		Escuela Politécnica Superior Ingeniería Informática Prácticas de Sistemas Informáticos 2			
Grupo	2311	Práctica	2	Fecha	04/06/2022
Alumno/a		Garitagoitia, Romero, David Teófilo			
Alumno/a					

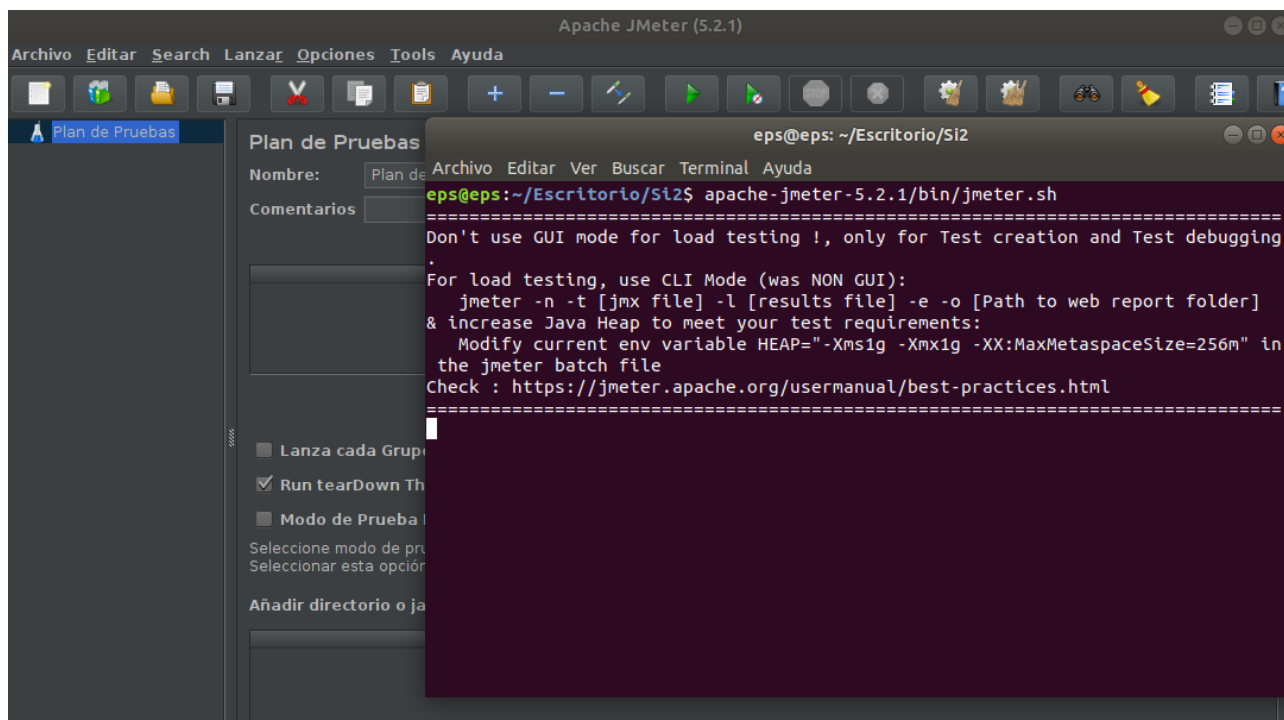
Práctica 2: Rendimiento

Ejercicio número 1:

Siguiendo todos los pasos anteriores, defina el plan completo de pruebas para realizar las tres ejecuciones secuenciales sobre los tres proyectos definidos hasta ahora (P1-base, P1-ws, P1-ejb). Adjunte el fichero generado P2.jmx al entregable de la práctica. Importante: Para comprobar el correcto funcionamiento de la simulación y detectar posibles fallos, se recomienda añadir también al elemento P2 Test un “árbol de resultados” (View Results Tree). Para ello, sobre el plan de pruebas, botón derecho, Add -> Listener -> View Results Tree. Una vez se tenga la certeza de que la simulación funciona correctamente se desactivará el “árbol de resultados” (pulsando encima con el botón derecho del ratón) y se realizará de nuevo la simulación. El árbol de resultados permite inspeccionar los datos enviados en cada petición HTTP y la respuesta obtenida del servidor, que deberán ser correctas. Por ejemplo, no deberá aparecer ningún pago incorrecto en las respuestas

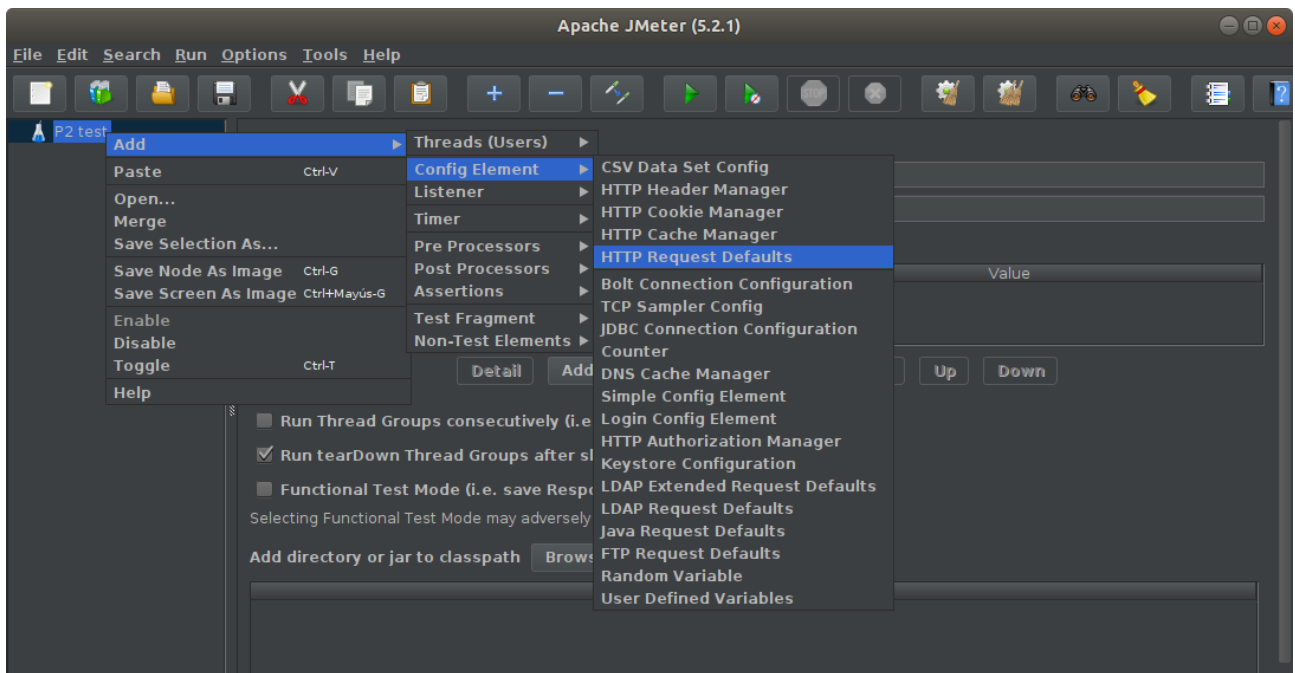
Para la realización de este ejercicio comenzamos con la configuración del JMeter.

Para ello, tras descargar el Jmeter lo abrimos de la siguiente forma:

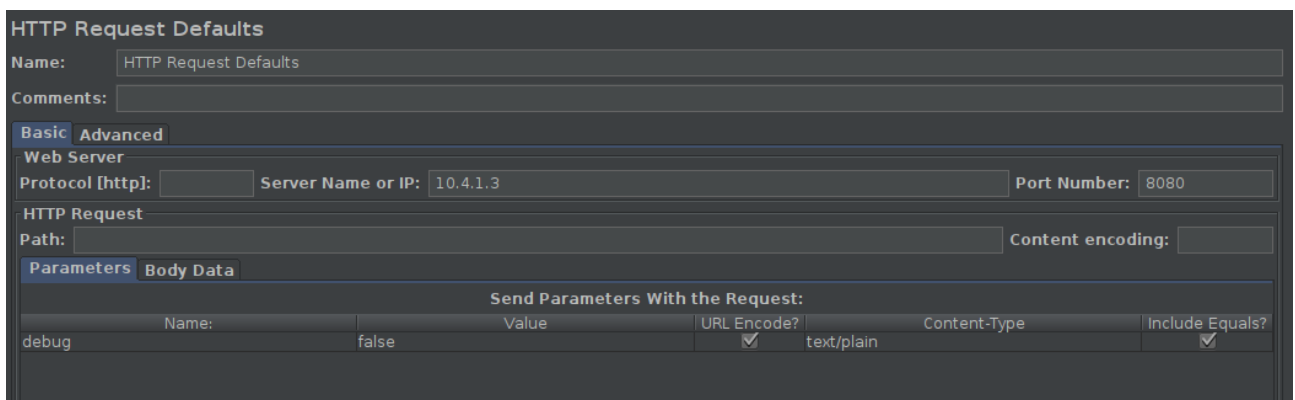


Una vez tengamos abierto el Jmeter, una vez hecho esto pondremos en opciones el idioma inglés.

