Pruebas de rendimiento de apliaciones Web con



Proyecto ASIR Miguel Ángel Ávila Ruiz







Índice de contenido

3
3
3
5
5
7
3
4
5
5
2
2
2
3
5
5
5
2
5
5
7
)
2
3



1. Introducción

Este documento se realiza para la entrega del proyecto integrado del ciclo superior de Administración de Sistemas informáticos en Red del centro I.E.S. Gonzalo Nazareno. En el se estudian varias herramientas, Apache JMeter e Hyperic que combinadas sirven para la realización de pruebas de rendimiento principalmente de aplicaciones web.

Con Apache JMeter se hará una carga personalizada de peticiones a la aplicación web, con Hyperic se monitorizará la maquina en la que esta desplegada dicha aplicación con lo que podremos observar si las características físicas son soportadas por dicha carga.

La finalidad de las pruebas que se realicen con este sistema entre otras podría ser la de poner a prueba la aplicación antes de su publicación para conocer si esta preparada para las peticiones que se prevean o bien para ir conociendo el rendimiento conforme se trata de mejorar el de alguna aplicación.

2. Apache JMeter

Apache JMeter es una herramienta diseñada para medir el rendimiento originalmente de aplicaciones Web pero además puede ser usado como una herramienta de pruebas para conexiones de bases de datos, FTP, LDAP, TCP, etc...

Se puede utilizar para simular una carga pesada en el servidor, grupo de servidores, la red o el objeto a probar su resistencia o para analizar el rendimiento general bajo diferentes tipos de carga. Se puede utilizar para hacer un análisis gráfico de rendimiento o para probar el comportamiento del servidor bajo una carga concurrentemente pesada.

Sus funcionalidades se pueden resumir en tres:

- **Diseñar** un **testplan** (plan de pruebas), en el que se genera un fichero ***.jmx.**
- Ejecutar un testplan .
- Ver las distintas formas los resultados de la ejecución de un testplan.

2.1 Instalación Apache JMeter

Para comenzar procederemos a instalar Java puesto que es necesario para el funcionamiento de la herramienta Apache JMeter:

Instalación de Java:

apt-get install openjdk-7-jre



Descargamos Apache JMeter y lo extraemos en el directorio deseado, lo propio sería hacerlo en /opt.



Posteriormente descargamos también el plugin **JmeterPlugins-Standard** que nos dará mas juego a la hora de realizar tests al permitirnos realizar peticiones mas complejas, y como pone en el fichero README descargamos el contenido en /opt/apache-jmeter-2.11/lib/ext/ para que JMeter cuente con este plugin.



Para poder ejecutar JMeter con el usuario con el que usaremos la herramienta habitualmente en mi caso "avila" le daremos propiedad:

chown -R avila:avila /opt/apache-jmeter-2.11

Para ejecutar JMeter teniendo en cuenta el **proxy** y el puerto que estamos usando para acceder a internet lo haremos de la siguiente manera:

avila@flamingos:/opt/apache-jmeter-2.11/bin\$./jmeter.sh -H 192.169.32.2 -P 8000

Si no es el caso para ejecutar JMeter de forma **normal** podemos ejecutarlo desde el directorio donde se encuentra:

avila@flamingos:/opt/apache-jmeter-2.11/bin\$./jmeter.sh

O podemos añadir el directorio en la **variable de entorno** para ejecutarlo desde cualquier ubicación:

echo 'export PATH=/opt/apache-jmeter-2.11/bin/:\$PATH' >> /home/avila/.bashrc



2.2 Realización de plan de pruebas

Un "plan de pruebas" es aquella configuración que define todos los aspectos de la prueba a realizar, como contra a qué se realizará, las peticiones que se harán y de qué modo, datos a introducir, definición de errores, que tipo de resultados obtendremos,... Los planes de pruebas generar ficheros de extensión .jmx.

En los siguientes puntos veremos como generar un plan de pruebas con las principales funcionalidades que JMeter cuenta.

2.2.1 Grupos de hilos

En un testplan (Plan de pruebas) lo habitual es añadir un "Grupo de Hilos" que representa el número de usuarios que ejecuta el test plan por ello es necesario que haya al menos uno en todo plan de pruebas. Cada uno de los hilos del grupo ejecutará el plan de forma completamente independiente a los demás hilos.

Lo que haremos será definir un grupo de hilos para indicar el numero de usuarios concurrentes y de repeticiones que se ejecutaran en el test de rendimiento:

👝 🏛 Plan d	e Druchae	8			_
Banco	Añadir	•	Hilos (Usuarios)	Grupo de Hilos	
(3E)	Pegar	Ctrl-V	Fragmento de Prueba	Montar grupo de Hilos	
	Resetear GUI		Elemento de Configuración	Tirar abajo grupo de Hilos	
	Abrir		Temporizador	jp@gc - Stepping Thread Group	
	Mozclar		Pre Procesadores	jp@gc - Ultimate Thread Group	lUsu
	Guardar selección como		Post Procesadores		
			Aserciones	· · · · · · · · · · · · · · · · · · ·	
	Guardar como imagen	Ctrl-G	Receptor	•	
	Guardar la pantalla como image	en Ctrl+Mayús-G		_	
	Habilitar				
	Deshabilitar				
	Toggle	Ctrl-T			
	Ayuda				
			-		



Después configuraremos el grupo de hilos, añadiendo una nueva linea en la que definiremos los parámetros:

- Start Threads Count: Los hilos con los que se van a comenzar a trabajar a darle peticiones.
- **Initial Delay, sec**: Los segundos que trascurrirán para que empiecen a cargarse los hilos tras arrancar la prueba.
- **Startup Time, sec**: El tiempo que tendrán los hilos para ir cargándose se incrementaran teniendo en cuenta el numero de hilos entre los segundos especificados.
- Hold Load For, sec: El tiempo que estarán todos los hilos activos.
- **Shutdown Time**: El tiempo que tardaran en ir cerrandose cada hilo en función de la cantidad.

Esta configuración simple y de poca carga se creará 10 clientes que se irán incrementando en un tiempo de 30 segundos y que se mantendrán haciendo peticiones 60 segundos y a partir de ahí irán decrementándose durante otros 10 segundos:





Entre otras cosas podemos hacer también que las peticiones en vez de crecer y decrecer tan solo una vez puedan comportarse de otra manera, en la siguiente configuración de grupos de hilos añadimos una nueva linea a la que que vez que se cargan todos los hilos de la primera linea se le suman los que especificamos de la segunda.

Así de este modo podremos predefinir una carga tan larga y variable como deseemos.



2.2.2 Grabación de la navegación

Para especificar en el plan de pruebas las peticiones a la aplicación web que se quiere someter a pruebas existe una forma automatizada que utiliza una herramienta que captura las peticiones HTTP que realiza el navegador, a medida que uno navega por la aplicación genera por cada una de estas peticiones HTTP un componente sampler en el árbol del testplan.

Existen dos herramientas que nos permiten generar de forma automática los samplers de un TestPlan:

- El componente HTTP Proxy Server de JMeter
- La utilidad **Badboy**.

El componente HTTP Proxy Server de JMeter graba los requests que son enviados el servidor pero no pueden ser grabados aquellos requests que utilicen **HTTPS**, ya que como es seguro el proxy no puede decodificar el trafico y grabar los parámetros que maneje el servidor.



Para realizar una grabación con el **HTTP Proxy Server de JMeter** en primer lugar configuramos el navegador desde donde vamos a realizar la simulación de la navegación a testear. En la configuración del proxy del navegador (en mi caso Iceweasel) definiremos la IP y el puerto desde donde se va a ir grabando lo que voy a ir realizando para definir el plan de pruebas es la siguiente:

Configuración de conexión							
Configurar proxies para el acceso a Internet							
○ Sin proxy							
\odot Autod <u>e</u> tectar configuración del proxy para esta red							
○ Usar la configuración <u>d</u> el proxy del sistema							
Configuración <u>manual</u> del proxy:							
Pro <u>xy</u> HTTP: 127.0.0.1 <u>P</u> uerto: 8765							
☑ U <u>s</u> ar el mismo proxy para todo							

En JMeter añadimos el Proxy para que reciba la navegación web desde el navegador y de este modo guarde los pasos realizados, para añadirlo:

ዮ 🏝 Plan de Prue ╦ jp@gc - L	bas Iltimate Thread Group	Banco de Tr	abajo	
🛱 Banco de Tra	oice	Nombre: Banco	de Trabaio	
	Añadir	•	Elementos NoDePrueba	Servidor espejo HTTP
	Pegar	Ctrl-V	Controlador Lógico	Servidor Proxy HTTP
	Resetear GUI		Elemento de Configuración)	Visualizador de propiedades
	Abrir		Temporizador 0	
	Mezclar		Pre Procesadores	
	Guardar selección como		Muestreador 0	
	Save as Test Fragment		Post Procesadores	
			Aserciones	
	Guardar como imagen	Ctrl-G	Receptor	
	Guardar la pantalla como ima	igen Ctrl+Mayús-G		
	Habilitar			
	Deshabilitar			
	Toggle	Ctrl-T		
	Ayuda			

Configuramos el Servidor Proxy HTTP, le podemos poner nombre si es conveniente, asignaremos el puerto en el que estará escuchando al navegador en mi caso "8765" que no esta cogido por ninguna otra herramienta, en el Controlador de Objeto elegimos "Plan de Pruebas > jp@gc – Ultimate Thread Group" y en el Tipo de "Parámetros muestra HTTP" seleccionamos "HttpClient4" Por ultimo en "URL Patrones a Excluir" podemos añadir por medio de una expresión regular los HTTP Request que no son relevantes para las pruebas y que por lo tanto no queremos grabar, en este caso excluimos las HTTP Request con extensión *.ico .



ar Cabeceras HTTP	🗌 Añad
🗌 Recuperar Todo	s los Re
rd	
jgested Excludes	
	ar Cabeceras HTTP Recuperar Todo d gested Excludes

Una vez arrancado el proxy abrimos el navegador para realizar los pasos que queremos que se graben en JMeter para que posteriormente pueda ser realizado pero de forma automática. En cuanto terminemos de realizar estos pasos con el navegador pulsaremos en el apartado del proxy de JMeter en "Parar" para que no siga grabando.

