un test se nos mostrará antes de realizarlo nuestra configuración completa (puedes acceder a ella con "phoronix-test-suite system-information"), y se nos preguntará si deseamos guardar la prueba (quedan guardadas en el home en .phoronix-test-suite/test-results), luego el test comenzará y una vez terminado se nos preguntará si queremos ver los resultados en el navegador, mostrándonos los datos y una breve explicación sobre ellos.

Phoronix Test Suite es un producto maduro, con una amplia base de pruebas, pero siempre queremos más; estas son fácilmente realizables, en un ambiente sencillo, reproducible y robusto, no obstante su promesa de ser multiplataforma está inconclusa, ya que en Windows depende de cpu-z para funcionar, rompiendo el paradigma de crear un ambiente aislado donde realizar las pruebas (solo existe soporte para Windows 7 64bits) .

Pros: Facilidad de instalación, Sencillo, Reproducible, Robusto, Base de pruebas amplia, Entrega resultados entendibles para cualquier persona.

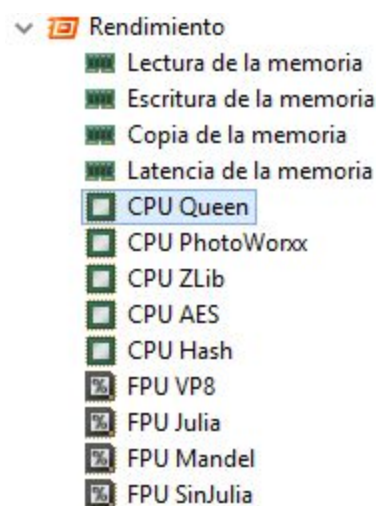
Contras: Multiplataforma inconclusa en Windows, Base de pruebas limitada.

1.2. Micro-benchmarks en Windows - Windows

Cuestión 2. Instale una de las aplicaciones y realice al menos 3 test distintos a Windows Server para medir la CPU y la memoria RAM. De los 3 test seleccionados indique qué pruebas hace el software, cómo las hace y muestre los resultados de 5 ejecuciones en una gráfica.



Instalamos AIDA64 y ejecutamos el programa.



Abrimos Rendimiento abajo a la izquierda y seleccionamos CPU Queen, que es el problema de las N reinas para evaluar el rendimiento de la CPU.



Pulsamos el botón de Iniciar.

12140	2x Core 2 Extreme X6...	2933 MHz	Abit AB9	P965	Dual DDR2-800
11236	2x Athlon64 X2 Black ...	3200 MHz	MSI K9N SLI Platinum	nForce570SLI	Dual DDR2-800
9614	2x Core 2 Duo P8400	2266 MHz	MSI MegaBook PR201	GM45 Int.	Dual DDR2-667
7485	2x Pentium EE 955 HT	3466 MHz	Intel D955XBK	i955X	Dual DDR2-667
7333	Core i7-3537U	3100 MHz	[TRIAL VERSION]	i440BX/ZX	
7304	2x Xeon HT	3400 MHz	Intel SE7320SP2	iE7320	Dual DDR333R
7300	2x Athlon64 X2 4000+	2100 MHz	ASRock ALiveNF7G-HDrea...	nForce7050-63...	Dual DDR2-700
5904	2x Atom D2500	1866 MHz	Intel D2500CC	NM10 Int.	DDR3-1066 SDR4
5452	Nano X2 L4350	1600 MHz	VIA EPIA-M900	VX900H Int.	DDR3-1066 SDR4
5164	2x E-350	1600 MHz	ASRock E350M1	A50M Int.	DDR3-1066 SDR4

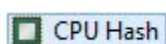
Y obtendremos una evaluación de nuestro CPU con respecto a otros, esto lo haremos 5 veces.



Seleccionamos ahora CPU PhotoWorxx, que consiste en un procesamiento de imágenes. Iniciamos la evaluación, como anteriormente.