Proseguimos añadiendo el HTTP Request Defaults el cual nos servirá para facilitar la migración del plan de pruebas, para ello add→config element→HTTP Request default.



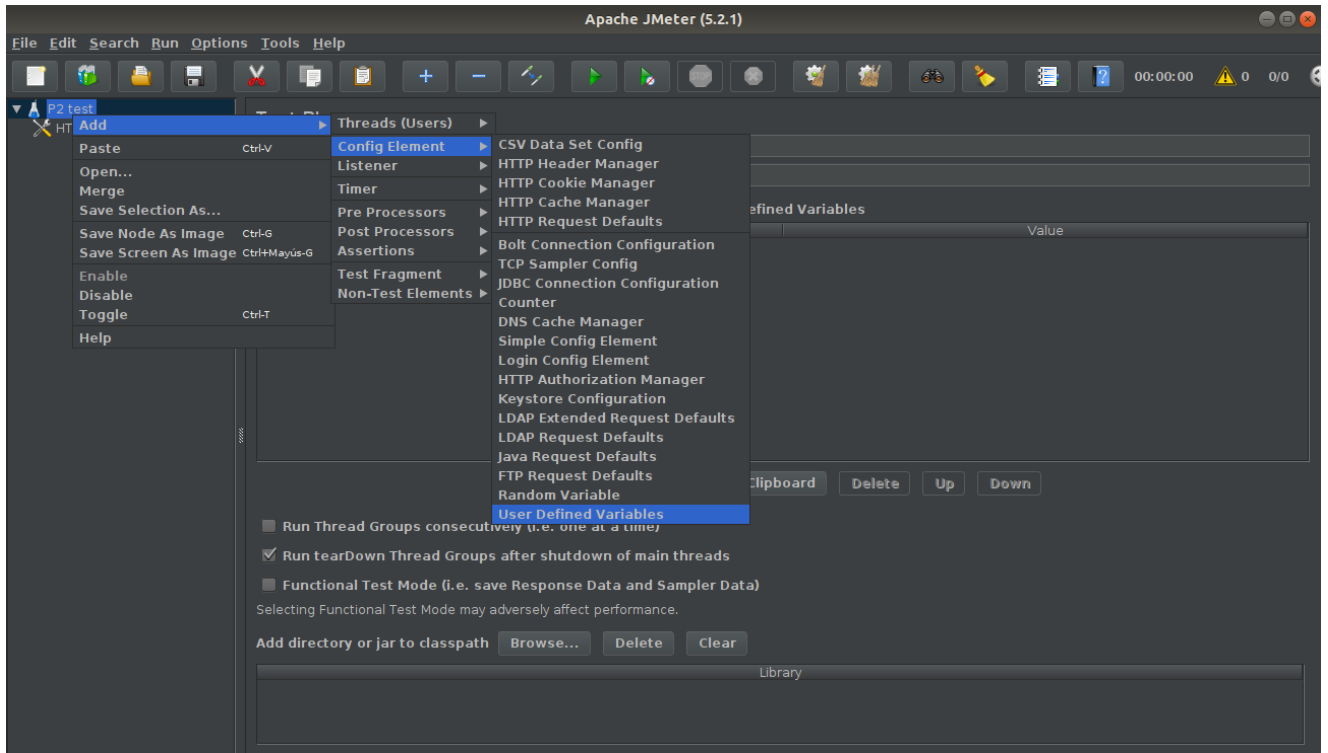
Configuramos los valores del HTTP Requests Defaults para que tengan la IP y puertos oportunos además de añadir el parámetro Debug, el cual pondremos a false para las pruebas (aunque podemos cambiarlo más adelante)



Lo siguiente que debemos hacer es añadir las variables para las pruebas.

Para ello nuevamente deberemos ir al P2 test y añadir un nuevo elemento de configuración, en esta ocasión un User Defined Variables.

Añadimos samples y colocamos a esta con un value de 1000



User Defined Variables

Name:

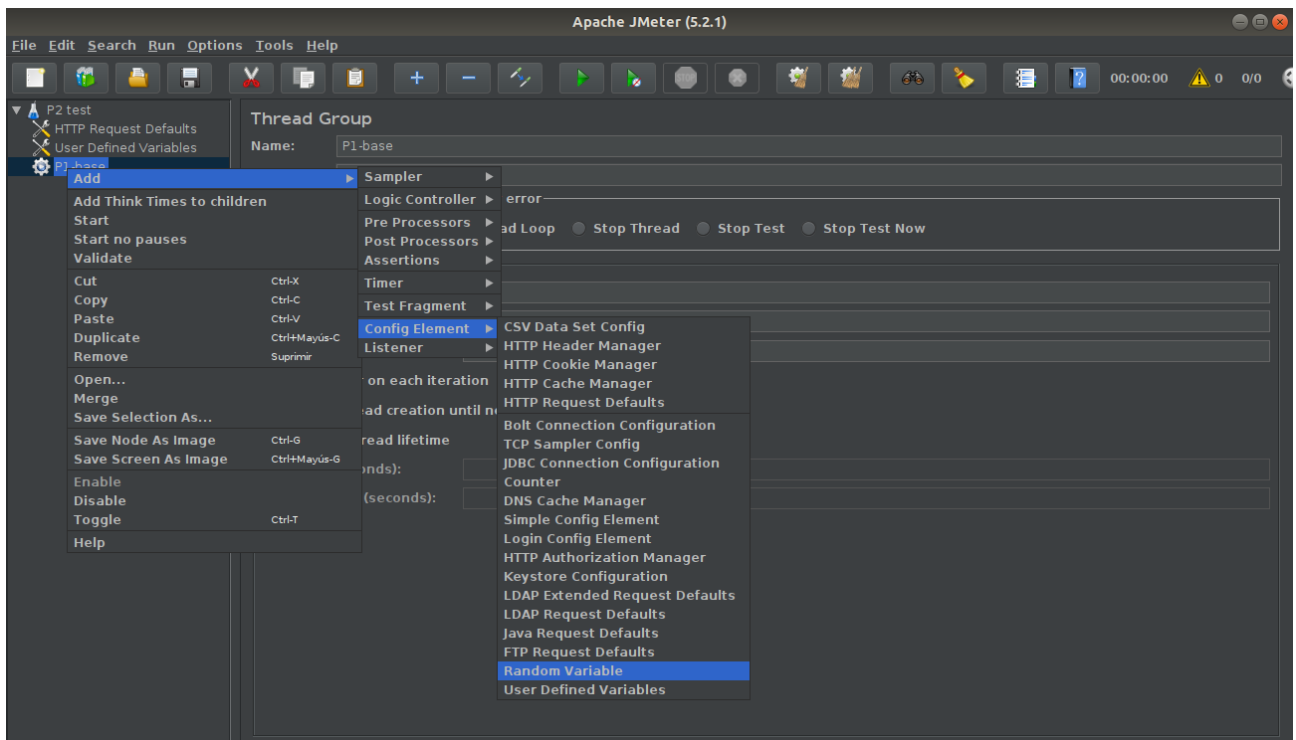
User Defined Variables

Comments:

User Defined Variables		
Name:	Value	Description
samples	1000	

Esta variable nos servirá para definir el número de repeticiones a realizar para cada una de las pruebas.

Ahora añadiremos los hilos volviendo a seleccionar P2 test-> add->Threads (Users)-> Thread Group y configuraremos esto de la forma que se nos indica.