En este caso utilizo una aplicación web que en la primera pantalla hay que picar en un botón para ver una lista de clientes en esta pantalla podremos pulsar un botón para añadir uno, en la siguiente pantalla añadiremos los campos y pulsaremos en "enviar".

En JMeter se nos registrará los pasos realizados:





Si queremos grabar la navegación de una pagina **HTTPS** tendremos que usar **BadBoy** para poder incluirla en el plan de pruebas. Badboy solo está desarrollado para Windows y con Wine tampoco podremos ejecutarlo desde Linux así que tendremos que usar la aplicación en Windows.

Para descargar Badboy lo haremos desde la página oficial <u>http://www.badboy.com.au/download/add</u> Durante la instalación simplemente hay que aceptar los términos y elegir la ubicación de la instalación.

Una vez instalada copiamos la URL de la página HTTPS que queremos grabar, ejecutamos Badboy y pegamos la URL en la parte superior como si de un navegador se tratase.





Desde la ventana derecha realizaremos la navegación a grabar. A la izquierda en la ventana "Script" podemos ver las peticiones que se están grabando



Una vez realizada la grabación exportaremos el plan de pruebas que podremos importar en Apache JMeter:







Esto nos generará un fichero jmx, ahora solo queda abrirlo con JMeter que por linea de comandos sería del siguiente modo:

jmeter -t Script.jmx

Al importar el fichero generado con Badboy podemos observar como ya contamos con las peticiones HTTPS y podemos trabajar con ellas:





2.2.3 Controladores

Los controladores nos permiten entre otras cosas organizar y detallar las peticiones del TestPlan permitiéndonos añadir dentro de ese controlador distintos elementos como por ejemplo temporizadores. Para añadir un indicie haremos un clic derecho sobre el Grupo de Hilos y añadimos lo siguiente:

] 🕼 👜 🌔		- 🍫 🕨	• 🗞 💿 😣 💺		ا الله الله الله	۵ 🏷 🗐 👔
9	Plan de Pruebas						
	👇 ّ jp@gc - Ultir	nate Thread Group	ge oninte	Controlador Lágico	, 	Conmutor Co	stealadae
	🗠 🧪 /Manteni	Anadir		Controlador Logico		Conmutar Con	
	🗠 🧪 /Manteni	Cortar	Ctrl-X	Elemento de Configu	iracion 🕨	Controlador A	leatorio
	🗣 🥒 /Manteni	Copiar	Ctrl-C	Temporizador	•	Controlador E	lucle
	🔶 🥒 /Manteni	Pegar	Ctrl-V	Pre Procesadores	•	Controlador o	le Módulo
	🔶 🥒 /Manteni	Duplicate	Ctrl+Mayús-C	Muestreador	•	Controlador F	orEach
	🔶 🥒 /Manteni	Resetear GUI		Post Procesadores	•	Controlador (Grabación
0	🗐 Banco de Traba	Borrar	Suprimir	Aserciones	•	Controlador I	F
Ľ	🗒 🗑 Servidor Pro	Abrir		Receptor	•	Controlador I	nterleave
	÷	Mozdar		count initiari	Delay, se	Controlador (Only Once
		Guardas calección como				Controlador (Orden Aleatorio
		Guardar selección como		-		Controlador S	imple
		Guardar como imagen	Ctrl-G			Controlador T	hroughput
		Guardar la pantalla como imagen	Ctrl+Mayús-G			Controlador T	iempo de Ejecución
		Habilitar				Controlador T	ransaction
		Deshabilitar				Controlador W	Vhile
		Toggle	Ctrl-T			Incluir Contro	lador
1							- B

Una vez que añadamos el controlador simple le daremos un nombre que identifique a las peticiones que englobe y meteremos dentro de el los registros oportunos, quedando algo así:





2.2.4 Contadores

Cuando uno de los pasos se trata de un formulario a rellenar y no podemos introducir valores constantes o bien no deseamos que siempre se registren los mismos valores nos podemos servir con un Contador sobre el Controlador simple deseado:

Plan de Pruebas	Contro	olador Simple	
Grupo de Hilos	Nombre	Bellenamos y enviamos formul	ario
Accediendo	Comont	aries	
/MantenimientoClientes	Coment	anos	
/MantenimientoClientes			
Accede al lietade			
Mantanimianta Cliantan /Cli	anta Condot? accion		
/MantenimientoClientes/clie	antel et ion		
Diagnage on affedin nueve alien	ta la		
Mantanimianta Cliantas (alia	Le la		
Mantenimientoclientes/cite	entemnt.jsp		_
Añadir	•	Controlador Lógico	•
🖉 🦉 🛺 Insertar Padre	Þ	Elemento de Configuración)	Configuración de la Conexión JDBC
Change Controller	•	Temporizador 0	Configuración de Muestreador TCP
	Ctrl-X	Pre Procesadores	Configuración del CSV Data Set
	Ctrl-C	Muestreador	Contador
Banco do Tro Pegar	Ctrl-V	Post Procesadores	Elemento de Configuración de Login
	Ctrl+Mayús-C	Aserciones	Elemento de Configuración Simple
Resetear GUI		Receptor	Gestionador de la Caché HTTP
Borrar	Suprimir		Gestor de Autorización HTTP
Abrir		-	Gestor de Cabecera HTTP
Mezclar			Gestor de Cookies HTTP
Guardar selección	como		Keystore Configuration
Save as Test Frage	como		MongoDB Source Config
Save as rest riagi	ient	-	Valores por defecto para petición FTP
Guardar como imag	jen Ctrl-G		Valores por Defecto para Petición HTTP
Guardar la pantalla	a como imagen Ctrl+Mayús-G	-	Valores por defecto para Petición Java
Habilitar			Valores por defecto Petición Extendidad LDAP
Deshabilitar			Valores por defecto Petición LDAP
Toggle	Ctrl-T		Variable aleatoria
Ayuda			Variables definidas por el Usuario
		-	

Por un lado observamos como en la ruta que corresponde al formulario van los valores a introducir, en ellos podemos poner el nombre de la variable del contador para que se vaya incrementando según especifiquemos con el fin de introducir distintos valores no repetitivos. La variable se añadirá del siguiente modo: \${nombrevariable}:

Plan de Pruebas	Petición HTTP							
🗣 🏠 Grupo de Hilos	Nombre: //MantenimientoClientes/ClienteServlet?accion=insertar&ruc=1&razSocial=12&direccion=13&telefono=14							
🐺 👹 Accediendo	ComentariosDetected the start of a redirect chain							
∽ 🧪 /MantenimientoClientes	Servidor Web		Timeout (milis					
- // MantenimientoClientes/	Nombre de Servidor o IP: 192.168.205.164	Puerto: 8080	Conexión:					
 Accede al listado /MantenimientoClientes/ClienteServie 	Petición HTTP							
🖙 🧪 /MantenimientoClientes/clienteLst.js;	Implementación HTTP: 🗾 👻 Protocolo: http	Método: POST < Codificacíon del contenido:	UTF-8					
 Picamos en añadir nuevo cliente 	Ruta: [lientes/ClienteServlet?accion=insertar&ruc=ruc\${mivariable}&r	razSocial= radsocial\$ {mivariable}&direccion=direccion\$ {m	ivariable}&telefo					
👇 👹 Rellenamos y enviamos formulario	🗌 Redirigir Automáticamente 📝 Seguir Redirecciones 📝 Utilizar Kee	epAlive 🔲 Usar 'multipart/form-data' para HTTP POST 🔲	Cabeceras compa					
	Parameters Body Data							
🗣 🧪 /MantenimientoClientes/clienteLst.js;	Envi	ar Parámetros Con la Petición:						
— 🔄 Informe Agregado	Nombre:	Valor	έCo					
🔤 Ver Árbol de Resultados			· · · · · · · · · · · · · · · · · · ·					
🗕 🏢 Banco de Trabajo								



Por otro lado en el contador definiremos varios elementos como:

- **Nombre**: Podremos un nombre a fin de identificar el contador.
- Arrancar: Aquí pondremos el valor variable por el que se va a iniciar.
- **Incrementar**: Se especifica el incremento del valor variable.
- Nombre de Referencia: Nombre de la variable:
- **Contador independiente para cada usuario**: Desactivar para que la variable se tenga en cuenta en todos los clientes por igual y no se incremente de forma independiente.

Image: Second	Contador Nombre: ContadorFormulario Comentarios
---	---

2.3.5 Temporizador

Para no ahogar al servidor y para que la realización de las pruebas tenga un comportamiento mas real podemos añadir temporizadores que hace que las peticiones no sean continuas y se tomen un tiempo allá en el punto que la coloquemos de la grabación. Para añadir este elemento:

🔶 💓 Picamos en a	añadir nuevo cliente			_
🦾 🧪 /Mantenir	Añadir	•	Controlador Lógico	•
🗣 🛞 Visualización	Insertar Padre	•	Elemento de Configuración	•
- 🚟 Contador	Change Controller	•	Temporizador	BeanShell Timer
🗠 🧪 /Mantenir	Cortar	Ctrl-X	Pre Procesadores	BSF Timer
🔶 🥒 /Mantenir	Copiar	Ctrl-C	Muestreador	jp@gc - Throughput Shaping Timer
🚟 Gestor d	Pegar	Ctrl-V	Post Procesadores	JSR223 Timer
🗕 🏢 Banco de Trabajo	Duplicate	Ctrl+Mayús-C	Aserciones	Poisson Random Timer
	Resetear GUI		Receptor	Temporizador Aleatorio Gaussiano
	Borrar	Suprimir		Temporizador Aleatorio Uniforme
	Abrir			Temporizador Constante
	Mezclar			Temporizador de Rendimiento Constante
	Guardar selección como			Temporizador sincronizado
	Save as Test Fragment			
	Guardar como imagen	Ctrl-G]	
	Guardar la pantalla como imagen	Ctrl+Mayús-G		
	Habilitar			
	Deshabilitar			
	Toggle	Ctrl-T		
	Ayuda			



En este caso el temporizador lo hemos puesto en el controlador que corresponde la inserción de datos para simular el tiempo que se lleva el usuario rellenándolo por ello le damos en el campo "Retardo de Hilo" un tiempo de 1200 milisegundos que equivalen a 20 segundos:

🔶 💓 Picamos en	añadir nuevo cliente			_
🔶 🧪 /Mantenir	Añadir	۱.	Controlador Lógico 🔹 🕨	
👇 🛞 Visualización	Insertar Padre	۱.	Elemento de Configuración 🕨	
— 🚟 Contador	Change Controller	Þ	Temporizador 🕨	BeanShell Timer
🗠 🥕 /Mantenir	Cortar	Ctrl-X	Pre Procesadores	BSF Timer
🗢 🧪 /Mantenir	Copiar	Ctrl-C	Muestreador 🕨	jp@gc - Throughput Shaping Timer
— 🚟 Gestor d	Pegar	Ctrl-V	Post Procesadores	JSR223 Timer
— 🏢 Banco de Trabajo	Duplicate	Ctrl+Mayús-C	Aserciones 🕨	Poisson Random Timer
	Resetear GUI		Receptor 🕨	Temporizador Aleatorio Gaussiano
	Borrar	Suprimir		Temporizador Aleatorio Uniforme
	Abrir		1	Temporizador Constante
	Mezclar			Temporizador de Rendimiento Constante
	Guardar selección como			Temporizador sincronizado

2.2.6 Receptores, obtención de resultados

JMeter dispone de gran variedad de salida de datos para analizar las medidas. Existen numerosos Receptores que nos permiten ver los resultados en distintos formatos o enviarlos hacia algún destino en fichero.

