

Rend.Lab1 - Pruebas de rendimiento

Introducción

Los requisitos de rendimiento establecen los niveles de servicio de un sistema informático en términos de tiempos de respuesta y productividad, y uno de los principales retos a los que se enfrentan diariamente los departamentos de tecnologías de la información es garantizar su cumplimiento.

La evaluación del rendimiento proporciona técnicas, métodos y herramientas cuyo propósito es la valoración del rendimiento del sistema. El diagnóstico se realiza en base a la cuantificación de un conjunto de índices que permiten determinar si las características de rendimiento se ajustan o no a los niveles de servicio requeridos. Por lo tanto, la evaluación del rendimiento es el instrumento fundamental para poder conocer, ajustar y/o predecir los niveles de servicio, desde la perspectiva del rendimiento y la escalabilidad, que el sistema está proporcionando o los que es capaz de proporcionar en situaciones de incremento de la demanda. El conjunto de técnicas disponibles para llevar a cabo la evaluación del rendimiento de un sistema puede clasificarse en técnicas de medida o empíricas y técnicas de modelado.

En las técnicas de medida, la evaluación se realiza a partir de un conjunto de datos obtenidos del sistema mientras desarrolla su actividad. Si la medición es realizada mientras el sistema desarrolla su actividad normal, estamos ante procesos de monitorización de la actividad. Si, por el contrario, la medición se realiza en escenarios de pruebas controlados, estamos ante procesos de benchmarking y/o de pruebas de rendimiento (*Performance testing*)¹.

En general, y en el contexto de la ingeniería del rendimiento de sistemas informáticos, se denominan pruebas del rendimiento al conjunto de pruebas destinadas a evaluar el ajuste del sistema o de alguno de sus componentes a los requisitos de rendimiento establecidos. Este tipo de pruebas nos permite validar o determinar la velocidad, la capacidad de respuesta (*Responsiveness*), la productividad (*throughput*), la interoperabilidad y la escalabilidad de un sistema y/o aplicación bajo unas ciertas condiciones de carga de trabajo.

Las pruebas de rendimiento son un elemento importante para,

- La detección de cuellos de botella o fallos del sistema. Ajuste de sistemas.
- El análisis y análisis comparativo de sistemas o aplicaciones.
- Determinar el paso a producción de una aplicación o sistema.
- La comparación de características de rendimiento. Cambios o ajustes en los parámetros de un sistema o una aplicación.
- La identificación de los niveles de productividad de un sistema.

Dependiendo del tipo de estudio y del objetivo, se dispone de diferentes tipos de pruebas [5].

¹Es importante distinguir entre *Performance testing* que hace referencia al proceso de evaluación de las características no funcionales del sistema del término *Performance test* que es uno de los cuatro tipos de pruebas de rendimiento que se pueden llevar a cabo, como veremos en adelante. En este contexto puede aparecer un tercer término, *performance test* que haría referencia a una prueba concreta de rendimiento ejercitada sobre un sistema o aplicación

Performance test Este tipo de test está enfocado a determinar o validar la velocidad, escalabilidad y/o estabilidad de un sistema o aplicación.

Load test El objetivo de las pruebas de carga es verificar el comportamiento de un producto en situaciones de carga normales y de picos de carga.

Stress test Con este tipo de pruebas se persigue determinar y/o validar el comportamiento de un sistema o aplicación, su capacidad de respuesta, cuando se somete el sistema a cargas extremadamente elevadas, muy por encima de los patrones de utilización normales. Sirven para determinar si el sistema o aplicación pueden mantener un cierto nivel de efectividad bajo condiciones desfavorables.

Capacity test Por último, las pruebas de capacidad están dirigidas a determinar cuántos usuarios y/o transacciones es capaz de soportar un determinado sistema sin que los objetivos de rendimiento se vean comprometidos.