Random Variable

Name: Random Variable

Comments:

Output variable

Variable Name: importe

Output Format:

Configure the Random generator

Minimum Value: 1

Maximum Value: 1000

Seed for Random function:

Options

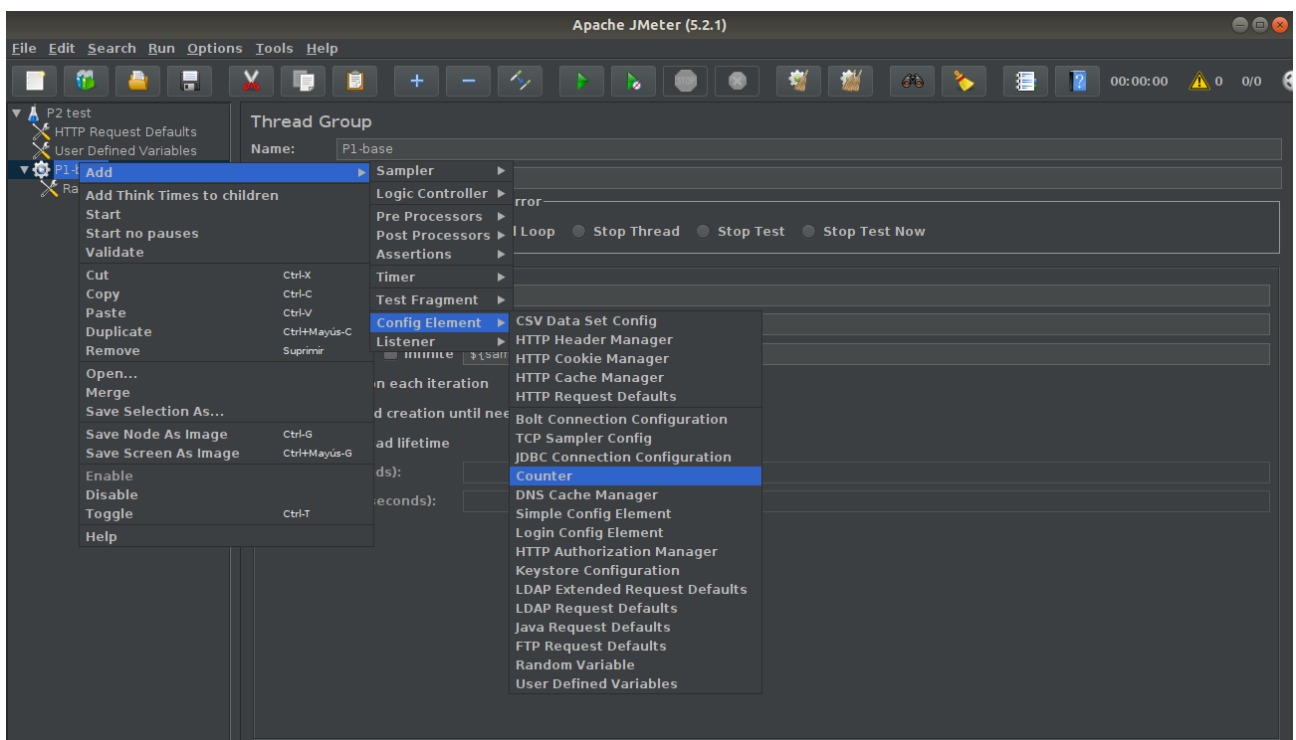
Per Thread(User)?: False

Añadimos Random Variable para introducir elementos dinámicos en las pruebas.

De esta forma haremos que el valor de importe pueda oscilar entre 1 y 1000 en las pruebas de forma aleatoria.

Después de esto añadimos la variable contador que nos servirá para el identificador de transacción que tendrá un valor que no puede repetirse.

Importante recordar poner el exported variable name.



Counter

Name:

Comments:

Starting value

Increment

Maximum value

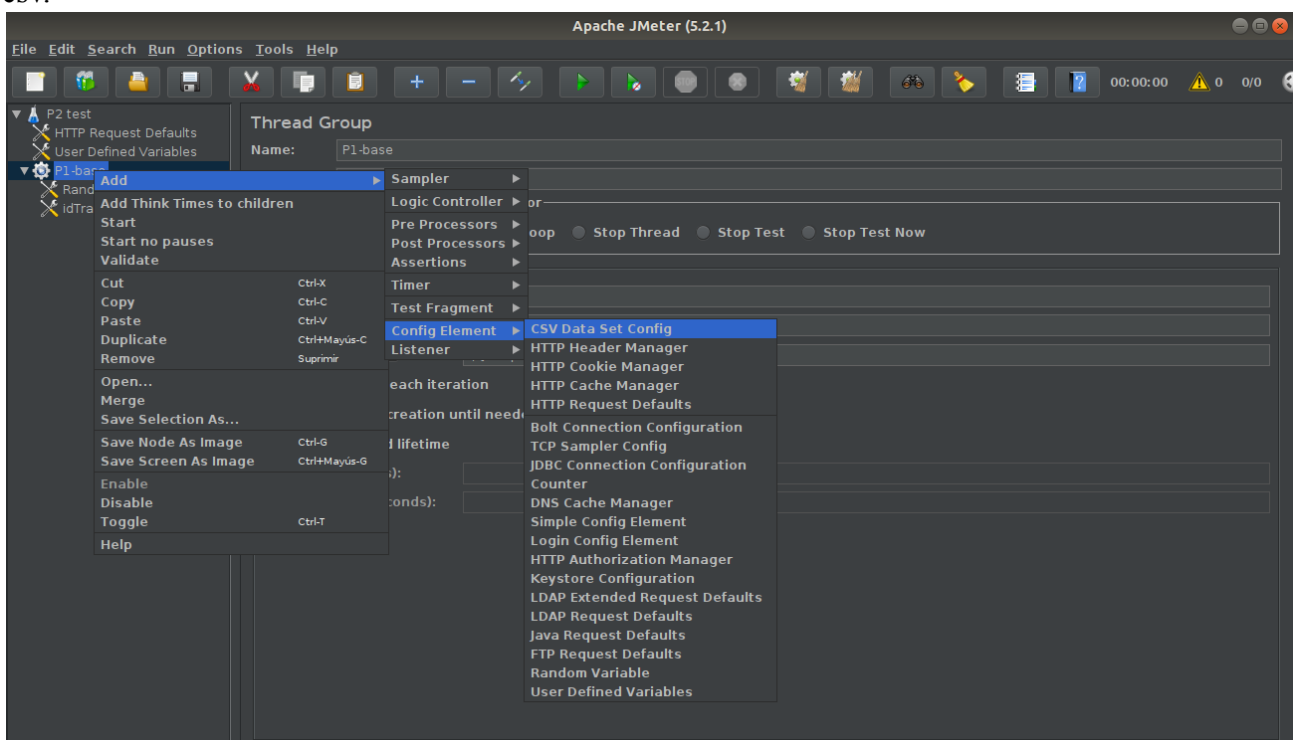
Number format

Exported Variable Name

☐ Track counter independently for each user

☐ Reset counter on each Thread Group Iteration

Añadimos el resto de variables necesarias para las pruebas, los cuales obtenemos directamente del CSV.



CSV Data Set Config

Name:

Comments:

Configure the CSV Data Source

Filename:

File encoding:

Variable Names (comma-delimited):

Ignore first line (only used if Variable Names is not empty):

Delimiter (use '\t' for tab):

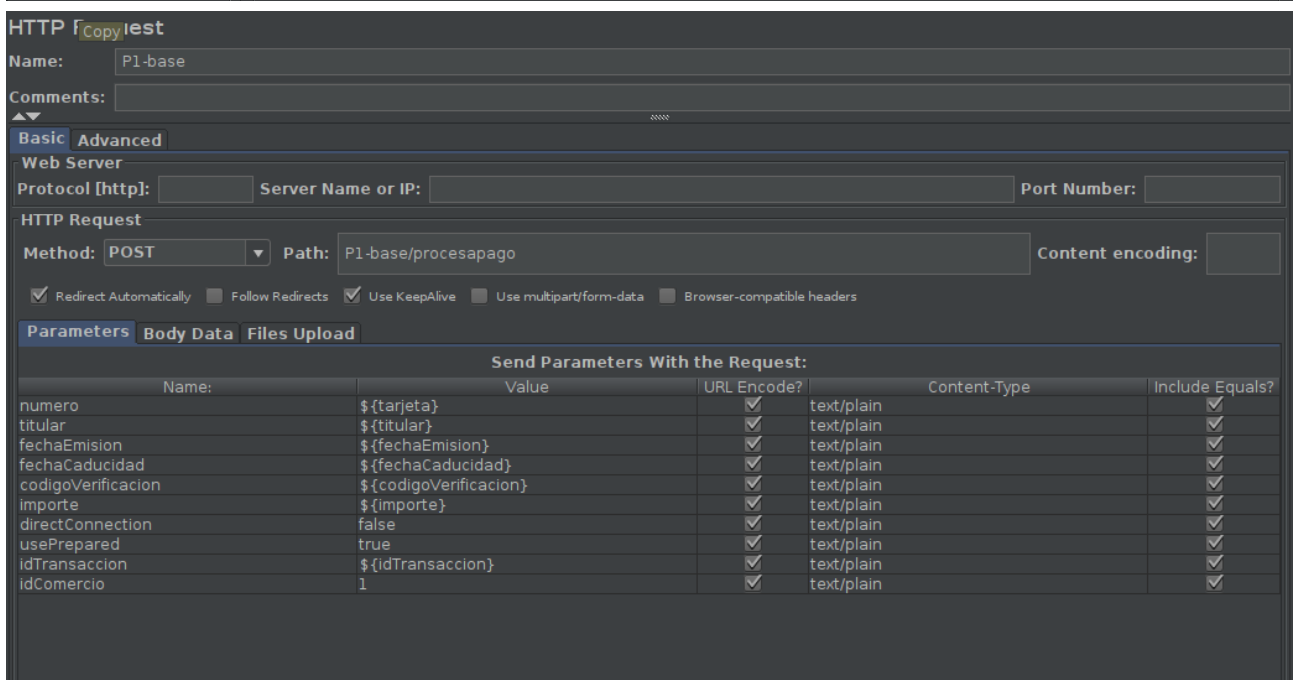
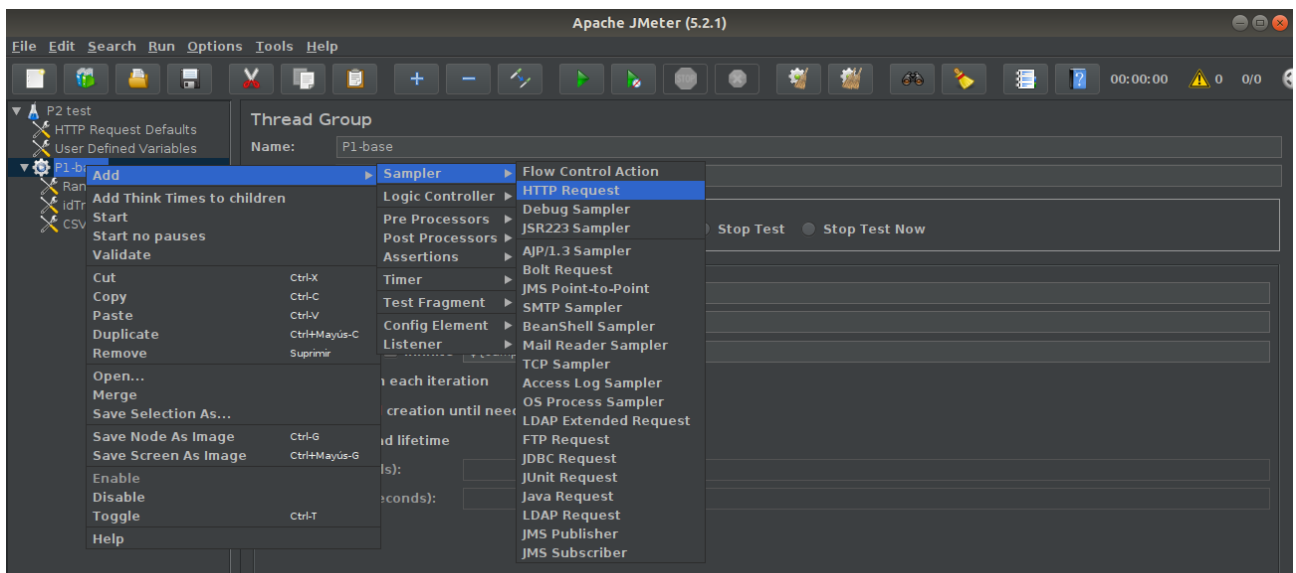
Allow quoted data?:

Recycle on EOF?:

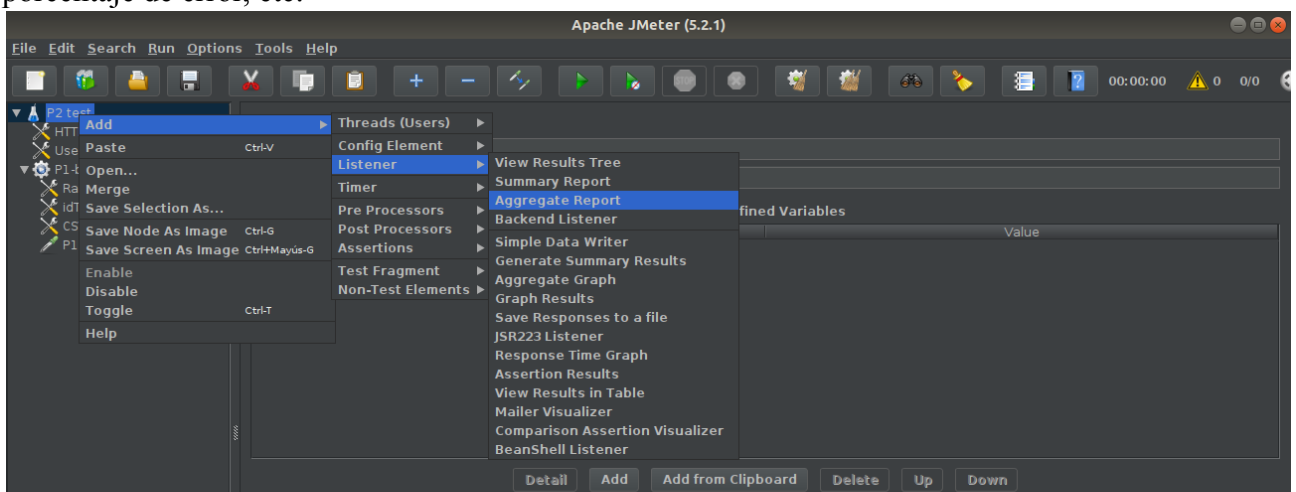
Stop thread on EOF?:

Sharing mode:

Ahora añadimos el elemento más importante para las pruebas, el generador de peticiones HTTP



Ahora al plan de pruebas le añadiremos el listener, en este caso Aggregate Report que nos servirá para extraer información interesante del plan de pruebas como puede ser el tiempo medio de respuesta, el tiempo de respuesta medio correspondiente a la mediana, valor máximo, mínimo, porcentaje de error, etc.



Ahora copiaremos el grupo de hilos para las pruebas en P1-ws-cliente/procesapago y P1-ejb-cliente-remoto/procesapago, para ello tras copiarlo, cambiamos el nombre del grupo de hilos así como los valores del http request para que el path se ajuste a cada uno de ellos, por último debemos modificar el contador idTransaccion para que cada grupo de hilos comience con un valor diferente con el fin de evitar colisiones.

También cambiaremos el idComercio para que sea diferente en cada uno de ellos.

Una vez hemos hecho esto, ya hemos terminado de configurar el P2 test,jmx

Ejercicio número 2:

Preparar el PC con el esquema descrito en la Figura 22. Para ello:

Anote en la memoria de prácticas las direcciones IP asignadas a las máquinas virtuales y al PC

Detenga el servidor de GlassFish del PC host

Inicie los servidores GlassFish en las máquinas virtuales

Repliegue todas las aplicaciones o pruebas anteriores (P1-base, P1-ws, etc), para limpiar posibles versiones incorrectas.

Revise y modifique si es necesario los ficheros build.properties (propiedad “nombre”) de cada versión, de modo que todas las versiones tengan como URL de despliegue las anteriormente indicadas.

Revise y modifique si es necesario el fichero glassfish-web.xml, para indicar la IP del EJB remoto que usa P1-ejb-cliente.

Despliegue las siguientes prácticas: P1-base, P1-ws, P1-ejb-servidor-remoto y P1-jeb-cliente-remoto, con el siguiente esquema:

o El destino de despliegue de la aplicación P1-base será PC2VM con IP 10.X.Y.2 (as.host)

o El destino del despliegue de la parte cliente de P1-ws y de P1-ejb-cliente-remoto será PC2VM con IP 10.X.Y.2 (as.host.client de P1-ws y as.host de P1-ejb-cliente-remoto)

o El destino del despliegue de la parte servidor de P1-ws y de P1-ejb-servidor-remoto será PC1VM con IP 10.X.Y.1 (as.host.server de P1-ws y as.host.server y as.host.client de P1- ejb-servidor-remoto)

o La base de datos en todos ellos será la de PC1VM con IP 10.X.Y.1 (db.host)

Tras detener/iniciar todos los elementos indicados, anotar la salida del comando “free” así como un pantallazo del comando “nmon” (pulsaremos la tecla “m” para obtener el estado de la RAM) tanto en las máquinas virtuales como en el PC host. Anote sus comentarios en la memoria.

Pruebe a ejecutar un pago “de calentamiento” por cada uno de los métodos anteriores y verifique que funciona a través de la página testbd.jsp.

Para la realización de este apartado se emplean las siguientes IPs:

->PC físico: 10.4.1.9

->PC1VM (máquina virtual): 10.4.1.1

>PC2VM (máquina virtual 2): 10.4.1.2

Se procede pues a la modificación de los ficheros de build.properties así como los postgresql.properties para que se adapten a la nueva configuración deteniendo el PostgreSQL de la máquina virtual 2 mediante el comando `sudo /etc/init.d/postgresql-8.4 stop`.

```
si2@si2srv02:~$ sudo /etc/init.d/postgresql-8.4 stop
[sudo] password for si2:
* Stopping PostgreSQL 8.4 database server
si2@si2srv02:~$
```

Tras configurar los build, procedemos a desplegar las aplicaciones y comprobar que se pueden realizar exitosamente los pagos.