En el grupo de hilos también añadiremos un "**Informe Agregado**" que es el receptor mas comun. Esto nos creará un informe con los resultados de la prueba. Hay otros muchos receptores que incluyen gráficas, detalles de la petición de repuesta, etc.

۴ 🛱	Plan de F	Pruebas	jp@gc - Ult	imate Thread Group	
۹ –	In Ineg	Añadir	•	Controlador Lógico	.]
		Cortar	Ctrl-X	Elemento de Configuración)	
	-	Copiar	Ctrl-C	Temporizador 0	
	- 1 A	Pegar	Ctrl-V	Pre Procesadores	lestreador
	-1	Duplicate	Ctrl+Mayús-C	Muestreador 0	estreador
	01	Resetear GUI		Post Procesadores	r siguiente iteración 🔾 Parar Hilo 🔾 Parar T
	- 🛞 Pi	Borrar	Suprimir	Aserciones 0	BeanShell Listener
	0-1	Abrir		Receptor 0	BSF Listener
	9- 🛞 R	Mezdar		ads count Initial Dela	Escritor de Datos Simple
		Guardar selección como		0	Generar Resumen de Resultados
	~ /	Guardar como imagon	chi c		Gráfico
	~ /	Guardar la pantalla como imagen	Ctri-G		Gráfico de Distribución (alfa)
	Banco de	Guardar la pantalla como image	en Ctri+Mayus-G		Gráfico de Resultados
		Habilitar			Guardar respuestas en archivo
		Deshabilitar			Informe Agregado
		Toggle	Ctrl-T		jp@gc - Active Threads Over Time
		Ayuda		Add Row	jp@gc - Composite Graph
			Expected p	arallel users count	jp@gc - Console Status Logger



También disponemos de la posibilidad de ver los resultados obtenidos en una gráfica a la que podemos configurar teniendo una vista mas clara y rápida de lo que queramos tener en cuenta o comparar. Añadiremos este receptor de la siguiente manear

r 蘽	Plan	de Pruebas	ip@ac -	Ultimate Thre	ad Group	
9	ni 🖵	Añadir	•	Controlador Lóg	jico	•
		Cortar	Ctrl-X	Elemento de Co	nfiguración	
	Ĭ	Copiar	Ctrl-C	Temporizador		•
	- 1	Pegar	Ctrl-V	Pre Procesador	es	
		Duplicate	Ctrl+Mayús-C	Muestreador		Muestreador
		Resetear GUI		Post Procesado	res	🕨 nzar siguiente iteración 🛛 Parar Hilo 🔾 Para
	- 11	Borrar	Suprimir	Aserciones		
	6	Abrir		Receptor		BeanShell Listener
	- 1	f Mezclar		Threads Count	Initial	BSF Listener
		Guardar selección como			0	Escritor de Datos Simple
		Guardar como imagen	Ctrl-G	1		Generar Resumen de Resultados
		Guardar la pantalla como image	n Ctrl+Mavús-G			Gráfico
	- 5	Habilitar		-		Gráfico de Distribución (alfa)
		Deshabilitar				Grafico de Resultados
	Banc		CHUT			Guardar respuestas en archivo
	,	Toggie	Curri			Informe Agregado
		Ayuda		Add Row	/	Jp@gc - Active Threads Over Time
			- Dynasty	ad parallal uppra a		jp@gc - Composite Graph
			Expecte	ed parallel users co	punt	Jp@gc - Console Status Logger
			10			jp@gc - Flexible File Writer
			200			jp@gc - Hits per Second

Otra gráfica que podemos obtener mas especifica para los tiempos de respuestas es la de "**Response Time Graph**" para ello añadiremos el receptor de la siguiente manera:

ዮ 🍈 P	lan de Prue	bas	ip@gc - Ultimate	Thread Group	
<u>۹</u>	[]p@gc -	Añadir	•	Controlador Lógico	·
9	- 🔘 Acce	Cortar	Ctri-X	Elemento de Configuración	•
		Copiar	Ctrl-C	Temporizador	•
	· · · · ·	Pegar	CtrI-V	Pre Procesadores	•
Ĭ.	Acce	Duplicate	Ctrl+Mayús-C	Muestreador	ador
	15	Resetear GUI		Post Procesadores	guiente iteración 🔾 Parar Hilo 🔾 Parar Test
	- C Picar	Borrar	Suprimir	Aserciones	
	- 1 IN	Abrir		Receptor	BeanShell Listener
	- 🕼 Relle	Mezclar		unt Initial Delay, see	BSF Listener
	- 40 C	Guardar selección com	0	0	Escritor de Datos Simple
	~ D IN	Guardar como imagen	Ctrl-G		Generar Resumen de Resultados
	~ / IN	Guardar la pantalla co	mo imagen Ctrl+Mayús-G		Gráfice de Distribución (elfe)
— 🗿 в	anco de Tr	Habilitar			Gráfico de Distribución (aira) Gráfico de Resultados
0.080		Deshabilitar			Guardar respuestas en archivo
		Toggle	Ctrl-T		Informe Agregado
		Avuda		dd Bow	ip@gc - Active Threads Over Time
	I	.,,		da kow	jp@gc - Composite Graph
			Expected parallel u	isers count	jp@gc - Console Status Logger
			20		jp@gc - Flexible File Writer
					jp@gc - Hits per Second
			10		jp@gc - Loadosophia.org Uploader
			10		jp@gc - PerfMon Metrics Collector
				· ۳	jp@gc - Response Times Over Time
			16	P	jp@gc - Transactions per Second
					JSR223 Listener
			14		Reporte resumen
			14	[Response Time Graph
			adi	4	Resultados de la Aserción
			6		Decultadas dal Manitas



Otro receptor que se suele usar sobre todo al crear los planes de prueba es el de "**Ver Árbol de Resultados**" que nos permite observar le resultado de cada uno de los requests, indicando el número de peticiones ("muestra"), en el momento de inicio de la petición o controlador "Etiqueta", el tiempo de respuesta, el resultado de la petición y del tamaño en bytes de petición:



Una vez que hemos configurado el plan de prueba podemos guardarlo por si queremos seguir configurándolo en otro momento o por si deseamos tenerlo disponible para otras ocasiones. Lo que guardaremos será un fichero con extensión .jmx. Se salvaría del siguiente modo:

Archivo	<u>E</u> ditar	Search	Lanza <u>r</u>	<u>O</u> pciones
<u>C</u> errar			С	trl-L
Abrir			С	trl-0
<u>T</u> emplat	es			
<u>M</u> ezclar				
Guarda	r		С	trl-S
Gu <u>a</u> rdai	r Plan de	e Prueba	s como c	trl+Mayús-S
Guarda	r selecci	ón como.		
Save as	Test Fra	agment		
<u>R</u> evertir				
<u>1</u> delvie	rnes.jm	ĸ		
<u>2</u> delvie	rnes.jm	¢		
<u>3</u> Servid	or Proxy	/ HTTP.jm	х	
<u>4</u> Ver Ár	bol de R	esultado	s.jmx	
<u>5</u> Proxy	JMeter.j	mx		
Salir			С	trl-Q
	-		reo en orrecoo	remonico coc.joj



Por ultimo ya terminado de configurar el plan de pruebas queda comenzarlo, para ello pulsamos en el icono de arrancar que se encuentra en la parte superior:

delvie	rnes.jmx (/home/avila/Escrit	orio/delviernes.jmx) – Apa	che JMeter (2.11 r155454	8)			o x
Archivo <u>E</u> ditar Search Lanza <u>r</u> Opciones	Ayuda						
	+ - 🔸 🕨 🛛		🧃 🎬 🎮 🏷 🗐	?	0	<u>A</u> (0 / 20
ዮ- 🍰 Plan de Pruebas ዮ- ፹͡ ʃp@gc - Ultimate Thread Group	jp@gc - Ultimate Thre	ead Group					
👇 👹 Accediendo	Nombre: jp@gc - Ultimate Th	nread Group					
🗠 🥕 /MantenimientoClientes	Comentarios						
- 🥕 /MantenimientoClientes/	Help on this plugin					`	v1.1.3
👇 👹 Accede al listado	Acción a tomar después d	le un error de Muestreado	or				
 // /MantenimientoClientes/ClienteServic // /MantenimientoClientes/clienteLst.js; 	Contin	iuar 🔾 Comenzar siguien	te iteración 🛛 🌣 Parar Hilo	🔾 Parar Test 🛛 🔾 Parar t	est ahora		
- Dicamos en añadir nuevo cliente	Threads Schedule				1		
 Bellenamos y enviamos formulario 	Start Threads Count	Initial Delay, sec	Startup Time, sec	Hold Load For, sec	Shutdown	Time	
ContadorFormulario	10	0	30	20	10		
MantenimientoClientes/ClienteServie							
- 🧷 /MantenimientoClientes/clienteLst.ist							
Banco de Trabajo							

Si lo hacemos mas de una vez y queremos limpiar todos los resultados obtenidos en los receptores pulsaremos el icono en el que aparecen un par de escobas situado en la parte superior de la aplicación:

Archivo Editar Search Lanzar Ope	iones Ayuda		
) 🖸 💠 🗕 🍫 🕨 💌 👟 🗞 🖋 🌌 🎮 🍆 📰	0 🔔	0 / 20 🗆
ዮ 🚨 Plan de Pruebas	Gráfico		

Tras la realización de la prueba obtenemos los resultados en los receptores (listeners) que hemos añadido, observamos el "**Informe agregado**":

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/MantenimientoClientes	1885	128	64	311	5	2442	0,00%	4,9/sec	5,5
/MantenimientoClientes/ClienteServlet?accion=buscar	1884	628	510	1286	47	3741	0,00%	4,9/sec	14,6
/MantenimientoClientes/clienteMnt.jsp	1878	84	38	199	3	1861	0,00%	4,9/sec	16,0
/MantenimientoClientes/ClienteServlet?accion=insertar&ruc	1878	1138	973	2279	91	4370	0,00%	4,9/sec	14,6
Total	7525	494	191	1444	3	4370	0.00%	19.4/sec	50.7

Nos saldrá una tabla en la que tenemos una fila para cada petición realizada para la que se nos detallarán los siguientes valores siendo milisegundos aquellos en los que se expresa en tiempo :

- Muestras: Cantidad de repeticiones de cada petición.
- Media: Media de tiempo invertido por una petición.
- Mediana: Significa que el 50% de las muestras tardaron menos del valor reflejado.
- Linea de 90%: Tiempo de respuesta en el que se encuentra el 90% de los requests.
- **Min**: Es el mínimo de tiempo invertido por una petición.
- **Max**: Es el máximo de tiempo invertido por una petición.
- %**Error**: Porcentaje de la petición fallida.
- Rendimiento: Numero de repeticiones por segundo.
- Kb/sec: Cantidad de datos procesados por segundos.



En los ajustes del receptor "**Gráfico**" podemos elegir que medidas queremos que se nos muestre gráficamente, en este caso marcamos todas:

Settings Graph
Mostrar gráfico Guardar gráfico Guardar la tabla de datos 🗌 Guardar la cabecera de la tabla
Column settings Columns to display: 🖉 Media 💶 🖉 Mediana 💶 🖉 Linea de 90% 💶 🖉 Mín 🔤 🖉 Máx 💶 Foreground color Value font: Serif 🛛 Size: 10 🗣 Style: Normal 🗣 🖉 Draw outlines bar? 🖉 Show number grouping? 🖉 Value labels vertical?
Column label selection: Apply filter Sensible a mayúsculas 🗹 Expresión regular
Title Título del gráfico Synchronize with name Font: Serif ▼ Size: 16 ▼
Graph size ☑ Dynamic graph size Anchura Altura
X Axis Y Axis (milli-seconds) Longitud máxima de la etiqueta del eje x Scale maximum value:
Legend Placement: Bottom ▼ Font: Serif ▼ Size: 10 ▼ Style: Normal ▼

Para observar la gráfica con las medidas a mostrar seleccionadas pulsamos en la pestaña "Graph":

Observamos que la petición que mas alto tiene los tiempos estadísticos es la que se realiza cuando se envía el formulario cumplimentado. Por el contrario la que menos tiempo ha supuesto es aquella que nos lleva a rellenar el formulario.





Observamos la salida del receptor "**Response Time Graph**" en la que podemos ver una gráfica de lineas que muestra la evolución del tiempo medio de repuesta durante la prueba para cada petición.

En este ejemplo vemos que a las 08:20:30 muy cerca del principio de la prueba es cuando mas han tardado las 4 peticiones en responder pues es el momento en el que mas hilos establecimos y que a partir de ahí se ha respondido mas rápido surgiendo dos pequeños picos mas adelante en a las 08:22:40 y a las 08:24:30. También podemos observar que la petición que mas tarda en responder es la que envía los datos del formulario mientras la que menos con gran diferencia es la que nos lleva a rellenar el formulario.



Miguel Ángel Ávila Ruiz

2.3 Inyectores

En este apartado veremos como arrancar peticiones desde distintas maquinas para conseguir con ello una prueba mas contundente o real.

Crearemos una plataforma de clientes para lanzar las pruebas, se requieren al menos de dos equipos independientes.

Estas maquinas tendrán Debian como sistema operativo y de identificarán así:

- Nombre: cliente1, IP: 192.168.205.191
- Nombre: cliente2, IP: 192.168.205.192

2.3.1 Instalación de clientes

Instalaremos entonces en cada maquina JMeter, para ello previamente instalaremos Java:

apt-get install openjdk-7-jdk

Y posteriormente instalamos JMeter de la misma manera que hicimos con el servidor.

2.3.2 Configuración de los clientes

Para conseguir el rendimiento necesario en muchas ocasiones se requiere al menos dos equipos clientes con suficientes recursos.

Para ello vamos a configurar dos instancias de JMeter que inyectarán las peticiones (cliente1 y cliente2) y la maquina con la que ya hemos trabajado anteriormente como consola de administración (debian) que tendrán que tener la misma versión de Apache JMeter y de Java. Para ello debemos realizar las siguientes configuraciones:

1.- Editamos en el equipo que ejercerá como administrador de JMeter el fichero /*opt/apache-jmeter-2.11/bin/jmeter.properties* y modificamos la linea remote_hosts por las direcciones IP de las maquinas que hemos creado para que realicen las peticiones a la aplicación web a testear junto al puerto de escucha:

remote_hosts=192.168.205.191:1099, 192.168.205.192:1099

2.- En los clientes de peticiones establecemos una variable como se indica a continuación:

export SERVER_PORT=1099

2.3.3 Ejecución de prueba con inyectores remotos

Procederemos a arrancar el el servicio en los inyectores ejecutando:

/opt/apache-jmeter-2.11/bin/jmeter-server

Por otro lado arrancamos tambíen la consola de administración con el comando.

/opt/apache-jmeter-2.11/bin/jmeter

O pasando como parámetros los hosts que harán las peticiones:

/opt/apache-jmeter-2.11/bin/jmeter -Jremote_hosts=192.168.205.191:3333,
192.168.205.192:3333

Ahora desde el administrador de JMeter abriremos el plan de prueba que deseamos realizar desde los los equipos clientes y para comenzar la prueba desplegaremos la pestaña "Lanzar" y posteriormente pulsaremos "Arrancar Todo Remoto" para iniciar las pruebas en todos los clientes remotos que hayamos especificado anteriormente.

	delviernes	s.jmx (/opt/a	pache-jmeter-2.11/	delviernes.jmx) – Apache
Archivo <u>E</u> ditar Search	Lanza <u>r</u> <u>O</u> pciones Ay	uda		
	Arrancar Inicio no se detiene	Ctrl-R		🔥 🗞 🗞 缓
ዮ 🚨 Plan de Pruebas	Arrancar Remoto	•	ic - Ultimate Th	read Group
👇 jp@gc - Ultimate Th	Arrancar Todo Remoto) Ctrl+Mayús-R		
🗠 🛞 Accediendo	Parar	Ctrl-Punto	🕅 e: jp@gc - Ultimate	Thread Group
🗠 🛞 Accede al listad	Interrumpir	Ctrl-Coma	ntarios	
🔶 🛞 Picamos en aña	Parar Remoto		lp on this plugin	
🗠 🛞 Rellenamos y er	Parar Todo Remoto	Alt-X	ón a tomar después	de un error de Muestrea
— 💽 Response Time	Apagar remoto			0.
— 💽 Gráfico	Apagar todo remoto	Alt-Z	Continua 🔍	ar 💛 Comenzar siguiente
- 🔄 Informe Agregad	Salir Remoto	•	ads Schedule	
🗌 🗆 🔄 Ver Arbol de Res	Salir de Todo Remoto		tart Threads Count	Initial Delay, sec
🗆 🗐 Banco de Trabajo	Limpiar	Ctrl+Mayús-E		0
	Limpi <u>a</u> r Todo	Ctrl-E		50
			-	

En las terminales de los inyectores se notificara el comienzo y el final de la prueba.

```
Starting the test on host 192.168.205.191:1099 @ Thu May 16 10:27:26 CEST
2014 (1398155246002)
Finished the test on host 192.168.205.191:1099 @ Thu May 16 10:33:57 CEST
2014 (1398155637482)
Starting the test on host 192.168.205.192:1099 @ Thu May 16 10:27:26 CEST
2014 (1398155246788)
Finished the test on host 192.168.205.192:1099 @ Thu May 16 10:33:57 CEST
2014 (1398155246788)
```

Una vez finalizado el test desde el Administrador de JMeter podemos observar los resultados obtenidos del mismo modo que si se hubiesen hecho las peticiones desde la misma maquina.

3 Hyperic

Describe el uso de un sistema que constantemente monitoriza una red de ordenadores en busca de componentes defectuosos o lentos.

Hyperic esta construido para funcionar en 75 componentes comunes tales como base de datos, dispositivos de red, servidores de red, etc. Detecta automáticamente todos los componentes de cualquier aplicación virtualizada.

Lo usaremos para monitorizar el servidor de aplicaciones web al que se le va a someter un proceso de testing. Cuenta con dos versiones una open soruce y una comercial, en nuestro caso usaremos la versión libre.

3.1 Instalación Hyperic

La instalación de Hyperic estará compuesta por varios elementos, uno: los agentes desde donde se quiera recibir la información del estado de esas maquinas y dos: el servidor desde donde se podrá observar dicho estado.

3.1.1 Instalación Hyperic Server

Para instalar la versión 5.8 de Hyperic hemos de contar previamente con una base de datos y usuario en PostgreSQL además de Java. Procedemos a instalar PostgreSQL:

aptitude install postgresgl-9.1

Lo primero que hay que hacer es cambiarle la clave al usuario "postgres" que se crea luego de haber instalado el paquete:

```
passwd postgres
Introduzca la nueva contraseña de UNIX:rootroot
Vuelva a escribir la nueva contraseña de UNIX: rootroot
passwd: contraseña actualizada correctamente
```

Accedemos a la consola de administración de PostgreSQL para cambiar la clave del usuario "postgres" con los siguientes comandos:

Ahora creamos el usuario administrador de Hyperic y le establecemos una clave:

postgres=# CREATE USER hqadmin WITH ENCRYPTED PASSWORD 'rootroot';

También creamos una base de datos para Hyperic

postgres=# CREATE DATABASE "HQ" OWNER hqadmin ENCODING 'UTF8';

Salimos de la sesión ejecutando lo siguientes comandos:

\q exit

Hyperic no está disponible en los repositorios de Debian, con lo que vamos a descargarlo de Sourceforge, lo descargaremos tanto en servidor como cliente para instalar en cada maquina de la manera pertinente.