6975 MPíxel/s	6x Phenom II X6 Blac...	3300 MHz	Gigabyte GA-890GPA-UD3...	AMD890GX Int.	Unganged Dual
6862 MPíxel/s	2x Core i5-650 HT	3200 MHz	Supermicro C7SIM-Q	Q57 Int.	Dual DDR3-1333
6131 MPíxel/s	8x Opteron 2344 HE	1700 MHz	Supermicro H8DME-2	nForcePro-3600	Unganged Quad
5627 MPíxel/s	4x Phenom II X4 Blac...	3000 MHz	Asus M3N78-EM	GeForce8300 Int.	Ganged Dual DD
5390 MPíxel/s	Core i7-3537U	3100 MHz	[TRIAL VERSION]	i440BX/ZX	
5276 MPíxel/s	Celeron J1900	2000 MHz	Gigabyte GA-J1900N-D3V	BayTrailD Int.	Dual DDR3-1333
4730 MPíxel/s	8x Xeon E5462	2800 MHz	Intel S5400SF	i5400	Quad DDR2-6400
4183 MPíxel/s	4x Core 2 Extreme QX...	3000 MHz	Gigabyte GA-EP35C-DS3R	P35	Dual DDR3-1066
4179 MPíxel/s	4x Athlon 5350	2050 MHz	ASRock AM1B-ITX	Yangtze Int.	DDR3-1600 SDR4
3840 MPíxel/s	4x Phenom X4 9500	2200 MHz	Asus M3A	AMD770	Ganged Dual DD

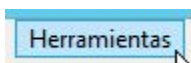
Y obtendremos otra evaluación, también lo haremos 5 veces.



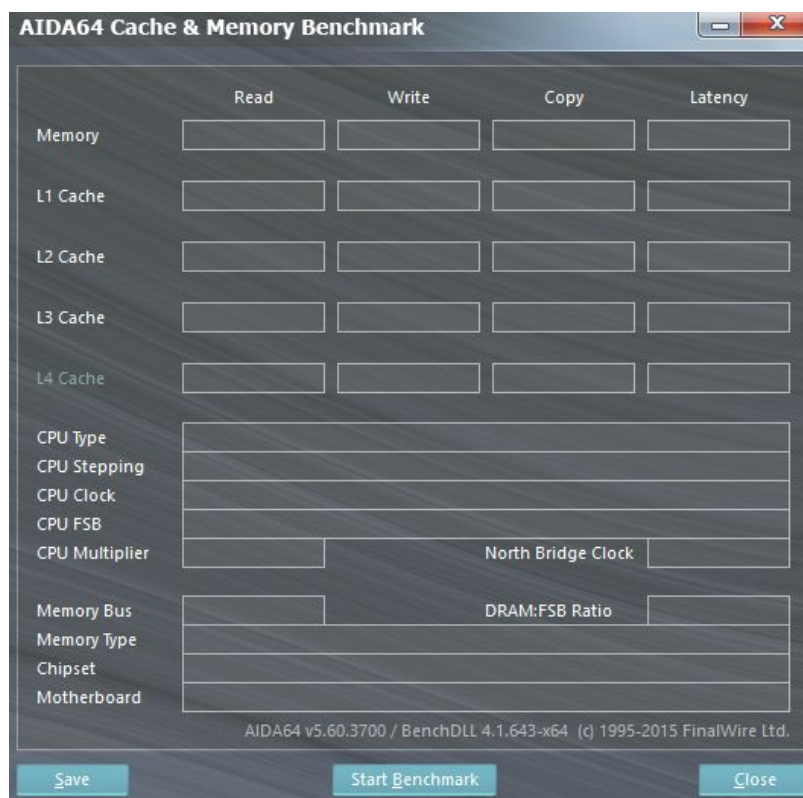
Pulsamos ahora la opción para evaluar con el algoritmo de Hash. Iniciamos la evaluación, como siempre.

828 MB/s	2x Pentium EE 955 HT	3466 MHz	Intel D955XBK	i955X	Dual DDR2-667
809 MB/s	2x Xeon HT	3400 MHz	Intel SE7320SP2	iE7320	Dual DDR333R
730 MB/s	2x Core 2 Duo P8400	2266 MHz	MSI MegaBook PR201	GM45 Int.	Dual DDR2-667
641 MB/s	2x Athlon64 X2 4000+	2100 MHz	ASRock ALiveNF7G-HDrea...	nForce7050-63...	Dual DDR2-700
580 MB/s	Core i7-3537U	3100 MHz	[TRIAL VERSION]	i440BX/ZX	
548 MB/s	2x Pentium D 820	2800 MHz	Abit Fatal1ty F-I90HD	RS600 Int.	Dual DDR2-800
489 MB/s	Nano L2200	1600 MHz	VIA VB8001	CN896 Int.	DDR2-667 SDRAM
443 MB/s	P4EE HT	3733 MHz	Intel SE7230NH1LX	iE7230	Dual DDR2-667
427 MB/s	2x Opteron 240	1400 MHz	MSI K8D Master3-133 FS	AMD8100	Quad DDR400R
350 MB/s	2x Atom D2500	1866 MHz	Intel D2500CC	NM10 Int.	DDR3-1066 SDRAM
336 MB/s	Opteron 248	2200 MHz	MSI K8T Master1-FAR	K8T800	Dual DDR266R

Ejecutaremos esto otras 5 veces.



