

# **Sistemas Informáticos II**

## **Práctica 2**

Prácticas 2016/17

Jaime López

# Objetivos P2

- **Objetivos**

- Análisis de rendimiento de las aplicaciones realizadas siguiendo la arquitectura JavaEE
- Monitorización del servidor de aplicaciones
- Análisis de la cadena de procesamiento dentro del servidor de aplicaciones.
- Elementos que la componen.
- Puntos de encolamiento.
- Configuración del servidor de aplicaciones de cara al rendimiento
- Mecanismos de evaluación del rendimiento, monitorización y realización de pruebas unitarias.
- Pruebas de stress de la aplicación, mediante el empleo de JMeter

- **Material entregado**

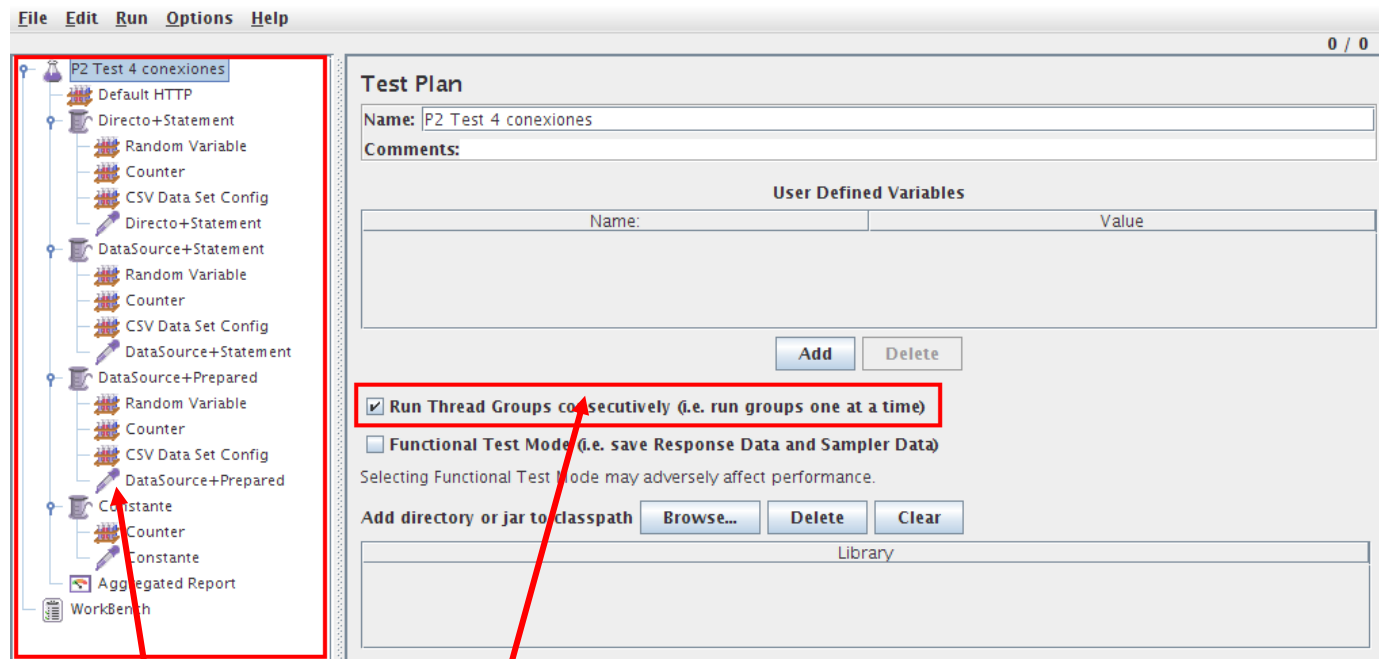
- **P2-alumnos.tgz** : Incluye nueva versión la base de datos y fichero .csv de pruebas
- **SI2-P2-curvaProductividad.ods** (hoja de cálculo)
- **SI2-P2-curvaProductividad.jmx** (fichero Jmeter)

# Pruebas a realizar

- Primera parte
  - Creación de un plan de pruebas con Jmeter
  - Pruebas **unitarias** (un solo usuario) con el fin de identificar la mejor solución en cuanto a rendimiento de las tres desarrolladas en la P1 (P1-base, P1-ws-cliente, P1-ejb)
- Segunda parte:
  - Pruebas de stress sobre una de las soluciones (P1-base)
  - Múltiples usuarios
  - Búsqueda del punto de saturación del sistema
  - Mejora del rendimiento

# Jakarta JMeter

Jmeter es una herramienta de test de carga, pruebas funcionales y medición de rendimiento de aplicaciones web, bases de datos y otros recursos



- Un plan de pruebas constará de varios elementos **organizados en forma de árbol.**
- **El árbol se ejecuta de arriba abajo,** en una secuencia de pruebas.