En cuanto al free y nmon:

1.4.1.9

```
eps@eps:~$ free
```

	total	usado	libre	compartido	búfer/caché	disponible
Memoria:	8144644	4721632	1474860	298312	1948152	2830964
Swap:	7999484	859904	7139580			

```
mmon-16g-----Hostname=eps-----Refresh= 2secs ---22:44.00---
```

Memory and Swap

PageSize:4KB	RAM-Memory	Swap-Space	High-Memory	Low-Memory
Total (MB)	7953.8	7812.0	- not in use	- not in use
Free (MB)	1436.2	6976.2		
Free Percent	18.1%	89.3%		

Linux Kernel Internal Memory (MB)

	Cached=	1535.1	Active=	4683.7
Buffers=	191.3	Swapcached=	81.1	Inactive =
Dirty =	0.7	Writeback =	0.0	Mapped =
Slab =	238.0	Commit_AS =	16269.0	PageTables=
				85.6

1.4.1.1

```
si2@si2srv01:/opt/glassfish4/glassfish/domains/domain1/logs$ free
```

	total	used	free	shared	buffers	cached
Mem:	767168	755176	11992	0	58840	280684
-/+ buffers/cache:		415652	351516			
Swap:	153592	724	152868			

```
si2@si2srv01:/opt/glassfish4/glassfish/domains/domain1/logs$
```

```
mmon-12f-----Hostname=si2srv01-----Refresh= 2secs ---13:59.21---
```

Memory Stats

	RAM	High	Low	Swap
Total MB	749.2	0.0	749.2	150.0
Free MB	9.6	0.0	9.6	149.3
Free Percent	1.3%	0.0%	1.3%	99.5%

	MB		MB	MB
		Cached=	274.6	Active=
Buffers=	57.5	Swapcached=	0.3	Inactive =
Dirty =	0.1	Writeback =	0.0	Mapped =
Slab =	21.7	Commit_AS =	1132.3	PageTables=
				2.2

1.4.1.2

```
si2@si2srv02:~$ free
```

	total	used	free	shared	buffers	cached
Mem:	767168	759120	8048	0	29572	136008
-/+ buffers/cache:		593540	173628			
Swap:	153592	12232	141360			

```
si2@si2srv02:~$
```


nmon-12f [H for help] Hostname=si2srv02 Refresh= 2secs 14:00.59					
Memory Stats					
	RAM	High	Low	Swap	
Total MB	749.2	0.0	749.2	150.0	
Free MB	8.4	0.0	8.4	138.0	
Free Percent	1.1%	0.0%	1.1%	92.0%	
	MB		MB		MB
		Cached=	132.3	Active=	357.4
Buffers=	27.5	Swapcached=	3.8	Inactive =	356.3
Dirty =	0.1	Writeback =	0.0	Mapped =	22.2
Slab =	17.9	Commit_AS =	1204.6	PageTables=	1.6

Ejercicio número 3:

Ejecute el plan completo de pruebas sobre las 3 versiones de la práctica, empleando el esquema de despliegue descrito anteriormente. Realice la prueba tantas veces como necesite para eliminar ruido relacionado con procesos periódicos del sistema operativo, lentitud de la red u otros elementos.

Compruebe que efectivamente se han realizado todos los pagos. Es decir, la siguiente consulta deberá devolver "3000":

```
SELECT COUNT(*) FROM PAGO;
```

Compruebe que ninguna de las peticiones ha producido un error. Para ello revise que la columna %Error indique 0% en todos los casos.

Una vez que los resultados han sido satisfactorios:

-> Anote los resultados del informe agregado en la memoria de la práctica.

-> Salve el fichero server.log que se encuentra en la ruta glassfish/domains/domain1/logs de Glassfish y adjúntelo con la práctica.

-> Añada a la memoria de prácticas la siguiente información: ¿Cuál de los resultados le parece el mejor? ¿Por qué? ¿Qué columna o columnas elegiría para decidir este resultado?

Incluir el directorio P2 en la entrega.

Repita la prueba de P1-ejb (inhabilite los „Thread Group" P1-base y P1-ws) con el EJB local incluido en P1- ejb-servidor-remoto. Para ello, cambie su „HTTP Request", estableciendo su „Server Name or IP" a 10.X.Y.1 (VM1) y su „Path" a „P1-ejb-cliente/procesapago". Compare los resultados obtenidos con los anteriores.

El fichero P2.jmx entregado no debe contener estos cambios, es decir, debe estar configurado para probar el EJB remoto.

Limpiamos la base de datos y empleamos el archivo bash para redespargar todas las aplicaciones.

Tras realizar las pruebas podemos ver como nos muestra un total de 0 errores lo que indica que las pruebas se pudieron realizar de forma satisfactoria, obteniendo la siguiente información tras la ejecución:

t.jmx (/home/eps/Escritorio/SI2/P2-alumnos/P2 test.jmx) - Apache JMeter (5.2.1)

Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Through...	Received...	Sent KB/...
P1-base	1000	10	7	13	22	48	5	1466	0,00%	21,4/sec	27,50	0,00
P1-ws	1000	100	87	140	158	292	57	1651	0,00%	2,5/sec	3,20	0,00
P1-ejb	1000	20	16	24	27	51	12	1655	0,00%	7,2/sec	9,44	0,00
Total	3000	43	17	107	130	190	5	1655	0,00%	5,1/sec	6,58	0,00

☐ Include group name in label? ☒ Save Table Header

```

21 2022-06-05 21:01:05,019 INFO o.a.j.t.JMeterInread: Inread is done: P1-ws 2-1
22 2022-06-05 21:01:05,019 INFO o.a.j.t.JMeterThread: Thread finished: P1-ws 2-1
23 2022-06-05 21:01:05,019 INFO o.a.j.e.StandardJMeterEngine: Starting ThreadGroup: 3 : P1-ejb
24 2022-06-05 21:01:05,019 INFO o.a.j.e.StandardJMeterEngine: Starting 1 threads for group P1-ejb.
25 2022-06-05 21:01:05,019 INFO o.a.j.e.StandardJMeterEngine: Thread will continue on error
26 2022-06-05 21:01:05,020 INFO o.a.j.t.ThreadGroup: Starting thread group... number=3 threads=1 ramp-up=5 dela
27 2022-06-05 21:01:05,020 INFO o.a.j.t.ThreadGroup: Started thread group number 3
28 2022-06-05 21:01:05,020 INFO o.a.j.e.StandardJMeterEngine: All thread groups have been started
29 2022-06-05 21:01:05,020 INFO o.a.j.t.JMeterThread: Thread started: P1-ejb 3-1
30 2022-06-05 21:03:23,884 INFO o.a.j.t.JMeterThread: Thread is done: P1-ejb 3-1
31 2022-06-05 21:03:23,884 INFO o.a.j.t.JMeterThread: Thread finished: P1-ejb 3-1
32 2022-06-05 21:03:23,884 INFO o.a.j.e.StandardJMeterEngine: Notifying test listeners of end of test
33 2022-06-05 21:03:23,884 INFO o.a.j.s.FileServer: Close: P2-alumnos/datagen/listado.csv
34 2022-06-05 21:03:23,885 INFO o.a.j.g.u.JMeterMenuBar: setRunning(false, *local*)
35

```

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Through...	Received...	Sent KB/...
P1-base	1000	10	7	13	22	48	5	1466	0,00%	21,4/sec	27,50	0,00
P1-ws	1000	100	87	140	158	292	57	1651	0,00%	2,5/sec	3,20	0,00
P1-ejb	1000	20	16	24	27	51	12	1655	0,00%	7,2/sec	9,44	0,00
Total	3000	43	17	107	130	190	5	1655	0,00%	5,1/sec	6,58	0,00

Para comprobar que efectivamente se insertan 3000 compras accedemos a pgadmin donde podemos comprobarlo:

Property	Value
Name	pago
OID	21148
Owner	alumnodb
Tablespace	pg_default
ACL	
Primary key	idautorizacion
Rows (estimated)	2717
Fill Factor	
Rows (counted)	3000
Inherits tables	No
Inherited tables count	0
Has OIDs?	No
System table?	No

Aquí podemos ver como la tabla de pagos cuenta con 3000 filas.

Sin embargo para asegurarnos realizamos la query para poder cerciorarnos.

Efectivamente el resultado muestra 3000.

SQL Editor Graphical Query Builder

Previous queries

```
select count(*) from pago;
```

Output pane

Data Output Explain Messages History

	count bigInt
1	3000

El server log lo guardamos y lo dejamos en la carpeta P2 dentro del fichero server.log.

Para poder dar respuesta a la siguiente pregunta primeramente explicaremos lo que son esas variables que vemos en el JMeter.

Error indica el porcentaje de errores que han tenido lugar en la ejecución de ese grupo de hilos.