Para las evaluaciones de la memoria RAM pulsamos Herramientas, Rendimiento de la memoria y la caché.



Pulsamos ahora Start Benchmark y nos mostrará la evaluación.

	Read	Write	Copy	Latency
Memory	13969 MB/s	TRIAL VERSION	TRIAL VERSION	67.8 ns
L1 Cache	TRIAL VERSION	61348 MB/s	117.49 GB/s	1.5 ns
L2 Cache	60656 MB/s	39023 MB/s	51971 MB/s	TRIAL VERSION
L3 Cache	49521 MB/s	35609 MB/s	36655 MB/s	TRIAL VERSION

Este es el resultado, lo ejecutaremos 5 veces.

Cuestión 3. Resuma la información de las 5 ejecuciones anteriores en un índice y realice una comparativa de su sistema frente a otros sistemas actuales en base a los test que usó en el punto anterior.

El resumen se encuentra en el punto anterior, siendo uno de las evaluaciones dadas.

Mi conclusión es que mi sistema está bastante por debajo de otros que muestra el programa.

2. Benchmarking de servicios

Cuestión 4. Liste las distintas opciones que dispone apache benchmark y qué hace cada una de las opciones.

ab - Apache HTTP server benchmarking tool

Available Languages: [en](#) | [ko](#) | [tr](#)

ab is a tool for benchmarking your Apache Hypertext Transfer Protocol (HTTP) server. It is designed to give you an impression of how your current Apache installation performs. This especially shows you how many requests per second your Apache installation is capable of serving.

Synopsis

```
ab [ -A auth-username:password ] [ -b window-size ] [ -c concurrency ] [ -C cookie-name=value ] [ -d ] [ -e csv-file ] [
-f protocol ] [ -g gnuplot-file ] [ -h ] [ -H custom-header ] [ -i ] [ -k ] [ -n requests ] [ -p POST-file ] [ -P
proxy-auth-username:password ] [ -q ] [ -r ] [ -s ] [ -S ] [ -t timelimit ] [ -T content-type ] [ -u PUT-file ] [ -v
verbosity ] [ -V ] [ -w ] [ -x <table>-attributes ] [ -X proxy[:port] ] [ -y <tr>-attributes ] [ -z <td>-attributes ] [
-Z ciphersuite ] [http[s]://]hostname[:port]/path
```



Options

- A **auth-username:password**
 Supply BASIC Authentication credentials to the server. The username and password are separated by a single : and sent on the wire base64 encoded. The string is sent regardless of whether the server needs it (*i.e.*, has sent an 401 authentication needed).
- b **window-size**
 Size of TCP send/receive buffer, in bytes.
- c **concurrency**
 Number of multiple requests to perform at a time. Default is one request at a time.
- C **cookie-name=value**
 Add a Cookie: line to the request. The argument is typically in the form of a *name=value* pair. This field is repeatable.
- d
 Do not display the "percentage served within XX [ms] table". (legacy support).
- e **csv-file**
 Write a Comma separated value (CSV) file which contains for each percentage (from 1% to 100%) the time (in milliseconds) it took to serve that percentage of the requests. This is usually more useful than the 'gnuplot' file; as the results are already 'binned'.
- f **protocol**
 Specify SSL/TLS protocol (SSL2, SSL3, TLS1, or ALL).
- g **gnuplot-file**
 Write all measured values out as a 'gnuplot' or TSV (Tab separate values) file. This file can easily be imported into packages like Gnuplot, IDL, Mathematica, Igor or even Excel. The labels are on the first line of the file.
- h
 Display usage information.
- H **custom-header**
 Append extra headers to the request. The argument is typically in the form of a valid header line, containing a colon-separated field-value pair (*i.e.*, "Accept-Encoding: zip/zop;8bit").
- i
 Do HEAD requests instead of GET.
- k
 Enable the HTTP KeepAlive feature, *i.e.*, perform multiple requests within one HTTP session. Default is no KeepAlive.
- 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.
- p **POST-file**
 File containing data to POST. Remember to also set -T.
- P **proxy-auth-username:password**
 Supply BASIC Authentication credentials to a proxy en-route. The username and password are separated by a single : and sent on the wire base64 encoded. The string is sent regardless of whether the proxy needs it (*i.e.*, has sent an 407 proxy authentication needed).
- q
 When processing more than 150 requests, ab outputs a progress count on stderr every 10% or 100 requests or so. The -q flag will suppress these messages.
- r
 Don't exit on socket receive errors.
- s
 When compiled in (ab -h will show you) use the SSL protected https rather than the http protocol. This feature is experimental and very rudimentary. You probably do not want to use it.
- S
 Do not display the median and standard deviation values, nor display the warning/error messages when the average and median are more than one or two times the standard deviation apart. And default to the min/avg/max values. (legacy support).