# Jakarta JMeter

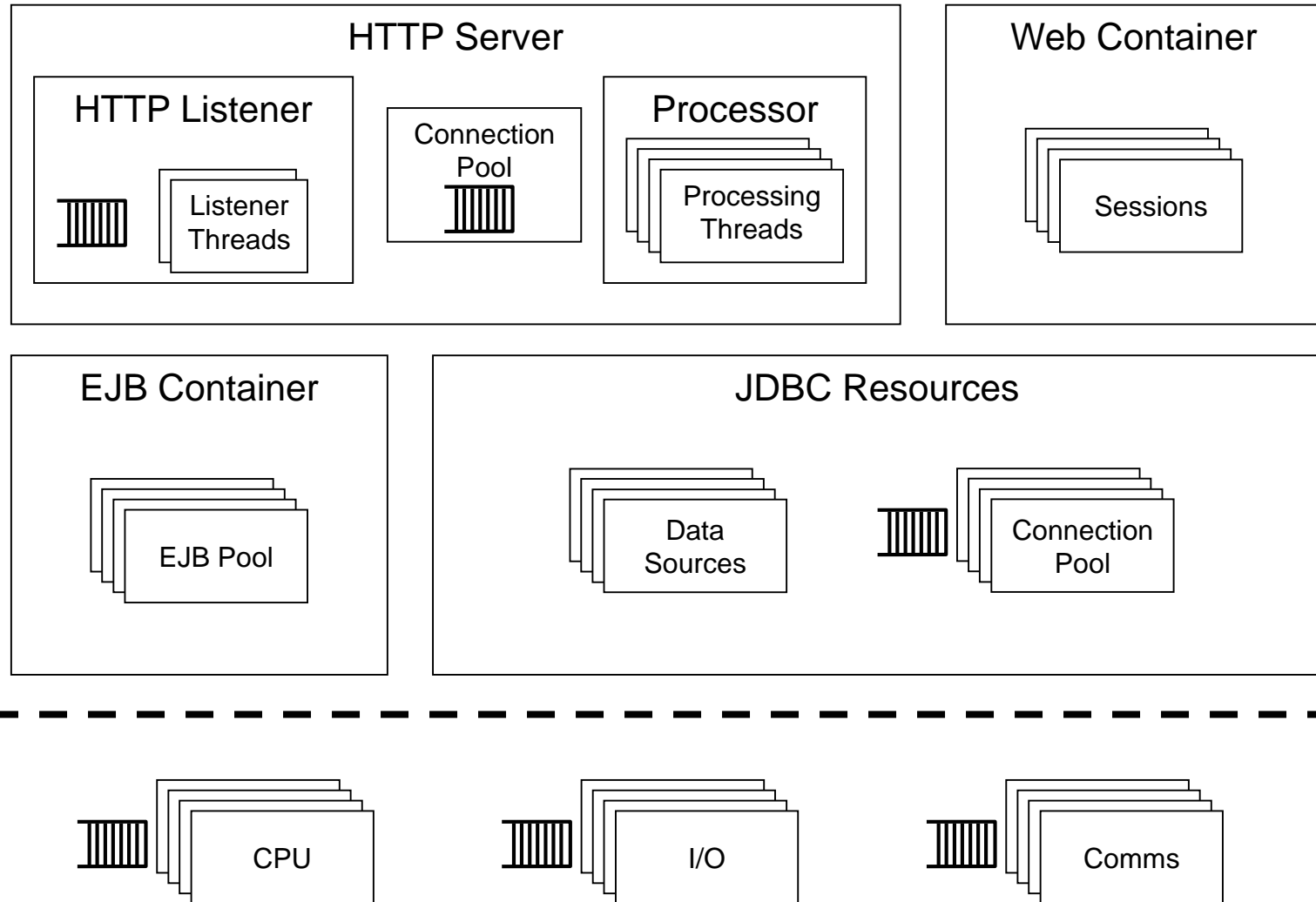
Podemos añadir los diferentes elementos al árbol