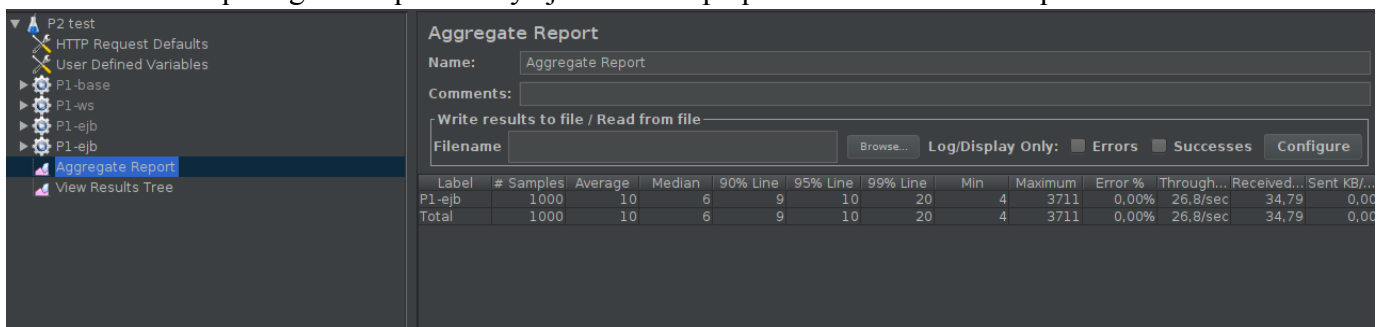
Average nos muestra el tiempo medio en ejecutar cada petición.

Throughput nos indica cual es la productividad.

Sabiendo esto podemos afirmar que el mejor resultado de todos es el de P1-base ya que es el que cuenta con menor Average y mayor Throughput.

Para terminar con el ejercicio se inhabilitan las pruebas de P1-base y P1-ws, repetimos la ejecución de P1-ejb en este caso local.

De esta forma obtenemos los siguientes valores como podemos ver se mejora reduciendo en la mitad el average lo cuál es una gran mejora la cual se debe a que en esta ocasión la ejecución al ser local, las peticiones son dirigidas directamente a la máquina virtual 1 donde también tenemos la base de datos lo que significa que no hay ejecución de paquetes fuera de esta máquina.



Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Through...	Received...	Sent KB/...
P1-ejb	1000	10	6	9	10	20	4	3711	0,00%	26,8/sec	34,79	0,00
Total	1000	10	6	9	10	20	4	3711	0,00%	26,8/sec	34,79	0,00

Ejercicio número 4:

Adaptar la configuración del servidor de aplicaciones a los valores indicados. Guardar, como referencia, la configuración resultante, contenida en el archivo de configuración localizado en la máquina virtual en \$opt/glassfish4/glassfish/domains/domain1/config/domain.xml3. Para obtener la versión correcta de este archivo es necesario detener el servidor de aplicaciones. Incluir este fichero en el entregable de la práctica. Se puede copiar al PC con scp.

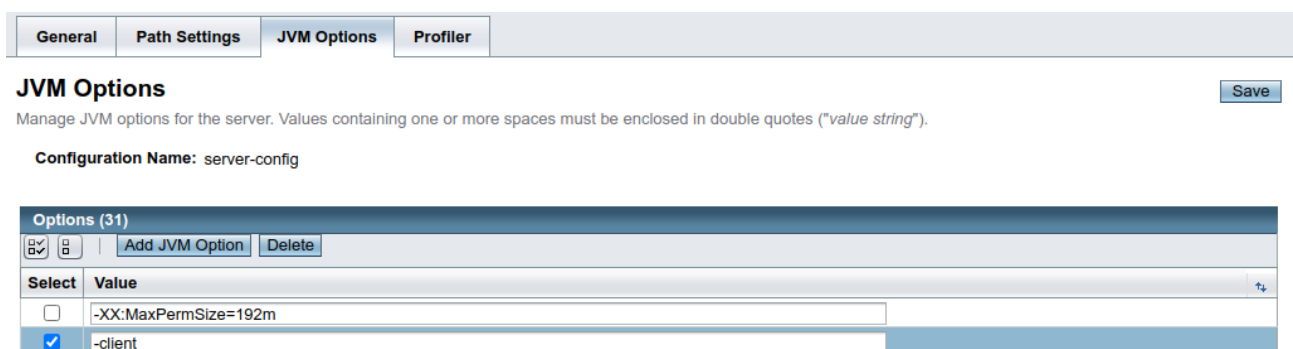
Revisar el script si2-monitor.sh e indicar los mandatos asadmin4 que debemos ejecutar en el PC host para averiguar los valores siguientes, mencionados en el Apéndice 1, del servidor PC1VM1:

1. Max Queue Size del Servicio HTTP
2. Maximum Pool Size del Pool de conexiones a nuestra DB

Así como el mandato para monitorizar el número de errores en las peticiones al servidor web.

Configuramos por tanto el servidor de aplicaciones (1.4.1.2) como se nos ha indicado.

Eliminamos la opción -client añadiendo -server



JVM Options [Save]

Manage JVM options for the server. Values containing one or more spaces must be enclosed in double quotes ("value string").

Configuration Name: server-config



Select	Value
<input type="checkbox"/>	-XX:MaxPermSize=192m
<input checked="" type="checkbox"/>	-client

Añadimos la opción -server así como el mínimo del heap.

JVM Options

Manage JVM options for the server. Values containing one or more spaces must be enclosed in double quotes ("value string").

Configuration Name: server-config

Options (32)	
 	<input type="button" value="Add JVM Option"/> <input type="button" value="Delete"/>
Select	Value
<input type="checkbox"/>	-Xms512m
<input type="checkbox"/>	-Dorg.glassfish.additionalOSGiBundlesToStart=org.apache.felix.shell,org.apache.felix.gogo.runtime,org.apache
<input type="checkbox"/>	-Djavax.management.builder.initial=com.sun.enterprise.v3.admin.AppServerMBeanServerBuilder
<input type="checkbox"/>	-Djavax.net.ssl.keyStore=\${com.sun.aas.instanceRoot}/config/keystore.jks
<input type="checkbox"/>	-server

Quitamos la selección de las opciones para despliegue y recarga automática de aplicaciones y activamos el precompile.

Reload: ☐ **Enabled**
Enables dynamic reloading of applications.

Reload Poll Interval: **Seconds**
Frequency for checking reload requests.

Admin Session Timeout: **Minutes**
A value of 0 means the session never times out.


Auto Deploy Settings

Auto Deploy: ☐ **Enabled**
Automatically deploys applications in the autodeploy directory.

Auto Deploy Poll Interval: **Seconds**
Frequency at which the autodeploy directory is checked for applications; interval does not affect amount of time to load the application or module.

Auto Deploy Retry Timeout: **Seconds**
Time to report failure after a file remains stable in size but cannot be opened.

Auto Deploy Directory:
Directory to monitor for autodeploy applications.

XML Validation: 
Type of deployment descriptor validation.

Verifier: ☐ **Enabled**
Performs detailed verification before deployment.

Precompile: ☒ **Enabled**
Precompiles JSPs, deploys only resulting class files.

Finalmente cambiamos las opciones de monitorización por los siguientes valores:

Monitoring MBeans: ☒ Enabled
Deploy all MBeans needed for monitoring

Component Level Settings (16)		
Select	Module	Monitoring Level
<input type="checkbox"/>	Jvm	HIGH
<input type="checkbox"/>	Transaction Service	OFF
<input type="checkbox"/>	Connector Service	OFF
<input type="checkbox"/>	Jms Service	OFF
<input type="checkbox"/>	Security	OFF
<input type="checkbox"/>	Web Container	HIGH
<input type="checkbox"/>	Jersey(RESTful Web Services)	OFF
<input type="checkbox"/>	Web Services Container	OFF
<input type="checkbox"/>	Java Persistence	OFF
<input type="checkbox"/>	Jdbc Connection Pool	HIGH
<input type="checkbox"/>	Thread Pool	HIGH
<input type="checkbox"/>	Ejb Container	OFF
<input type="checkbox"/>	ORB (Object Request Broker)	OFF
<input type="checkbox"/>	Connector Connection Pool	OFF
<input type="checkbox"/>	Deployment	OFF
<input type="checkbox"/>	Http Service	HIGH

Como se nos indica en el pdf, detenemos el servidor de aplicaciones para obtener la versión correcta y mediante el comando SCP obtenemos el valor del domain.xml empleando el siguiente comando:

```
scp si2@10.4.1.2:/opt/glassfish4/glassfish/domains/domain1/config/domain.xml ./
```

Como podemos ver al examinar el script:

```
asadmin --host $HOST --user $GFUSER --passwordfile $GFPASSFILE |
```

```
mawk -W interactive -vLOGFILE=$LOGFILE '...'
```