Dada la importancia del *Performance testing* en el análisis y la evaluación del rendimiento se propone una primera práctica de laboratorio en la que se desarrolle la planificación y ejecución de pruebas de rendimiento. Se proponen dos actividades, en la primera, las pruebas de rendimiento serán la base para la evaluación del rendimiento de un servidor web; en la segunda, las pruebas de rendimiento servirán como soporte al análisis del rendimiento de una sencilla aplicación web. En ambos casos el estudio estará enfocado al análisis de los tiempos de respuesta y los niveles de productividad en función de la intensidad de la carga.

El desarrollo de la práctica adoptará un procedimiento reducido y simplificado del proceso de pruebas de rendimiento. Se tendrá un primer contacto con el diseño no formal (no se aplicarán técnicas estadísticas de diseño de experimentos) de los casos de prueba, la ejecución de las pruebas, y la recogida y el análisis de los datos resultantes.

En ambas prácticas el entorno y el diseño de las pruebas están fijados. Dada la naturaleza y el objetivo de la práctica no se dispone de criterios de aceptación ni de niveles de servicio que cumplir. Por lo tanto, las tres primeras fases de todo procedimiento de pruebas de rendimiento: identificación del entorno de pruebas, identificación de los criterios de aceptación y planificación y diseño de las pruebas; no se llevarán a cabo. Para el desarrollo de las prácticas se dispone de herramientas de libre disposición que permiten, como mínimo, recoger datos acerca de los índices de rendimiento de interés para el sistema en estudio.

Práctica LRdto.PT1 - Evaluación del rendimiento de un servidor web. En esta primera parte de la práctica se evaluará la productividad del servidor web Apache instalado en la máquina virtual asignada al grupo. La evaluación se realizará mediante pruebas de carga.

Objetivos

- Familiarizarse con los aspectos básicos de prueba y medición del rendimiento de un sistema.
- Familiarizarse con los procedimientos y herramientas para la realización de pruebas de carga.
- Familiarizarse con una herramienta sencilla de benchmarking y el análisis de los resultados que proporciona.
- Obtener una primera aproximación a las capacidades de un servidor web.

Desarrollo de la práctica. Fases

Entorno o banco de pruebas Fase de adquisición del conocimiento preciso acerca de las configuraciones hardware, software y de comunicaciones del sistema a probar. ²

Sistema en estudio Máquina virtual de laboratorio asignada al grupo.

- Servidor: Hardware emulado.
- Sistema Operativo – Ubuntu Server 16.04 LTS
- Software servidor web: Apache

Actividades Las actividades a desarrollar en esta fase son:

- Utilización de los comandos Unix o herramientas de consulta de información del hardware y software del sistema para obtener la información precisa acerca de las características hard y soft del entorno de pruebas([4] cap. 2).
- Verificar que la opción *keep alive* está habilitada en el servidor Apache
- El grupo puede realizar ajustes de los parámetros del sistema operativo y del servidor web

Planificación y diseño de las pruebas Conocimiento de los patrones de utilización y diseño de los escenarios de utilización. En la práctica las condiciones del experimento y el conjunto de pruebas a ejercitar sobre el sistema serán los que se indican a continuación:

Condiciones de ejecución del experimento .

- La ejecución de los test se realizará con la máquina dedicada para evitar interferencias.
- Todos los test se ejecutarán con el servidor Apache en modo Keep-alive.
- El servidor se someterá a los niveles de intensidad de carga **IC** que se indican a continuación-

		Número de peticiones				
		1000	5000	10000	50000	100000
Clientes	50	IC1				
	100	IC2	IC3	IC5	IC7	IC10
Concurrentes	250		IC4	IC6	IC8	IC11
	500				IC9	IC12

- Para cada nivel de intensidad de carga, cada caso de prueba se ejecutará al menos 5 veces no consecutivas.

Casos de prueba Cada grupo desarrollará y desplegará en el servidor que tiene asignado los siguientes elementos de prueba, representativos de los tipos de carga de trabajo que ha de procesar un servidor Web.

- Carga de trabajo estática.
 - Página html que incluya texto y una imagen pequeña, menor de 50 Kb
 - Página html que incluya texto y una imagen grande, mayor de 250 Kb

²Aunque la red es un factor importante a la hora de evaluar el rendimiento obtenido por los clientes de un servidor web, debido a que las pruebas se ejecutarán en la red local del centro y, en teoría, se dispone de un ancho de banda suficiente, este aspecto no será tenido en cuenta a la hora de evaluar los resultados. Se supone que la red no constituye un cuello de botella.

