

# Práctica 4, Sesión 1 – Benchmark

## - Apache Benchmark (ab)

Nos conectamos por ssh a ubuntu o CentOS. La herramienta ab (apache benchmark) nos permite realizar peticiones desde un cliente a nuestro servidor, simulando una carga en el mismo.

Apache Benchmark se distribuye como parte del servidor de Apache y se basa en el protocolo http. Una cosa es que se distribuya junto con apache y otra muy distinta es que sirve para testear cualquier servidor web, no es exclusivo de apache.

Lo usamos para simular cargas sobre cualquier servidor web (ejemplo → nginx(Usado por la ugr)).

Tiene un funcionamiento bastante sencillo. Si ponemos en el terminal “ab” nos muestra una ayuda bastante descriptiva de las posibles cosas que podemos hacer. De los muchos parámetros que tiene para ajustar su funcionamiento, nosotros sólo vamos a usar los dos primeros.

1. ab -n 100 -c 10 <http://www.ugr.es> //n = número de solicitudes a pedir, c = número de hebras  
la concurrencia que genera ab no es real, es decir, no se crean hebras para procesar las peticiones. Si especificamos un nivel de concurrencia N, lo que ocurre es que se mantienen siempre N peticiones activas, y cuando acaba una comienza otra, hasta llegar al límite.

Nginx va muy bien para proxy porque carga muy poco el sistema operativo para acceder a ficheros. Uno de los problemas de apache benchmark es que nos deja pasarle una url como parámetro. Con Jmeter veremos que eso no sirve para cargar el sistema.

2. curl <http://www.ugr.es> (nos muestra el código en html de la pagina con nginx)
3. curl -h (Nos muestra las opciones)
4. curl -I (Nos muestra la información de la página a visitar)

Connect → Tiempo en hacer la petición.

Waiting → Tiempo que transcurre desde que enviamos la petición hasta que nos llega el primer byte.

## - Phoronix

Phoronix-suite es un grupo de benchmark con un repositorio en el que cualquiera puede contribuir. Funciona de manera similar a los paquetes de linux. Para instalarlo hacemos lo siguiente:

1. sudo apt-cache search phoronix
2. sudo apt-get install phoronix-test-suite

Si ponemos sólo “phoronix-test-suite” podemos ver las múltiples opciones como las siguientes:

3. phoronix-test-suite **system-info** (Nos muestra información resumida del sistema)
4. phoronix-test-suite **system-sensors** (Nos muestra los sensores hardware instalados)

Ahora vamos a mirar los benchmarks que tenemos disponibles, que están publicados en un repositorio remoto. El procedimiento es primero buscarlo con apt-cache y nos bajamos el que más nos interese.

5. phoronix-test-suite **list-available-suites** (conjuntos de test con sentido común)
6. phoronix-test-suite **list-available-tests** (Aquí se muestran los benchmarks propiamente dichos)

Antes de bajar algún benchmark podemos consultar su información.

7. phoronix-test-suite info <**nombre del benchmark**>

Para bajar el benchmark primero hay que hacer **install** y luego **run**. También se puede hacer directamente **run**, ya que se instala y se arranca directamente. Con **install** podemos ir viendo las dependencias a instalar.

En este caso, vamos a instalar uno de tipo Procesador, concretamente **pts/compress-gzip**.

8. phoronix-test-suite run **pts/compress-gzip**

Para la instalación seguimos los pasos: Yes → iseA2 → Vbox VM sobre Ubuntu → Test de Prueba

9. phoronix-test-suite gui (es un servidor gráfico que facilita la forma de ejecutar los paquetes.)

Al terminar el benchmark nos devuelve unos números. Con esto solo no podemos realizar nada, necesitamos compararlos así que accederíamos a phoronix y a la web del test para comparar resultados.

Funcionalidad ssh -x → redirección de x-Windows (sistema de ventanas estándar de Linux)

Tanto el cliente como el servidor tienes que tener x-Windows (Ubuntu lo tiene por defecto)

## - Ejercicio JMeter

Es un programa para la simulación de carga. Lo que vamos a utilizar es tener un contenedor con Docker el cual estará gestionando una API. JMeter es tan estándar que los ficheros generados .jmx se pueden simular con la ejecución de ciertos benchmarks en servidores subiéndolo a la nube. Si buscamos jmeter cloud load test, nos sale BlazeMeter. En el apartado Features, se puede subir un archivo jMeter.

Vamos a iniciarnos con JMeter, lo instalamos y se nos lanza una interfaz. Test Plan es la raíz del proyecto y le ponemos nuestro nombre. WorkBench es un cajón para guardar cosas aunque no se vaya a utilizar, en Variables se le puede añadir y se le asigna un nombre y una IP. Para la visualización es como los parámetros de ejecución Debajo de un test lo primero que habría que añadir sería Thread Group. Ramp-Up es el crecimiento del número de hebras.

Ahora tras añadir la hebra hay que añadirle una función. en el ejemplo hemos utilizado un HTTP Request. Se le asigna la IP, el Puerto. Debajo está el request, en este caso vamos a añadirle los parámetros, añadimos los nombres de las variables y de que tipo son. En Add->Sampler->Access Log Sampler. Sirve por ejemplo para ejecutar los logs. Coges los últimos 1000 y los repartes entre las 1000 hebras para coger y simular esos usuarios. La documentación está situada en la ventana de la ? o la busquemos en internet.

```
jmeter -n -t ./basico.jmx -l salita.jtl # -n sin entorno gráfico
```

Hay una fórmula para vagos, para el modo proxy. En resumen lo que se hace es que el navegador de firefox, al indicarle el acceso a una pagina https, este necesita una llave de configuración, la cual JMeter lo engaña, ya que establecemos que Firefox se crea todo lo que diga este.