En la primera parte se indica el host así como usuario y contraseña, mientras que la segunda parte sirve para indicar la acción a realizar, en este caso obtener (get) una variable seguido del indicador de lo que se desea obtener, es decir, como nos piden la máquina virtual 1, la ip sería la 10.4.1.1, los comandos que se nos piden serían los siguientes:

```
asadmin --host 10.4.1.1 --user admin --passwordfile ./passwordfile get
server.thread-pools.thread-pool.http-thread-pool.max-queue-size
```

```
eps@eps:~/Escritorio/Si2/P2-alumnos$ asadmin --host 10.4.1.1 --user admin --passwordfile ./passwordfile get server.thread-pools.thread-pool.http-thread-pool.max-queue-size
server.thread-pools.thread-pool.http-thread-pool.max-queue-size=4096
Command get executed successfully.
```

```
asadmin --host 10.4.1.1 --user admin --passwordfile ./passwordfile get
resources.jdbc-connection-pool.VisaPool.max-pool-size
```

```
eps@eps:~/Escritorio/Si2/P2-alumnos$ asadmin --host 10.4.1.1 --user admin --passwordfile ./passwordfile get resources.jdbc-connection-pool.VisaPool.max-pool-size
resources.jdbc-connection-pool.VisaPool.max-pool-size=32
Command get executed successfully.
```

```
asadmin --host 10.4.1.1 --user admin --passwordfile ./passwordfile monitor --type httplistener server
```

```
eps@eps:~/Escritorio/Si2/P2-alumnos$ asadmin --host 10.4.1.1 --user admin --passwordfile ./passwordfile monitor --type httplistener server
Error encountered when making remote call: remote failure: Monitoring Registry does not exist. Possible causes are 1) Monitoring is not turned on or at a lower level 2) The corresponding container (web, ejb, etc.) is not loaded yet
Command monitor failed.
```

(aunque se nos indica la máquina virtual 1, este comando lo realizaremos sobre el 10.4.1.2 ya que los cambios para la monitorización se ejecutaron sobre el servidor de aplicaciones)

```

eps@eps:~/Escritorio/Si2/P2-alumnos$ asadmin --host 10.4.1.2 --user admin --passwordfile ./passwordfil
e monitor --type httpListener server
ec  mt  pt  rc
0   0   0.00  0
0   0   0.00  0
0   0   0.00  0

```

Ejercicio número 5:

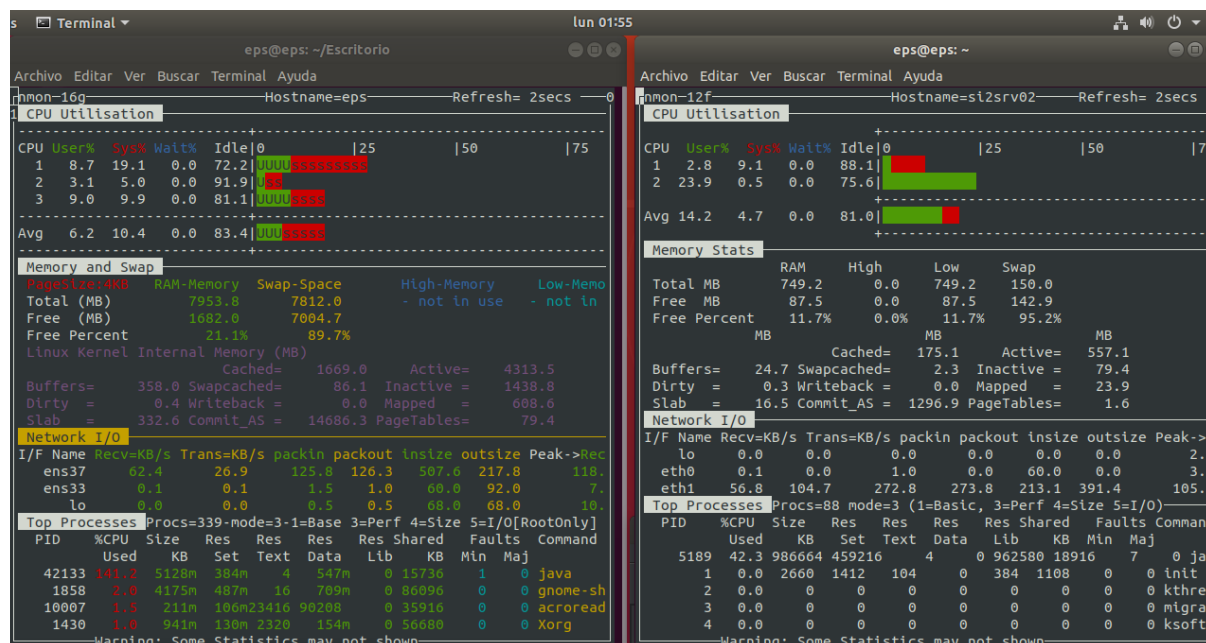
Registrar en la hoja de cálculo de resultados los valores de configuración que tienen estos parámetros.

Valores máximo y mínimo del heap de memoria que utiliza la máquina virtual Java	512m ambos
Máximo número de conexiones a procesar simultáneamente por el servidor web	5
Tamaño máximo de la cola de conexiones pendientes de servicio	4096
Máximo número de sesiones en el contenedor Web. (Valor por defecto -1, ilimitadas)	-1
Máximo número de conexiones en pools JDBC	32

Ejercicio número 6:

Tras habilitar la monitorización en el servidor, repita la ejecución del plan de pruebas anterior. Durante la prueba, vigile cada uno de los elementos de monitorización descritos hasta ahora. Responda a las siguientes cuestiones:

Adjuntamos la captura de pantalla donde mostramos enfrentados los resultados de las pruebas entre ambas capturas de monitorización.



También adjuntamos los valores devueltos por el JMeter

Aggregate Report

Name:Aggregate Report

Comments:

Write results to file / Read from file

Filename

Browse...

Log/Display Only:

Errors

Successes

Configure

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Through...	Received...	Sent KB/...
P1-base	1000	7	6	9	10	13	4	758	0,00%	38,4/sec	49,20	0,00
Total	1000	7	6	9	10	13	4	758	0,00%	38,4/sec	49,20	0,00

Por último, los valores devueltos por el script son los siguientes y podemos ver que efectivamente se realizan los pagos:

eps@eps:~/Escritorio/SI2/P2-alumnos\$./si2-monitor.sh 10.4.1.2			
#Muestra	numJDBCCount	numHTTPCount	numHTTPQ
0	0	1	0
1	0	0	0
2	0	1	0
3	0	1	0
4	0	1	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	1	0
9	0	0	0
10	0	0	0
11	0	1	0
12	0	0	0
13	0	0	0
14	0	1	0
15	0	0	0
16	0	0	0
17	0	0	0
18	0	0	0
19	0	0	0
20	0	1	0
21	0	0	0
22	0	0	0
23	0	0	0
24	0	0	0
25	0	0	0
26	0	0	0
TOT.MUESTRAS	MEDIA:		
27	0	0.296296	0

SQL Editor	Graphical Query Builder
Previous queries	
select count(*) from pago;	
Output pane	
Data Output	Explain Messages
count	bigint
1	1000

Aggregate Report

Name:

Aggregate Report

Comments:

Write results to file / Read from file

Filename

Browse...

Log/Display Only: ☐ Errors ☐ Successes

Configure

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Through...	Received...	Sent KB/...
P1-base	1000	20	9	21	29	48	5	8106	0.00%	12.7/sec	16.27	0.00

- A la vista de los resultados, ¿qué elemento de proceso le parece más costoso? ¿Red? ¿CPU? ¿Acceso a datos? En otras palabras, ¿cuál fue el elemento más utilizado durante la monitorización con nmon en un entorno virtual? (CPU, Memoria, disco ...)

Por los resultados arrojados por nmon podemos afirmar que el proceso con más costo es el de la CPU

- ¿Le parece una situación realista la simulada en este ejercicio? ¿Por qué?

La situación no puede ser considerada realista pues esta es simulada de manera que al estar muy controlado el entorno donde ejecutamos las pruebas y en esta la respuesta del servidor y la red es instantánea además de únicamente contar con un cliente con peticiones secuenciales (se puede ver en el monitor como el numHTTPCount está siempre entre 0 y 1), es por ello que no puede considerarse una situación realista al no contar con las variaciones impredecibles a las que estaría sometido en condiciones reales.

- Teniendo en cuenta cuál ha sido el elemento más saturado, proponga otro esquema de despliegue que resuelva esa situación.

El elemento más saturado como ya hemos mostrado es la CPU, la primera opción que se nos viene a la cabeza es simplemente dirigir más recursos de procesamiento aumentando los cores destinados a estas labores, también podría sustituirse la MV por otra computadora.

Ejercicio número 7:

Preparar el script de JMeter para su ejecución en el entorno de pruebas. Cambiar la dirección destino del servidor para que acceda al host en el que se encuentra el servidor de aplicaciones. Crear también el directorio datagen en el mismo directorio donde se encuentre el script, y copiar en él el archivo listado.csv, ya que, de dicho archivo, al igual que en los ejercicios anteriores, se obtienen los datos necesarios para simular el pago.

A continuación, realizar una ejecución del plan de pruebas con un único usuario, una única ejecución, y un think time bajo (entre 1 y 2 segundos) para verificar que el sistema funciona correctamente. Comprobar, mediante el listener View Results Tree que las peticiones se ejecutan correctamente, no se produce ningún tipo de error y los resultados que se obtienen son los adecuados.