- **Thread Group (Grupo de Threads):** Simula un conjunto de usuarios concurrentes que accederán a nuestra aplicación. En el ejemplo de esta práctica únicamente usaremos *thread groups* con un único hilo (un usuario).
- **Config Element (Elemento de configuración):** Define un conjunto de parámetros que pueden ser reutilizados en diferentes grupos de hilos.
- **Sampler (Muestreador):** Es un elemento cuyo cometido es generar una petición a un servidor. La tarea de recoger la respuesta queda relegada a un *listener* (escuchador) que veremos a continuación.
- **Listener (Escuchador):** Recoge la respuesta de la petición anterior, y realiza algún tipo de cálculo o representación de la misma para su posterior análisis. Los *listeners* pueden ser locales al grupo de hilos o globales al plan de pruebas

**En cuanto al rendimiento, el enunciado de la práctica contiene un apéndice en la que se explican los diversos factores que influyen en el rendimiento de una aplicación JavaEE, que el alumno debe leer en detalle y comprender para abordar la ejecución de la práctica**

- Aspectos generales del rendimiento en aplicaciones JavaEE
  - *Best Practices* de Java orientadas al rendimiento
  - Configuración de la *Java Virtual Machine*.
  - Configuración del *hardware* y del sistema operativo.
- Cadena de procesamiento de una aplicación JavaEE.
- Parámetros de configuración de los elementos de la cadena de procesamiento de cara al rendimiento.
  - Servicio HTTP.
  - Contenedores Web.
  - Contenedores de EJBs.
  - Recursos JDBC

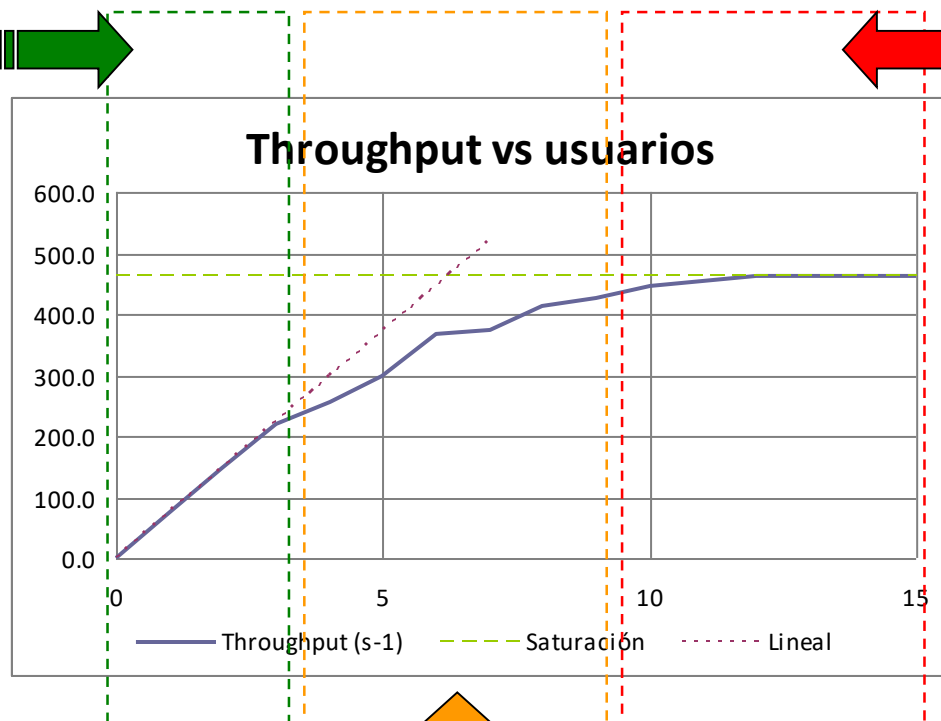
**Los elementos de la cadena de procesamiento no son independientes, ya que comparten recursos en un ordenador. El sistema global es un sistema de colas complejo, en el que el primer elemento que se sature fijará el rendimiento global**



**La curva de rendimiento presenta la productividad (*throughput*) de una aplicación en función de el número de usuarios conectados. Presenta, por tanto, el número de transacciones por segundo que procesa a diferentes niveles de carga**

### 1. Zona lineal

- Bajo número de usuarios
- *Throughput* crece linealmente con el número de usuarios
- No hay saturación en ningún punto de la cadena de procesamiento.
- La latencia del sistema permanece constante al aumentar los usuarios.



