Guía desarrollo de Prácticas

ISE

Práctica 4: Benchmarks

Prof. David Palomar Sáez (dpalomar@ugr.es)

Índice

Objetivo

La práctica 4 se desarrolla en una unica lección. Evaluaremos las herramienta Phoronix para la ejecución de Benchmarks y Apache Benchmark junto a Jmeter para la simulación de cargas.

El alumno realizará un ejercicio obligatorio consistente en la definición de una prueba de carga empleando Jmeter sobre una interfaz Rest.

Benchmarking con Phoronix

Phoronix Test Suite es una solución de código abierto para el desarrollo comunitario de bechmark, su gestión y ejecución en entornos Linux.

El alumno debe ser capaz de Instalar la aplicación Phoronix Suite en las dos distribuciones Linux empleadas en prácticas y realizar las siguientes tareas:

- Emplear Phoronix para consultar la información del sistema y los sensores hardware disponibles.
- Consultar las Suites y Test disponibles sin instalarlos.
- Instalar y ejecutar Suites y Test individuales.
- Consultar los resultados históricos de la ejecución de test.

Para demostrar los conocimientos adquiridos, el alumno debe elegir un test de memoria o cpu, ser capaz de explicar el objetivo del test, ejecutarlo y poner de manifiesto la carga del sistema empleando la herramienta top vista en la Práctica 3.

Referencias:

- Web del poyecto Phoronix: https://www.phoronix-test-suite.com
    - Consultar la documentación disponble en PDF: https://www.phoronix-test-suite.com/documentation/phoronix-test-suite.pdf

Apache Benchmark

Apache Benchmark o “ab” es una herramienta distribuida junto con el servidor http Apache Http Server para realizar simulaciones de cargar HTTP.

La gran difusión del servidor Http de Apache ha facilitado que ab se encuentre disponible para la mayor parte de los sistemas operativos. Se trata de una herramienta muy sencilla que basa su éxito, precisamente, en su facilidad de uso. Permite simular cargas de trabajo simples de forma inmediata por línea de comandos.

Ab puede combinarse con scripting para realizar pruebas más sofisticadas, aunque para estos casos de uso, son más adecuadas las soluciones especializadas como Jmeter que veremos más adelante.

La alunna debe ser capaz de:

- Definir una carga de trabajo Http empleando las opciones básicas de línea de línea de comandos de ab: Concurrencia y Número de peticiones.

- Interpretar la información y métricas resultado de la ejecución de ab.
- Ejcutará la prueba de carga desde cualquiera de los servidores disponibles (VM Ubumtu, VM CentOS, Host) contra un servidor Http ejecutándose en un servidor distinto al que corre ab.

Referencias:

- Documentación de AB: https://httpd.apache.org/docs/2.4/programs/ab.html

JMeter

Las soluciones de test de carga permite reproducir cargas de trabajo sobre los sistemas a testar y así evaluar su capacidad para soportar las cargas esperadas en producción. Las cargas de trabajo pueden ser reales (recogidas del comportamiento de los usuarios en producción), sintéticas (generadas artificialmente) o una mezcla de ambas. Por ejemplo, es muy común tomar una carga real de trabajo y multiplicarla por un factor dado para simular la demanda futura esperada.

Los test de carga son fundamentales para los desarrolladores y operadores de sistemas. Les permiten acotar los riesgos del paso a producción de nuevos servicios o estimar la capacidad de los actuales para soportar la demanda futura esperada. Variables como throughput y latencia son métricas habitualmente objetivo de estos tests.

Existen numerosas soluciones en el mercado para la realización de test de carga. Algunas muy relevantes son Gatling y Locus.

En prácticas emplearemos Apache Jmeter. Se trata de una solución ampliamente consolidada en la industruia. Aun tratándose de una aplicación relativamente antigua, su amplia difusión y versatilidad hacen que se sigan demandando en los perfiles profesionales de DevOps y SysAdmin. Servicios en la nube como flood.io o loadview admiten Jmeter como formato de definición de pruebas de carga.

Jmeter es una aplicación Java por lo que requiere disponer de una máquina virtual java instalada (JVM). La forma más común de definir los test, y que emplearemos en clase, es con la consola gráfica. Para evitar tener que instalar un servidor de Xwindows en las Vms, vamos a ejecutar Jmeter en el ordenador anfitrión. Los pasos básicos son:

1. El alumno descargará e instalará una máquina virtual java 11 (si no dispone ya de ella): https://www.java.com/
2. Después descargará Jmeter desde un mirror cercano: https://jmeter.apache.org/download_jmeter.cgi
3. Seguirá el tutorial para realizar una prueba de carga básica: https://jmeter.apache.org/usermanual/build-web-test-plan.html

El alunmo debe conocer:

- Las opciones para lanzar Jmeter por línea de comandos con interfaz gráfica y sin ella (modo de ejecución de test).
- Test Plan y Variables de Usuario

- Thread Group: Significado de sus propiedades fundamentes “Thread Properties”.
- Sampler Http Rquest: Configuración de peticiones Get y Post. Access Log Sampler.
- Config Elements: Http Header Manager, Http Cookie Manager, Http Cache Manager y Http Request Defaults. Http Authorization Manager, CSV Data Set Config.
- Listenters: View Results Tree, Summary Report y Aggregate Report
- Post Processor Elements: Regular Expression Extractor.
- Timers.
- Assertions: JSON, Timer.