- t *timelimit***
Maximum number of seconds to spend for benchmarking. This implies a -n 50000 internally. Use this to benchmark the server within a fixed total amount of time. Per default there is no timelimit.
- T *content-type***
Content-type header to use for POST/PUT data, eg. application/x-www-form-urlencoded. Default: text/plain.
- u *PUT-file***
File containing data to PUT. Remember to also set -T.
- v *verbosity***
Set verbosity level - 4 and above prints information on headers, 3 and above prints response codes (404, 200, etc.), 2 and above prints warnings and info.
- V**
Display version number and exit.
- w**
Print out results in HTML tables. Default table is two columns wide, with a white background.
- x *<table>-attributes***
String to use as attributes for <table>. Attributes are inserted <table here >.
- X *proxy[:port]***
Use a proxy server for the requests.
- y *<tr>-attributes***
String to use as attributes for <tr>.
- z *<td>-attributes***
String to use as attributes for <td>.
- Z *ciphersuite***
Specify SSL/TLS cipher suite (See openssl ciphers).

Cuestión 5. Elija 2 de las opciones de ab que considere más relevantes para medir el rendimiento del servicio web http. Realice una ejecución de ab con las 2 opciones que ha elegido contra alguno de los servicios http (httpd en CentOS, apache en Ubuntu e IIS en Windows Server) de las máquinas virtuales que creó y configuró en las prácticas anteriores. Incluya los resultados en la memoria así como el comando que usó para la ejecución. ¿Qué información muestra ab como resultado de la ejecución? Haga un resumen.

```
ab -n 2000 -c 50 http://localhost:80/
```

Este comando realiza 2000 peticiones con una concurrencia de 50 al servidor que tenemos en el puerto 80.

```

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

Document Path:        /
Document Length:      11510 bytes

Concurrency Level:    50
Time taken for tests:  0.808 seconds
Complete requests:    2000
Failed requests:       0
Total transferred:    23566000 bytes
HTML transferred:     23020000 bytes
Requests per second:  2475.35 [#/sec] (mean)
Time per request:     20.199 [ms] (mean)
Time per request:     0.404 [ms] (mean, across all concurrent requests)
Transfer rate:        28483.50 [Kbytes/sec] received

Connection Times (ms)
              min    mean[+/-sd] median    max
Connect:        0      0   0.8      0      13
Processing:      6     20  49.2      8     335
Waiting:        6     19  49.1      8     334
Total:          6     20  50.0      8     340

Percentage of the requests served within a certain time (ms)
 50%      8
 66%      9
 75%     21
 80%     21
 90%     22
 95%     23
 98%    322
 99%    332
100%   340 (longest request)
  
```

Y este es el resultado.

Cuestión 6. Realice al menos 10 ejecuciones de ab con las mismas opciones que eligió anteriormente sobre el index.html que creó en la práctica 3 y que sirve cada uno de los servicios http de las máquinas virtuales de las prácticas anteriores. Elija 2 índices de rendimiento que considere relevantes para medir el rendimiento del servicio web http y muestre una tabla resumen con la siguiente información:

	CentOS		Ubuntu		Windows	
Ejecución nº	Índice de rendimiento 1	Índice de rendimiento 2	Índice de rendimiento 1	Índice de rendimiento 2	Índice de rendimiento 1	Índice de rendimiento 2
1						
2						
...						
RESUMEN	resumen índice 1*	resumen índice 2*	resumen índice 1*	resumen índice 2*	resumen índice 1*	resumen índice 2*

* Indique con que medida estadística ha resumido el índice de rendimiento.

¿Cuál de los tres servicios web (httpd en CentOS, apache en Ubuntu e IIS en Windows Server) proporciona mejores resultados según los experimentos que ha realizado? Realice una crítica a las conclusiones obtenidas. Puede ilustrar los resultados con gráficas.

El mejor servicio web de los tres es apache en Ubuntu, tiene mejor tiempo de respuesta y admite mayor número de ellas por segundo.