### 3. Zona de saturación

- Alcanzada o excedida la capacidad de proceso en algún punto de la cadena de procesamiento.
- *Throughput* constante.
- La latencia del sistema aumenta significativamente con el número de usuarios.
- Se rechazan peticiones. Impacto en la disponibilidad.

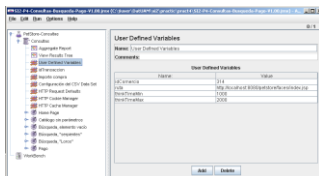
### 2. Zona de transición

- Comienza a haber esperas en elementos de la cadena de procesamiento.
- Crece el *throughput*, pero pierde la linealidad.
- Aumenta la latencia del sistema.



El entorno de pruebas de que se dispone es limitado. Por este motivo, en la práctica se prescinde de la realización de estimaciones de escalado para reflejar un entorno real, limitándonos a obtener la curva como si el entorno utilizado fuera idéntico al de producción final

Herramienta de pruebas



Monitorización

Monitorización valores de rendimiento de Glassfish

Pulse Control-C para terminar

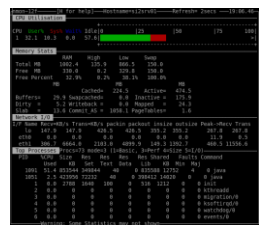
	QCOUNT	QOVERFLOWS	QIMNAVE	PETSPPOOL	VISAPPOOL
HTTPTHREADS	2	0	0	1	0
	3	0	0	2	1
	2	0	0	4	1
	2	0	0	1	0

Fin de la monitorización. Valores medios:

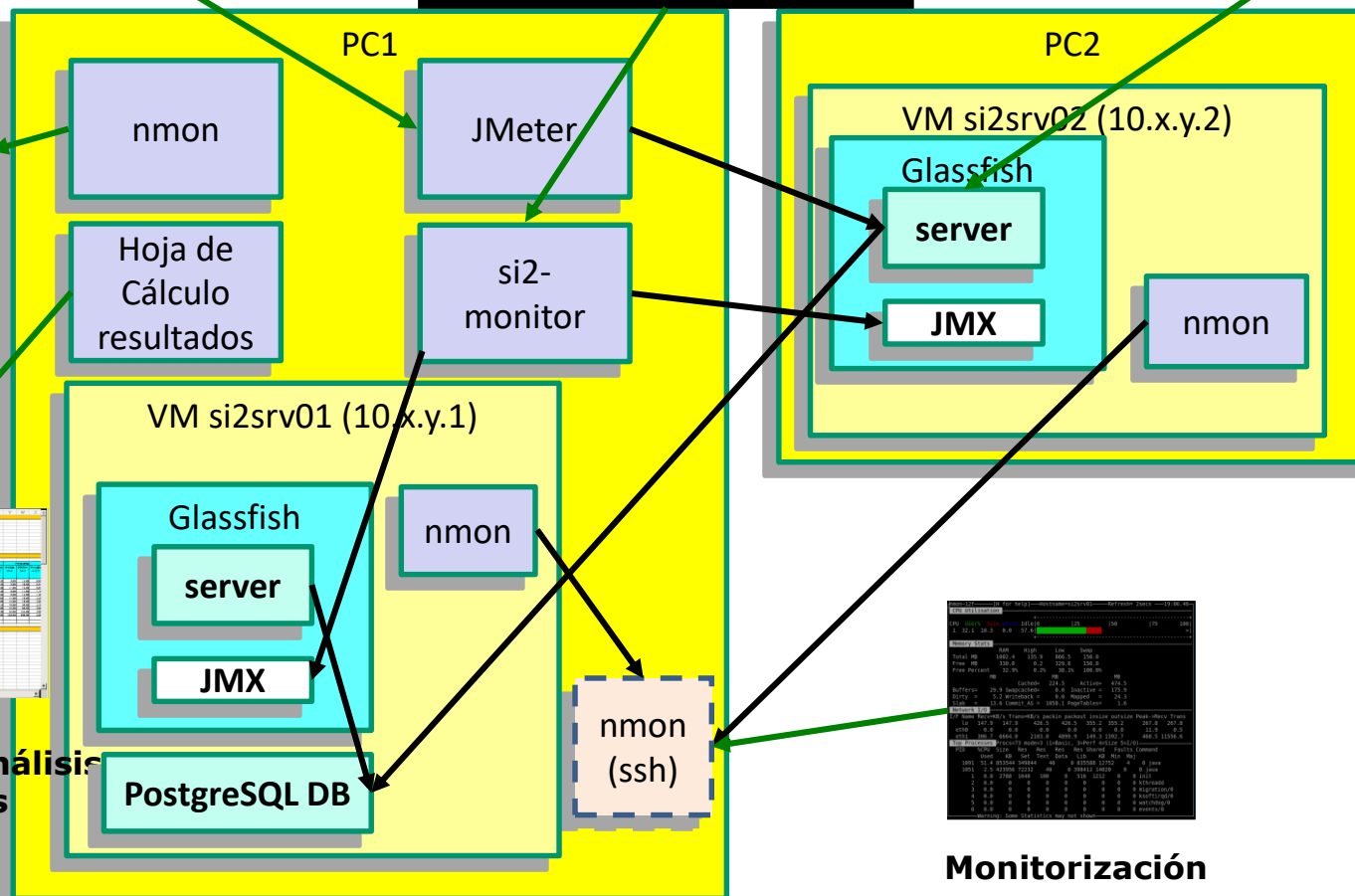
	QCOUNT	QOVERFLOWS	QIMNAVE	PETSPPOOL	VISAPPOOL
HTTPTHREADS	2.25	0	0	1.25	0.25



Proceso



Monitorización



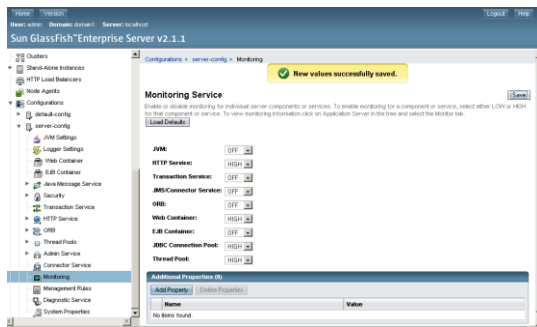
Recogida y análisis de datos



Monitorización

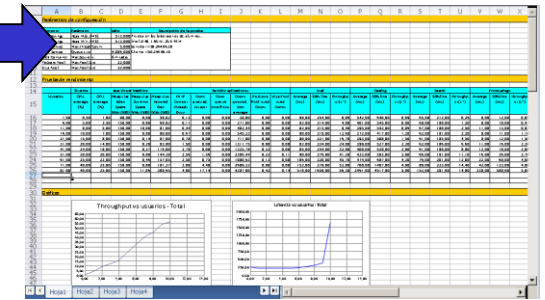
**Antes de comenzar las pruebas, es necesario realizar una configuración adecuada del servidor orientada a rendimiento. Se proporcionan los principales parámetros que es necesario modificar sobre una configuración por defecto de Glassfish. Esta configuración debe permanecer estable a lo largo de todas las pruebas**

1. Editar los parámetros en la consola de administración de Glassfish.



2. Registrar los parámetros de configuración requeridos

Parámetro	Posición en consola administración
Valores máximo y mínimo del heap de memoria que utiliza la máquina virtual Java	Application Server -> Pestaña JVM Settings -> Pestaña JVM Options Parámetro -Xmx y -Xms
Máximo número de conexiones a procesar simultáneamente por el servidor web	Configuration -> HTTP Service -> Pestaña Request Processing -> Parámetro Thread Count
Tamaño máximo de la cola de conexiones pendientes de servicio	Configuration -> HTTP Service -> Pestaña Connection Pool -> Parámetro Queue Size
Máximo número de sesiones en el contenedor Web. (Valor por defecto = 1, limitadas)	Configuration -> Web Container -> Pestaña Manager Properties -> Parámetro Max Sessions
Máximo número de conexiones en pools JDBC	Resources -> JDBC -> Connection Pools Petstore Pool -> Parámetro Maximum Pool Size Visa Pool -> Parámetro Maximum Pool Size