Referencias:

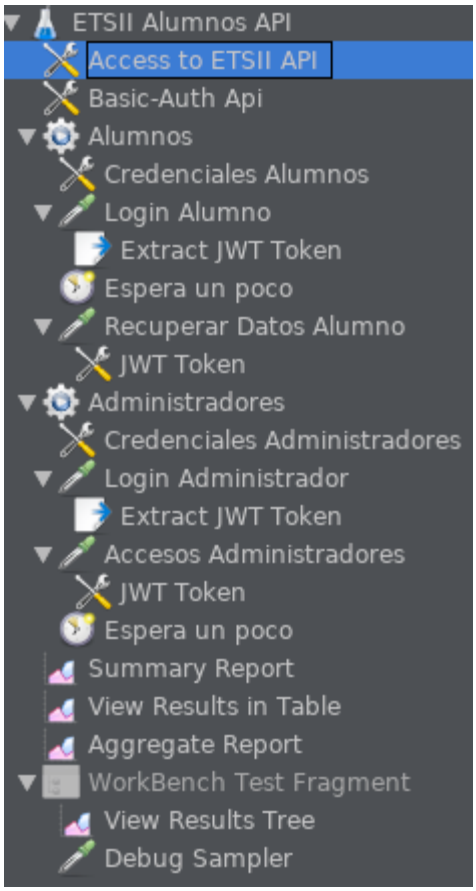
- Manual de Usuario de Jmeter: https://jmeter.apache.org/usermanual/index.html
    - Este mismo manual está disponible en la consola de Jmeter. Se recomienda el uso de la interfaz y documentación en inglés ya que los nombres de los elementos de test pueden tener traducciones inconsistentes a español.

Prueba de carga sobre API Rest

La alumna debe realizar un test de carga sobre la API Rest implementada por el servicio disponible en el repositorio GitHub: https://github.com/davidPalomar-ugr/iseP4JMeter.

En el anterior repositorio dispone de la documentación necesaria para instalar y ejecutar el servicio. Siguiendo las indicaciones disponibles en el repositorio de de GitHub la aluna debe crear un test de carga que simule el acceso de usuarios de tipo administrador y alumno a la consulta de expedientes.

La prueba de carga debe contener, al menos, los elementos de test que aparecen en la figura siguiente:



Indicaciones:

Parametrice el Host y el Puerto en el Test Plan (puede hacer referencia usando ${param})

Debe hacer dos grupos de hebras distintos para simular el acceso de los alumnos y los administradores. Las credenciales de alumno y administrador se cogen de los archivos: alumnos.csv y administrador.csv respectivamente.

El login de alumno, su consulta de datos (recuperar datos alumno) y login del adminsitrador son peticiones HTTP.

El muestreo para simular el acceso de los administradores lo debe coger el archivo apiAlumnos.log (usando un Acces Log Sampler).

Los tiempos de espera aleatorios se definen empleando un Gaussian random timer.

Para extraer los token JWT puede emplear un postprocesador de tipo Regular Expression Extractor.

El servidor de API Rest se ejecutará sobre una VM Ubuntu Server y la prueba de carga se ejecutará empleando línea de comandos (sin consola gráfica) desde el anfitrión.

El ejercicio práctico se entregará aportando una memoria descriptiva del proceso de resolución del ejercicio: pasos, problemas solucionados, resultados obtenidos y referencias empleadas. Esta memoria se adjuntará al plan de test (JMX) y los resultados (JTL).

La entrega del ejercicio se realizará antes del 16 de Enero y se valorará con un máximo de 2 puntos sobre la nota de la práctica 4. El examen de la práctica se valorará sobre 8.

Aunque no son objeto del ejercicio a desarrollar, la alumna debe entender como funciona:

- La ejecución del servidor con Docker y Docker Compose.
- El funcionamiento básico de los token JWT como mecanismo de autenticación.
- El uso que de curl se hace para probar el funcionamiento del servidor.
- El papel de los dos mecanismos de autenticación empleados en el servicio: basic y basado en token.

Referencias:

- Tutorial de Curl: https://curl.haxx.se/docs/httpscripting.html
- Guía de Instalación de Docker en Ubuntu 16: https://www.digitalocean.com/community/tutorials/como-instalar-y-usar-docker-en-ubuntu-16-04-es
- Tutoria Básico de Docker: https://docs.docker.com/get-started/

- JWT Documentación básica y consola de verificación de token online: https://jwt.io/introduction/
- REpresentantion State Transfer (REST): https://en.wikipedia.org/wiki/Representational_state_transfer