Franxx
www.wuolah.com/student/Franxx
1661

RodríguezHernándezAlbertoP5.pdf

PRÁCTICA 5

3° Configuración y Evaluación de Sistemas Informáticos

Grado en Ingeniería Informática

Escuela Politécnica Superior de Córdoba
UCO - Universidad de Córdoba

BENCHMARKING

Práctica 5

Alberto Rodríguez Hernández

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

Características de Phoronix Suite:

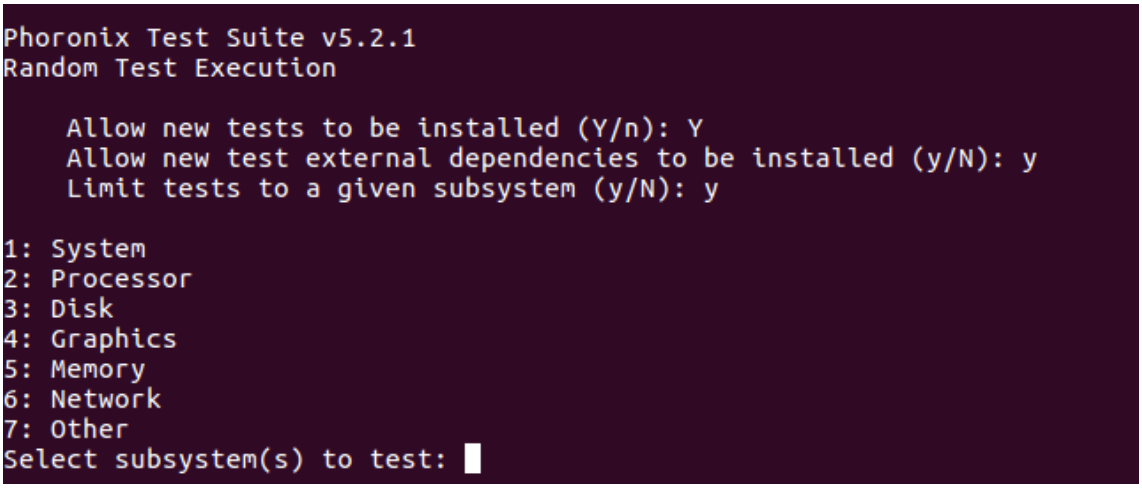
1. Sencilla de usar: La plataforma se encarga de proporcionarte los benchmarks necesarios y del proceso completo desde descargar hasta correr el benchmark.
2. Existen todo tipo de benchmarks proporcionados por http://openbenchmarking.org y en caso de que no lo encuentres siempre está la opción de crearlo tú mismo y subirlo inmediatamente.
3. Además de proporcionarte estos benchmarks, también te entrega los resultados de una forma sencilla de comprender, con gráficos entendibles.
4. Otra característica importante a mencionar sería que puedes compartir tus datos tanto anónimamente como no, siendo así una plataforma con una comunidad bastante activa.

A continuación, listaré algunos de los parámetros que se pueden añadir al comando *phoronix-test-suite.* Tener en cuenta que los comandos se dividen en varios tipos dependiendo de si se encargan del manejo de resultados, de la información, del sistema, del manejo de la instalación, entre otras características :

- *Batch-benchmark[benchmark/test]:* Este comando instala y ejecuta los tests en el modo batch de phoronix-suite.
- *Run-random-tests:* Este comando te ejecuta tests aleatorios de http://openbenchmarking.org y compara los resultados de esos tests.
- *Openbenchmarking-repositories:* Este comando muestra los repositorios de http://openbenchmarking.org conectados al cliente de *Phoronix Test Suite*.
- Refresh-graphs [test result]: Este comando te re-renderiza y guarda los gráficos junto con el resultado del test que tú le hayas indicado previamente.

La carga de prueba usada por los tests son archivos que no pesen mucho. Por ejemplo, he probado a correr un conjunto de tests sobre mi procesador y estos se descargaban 4000 ficheros de 1MB cada uno para comprobar el estado de este. Añadir que cada test tendrá una carga de prueba diferente dependiendo de que parte de tu ordenador estés poniendo a prueba.

