

Ingeniería de Servidores (2014-2015)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Memoria Práctica 4

Jose Arcos Aneas

10 de diciembre de 2014

Índice

1. Instale la aplicación. ¿Qué comando permite listar los benchmarks disponibles? 3
2. De los parámetros que le podemos pasar al comando ¿Qué significa -c 30 ? ¿y -n 1000? 3
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 las estadísticas. ¿Cuál es la que proporciona mejores resultados? Fíjese en el número de bytes transferidos, ¿es igual para cada máquina? 3
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.). 4
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 4

Índice de figuras

Índice de tablas

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 Para listar los benchmark que tenemos hemos de utilizar el comando: `>phoronix-test-suite list-tests`

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

La opción -c concurrencia 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.

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) y muestre las estadísticas. ¿Cuál es la que proporciona mejores resultados? Fíjese en el número de bytes transferidos, ¿es igual para cada máquina?

Windows

CentOs

Ubuntu

Se ha realizado 3 gráficas una por cada servidor, con su tiempo de respuesta. Con el comando “`ab -c 30 -n 1000 http://<servidor>`”

Puede observarse que se han puesto las gráficas en orden, de mayor tiempo a menor tiempo. Se puede ver como el que peor tiempo de respuesta es el de Windows server. (Aunque no se distinga exactamente los valores que toma) Al final ser sobrecarga demasiado y tarda hasta 40-50 veces más de lo normal. Y después le sigue, con un tiempo un poco más constante la máquina CentOs que aunque también se le envíen 1000 peticiones no llega a sobrecargarse tanto como la de Windows. Pero de media tiene unos tiempo un poco peores que los de Ubuntu. Y finalmente Ubuntu Server tiene unos tiempo constantes inferiores a 5 ms, pero llega un momento en el que se sobrecarga y comienza a tardar hasta 20 veces más lento.

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.).**

Después de instalarlo, con los comando 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 parametros:

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:

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:

5. **Programe un benchmark usando el lenguaje que desee. Elbenchmark debe incluir: 1) Objetivo del benchmark 2) Métricas (unidades, variables, puntuaciones, etc.) 3) Instrucciones para su uso 4) Ejemplo de uso analizando los resultados**

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

Por ejemplo vamos a instalar con la orden: - phoronix-test-suite install tachyon el benchmark tachyon para procesadores. Ynos saldria un resultado como el que sigue:

Tenemos una media de 122.39 segundos. Otro podría ser el de "blogbench": - phoronix-test-suite install blogbench

Y el último que vamos a probar es otro para procesadores llamado "bork phoronix-test-suite benchmark bork

¿Qué es Scala? Instale Gatling y pruebe los escenarios por defecto.

En primer lugar “Scala” es un lenguaje de programación de propósito general diseñado para expresar patrones comunes de programación de forma concisa y elegante. Se integran las características de los lenguajes orientados a objetos y funcional. Scala no es una extensión de Java pero es totalmente interoperable con él. Scala se traduce a bytecodes Java y la eficiencia de los programas compilados por lo general es igual que Java. Bueno pasando un poco de la teoría, vamos a instalarlo en Ubuntu con el siguiente comando: “sudo apt-get install scala”

El código de hola mundo sería el siguiente:

Guardaremos el código anterior para poder compilarlo con el siguiente comando. Una vez compilado obtendremos dos nuevos archivos `holaMundo.class` u `holaMundo$.class`. Si hay errores durante la compilación serán mostrados en el terminal. Cuando no haya errores podremos ejecutar el programa con la siguiente orden. El terminal nos muestra la salida del programa. `List()` nos indica los parámetros que le pasamos a la función. Esto se muestra por pantalla por la línea:
`println("Hola Mundo! - args.toList");`
El resultado es el siguiente:
`Hola Mundo! List(Parametro1, Parametro2)`

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

He elegido SisoftSandra.

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:

Cómo vemos, la configuración de dos procesadores es inferior a la de 4 en rendimiento del procesador (lógicamente) pero hay algunas áreas en las que se nota el aumento en la memoria RAM. Añado el resultado resumido de los test al final, a modo de apéndice, para no alargar en exceso la lectura.