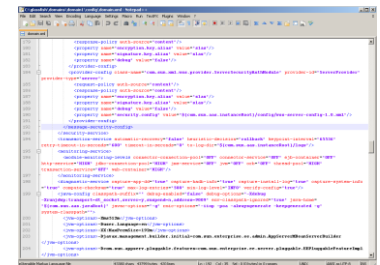
3. Desplegar las aplicaciones

4. Rearrancar el servidor

5. Guardar domain.xml

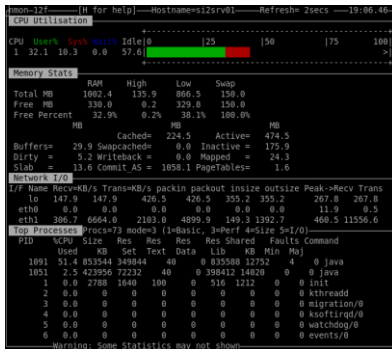
**iOjo! Configurar práctica 2:**

- debug=off,
- uso de DataSource y Prepared Statements,
- ProcesaPago lee parámetros recogidos por ComienzaPago

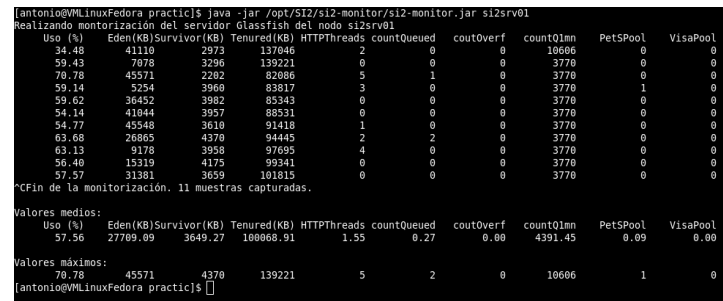


# Tras la configuración del servidor de aplicaciones, es necesario realizar una serie de operaciones para preparar el entorno de pruebas y monitorizar adecuadamente la misma

1. Arrancar la herramienta de monitorización del sistema en ambos ordenadores (*nmon*)



2. Preparar el arranque del programa de monitorización del servidor de aplicaciones (*si2-monitor*)



3. Poner la dirección del servidor de destino en el *script* de pruebas de JMeter

The screenshot shows the 'HTTP Request Defaults' configuration dialog in JMeter. The 'Name' field is set to 'HTTP Request Defaults'. The 'Server Name or IP' field is highlighted with a red circle and contains the value '150.244.64.39'. Other fields like 'Protocol' and 'Path' are also visible.

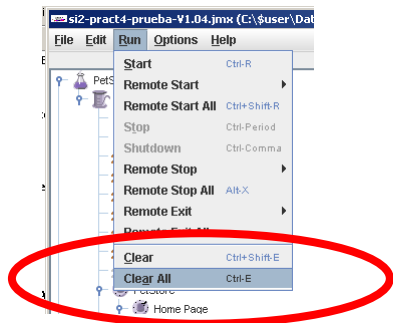
4. Ejecutar unos ciclos de prueba con JMeter sin tomar resultados para preparar todo el entorno

Comprobar con el listener *View Results Tree* que la ejecución es correcta y no hay errores.

**iOjo! Copiar directorio datagen**

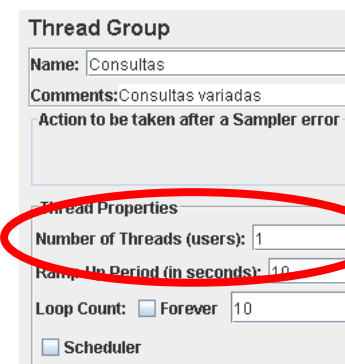
**La curva de rendimiento se obtendrá punto a punto mediante distintas ejecuciones del *script* de pruebas de Glassfish. Cada ejecución nos dará un punto de la curva. Los pasos a seguir para cada ejecución son los siguientes:**