- Carga de trabajo dinámica.
 - Página web dinámica ejecutando código: php, perl, python,

Actividades Las actividades a desarrollar en esta fase son:

- Desarrollo, despliegue y verificación del correcto funcionamiento de las páginas web indicadas.

Preparación del entorno de pruebas Configurar los recursos necesarios (hard, soft, red, herramientas, etc) para llevar a cabo los escenarios de prueba. Hay que instrumentar el entorno de pruebas para la monitorización de recursos y para favorecer un buen análisis de los resultados

Configuración .

- Cliente: Máquina física del laboratorio o PC portátil conectado a la red de la escuela.
- Servidor: Máquina virtual del grupo de prácticas.
- Herramientas:
 - Apache Benchmark (ab)[1] para la ejecución de las pruebas de carga.
 - Monitor sar (system activity reporter)[6], y utilidades como *vmstat* o *iotstat* si se desea monitorizar la actividad del servidor. En esta parte de la práctica no es obligatorio monitorizar el comportamiento del servidor.
 - Herramientas de análisis estadístico, de elaboración de gráficos y de tratamiento de textos para el análisis de los datos y la elaboración del informe final.

Preparar el plan de pruebas Dependiendo de la herramienta utilizada para llevar a cabo las pruebas puede ser necesario preparar y grabar el plan de pruebas en la propia herramienta. En este caso la ejecución de *ab* supone la ejecución de un comando unix al que se le pueden indicar opciones de ejecución y, por lo tanto, no es precisa una preparación previa. Puede ser conveniente elaborar un script que recoja el plan de pruebas completo.

Actividades Las actividades a desarrollar en esta fase son:

- Establecer los parámetros de ejecución de *ab* (Apache Benchmark). Se debe incluir la opción *KeepAlive* ya que el servidor está configurado con esta opción.
- Prepare un script que incluya las órdenes de Apache Benchmark que ejecutan las pruebas de carga.
- Prepare la recogida de los resultados de la ejecución de las pruebas en un fichero. Esto facilita el posterior análisis de los resultados y la elaboración del informe final.

Ejecutar los test Una vez preparado el entorno y el plan de prueba, y comprobado que se cumplen las condiciones de prueba establecidas, se ejecutan las pruebas. Es conveniente lanzar una primera prueba con carga ligera para cada uno de los test para ver que todo funciona correctamente.

Análisis Análisis de los resultados y elaboración del informe a partir de los datos recogidos en la ejecución de las pruebas. En este apartado, y a partir de los datos obtenidos, se analiza el comportamiento del servidor, tiempos de respuesta y productividad, en función del nivel de carga soportada por el servidor.

Actividades Las actividades a desarrollar en esta fase son:

- Análisis exploratorio de los datos. Tenga en cuenta que para los valores medios, y teniendo en cuenta que ha realizado 5 ensayos independientes, deberá proporcionar el intervalo de confianza de la media a un nivel de confianza del 95 %.
- Elaboración y análisis de las tablas y gráficas que describan el comportamiento del servidor en función de la carga.
- Elaboración de las conclusiones acerca de la capacidad del servidor en función de los resultados obtenidos y analizados.

Informe

Una vez finalizada la fase experimental elabore un informe en el que se recojan los detalles, resultados, análisis y conclusiones acerca de la capacidad teórica del servidor estudiado. Se elaborará un informe por grupo e incluirá las secciones que se indican a continuación. No obstante, si el grupo lo cree conveniente, se pueden añadir secciones y/o anexos.

1. Introducción.
2. Sistema en evaluación.
3. Resultados y análisis de los resultados.
4. Conclusiones
5. Bibliografía
6. Anexo I - Configuraciones (servidor, apache, ...)
7. Anexo II - Casos de prueba.
8. Anexo III - Tablas de Resultados. Para cada uno de los 4 casos de prueba incluya una tabla resumen con los resultados obtenidos para cada una de las intensidades de carga. A continuación se presenta un ejemplo de formato de tabla de presentación de datos. Puede utilizar uno alternativo siempre que se refleje toda la información.