Descargamos la ultima versión actual, se trata de la versión 5.8.1 del siguiente enlace <u>http://sourceforge.net/projects/hyperic-hq/files/Hyperic 5.8.1/hyperic-hq-installer-x86-64-linux-5.8.1.tar.gz/download</u>

Una vez finalizada la descarga descomprimimos el fichero:

tar -xvzf hyperic-hq-installer-x86-64-linux-5.8.1.tar.gz.crdownload

Antes de ejecutar el script de instalación hay que crear el directorio en el que va a instalarse Hyperic y cambiar el propietario (por defecto /home/hyperic):

mkdir /home/hyperic
chown avila:avila /home/hyperic/

Ahora procedemos con la instalación del **servidor**:

```
avila@flamingos:~/Descargas/hyperic-hq-installer-5.8.1$ ./setup.sh -full
Unpacking JRE to temporary directory /tmp/jre
Please ignore references to missing tools.jar
Unable to locate tools.jar. Expected to find it in /tmp/lib/tools.jar
Loading taskdefs...
Taskdefs loaded
Initializing Hyperic HQ 5.8.1 Installation...
Choose which software to install:
1: Hyperic HQ Server
2: Hyperic HQ Agent
You may enter multiple choices, separated by commas.
```

Aquí tenemos que elegir que paquete queremos instalar (si es mas de un paquete debemos separarlo por comas). En esta caso estamos instalando el servidor pero tambíen instalamos el agente por si por algún motivo nos pudiera hacer falta monitorizar nuestro servidor.

HQ server installation path [default '/home/hyperic']:

El directorio de la instalación será el que preparamos que es el que nos proponen por defecto, pulsamos "Enter".

What port should the HQ server's web-based GUI listen on for http communication? [default '7080']:

Establecemos el puerto por defecto de escucha para la web del servidor Hyperic HQ, pulsamos "Enter".

What port should the HQ server's web-based GUI listen on for secure https communication? [default '7443']:

Establecemos el puerto por defecto de escucha para la web del servidor Hyperic por https , pulsamos "Enter".

Aquí pulsamos la opción por defecto para no utilizar nuestro almacén de claves de Java.

Enter the base URL for the HQ server's web-based GUI [default 'http://192.168.205.163:7080/']:

Dejamos la URL por defecto para la web de Hyperic, pulsamos "Enter".

Enter the fully qualified domain name of the SMTP server that HQ will use to send email messages [default '192.168.205.163']:

Después nos pregunta por el servidor SMTP que Hyperic usará para enviar los mensajes de email que después configuraremos en esta misma maquina si se cree oportuno, con lo que nos habrá salido el FQDN de nuestro localhost, con lo que pulsamos de nuevo "Enter".

Enter the email address that HQ will use as the sender for email messages [default 'hqadmin@168.205.163']:

Especificamos el correo con el que se van a enviar las notificaciones por email, dejamos el que viene por defecto pulsando "Enter".

Choices: 1: HQ Built-in Database 2: PostgreSQL The HQ built-in database is provided for EVALUATION PURPOSES ONLY. For production purposes use vPosgreSQL. What backend database should the HQ server use? [default '1']:

Seleccionamos que base de datos queremos utilizar, en nuestro usaremos PostgreSQL pulsamos entonces "2".

Enter the vPostgres DB hostname [default 'localhost']:

Especificamos la maquina donde se encuentra la base de datos que es en la misma, como ya viene por defecto así pulsamos "Enter".

Enter the vPostgres DB port [default '5432']:

Aquí indicamos el puerto de PostgreSQL que es el mismo que viene por defecto por lo que volvemos a pulsar "Enter".

Enter the vPostgres DB name [default 'HQ']:

El nombre de la base de datos que nos propone es el mismo que ya habíamos creado anteriormente, por lo que pulsamos "Enter".

Override the JDBC connection URL for the PostgreSQL database [default 'jdbc:postgresql://localhost:5432/HQ?protocolVersion=2']:

Seleccionamos las opciones por defecto, pulsamos "Enter".

```
Enter the username to use to connect to the database:
hqadmin
```

Escribimos el nombre del usuario que se conectará a la base de datos "HQ" que será el que creamos "hqadmin".

Enter the password to use to connect to the database.: rootroot (again): rootroot

Introducimos la clave del usuario hq de la base de datos de PostgreSQL (rootroot):

Para no usar una clave autogenerada y establecer una personal pulsamos "2".

```
Enter an encryption key to use to encrypt the database password.: rootroot
```

Introducimos una clave para cifrar la base de datos.

What should the username be for the initial admin user? [default 'hqadmin']:

El usuario del administrador principal de Hyperic es el que viene por defecto, pulsamos "Enter".

What should the password be for the initial admin user?: rootroot (again): rootroot

Introducimos una clave para el administrador principal de Hyperic y la confirmamos.

What should the email address be for the initial admin user? [default 'hqadmin@168.205.163']:

Dejamos como correo electrónico del administrador principal como el que ya viene por defecto, pulsamos "Enter".

HQ agent installation path [default '/home/hyperic']:

La instalación del agente de Hyperic se hará en la ruta que ya creamos que es la misma que viene por defecto. Pulsamos "Enter".

```
Loading install configuration...
Install configuration loaded.
Preparing to install...
Validating agent install configuration...
Validating server install configuration...
Checking server webapp port...
Checking server secure webapp port...
Checking database permissions...
Verifying admin user properties
Validating server DB configuration...
Installing the agent...
Looking for previous installation
Unpacking /home/avila/Descargas/hyperic-hq-installer-5.8.1/hyperic-hq-agent-
x86-64-linux-5.8.1.tar.gz to: /home/hyperic/agent-5.8.1...
Setting permissions on /home/hyperic/agent-5.8.1...
Setting permissions on /home/hyperic/agent-5.8.1/conf/agent.properties...
Setting permissions on agent binaries...
Fixing line endings on text files...
Installation Complete:
 Agent successfully installed to: /home/hyperic
You can now start your HQ agent by running this command:
      /home/hyperic/agent-5.8.1/bin/hq-agent.sh start
Installing the server...
Unpacking server to: /home/hyperic...
Creating server configuration files...
Using "small" installing profile...
Copying binaries and libraries to server installation...
Copying server configuration file...
Copying server db-upgrade files...
Copying server libs...
Setting up server database...
Loading taskdefs...
Taskdefs loaded
```

Miguel Ángel Ávila Ruiz

Loading taskdefs... Taskdefs loaded Loading taskdefs... Taskdefs loaded Installing the server JRE ... Unpacking JRE amd64-linux-1.7 40.tar.gz to: /home/hyperic/server-5.8.1... Setting permissions on /home/hyperic/server-5.8.1... Setting permissions on server binaries... Fixing line endings on text files... Installation Complete: Server successfully installed to: /home/hyperic/server-5.8.1 You can now start your HQ server by running this command: /home/hyperic/server-5.8.1/bin/hq-server.sh start Note that the first time the HQ server starts up it may take several ninutes to initialize. Subsequent startups will be much faster. Once the HQ server reports that it has successfully started, you can log in to your HQ server at: http://192.168.205.163:7080/ username: hqadmin To change your password, log in to the HQ server, click the Administration" link, choose "List Users", then click on the "hqadmin" user. Setup completed. A copy of the output shown above has been saved to: /home/avila/Descargas/hyperic-hq-installer-5.8.1/installer/logs/hqinstall.log

Deleting temporary JRE

3.1.2 Instalación Agente Hyperic

Ahora instalaremos el agente de Hyperic en la maquina donde se servirá la aplicación web a la que se le está realizando el proceso de de testing con Apache JMeter.

Nos ubicamos en el directorio /tmp para descargar el paquete de instalación de Hyperic ahí:

cd /tmp

Descargamos la ultima versión actual, se trata de la versión 5.8.1:

wget http://sourceforge.net/projects/hyperic-hq/files/Hyperic_5.8.1/hyperichq-installer-x86-64-linux-5.8.1.tar.gz/download

Una vez finalizada la descarga descomprimimos el fichero:

tar -xvzf hyperic-hq-installer-x86-64-linux-5.8.1.tar.gz.crdownload

Antes de ejecutar el script de instalación hay que crear el directorio en el que va a instalarse Hyperic y cambiar el propietario (por defecto /home/hyperic):

mkdir /home/hyperic
chown avila:avila /home/hyperic/

Ejecutamos el script de instalación con un usuario no privilegiado:

Introducimos "2" para instalar el paquete del Agente Hyperic .