1. Borrar los resultados de la ejecución anterior



2. Borrar el contenido de la tabla de pagos de la base de datos Visa

3. Seleccionar el número de usuarios para la prueba



4. Conmutar el JMeter a la ventana Aggregate Report y comenzar la prueba

Aggregate Report		
Name:	Aggregate Report	
Comments:		
Write results to file / Read from file		
Filename		
Label	# Samples	A
/petstore/aces/...	68	
/petstore/aces/...	35	
Categorías	33	
Gatos 1, "haired"	32	
Gatos 1 parte 2	30	

5. Arrancar el programa de monitorización al terminar la rampa de subida de usuarios

6. Pararlo (Ctrl-C) al comenzar a decrecer el número de usuarios.

```
[antonio@VMLinuxFedora practica]$ java -jar /usr/local/bin/si2-monitor/si2-monitor.jar si2srv01
Realizando monitorización del servidor glassfish de si2srv01
Uso (%)      Eden(KB) Survivor(KB) Tenured(KB) HTTPThreads countQueued countOverf countQlmm PetSPool VisaPool
34.48        41110      2973    137046      2          0          0      10666      0      0
59.43        7078      3296    139221      0          0          0      3770      0      0
70.78        45571      2202    82086       5          1          0      3770      0      0
59.14        5254      3900    83817       3          0          0      3770      1      0
59.62        36452     3982    85343       0          0          0      3770      0      0
54.14        41044     3957    88531       0          0          0      3770      0      0
54.77        45548     3610    91418       1          0          0      3770      0      0
63.68        26065     4370    94445       2          2          0      3770      0      0
63.13        9170      3950    97095       4          0          0      3770      0      0
56.40        15319     4175    99341       0          0          0      3770      0      0
57.57        31381     3659   101815       0          0          0      3770      0      0
Fin de la monitorización. 11 muestras capturadas.

Valores medios:
Uso (%)      Eden(KB) Survivor(KB) Tenured(KB) HTTPThreads countQueued countOverf countQlmm PetSPool VisaPool
57.56        27709.09   3649.27 100068.91   1.55       0.27       0.00    4391.45    0.09    0.00

Valores máximos:
70.78        45571     4370    139221       5          2          0      10666      1          0
[antonio@VMLinuxFedora practica]$
```

7. Salvar el *Aggregated Report* de JMeter y registrar resultados en la hoja de cálculo 12

**Para interpretar los resultados de la prueba, es necesario tener en cuenta las medidas realizadas con la monitorización, y analizar su posible impacto en la saturación del sistema. Algunas preguntas a realizarse para este análisis son:**

- ¿Hay CPU suficiente en el servidor para todo el trabajo que se está realizando?
- ¿Causa algún problema la red?
- ¿Tiene suficiente CPU la JVM para el trabajo del servidor de aplicaciones?
- ¿Se llena la memoria de la JVM?
- ¿Se satura el *pool* de proceso del servidor de aplicaciones?
- ¿Se satura el *pool* de conexión a la base de datos?
- ¿Se alcanza el valor máximo de sesiones permitido por el servidor de aplicaciones?

... y nunca hay que descartar que la prueba no se esté realizando bien...

- ¿Tiene CPU suficiente el ordenador donde se ejecuta el JMeter para crear todos los hilos que emulan a los clientes?
- ¿Es la monitorización la que hace que el servidor se sature?

**Analizar los resultados, y estimar las causas**

# Entrega

- La entrega de los ejercicios de esta práctica se regirá por las normas expuestas de la asignatura.
- Nomenclatura del fichero a entregar  
SI2P1B\_<grupo>\_<pareja>.zip  
(ejemplo: SI2P1B\_2311\_1.zip)
- Contenido del fichero:
  - Código resultante de todas las modificaciones sugeridas: directorio P2.
  - Archivo JMX con el guion de pruebas (P2.jmx) y archivo P2-curvaProductividad.jmx.
  - Archivo SI2-P2-curvaProductividad.ods con los datos recogidos y la curva de productividad.
  - Archivo domain.xml con las modificaciones de la segunda parte de la práctica.
  - Memoria de la ejecución de la práctica con los resultados obtenidos en las 3 partes de la práctica.
- La entrega se realizará la semana del **3 al 7 de abril de 2017**.
- En todos los casos, la entrega se realizará antes del comienzo de la siguiente sesión de prácticas.