Cuadro 2: Tiempo de Respuesta

	Caso de Prueba [Html texto— ...]						
	Ejec 1	Ejec 2	Ejec 3	Ejec 4	Ejec 5	Media	IC 95 %
IC1
IC2
...

El documento final ha de ajustarse a las normas de estilo elaboradas para las prácticas de laboratorio de esta asignatura.

Prueba de rendimiento de una aplicación web

Práctica LRdto.PT2 - Evaluación del rendimiento de una aplicación web. En esta segunda parte de la práctica se evaluará el rendimiento de una aplicación en ejecución. La evaluación del rendimiento se realizará mediante pruebas de carga. En este caso, al no tener que evaluar el cumplimiento de los requisitos de rendimiento o los niveles de servicio establecidos (*SLA*'s), el estudio se limitará a determinar los tiempos de respuesta y la productividad para los diferentes tipos de sesión (escenarios) bajo diferentes condiciones de intensidad de carga.

Objetivos

- Familiarizarse con los aspectos básicos de prueba y medición del rendimiento de una aplicación software.
- Familiarizarse con una herramienta de pruebas de rendimiento software y el análisis de los resultados que proporciona.

Desarrollo de la práctica. Fases

Entorno o banco de pruebas Fase de adquisición del conocimiento preciso acerca de la configuración del sistema en estudio.

Sistema en estudio Aplicación web *WebCalendar*[7].

- Servidor: Máquina virtual asignada al grupo.
- Sistema Operativo – Ubuntu Server 16.04 LTS
- Software servidor web: Apache
- Aplicación software: Sistema de agenda basada en web. La agenda contiene datos de prueba en diferentes días y diferentes tipos de actividad. La agenda está soportada por la aplicación web *WebCalendar*.

Actividades Las actividades a desarrollar en esta fase son:

- Si se ha desarrollado correctamente la primera parte de esta práctica, únicamente quedaría incluir en la descripción del sistema en estudio las características de la versión y configuración de *Php* y de *Webcalendar*.

Planificación y diseño de las pruebas Conocimiento de los patrones de utilización y diseño de los escenarios de utilización. Las condiciones del experimento y el conjunto de pruebas a ejercitar para esta práctica serán los que se indican a continuación:

Condiciones de ejecución del experimento .

- La ejecución de los test se realizará con la máquina dedicada para evitar interferencias.
- Todos los test se ejecutarán con el servidor Apache en modo Keep-alive.

Casos de prueba La carga de trabajo de un sistema web se caracteriza en términos de sesiones de usuario. Una sesión es un conjunto de peticiones de usuario-respuesta del sistema, encadenadas y relacionadas, con el fin de obtener un servicio del sistema. En este sentido, *WebCalendar* presenta la funcionalidad básica y típica de una aplicación de agenda: login, consulta, inserción de un nuevo apunte, edición de una entrada existente y borrado. Por razones de simplificación y dado que el objetivo de la práctica no es

hacer un estudio completo de rendimiento, solamente se van a considerar los escenarios de consulta e inserción de un nuevo apunte.

- Consulta de apunte. Desarrolle un escenario en el que se incluya la consulta de dos actividades programadas. Por ejemplo:
 1. El usuario consulta las actividades a realizar en una semana
 2. El sistema muestra la semana elegida
 3. El usuario selecciona un evento
 4. El sistema muestra los detalles del evento
 5. El usuario consulta cuándo tendrá lugar una actividad (ej:Tuthora)
 6. El sistema muestra todas las fechas en las que tiene lugar esa actividad
 7. El usuario selecciona *Home*
 8. El sistema muestra la página principal de la aplicación
- Modificar apunte. Desarrolle un escenario en el que se modifiquen los datos de alguno de los apuntes del calendario. Por ejemplo:
 1. El usuario se identifica (login) en el sistema.
 2. El sistema le presenta su calendario/agenda
 3. El usuario selecciona una semana
 4. El sistema muestra la semana elegida
 5. El usuario selecciona editar uno de los eventos de la semana elegida
 6. El sistema muestra el formulario de edición del evento
 7. El usuario modifica la información asociada al evento.
 8. El sistema muestra la semana con el evento modificado.
 9. El usuario sale del sistema (logout)
 10. El sistema muestra la página principal de la aplicación.