HQ agent installation path [default '/home/hyperic']:

El directorio de instalación que viene por defecto es el que creamos anteriormente por lo que pulsamos "Enter" para establecer el valor por defecto.


```
Loading install configuration...
Install configuration loaded.
Preparing to install...
Validating agent install configuration...
Installing the agent...
Looking for previous installation
Unpacking /tmp/hyperic-hq-installer-5.8.1/hyperic-hq-agent-x86-64-linux-
5.8.1.tar.gz to: /home/hyperic/agent-5.8.1...
Setting permissions on /home/hyperic/agent-5.8.1...
Setting permissions on /home/hyperic/agent-5.8.1/conf/agent.properties...
Setting permissions on agent binaries...
Fixing line endings on text files...
Installation Complete:
 Agent successfully installed to: /home/hyperic
You can now start your HQ agent by running this command:
       /home/hyperic/agent-5.8.1/bin/hq-agent.sh start
Setup completed.
A copy of the output shown above has been saved to:
 /tmp/hyperic-hq-installer-5.8.1/installer/logs/hq-install.log
Deleting temporary JRE
```

Ahora arrancamos el servidor Hyperic, la primera vez que se inicia tarda bastante en arrancar ya las siguientes veces no tanto:

```
avila@flamingos:~/$ /home/hyperic/server-5.8.1/bin/hq-server.sh start
-e Starting HQ Server...
```

Luego arrancamos los Agentes que hemos instalado:

En la maquina donde esta instalado el servidor Hyperic y JMeter:

```
avila@flamingos:~$ /home/hyperic/agent-5.8.1/bin/hq-agent.sh start
Starting HQ Agent..... running (7214).
[ Running agent setup ]
What is the HQ server IP address: 192.168.205.163
Should Agent communications to HQ always be secure [default=yes]:
What is the HQ server SSL port [default=7443]:
- Testing secure connection ... Success
What is your HQ login [default=hqadmin]:
What is your HQ password:
What IP should HQ use to contact the agent [default=127.0.1.1]:
192.168.205.163
```


Wł	nat port should HQ use to contact the agent [default=2144]:
-	Received temporary auth token from agent
_	Registering agent with HQ
_	HQ gave us the following agent token
	1396511318832 - 6129476998079249628 - 4954716731030195406
_	Informing agent of new HQ server
_	Validating
_	Successfully setup agent

En la maquina donde se servirá la aplicación web a testear:

```
root@debian:~# /home/hyperic/agent-5.8.1/bin/hq-agent.sh start
Starting HQ Agent..... running (6047).
Running agent setup ]
What is the HQ server IP address: 192.168.205.163
Should Agent communications to HQ always be secure [default=yes]:
What is the HQ server SSL port [default=7443]: ^C
root@debian:~# /home/hyperic/agent-5.8.1/bin/hq-agent.sh start
Starting HQ Agent...HQ Agent is already running.
root@debian:~# /home/hyperic/agent-5.8.1/bin/hq-agent.sh stop
Stopping HQ Agent...
Stopped HQ Agent.
root@debian:~# /home/hyperic/agent-5.8.1/bin/hq-agent.sh start
Starting HQ Agent..... running (5389).
Running agent setup ]
What is the HQ server IP address: 192.168.205.163
Should Agent communications to HQ always be secure [default=yes]: no
What is the HQ server port
                            [default=7080]:
- Testing insecure connection ... Success
What is your HQ login [default=hqadmin]:
What is your HQ password:
What IP should HQ use to contact the agent [default=192.168.205.21]:
192.168.205.164
What port should HQ use to contact the agent [default=2144]:
- Received temporary auth token from agent
- Registering agent with HQ
The server to agent communication channel is using a self-signed certificate
and could not be verified
Are you sure you want to continue connecting? [default=no]: yes
- HQ gave us the following agent token
   1396512359668-1266683792848212614-3358222710918894051
 Informing agent of new HQ server
 Validating
 Successfully setup agent
```


3.2 Acceso a Hyperic

Ahora accedemos a la interfaz web, nos conectamos al puerto 7080 de nuestra maquina, pondremos el nombre de usuario y contraseña del usuario administrador principal que especificamos durante la instalación de Hyperic:

🐳 Sign In - Hyperic 🛛 🗣			
Contemporary 192.168.205.163:7080/app/login;jsessionid=84B25C608	1E63FF76615DA46A888E175	☆♥@ 🕙 Google	۵ 🕹 🙆
HYPERIC HQ			Screencasts Solution Help
	Sign in to Hyperic		
	lisername		
	Password		
	Sign in		

Podemos observar como Hyperic nos ha detectado automáticamente los agentes instalados y activados con sus respectivos servicios:

📫 HQ Dashboard 😽					
3 192.168.205.163:7080		☆ ~ ©	8 V Google		Q 🕹
Recent Alerts: (There have be	en no alerts in the last 2 hours.)		Welcome, HQ Sign Out Scr	reencasts	Solution Hel
	nalvze Administration		0.		
Dusinista a Resources 1	nayze naniniseauon		~		
Search Resources	Auto-Discovery				÷ ¥
	Resource	Name		Status	Changes
Resource Name Platforms	le debian - De	abian 7.4		new	N/A
Sauad Charte	HQ Agent 5	.8.1 /home/ł	hyperic/agent-5.8.1	new	N/A
Saved chaits	Apache Ton	ncat 7.0 /var/lib/	/tomcat7	new	N/A
	MySQL Stat	s 5.5.x /usr		new	N/A
Recently Added	Apache 2.2	.22 /etc/ap	ache2	new	N/A
No resources to display	Samba 3.x	/usr/sb	in	new	N/A
Availability Summary	🔅 🗙 🔿 🛛 flamingos	- Debian 7.4		new	N/A
Resource Type Availability	Apache 2.2	.22 /etc/ap	ache2	new	N/A
No resources to display, please click the 💮 icon above to add resources to portlet.	PostgreSQL	9.x localhost:5432 /usr/lib/	/postgresql/9.1/bin/postgres	new	N/A
Add content to this column:	HQ Agent 5	.8.1 /home/ł	hyperic/agent-5.8.1	new	N/A
Select Portiet	Hyperic - A	pache Tomcat 6.0 /home/ł	hyperic//hq-engine/hq-server	new	N/A
	MySQL Stat	s 5.5.x /usr		new	N/A
	Add to Invent	ory Skip Checked Resour	ces		
	🔝 Favorite Res	ources			* ×
	Resource Name		Resource Type	Availa	bility Alerts
	No resources to dis	play, please click the 🎡 icon above t	to add resources to portlet.		
				Upd	lated: 10:28 AM
	🔝 Recent Alerts	5			\$ X
	🗆 Date / Time –	Alert Name	Resource Name	Fired	Ack

3.3 Acondicionando Hyperic

Ahora añadiremos al Inventario la maquina a monitorizar, en nuestro caso es "debian" así que la marcamos y el panel "Auto-Discovery" y pulsamos en "Add to Inventory". Nos aparecerá en "Recently Added":

🖶 HQ Dashboard 🕹								
💮 🛞 🕑 192.168.205.163:7080/Dashboard.do	?org.apache.catalina.filt	ers.CSRF_N	ONCE=E68	D9C56A88C06F4	63BF580C ☆ ✔ @	🗿 🚺 🗙 Google		Q 🕹 🏠
	Dashboard Resources	Analyze	Administrati	on		Q.		
Search Resources			💢 Auto	-Discovery				* X
Resource Name Platforms T				Resource Name			Status	Changes
			۲	flamingos - Debian 7.4			new	N/A
Saved Charts			× ¥	Apache 2.2.22	/etc/a	pache2	new	N/A
No charts to display			× •	PostgreSQL 9.x localhos	st:5432 /usr/lit	p/postgresql/9.1/bin/postgres	new	N/A
			V	HQ Agent 5.8.1	/home	/hyperic/agent-5.8.1	new	N/A
Availability Summary			× ×	Hyperic - Apache Tomca	at 6.0 /home	/hyperic//hq-engine/hq-serv	er new	N/A
Resource Type	Availability	y	V	MySQL Stats 5.5.x	/usr		new	N/A
No resources to display, please click the opticon above to add resourc	es to portiet.		A	dd to Inventory	Skip Checked Resou	rces		
Recently Added		ł	÷ 🗙					
Resource Name		Time	🔊 F	avorite Resources			_	🌣 🗙
debian	9 r	minutes ago	Reso	urce Name			Resource Avai Type Avai	lability Alerts
Availability Summary			🔅 💢 🛛 No re	sources to display, please	click the 🎡 icon above	to add resources to portlet.		
Resource Type	Availabilit	y					U	pdated: 11:39 AM
No resources to display, please click the 🏐 icon above to add resourc	es to portlet.			lecent Alerts				* ×
Add content to this column:				ate / Time 🔻	Alert Name	Resource Name	Fixed	Ack
Select Portiet			No re	cent alerts to display				
			FIXE	DACKNOWLEDGE			U	odated: 11:39 AM
			M	ontrol Actions				* *
			Re	cent Control Actions				
			No	resources to display				
			Qui	ck Control Frequency				

Si pulsamos sobre "debian" entraremos en la pantalla de monitorización de dicha maquina, podemos observar los servicios del equipo, si nos fijamos en Apache Tomcat 7.0 y MySQL Stats 5.5x nos aparece un icono de interrogación de color gris a su derecha. Vamos a tener que hacer alguna tarea adicional para tener el completo monitorio de estos recursos.

HQ View Platform Monitor Cur	+										-	8
3 192.168.205.163:7080/reso	ource/pla	tform/monitor/Visibil	ity.do?mode=curr	entHealth&eid=1%	3a10001&o	g.apache.cata	atir 🖙 🗸 🕲 🚺	8 ∨ Google	2		Q	J. 1
HYPERIC H	-19	Recent Alert Dashboard	s: (There have been Resources Ana	no alerts in the last 2 ho lyze Administration	urs.)			Welcome, HQ) Sign Out s	Screencasts	Solution I	Help
Browse > debian Return to debian Apache Tomcat 7.0 Description: Debian 7.4 Default Gatway: 132 188 205 1 IP Address : 132 188 205 164 OS Version : 3 2.0-4-amd64 Map © Tools Menu ©		Owner: HQ A Vendor : Deb Primary DHS RAM : 1024 №	dministrator (hqadmin) ian : 194.140.1.51 B) - Change			Vendor Vers CPU Speed : Architecture	ion : 7.4 1895 MHz 9 : x86_64				
Monitor Inventory RESOURCES	Alert Metric Di	Views splay Range: 4 Last 8 INDICATORS	Hours I	Advanced Settings								
Platform Services Health C NetworkServer Interface Process	Avail	Indicator Charts Sho	w Last 8 Hours			View:	Update Default 100,0% — Availability					
debian Linux CPU 1 (1895Mhz AMD Athlon) debian Linux File System /dev/disk /by-uuid/3a754de6-5109-4ad8- acb1-610e0eatLa5c mounted on / (local/ext4)	0 5 0 5	Cpu Usage (Linux)			LOW: 5,8%	AVG: 12,9%		× ^				
Deployed Servers Health Samba 3.x	Avail				•	•	50% 25%					
debian Apache 2.2.22 debian Apache Tomcat 7.0 debian HQ Agent 5.8.1	0 9	Free Memory			LOW: 59,6 MB	AVG: 85,1 MB	• _ 0%	≍ ₌				
debian MySQL Stats 5.5.x Select Resources above & Click button to view metrics View Metr	ics 🕕	-			•••••		116 MB 101 MB 86 MB					
All Metrics Cpu Usage Free Memory	90						72 MB					
Load Average 5 Minutes Swap Used Availability	9 D 9 D 9 D	E Load Average 5 Mir (Linux)	utes		LOW: 0,0	AVG : 0,0	1 4 PEAK: 0,0	×				
												ļ