Cuestión 7. Ejecute ab sobre distintas páginas web contra el servicio http que concluyó que era mejor según el estudio realizado en la cuestión 6. Por ejemplo, sobre páginas .html con varias imágenes, con varios hipervínculos y con páginas .php o .asp. Analice los resultados y exponga una conclusión.

```

Server Hostname:      www.youtube.es
Server Port:         443
SSL/TLS Protocol:    TLSv1.2,EC,DHE-RSA-AES128-GCM-SHA256,2048,128

Document Path:      /
Document Length:    0 bytes

Concurrency Level:   50
Time taken for tests: 13.398 seconds
Complete requests:   2000
Failed requests:     0
Non-2xx responses:   2000
Total transferred:   1116000 bytes
HTML transferred:    0 bytes
Requests per second: 149.27 [#/sec] (mean)
Time per request:    334.962 [ms] (mean)
Time per request:    6.699 [ms] (mean, across all concurrent requests)
Transfer rate:       81.34 [Kbytes/sec] received

Connection Times (ms)
      min  mean[+/-sd] median   max
Connect:  139   236 107.5    219   1249
Processing:  66    92  33.7     84    428
Waiting:    65    91  33.5     83    428
Total:      209   328 114.3    307   1337

Percentage of the requests served within a certain time (ms)
 50%    307
 66%    328
 75%    347
 80%    354
 90%    376
 95%    402
 98%    636
 99%    701
100%   1337 (longest request)

```

```

Server Port:          443
SSL/TLS Protocol:    TLSv1.2,ECDHE-RSA-AES128-GCM-SHA256,2048,128

Document Path:       /
Document Length:     155444 bytes

Concurrency Level:    50
Time taken for tests: 37.584 seconds
Complete requests:    2000
Failed requests:      1995
    (Connect: 0, Receive: 0, Length: 1995, Exceptions: 0)
Non-2xx responses:    1546
Total transferred:    72327299 bytes
HTML transferred:     71078596 bytes
Requests per second:  53.21 [#/sec] (mean)
Time per request:     939.602 [ms] (mean)
Time per request:     18.792 [ms] (mean, across all concurrent requests)
Transfer rate:        1879.31 [Kbytes/sec] received
    
```

```

Connection Times (ms)
      min   mean[+/-sd] median   max
Connect:   145   310 356.6   208   6577
Processing: 260   621 513.7   471   4856
Waiting:    87   390 113.5   394    805
Total:     423   931 755.6   679  11432
    
```

```

Percentage of the requests served within a certain time (ms)
 50%    679
 66%    739
 75%    842
 80%   1165
 90%   1734
 95%   2060
 98%   3484
 99%   3881
100%  11432 (longest request)
    
```

```

Server Software:      nginx/1.9.9
Server Hostname:      www.marca.com
Server Port:          80

Document Path:       /
Document Length:     383876 bytes

Concurrency Level:    50
Time taken for tests: 163.059 seconds
Complete requests:    2000
Failed requests:      1821
    (Connect: 0, Receive: 0, Length: 1821, Exceptions: 0)
Total transferred:    770256643 bytes
HTML transferred:     769200533 bytes
Requests per second:  12.27 [#/sec] (mean)
Time per request:     4076.478 [ms] (mean)
Time per request:     81.530 [ms] (mean, across all concurrent requests)
Transfer rate:        4613.07 [Kbytes/sec] received
    
```

```

Connection Times (ms)
      min   mean[+/-sd] median   max
Connect:    26   176 785.7     80  15178
Processing: 426  3873 2428.6   3165  21770
Waiting:    40   221 128.3    217   3743
Total:     523  4050 2560.7   3278  21952
    
```

```

Percentage of the requests served within a certain time (ms)
 50%    3278
 66%    3727
 75%    4284
 80%    4730
 90%    7020
 95%    8924
 98%   11998
 99%   16654
100%  21952 (longest request)
    
```

3. Cuestiones propias.

Cuestión 8. ¿Qué ocurre si el servidor al que hacemos el test con ab no existe?

```
apr_sockaddr_info_get() for fasdf.asdf: Name or service not known (670002)
```

Se nos da un error de desconocimiento del servicio.

Cuestión 9. ¿Por qué el valor que nos da una evaluación de un servicio o sistema no es siempre el mismo?

Porque depende del estado del sistema en concreto, los procesos que se estén ejecutando en ese momento o miles de cosas que influyen, que lo hacen casi aleatorio dentro de unos rangos.

Cuestión 10. ¿Qué diferencia existe entre peticiones por segundo y tiempo por petición?

Peticiones por segundo indica la cantidad de peticiones que se pueden recibir en un segundo mientras que tiempo por petición indica el tiempo que tarda en contestar esta petición.