Actividades Las actividades a desarrollar en esta fase son:

- Definición de las sesiones de usuario que reproducen los escenarios de prueba. Utilice diagramas de actividad de UML2 o la técnica de descripción de escenarios de Casos de Uso - UML2 para documentar estos escenarios.

Preparación del entorno de pruebas Configurar los recursos necesarios (hard, soft, red, herramientas, etc) para llevar a cabo los escenarios de prueba establecidos. Hay que instrumentar el entorno de pruebas para la monitorización de recursos y para favorecer un buen análisis de los resultados

Configuración .

- Cliente: Máquina física del laboratorio o PC portátil conectado a la red de la escuela.
- Servidor: Máquina virtual del grupo de prácticas.
- Herramientas:
 - Apache JMeter [3] para la ejecución de las pruebas de carga.
 - Monitor sar (system activity reporter)[6] o el plugin PerfMon Server Agent [2] para monitorizar el rendimiento del servidor
 - Herramientas de análisis estadístico, de elaboración de gráficos y de tratamiento de textos para el análisis de los datos y la elaboración del informe final.

Preparar el plan de pruebas Dependiendo de la herramienta utilizada para llevar a cabo las pruebas puede ser necesario preparar y grabar el plan de pruebas en la propia herramienta. En el caso de JMeter es preciso grabar previamente el escenario que se puede almacenar en un fichero con extensión *jmx*, que podrá ser editado y utilizado tantas veces como sea preciso.

Actividades Las actividades a desarrollar en esta fase son:

- Grabar los escenarios definidos en la fase anterior utilizando la función de *Grabación de Sesiones* incorporada en la herramienta JMeter
- Configure el plan de pruebas obtenido a partir de la grabación de la sesión con JMeter.
 - Añadir controladores
 - Añadir receptores (Ver Resultados en Árbol, Informe Agregado, etc).
 - Añadir ruta y nombre del fichero en el que se recogerán los resultados de la prueba.

Ejecutar los test Una vez preparado el entorno y el plan de prueba, y comprobado que se cumplen las condiciones de prueba establecidas, se ejecutan las pruebas. Es conveniente lanzar una primera prueba con carga ligera para cada uno de los escenarios para ver que todo funciona correctamente.

Configuración . En el grupo de hilos o en el controlador definido en JMeter establecer valores para los siguientes parámetros.

- Número de hilos. Representa el conjunto de usuarios que se desea simular para la prueba.
- Periodo de subida (en segundos). Tiempo sobre el que se distribuye el comienzo de ejecución de los hilos.
- Contador de bucle. El número de veces que cada hilo ejecuta el test (el escenario completo)
- Duración(segundos) de la ejecución de la prueba.
- Retardo de arranque (segundos). Tiempo de demora antes de iniciar el test-

Actividades En esta fase se ejecutarán las pruebas para cada uno de los escenarios con los siguientes valores de los parámetros.

Consulta de apunte Este escenario será utilizado para someter al sistema a una prueba de carga que permita verificar el comportamiento de la instalación de *Webcalendar* en nuestro servidor web en una situación de carga normal. La variación en los parámetros del test servirá para familiarizarse con el funcionamiento y los diferentes resultados que se obtienen de la ejecución de los planes de prueba en función de los parámetros de JMeter. Los diferentes escenarios a ejecutar son los que se enumeran a continuación. En todos los casos el retardo de arranque se establecerá en 0 segundos (comienzo inmediato).