3.3.1 Configuración Tomcat para monitorio

Para poder monitorizar nuestro Tomcat, deberemos habilitar JMX. Modificaremos el fichero catalina.sh para agregar estos parámetros en la variable CATALINA_OPTS, añadiremos las siguientes lineas en el fichero "/usr/share/tomcat7/bin/catalina.sh":

Luego reiniciamos el servicio de Tomcat para que los cambios surtan efecto:

Una vez hecho esto, picamos en nuestro recurso monitorizado Tomcat de la lista:

🖶 HQ View Platform Monitor Cur	+								
	0/res	ourc	e/platform/monitor/Visibility.do?mode=currentHealth8	keid=1%3a100	01&org.apac	h☆ ∨ @ <mark>8</mark> ∨ 0	Google	Q 🕹 🍪	E.
Browse • debian									
Description: Debian 7.4 Default Gateway : 192.168.205.1 IP Address : 192.168.205.164 OS Version : 3.2.0-4-amd64			Owner: HQ Administrator (hqadmin) - Change Vendor : Debian Primary DNS : 194.140.151 RAM : 1024 MB			Vendor Version : CPU Speed : 1895 Architecture : x86	7.4 MHz i_64		
Map 🗊 Tools Menu 🗉 Monitor Inventory	Alert		Views						
	Metrie	: Disj	olay Range: 🔳 Last 8 丈 Hours 丈 🚺 Advanced Settings						
RESOURCES			INDICATORS METRIC DATA						=
Platform Services Health	A vail		Indicator Charts Show Last & Hours		View:	Undate Default			1
IntworkServer Interface	٢	9			*****·L				
Process	٢	9				Availability			
debian Linux CPU 1 (1895Mhz AMD Athlon)	٢	Q	Cpu Usage	LOW: 5.8%	AVC: 12.6%	1 J X			
debian Linux File System /dev/disk /by-uuid/3a754de6-5109-4ad8- acb1-610clcaf1a5c mounted on / (local/ext4)	0	9	(4004)	•	H00. 12,070	100% 75%			
Deployed Servers Health	Avail					50%			
Samba 3.x	٢	∇			•				
debian Apache 2.2.22	٢	∇				• 0%			
debian Apache Tomcat 7.0	Θ	∇	Eree Memory			î I 🗙			
debian HQ Agent 5.8.1	٢	∇	(Linux)	LOW: 59,6 MB	AVG: 83,9 MB	PEAK: 112,9 MB	=		
debian MySQL Stats 5.5.x	Θ	∇				116 MB			
Select Resources above & View Metro	rics	0				101 MB			
All Metrics				•	-	86 MB			
Cpu Usage	9					• 72 MB			
Free Memory	∇								~

Configuraremos las propiedades, para ello pulsamos en "Configuration Properties":

🖷 HQ View Server Monitor Curre 🗣	
	ce/server/monitor/Visibility.do?mode=currentHealth&eid=2%3a10004&org.apache. 🗇 🕫 🚺 🖉 Google 🔍 🌷
Map 😨 Tools Menu 😨	
Monitor Inventory Alert	Control Views
This resource is turned off or has not been configured p The problem is: Error contacting resource: Can't connect & Root exception is javarini. Connect: Exception: Connection java.net. ConnectException: Con <u>ection rehusadal</u> Please turn on the server, or fix ts Configuration Prop	roperly. To MBeanServer url [service;imx:mi://ijndi/mi://localhost.6969/jmx:mi] port [null] usernane [system]: java.io JOException: Failed to retrieve RMIServer stub: javax.naming.ServiceUnavailableException refused to host: localhost; nested exception is: Territes]
Metric Dis	splay Range: Alast 8 - Hours - Control Advanced Settings
RESOURCES	INDICATORS METRIC DATA
Services	Indicator Charts Show Last 8 Hours View: Update Default -
No health data is available for this resource.	
Host Platform Avail	
🗌 debian 🛛 🔘 🖓	
Select Resources above & View Metrics	
All Metrics	· · · · · · · · · · · · · · · · · · ·
No metrics to display	There are no indicator metrics data to display for time period between 04/03/2014 05:26 AM and 04/03/2014 01:26 PM

Cambiamos la ruta por "/usr/share/tomcat7/bin/catalina.sh" y pulsamos "OK":

Or 2012 06.205.103.7080/resource/server/liventary.do?mode=edtConfig&type=2&rid=10004 Or	🖷 HQ Edit Server Configuration P 🚭							
Recent Alere: There have been no alers in the last 2 hours.) Dashboar Tesource Deshboar Tesource <	@ 192.168.205.163:7080/resource/server	/Inventory.do?mode=edit	Config&type=2&rid=100	04	☆ ~ ©	8∨ Google		Q 🕹 🏠
Dashboard Resource Analyza Administration debian Apache Toncat 7.0 Please verify that this resource has been enabled for monitoring by following the directions below Configuration Properties Sheed	HYPERIC HQ	Recent Alerts: (There h	nave been no alerts in the last 2	hours.)	_	Welcome, HQ Sign Out	Screencasts	Solution Help
debian Apache Tomcat 7.0 Please verify that this resource has been enabled for monitoring by following the directions below Conditionation Definition Shared JMX URL to MBeanSewer ServerLog_TackLevel Enable Cog TackIng Batter Mark ServerLog_TackLevel Log Pattern Mark Se		Dashboard Resource	es Analyze Administratio	n				
Please verify that this resource has been enabled for monitoring by following the directions below Diffuration Properties Shared jmxusemane gutem JMK URE to Missioner jmmune gutem gutem JMK URE to Missioner jmmune gutem gutem gutem Servel.og match.exeline immune gutem gutem gutem gutem gutem gutem gutem gutem gutem gute	debian Apache Tomcat 7.0							
Configuration Properties Shared JMX URL to MBeanSever JMX Departments JMX Departments JMX Departments JMX Departments State Atomate Log File Server.log Track.level Log File Server.log Track.level Server.log Track.level </td <td>Please verify that this resource has been</td> <td>enabled for monitoring</td> <td>by following the directio</td> <td>ns below</td> <td></td> <td></td> <td></td> <td></td>	Please verify that this resource has been	enabled for monitoring	by following the directio	ns below				
Shared imx.ung imx.ung.te wide anservet imx.process.query imax.password imx.password imx.password imx.password imx.password imx.password imx.password imax.password	Configuration Properties		, in the second					
JMX UR to the server log fink in jimk in ji	Shared							
intrapassword	★jmx.url JMX URL to MBeanServer	service;jmx:rmi:///jndi/rmi://loc;			jmx.usernan JMX usernar	ne system		
Listeniap Ports B006,8080 Monitoring Server.log_track.level Enable Log Tracking Error ↓ Server.log_track.nebide Log File Server.log_track.level Dep Attein Exitive Error ↓ Server.log_track.nebide Log File Server.log_track.enable Dep Attein Exitive Server.log_track.enable Enable Config_track.files Server.comfig_track.files Configuration File Server.comfig_track.files Server.comfig_track.files Configuration Files Server.comfig_track.files Server.comfig_track.files Configuration Files Server.comfig_track.files Server.comfig_track.files Configuration Files Server.comfig_track.files Server.comfig_track.files Configuration Files Server.comfig_track.files Server.comfig_track.files Server.comfig_track.files Server.comfig_track.files Full path to Apache Tomcat?/Dem/Catalina.st Prefits arguments to control program	j mx.password JMX password				*process.que PTQL for Tomcat Proce	State.Name.re=javaljsvc,Sta	te	
Monitoring Server.log_track.enable Enable log Tracking Server.log_track.evel Track vent log review Error → Track vent log review Server.log_track.inextde Log Platen Nakch, lies Log Files Server.log_track.evel Deg Platen Skettede Enable Config Track.enable Enable Config Track.enable Enable Config Tra	listen.ports Listening Ports	8005,8080						
server.log_track.level Error ■ Enable Log Trackinetide Track verting level Log Patem Nacht Server.log_track.evel Log Patem Nacht Log Patem Nacht Server.log_track.evel Error ■ Log Patem Nacht Server.log_track.evel Log Patem Nacht Server.log_track.evel Server.log_track.files Opplicatalina.out Server.config_track.enable Enable Config Track.enable Configuration Files Configuration Files Control #program Prefix Prefix Full path to Apache Tomcat7.0 control pregram Prefix arguments to control pregram Timeout of control operations (in second) 30 Ok Reset	Monitoring							
server.log_track.inelude	server.log_track.enable Enable Log Tracking				server.log_track.lev Track event log lev	Error 💌		
Server.log, Track.files Log Files Iog Slotataina.out Server.config_track.enable Enable Config Tracking Server.config_track.files configuration Files Controure Pools, and other services? Controu Profigram Ivar/libtomeat7/bin/cataina.st Full path to Apache Tomeat 7.0 control program Profix Timeout of control operations (in second) 30 Ok Reset	server.log_track.include Log Pattern Match				server.log_track.exclud Log Pattern Exclu	le de		
Server.config.urack.files conf.true;"\properties[.*].xml ✓ Auto-Discover Caches, DataSource Pools, and other services? Control *Program Full path to Apache Tomcat 7.0 control program /var/lib/tomcat7/Din/catalina.sl Full path to Apache Tomcat 7.0 control program /var/lib/tomcat7/Din/catalina.sl Timeout of control operations (in seconds) 30 Ok Reset	server.log_track.files Log Files	logs/catalina.out			server.config_track.enab Enable Config Tracki			
Auto-Discover Caches, DataSource Pools, and other services? Control Full path to Apache Tomcat 7.0 control program Full path to Apache Tomcat 7.0 control program Full path to Apache Tomcat 7.0 control program Imeout of control operations (in seconds) 0k Cancel	server.config_track.files Configuration Files	conf;true;.*\.properties .*\.xml						
Control Program Full path to Apache Tomicat 7.0 control program Full path to Apache Tomicat 7.0 control program Full path to Apache Tomicat 7.0 control program fineout of control operations (in seconds)	☑ Auto-Discover Caches, DataSource Pools, and other	services?						
program prefix Full path to Apache Tomicat 7.0 control program Minimum Timeout of control operations (in seconds) 30 Ok Reset Cancel	Control							
• timeout 30 Timeout of control operations (in seconds) 30 Ok Reset	★ program Full path to Apache Tomcat 7.0 control program	/var/lib/tomcat7/bin/catalina.sł			pret Prefix arguments to control progra	ix m		
Ok Reset Cancel	timeout Timeout of control operations (in seconds)	30						
			Ok Reset	Cancel				