Respecto a los tipos de test, añadir que puedes encontrar de todo tipo, hasta suites que puedes ejecutar que contienen tests que ponen a prueba varias características de tu ordenador, pero lo más común es encontrarte tests que afecten a subsistemas como los siguientes expuestos en la foto:



Esta foto es una captura de la ejecución del comando *phoronix-test-suite run-random-tests* en mi máquina virtual.

Fuentes: [Características] https://www.phoronix-test-suite.com/?k=features

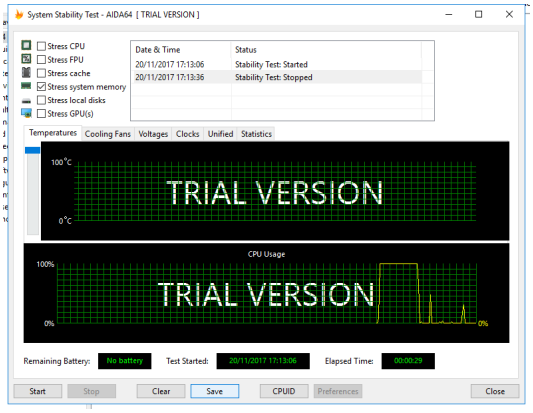
[Parámetros]: https://www.phoronix-test-suite.com/documentation/phoronix-test-suite.html

Cuestión 2: Instale una de las aplicaciones que realice al menos 3 test distintos de Windows Server para medir 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.

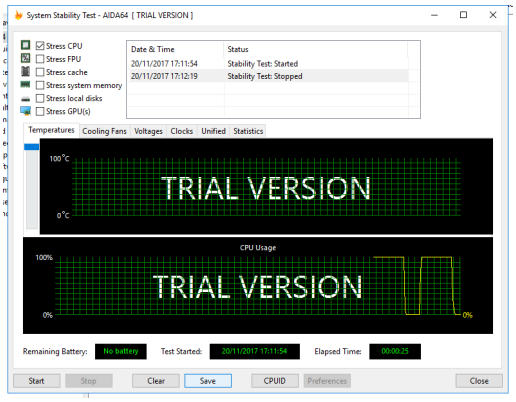
En mi caso, probé ambos softwares pero debido a que SiSoft no lo terminé de comprender decidí optar por Aida64.

Los tests que usé tanto para la memoria RAM como para la CPU fueron los llamados 'stress test' que consisten en probar el equipo en cuestión por encima de la capacidad operacional que es llevada a cabo. El resultado de las 5 ejecuciones que llevé acabo es el siguiente:

MEMORIA RAM



CPU



Fuentes: https://www.aida64.com/products/aida64-extreme

https://www.aida64.com/products/aida64-extreme

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.

…

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

En el manual de apache benchmark vienen todas las opciones, dentro de las cuales se encuentran las que voy a listar:

-k: Validar la característica HTTP KeepAlive.

-B *local-address:* Dirección a la que enlazar cuando se hacen conexiones externas.

-z *<td>-attributes:* Cadena a usar como atributo para <td>.

-r: No salir en los errores de recepción del socket.

Fuente (manual online de apache benchmark): https://linux.die.net/man/1/ab

Cuestión 5: Elija 2 opciones de ab que considere más relevantes para medir el rendimiento del servicio web http. Realice una ejecución ab con las dos opciones que ha elegido contra alguno de los servicios http 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.

Este ejercicio lo he llevado a cabo en CentOs. Para ello he seguido unos pasos para poder hacer uso del servicio *httpd*.

Primero hay que iniciar el servicio *httpd* con: *service httpd start*.

Segundo usamos el comando *service httpd status* para comprobar que se ha iniciado el servicio.

Una vez tenemos el servidor *httpd* en marcha, nos vamos a la terminal del sistema operativo natal y ahí ejecutamos *ab* sobre la ip de la máquina virtua. Por ejemplo: *ab -n 1000* http://192.168.160.128/

Los parámetros del comando que usé para este ejercicio fueron:

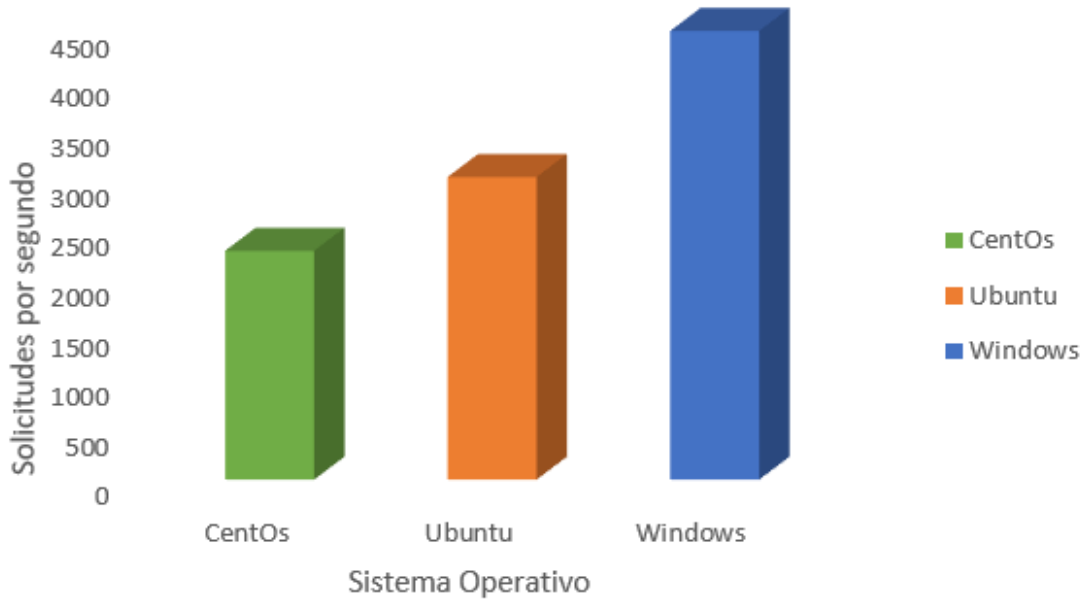
- n -> número de peticiones que le pides al servidor en cuestión
- c -> concurrencia del servidor
- g [filename] -> Guarda los valores medidos como 'gnuplot'

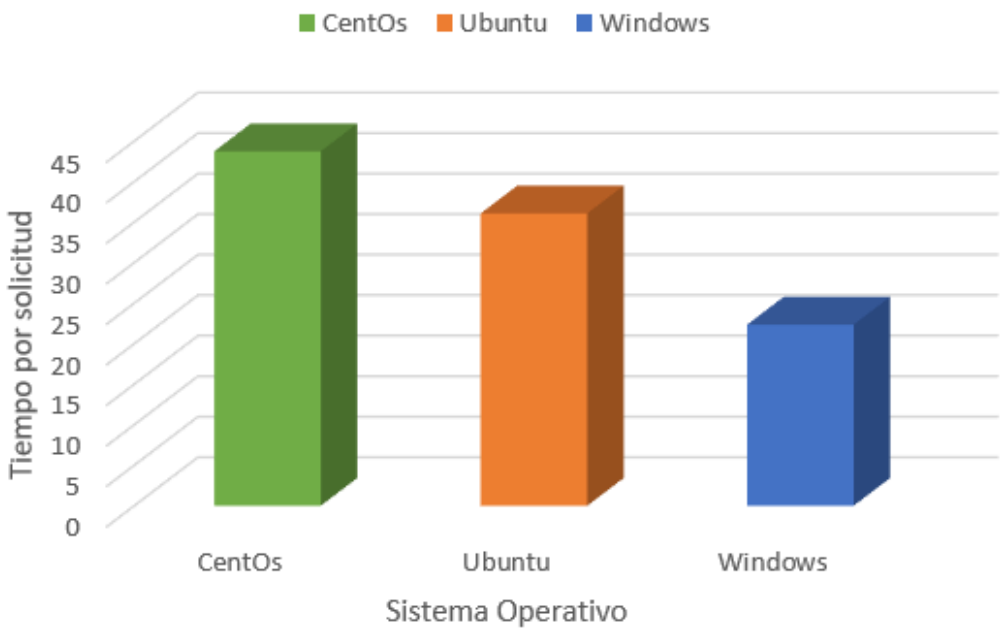
Fuente: https://www.nireleku.com/2012/08/como-medir-el-rendimiento-servidor-apache-ab/

Cuestión 6: Realice al menos 10 ejecuciones de ab con las mismas opciones que el ejercicio anterior 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 con la siguiente información.