1. Número de hilos: 1; Contador de bucle: 150, Periodo de subida: 10 segundos
2. Número de hilos: 150; Contador de bucle: 1, Periodo de subida: 10 segundos
3. Número de hilos: 150; Contador de bucle: 1, Periodo de subida: 1 segundo
4. Número de hilos: 150; Contador de bucle: «Sin fin», Periodo de subida: 5 segundos, Duración: 45 segundos

Modificar apunte Este escenario será utilizado para someter al sistema a una prueba de capacidad. Permitirá analizar el comportamiento del sistema en función del

incremento de la demanda y determinar el nivel de demanda que es capaz de soportar el sistema sin que se degrade su rendimiento.

- Número de hilos: 10, 20, 40, 70, 130, 240, 440
- Periodo de subida. A elección del grupo.
- Contador de bucle. Dado que vamos a establecer un periodo de ejecución estacionaria de 10 minutos, el contador de bucle se establecerá en la opción «*Sin fin*»
- Duración (segundos): 600 + periodo de subida
- Retardo de arranque (segundos): 0 (comienzo inmediato)

Análisis Análisis de los resultados y elaboración del informe a partir de los datos recogidos en la ejecución de las pruebas. En este apartado, y a partir de los datos obtenidos, se analizarán tiempo de respuesta y productividad. El tiempo de respuesta se puede analizar considerando el escenario completo, o el comportamiento de alguna de las peticiones del escenario que supongan un mayor peso. Es importante completar el estudio con el análisis de comportamiento del servidor para detectar posibles cuellos de botella y puntos de saturación del sistema.

Actividades Las actividades a desarrollar en esta fase son:

- Análisis exploratorio de los datos. .
- Elaboración y análisis de las tablas y gráficas que describan el comportamiento de la aplicación en función de la carga. Tenga en cuenta que el propio JMeter proporciona gráficos y tablas que pueden ser aprovechados en el informe final.
- Elaboración de las conclusiones acerca del rendimiento de la aplicación y de los niveles de servicio que se pueden alcanzar con la configuración de sistema sometida a estudio.

Informe

Una vez finalizada la fase experimental y de análisis elabore un informe en el que se recojan los detalles, resultados, análisis y conclusiones acerca del rendimiento y los niveles de servicio de la aplicación en estudio. Complete la información anterior con el análisis de la utilización de recursos en función de los niveles de carga a los que se ha sometido el servidor. Se elaborará un informe por grupo e incluirá las secciones que se indican a continuación. No obstante, si el grupo lo cree conveniente, se pueden añadir secciones y/o anexos.

1. Introducción.
2. Sistema en evaluación.
3. Plan y diseño de las pruebas.
4. Resultados y Análisis de los resultados.
5. Conclusiones
6. Bibliografía
7. Anexo I - Configuraciones (servidor, apache, webcalendar,...)

8. Anexo II - Casos de prueba.

9. Anexo III - Tablas de Resultados. Incluya las tablas con los resultados de los test que crea necesarias como soporte al análisis y conclusiones incorporados en el cuerpo del informe.

El documento final ha de ajustarse a las normas de estilo elaboradas para las prácticas de laboratorio de esta asignatura.

Bibliografía

- [1] ab - apache HTTP server benchmarking tool - apache HTTP server version 2.4. [Online]. Available: <http://httpd.apache.org/docs/current/programs/ab.html>
- [2] Perfmon server agent. [Online]. Available: <https://jmeter-plugins.org/wiki/PerfMonAgent/>
- [3] Apache JMeter - apache JMeter™. [Online]. Available: <http://jmeter.apache.org/>
- [4] X. Molero, C. Juiz, and M. Rodeño, *Evaluación y Modelado del Rendimiento de los Sistemas Informáticos*. Pearson Educación S.A., 2004. [Online]. Available: <http://www.pearsoneducacion.com/molero/>
- [5] Meier, J.D., Farre, Carlos, Bansode, Prashant, Barber, Scott, and Rea, Dennis. Performance testing guidance for web applications. [Online]. Available: <https://msdn.microsoft.com/en-us/library/bb924375.aspx>
- [6] S. Godard. SYSSTAT. [Online]. Available: <http://sebastien.godard.pagesperso-orange.fr/>
- [7] WebCalendar. [Online]. Available: <https://sourceforge.net/projects/webcalendar/>