Dentro de unos instantes observaremos como el servicio de Tomcat ya se está monitorizando correctamente, pulsamos sobre él para comprobarlo entre otros parámetros podemos observar la CPU usada y la memoria libre:

🖶 HQ View Platform Monitor Cur	*							
192.168.205.163:7080/res	ource/pla	atform/monitor/Visibility.do	o?mode=current	Health&eid=1%3a100018	korg.apache.cat	☆ ヾ @ <mark>8</mark> ▼ Goog	gle (🔍 🕹 🍙
	ΗQ	Recent Alerts: (The Dashboard Reso	re have been no ale urces Analyze	erts in the last 2 hours.) Administration		Welcome	e, HQ Sign Out Screencasts So	lution Help
Browse - debian Description: Debian 7.4 Default Gateway: 192,168,205.1 IP Address: 192,168,205,1.64 OS Version: 3.2.0-4-amd64 Map : Tools Menu :		Owner: HQ Administr; Vendor : Debian Primary DIS : 194,14 RAM : 1024 MB	ator (hqadmin) - Cha 40.1.51	nge		Vendor Version : 7.4 CPU Speed : 1895 MHz Architecture : x86_64	:	
Monitor Inventory Alert Views Metric Display Range: Last 8 Hours 3 Advanced Settings NUCLATORS METRIC DATA 1								
Interface	0 9	Indicator Charts Show Last	8 Hours		View: U	100.0% -		
Process	0 🖓					Availability		
debian Linux CPU 1 (1895Mhz AMD Athlon) debian Linux File System /dev/disk /by-uuid/3a754de6-5109-4ad8-	0 9	⊠ Cpu Usage (Linux)		LOW: 8,6%	AVG: 19,7%			-
acbf-610c0caf1a5c mounted on / (local/ext4)			_			- 75%		
Deployed Servers Health	Avail				•	50%		
Samba 3.x	0 🖓				÷.,	- 25%		
debian Apache 2.2.22	0 🖓					- 0%		
debian Apache Tomcat 7.0		🔤 Free Memory				î 🕹 🗙 👘		
debian HQ Agent 5.8.1		(Linux)		LOW: 63,9 M	B AVG: 72,8 MB	PEAK: 83,2 MB		
debian MySQL Stats 5.5.x	0 🖓					84 MB		
Select Resources above & View Metr	ics 🕕		_			79 MB		

IVPERIC HC				welcome, no sign o
	Dashboard Resources	Analyze Administration		
Prowse , debian , debian Anacha Tamaa	+ 7.0			
escription:	ner: HO Administrator (bradmin) - Change			
	indiring standard day (repairing) of angoing			
Map 🔄 Tools Menu 🛡				
Monitor Inventory Alert	Control Views			
Metric D	Nisplay Range: ◀ Last 🛛 🛨 Hours	Advanced Settings		
RESOURCES	INDICATORS METRIC DATA			
Services Avail	Indicator Charts Show Last 8 Hours		View: Update Default	-
🗌 🗀 Apache Tomcat 7.0 JSP Monitor 👘 📀 🤅			100.0%	
🗌 🗀 Apache Tomcat 7.0 Servlet Monitor 🛛 📀 🔍	2		Availability	у
Apache Tomcat 7.0 Web Module 🛛 🖉 🕬	Heap Memory Free	LOW: 29.5 MR	▲V/C: 59 3 MB PEAK: 70 3 MB	×
debian Apache Tomcat 7.0 "http-		LOW: 23,5 WD	72 MB	
Processor			61 MB	
debian Apache Tomcat 7.0 "http- bio-8080" Thread Pools	▽		50 MB	
Host Platform Avail	-		39 MB	
🗌 debian 🖉 🖉	≂		◆ 27 MB	
Select Resources above & View Metrics) Process Cpu Time per Minute		1 🌡	×
All Metrics	(Apache Tomcat 7.0)	LOW: 1,105s	AVG: 2,062s PEAK: 2,864s	
Heap Memory Free 🖓 🕻	2		• 3s	
Process Cpu Time per Minute 🤤 🕻	3		2,5s	
UpTime 🖓 🕻	3		2S	
Availability	2		• 1,55 • 15	
	(Apache Tomcat 7.0)	LOW: 00:01:55.86	8 AVG: 00:01:55.868PEAK: 00:01:55	5.868
			00:03:51.7	
			00:02:53.8	
			• 00:01:55,9	
			00:00:57,9	
			00:00:00	
	04/04/2014 01:58 AM	0	4/04/2014 09:50 AM	

Vemos como ya estamos obteniendo resultados de monitorización de Apache Tomcat:

3.3.2 Configuración MySQL para monitorio

Para que también nos monitorice correctamente MySQL en primer lugar crearemos un usuario en MySQL para las cuestiones de monitorización

Entramos en MySQL con un usuario con privilegios:

```
mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 815
Server version: 5.5.35-0+wheezy1 (Debian)
Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.
```


Creamos el usuario para la monitorización:

```
mysql> create user 'hqagent'@'localhost' identified by 'hqagent';
Query OK, 0 rows affected (0.38 sec)
```

Para habilitar la supervisión de procesos completa de MySQL el usuario del agente debe de ser propietario del proceso o tener permisos para ver todas las estadísticas de procesos por ello le otorgamos los siguientes privilegios:

```
mysql> grant show databases on *.* to 'hqagent'@'localhost';
Query OK, 0 rows affected (0.01 sec)
mysql> grant select on *.* to 'hqagent'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Salimos de la consola de MySQL:

Luego en la interfaz web de Hyperic picaremos en MySQL, después en "Configuration Properties", introducimos el usuario que hemos creado y la clave del usuario por ultimo pulsamos en "OK":

HYPERIC H	0	Recent Alerts: (There have been no alerts in the last 2	nours.)			Velcome, HQ Sign Out Screencasts Solution Help
	. – «	Dashboard Resources Analyze Administration	1			
Browse • debian • debian MySQL Si	tats 5.5	x				
Description: MySQL Version : 5.5.x		Owner: HQ Administrator (hqadmin) - Change				
Map 🐨 Tools Menu 🛡						
Monitor Inventory Al	ert	Views				
м	etric Dis	play Range: 4 Last 8 Hours Advanced Settings				
RESOURCES		INDICATORS METRIC DATA				
Services		Indicator Charts Show Last & Hours		View:	Indate Default	
No health data is available for this resource.				vic.vv. [100.0%	
Host Platform	Vail				Availability	
🗌 debian	0 🖓	M Process Cpu Usage			î 🛛 🗙	
Select Resources above & View Metrics	0	(MySQL Stats 5.5.x)	LOW: 0,3%	AVG: 0,3%	PEAK: 0,4%	
All Metrics					100%	
Process Cpu Usage	🗩 🖪				75%	
Process Resident Memory Size	🗩 🖪				50%	
Slow Queries per Minute	🗩 🖪				25%	
Temporary Disk Tables Created per Minute	🗩 🗖				• 0%	=
Temporary Files Created per Minute	🗩 🗖	Memory Size			î 🖡 🗙	
Availability	🗩 🖪	(MySQL Stats 5.5.x)	LOW: 42,9 MB	AVG: 43,0 MB	PEAK: 43,2 MB	
				**	43,20 MB	
					43.13 MB	
					43,06 MB	
					42,99 MB	
				•••••	42,92 MB	
		Sieve Overlag per Minute			1 I V	
		(MySQL Stats 5.5.x)	LOW: 0,0	AVG: 0,0	PEAK: 0,0	
					4	
					3	
					2	
					1	
				• •••• •••	0	
		27 Tomperatu Dick Tables Created nor Minute			î I 🖌	~
		04/04/2014 02:41 AM		4/04/2014 11:22 AI		

Comprobamos que ya se esta monitorizando correctamente MySQL:

Ya tenemos la maquina donde se va a servir la aplicación web a someter a pruebas incluyendo todos los servicios monitorizados para que al someterlas a cargas con Apache JMeter podamos comprobar si los recursos con el que cuenta ese equipo son suficientes o no, ya sea de procesador o de memoria RAM.

4. Conclusión

Tras finalizar el proyecto y haber hecho diversas pruebas con este sistema de pruebas de rendimiento de aplicaciones web quedo fascinado por la aplicación de Apache JMeter con la que se pueden hacer pruebas muy personalizadas pudiendo llegar a simular muy bien la carga que se desee y que reciba una vez publicada la aplicación web.

Hay que recordar que con esta herramienta también se pueden hacer pruebas de rendimiento en otros campos como en bases de datos, TCP, LDAP,... Por lo que no descarto en cuanto surja la ocasión de retomar JMeter y de seguir estudiando el programa en estos otros ámbitos.

5. Referencias

Estas son tan solo algunas páginas de las que me serví para el desarrollo del proyecto:

Introducción a JMeter: Conceptos Básicos http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/388

Performance Testing con JMeter: Paso por Paso para grabar escenarios http://josepablosarco.wordpress.com/2009/02/03/performance-testing-con-jmeter-paso-por-paso-para-grabar-escenarios/

Construcción de Planes de Prueba con JMeter <u>http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/391#Grabacion de un testplan para aplicaciones</u> <u>web</u>

Pruebas de carga con el incomprendido JMeter - Parte 1. http://coudmonqui.blogspot.com.es/2013/06/pruebas-de-carga-con-el-incomprendido.html

Usando JMeter para el estrés Web http://rafaeska.es/algo-que-anadir/-/blogs/usando-jmeter-para-el-estres-web

JMeter: Tests de rendimiento usando varios clientes distribuidos http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=jmeter_remote

Hyperic HQ Instalación en GNU Linux #Monitoreo http://www.leobravo.cl/?p=2096#respond

Instalación de Hyperic HQ 4.5 en Ubuntu Server 10.10 http://www.openredes.com/2010/12/31/instalacion-de-hyperic-hq/