	CentOS		Ubuntu		Windows	
Número De Ejecución	Solicitudes por segundo	Tiempo por solicitud	Solicitudes por segundo	Tiempo por solicitud	Solicitudes por segundo	Tiempo por solicitud
1	2326,73	42,979	2882,16	34,696	4001,07	24,993
2	2307,73	43,333	3027,67	33,029	4337,66	23,054
3	2286,79	43,729	3021,91	33,091	4457,86	22,432
4	2316,35	43,171	2997,84	33,357	4499,25	22,226
5	2298,17	43,513	3038,01	32,916	4452,72	22,458
6	2292,7	43,617	3010,27	33,220	4525,27	22,098
7	2250,71	44,43	3070,04	32,571	4773,35	20,95
8	2279,31	43,873	3069,85	32,575	4653,45	21,489
9	2279,36	43,872	3066,41	62,611	4529,78	21,684
10	2232,61	44,791	3063,62	32,641	4611,74	22,076
MEDIAS	2287,046	43,7308	3024,778	36,0707	4484,215	22,346

¿Cuál de los tres servicios proporciona mejores resultados según los experimentos que ha realizado? Realice una crítica a las conclusiones obtenidas. Puede ilustrar los resultados con gráficas





Observando los datos de la tabla y los gráficos, destacamos que CentOs tiene un mayor rendimiento en cuanto al envía de solicitudes por segundo pero a la hora del tiempo que le reserva a cada solicitud es mayor con respecto a Ubuntu y Windows. A pesar de esto yo valoraría por encima el nº de solicitudes por segundo que el tiempo que le reserva a cada una de ellas, siendo así *httpd* mejor servicio que los respectivos a Ubuntu y a Windows.

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

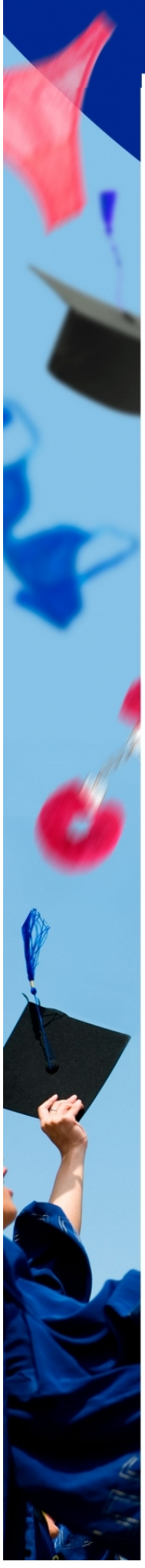
El comando que utilicé fue *ab -n 100 -c 10 http…* Las página que examiné y los datos recopilados son los siguientes:

http://moodle.uco.es/m1718/
Requests per second: 30.51 [#/sec] (mean)
Time per request: 327.761 [ms] (mean)

http://beta.manganimax.com/
Requests per second: 5.51 [#/sec] (mean)
Time per request: 1814.326 [ms] (mean)

https://www.youtube.com/watch?v=pMhfbLRoGEw
Requests per second: 128.24 [#/sec] (mean)
Time per request: 77.978 [ms] (mean)

Observando los resultados vemos que a más tiempo que se le dedica a una solicitud menos solicitudes por segundo se llevan cabo por lo que la conclusión que puedo sacar de poner a prueba estas tres páginas web, es que dependiendo de los elementos que tiene la página web en cuestión, esta tarda más en procesar una solicitud que las demás, siendo el vídeo de la plataforma YouTube el que menos tarda.



Cuestiones del alumno:

1) ¿Resulta fácil el uso de SiSoftware Sandra?

Creo que la parte de instalación y uso es fácil pero a la hora de sacar los datos y analizarlos es tema a parte, ya que apenas hay información de como sacarlos en internet y hacerlo por tu cuenta es muy complicado. Añadir que conseguí sacar datos de este software pero no he logrado entenderlos.

2) ¿Qué opciones del comando de apache benchmark encuentras más útiles?

En mi opinión las dos opciones que yo encuentro más útiles son *-n* y -g. *-n* nos permite pedir un número de peticiones determinadas al servidor y *-g* nos permite guardar los datos que se nos muestran por pantalla al ejecutar el comando en un fichero que nosotros queramos.