Una vez comprobado que todo el proceso funciona correctamente, desactivar dicho listener del plan de pruebas para que no aumente la carga de proceso de JMeter durante el resto de la prueba.

Este ejercicio no genera información en la memoria de la práctica, realícelo únicamente para garantizar que la siguiente prueba va a funcionar.

Se realizan las pruebas satisfactoriamente, adjuntamos las pruebas:

Aggregate Report

Name:

Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Through...	Received...	Sent KB/s
/P1-base...	10	52	3	21	21	472	3	472	0,00%	13,0/min	0,42	0,00
/P1-base...	10	50	9	13	13	419	7	419	0,00%	13,1/min	0,27	0,00
Total	20	51	7	21	419	472	3	472	0,00%	24,4/min	0,64	0,00

[illegible]

Ejercicio número 8:

Obtener la curva de productividad, siguiendo los pasos que se detallan a continuación:

Previamente a la ejecución de la prueba se lanzará una ejecución del script de pruebas (unas 10 ejecuciones de un único usuario) de la que no se tomarán resultados, para iniciar el sistema y preparar medidas consistentes a lo largo de todo al proceso.

Borrar los resultados de la ejecución anterior. En la barra de acción de JMeter, seleccionar Run -> Clear All.

Borrar los datos de pagos en la base de datos VISA.

Ejecutar la herramienta de monitorización nmon en ambas máquinas, preferiblemente en modo "Data-collect" (Ver 8.2.2).

Seleccionar el número de usuarios para la prueba en JMeter (parámetro C de la prueba)

Conmutar en JMeter a la pantalla de presentación de resultados, Aggregate Report.

Ejecutar la prueba. En la barra de acción de JMeter, seleccionar Run -> Start.

Ejecutar el programa de monitorización si2-monitor.sh

o Arrancarlo cuando haya pasado el tiempo definido como rampa de subida de usuarios en JMeter (el tiempo de ejecución en JMeter se puede ver en la esquina superior derecha de la pantalla).

o Detenerlo cuando esté a punto de terminar la ejecución de la prueba. Este momento se puede detectar observando cuando el número de hilos concurrentes en JMeter (visible en la esquina superior derecha) comienza a disminuir (su máximo valor es C).

o Registrar los resultados que proporciona la monitorización en la hoja de cálculo.

Durante el periodo de monitorización anterior, vigilar que los recursos del servidor si2srv02 y del ordenador que se emplea para realizar la prueba no se saturen. En caso de usar nmon de forma interactiva, se deben tomar varios pantallazos del estado de la CPU durante la prueba, para volcar en la hoja de cálculo del dato de uso medio de la CPU (CPU average %). En caso de usar nmon en modo "Data-collect", esta información se puede ver posteriormente en NMonVisualizer. Una tercera opción (recomendada) es ejecutar el comando vmstat en una terminal remota a la máquina si2srv02, para extraer directamente el valor de uso medio de su CPU 5.

Finalizada la prueba, salvar el resultado de la ejecución del Aggregate Report en un archivo, y registrar en la hoja de cálculo de resultados los valores Average, 90% line y Throughput para las siguientes peticiones:

o ProcesaPago.

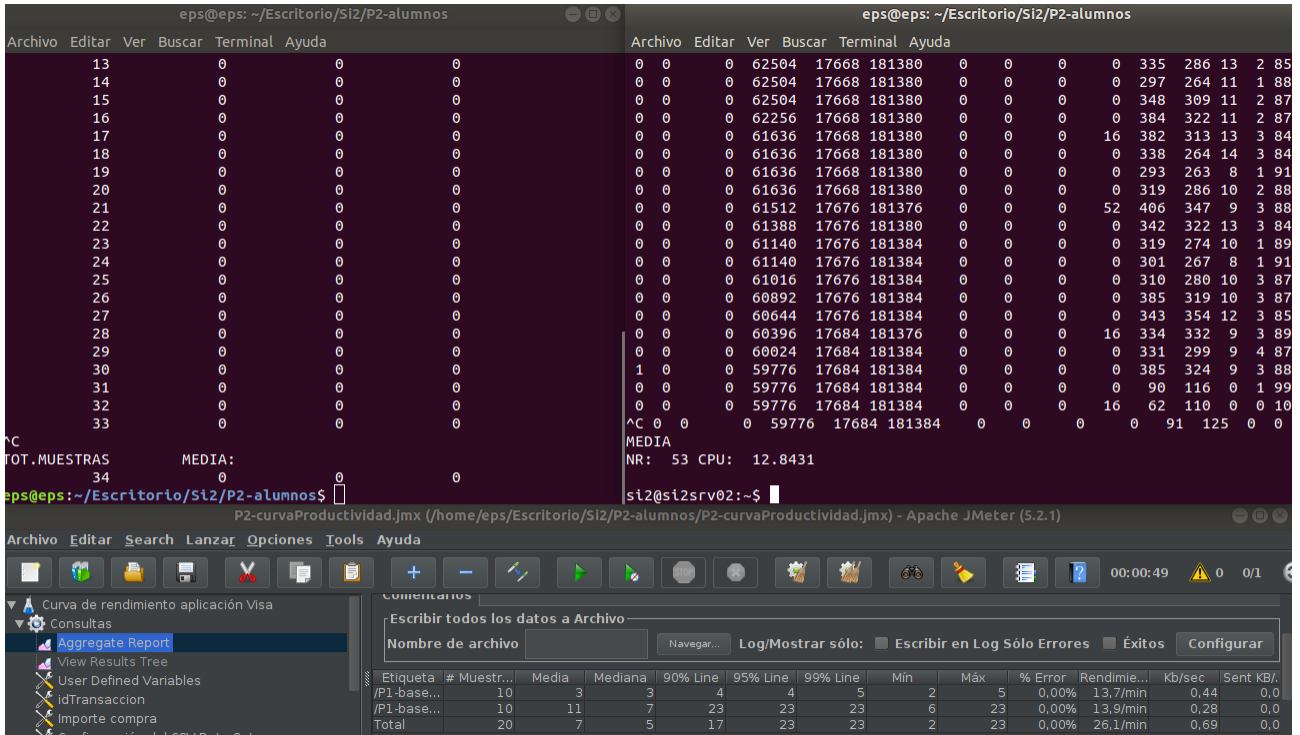
o Total.

Una vez realizadas las iteraciones necesarias para alcanzar la saturación, representar la curva de Throughput versus usuarios. Incluir el fichero P2-curvaProductividad.jmx en la entrega.

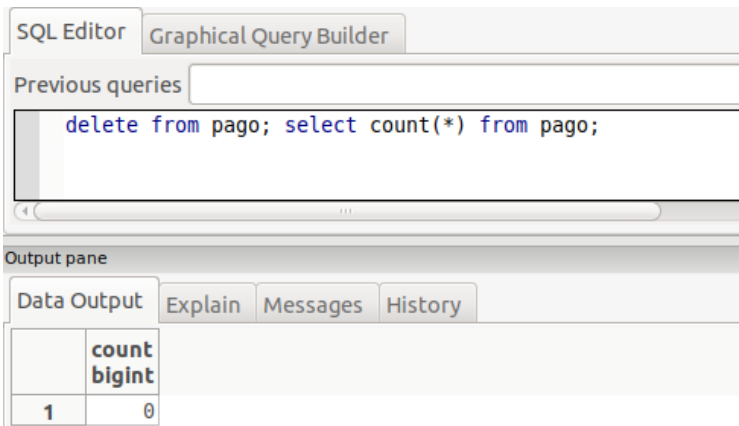
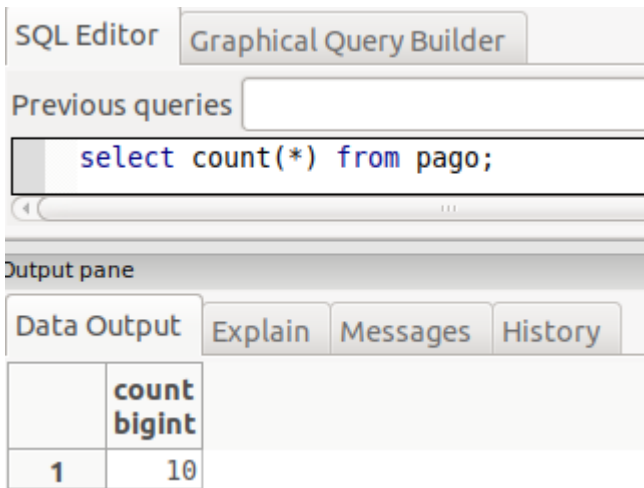
Para la realización de este apartado iremos guardando los valores resultado de la ejecución de la prueba para diferentes.

Para la recolección de la información se emplea la herramienta vmstat acompañada de los resultados arrojados por el monitor.sh y el Aggregate Report.

Los resultados obtenidos para 1 usuario:



Como podemos ver los valores de pago de la base de datos han sido actualizados, así que los borramos para continuar con la siguiente prueba, repitiendo este proceso con el resto de valores:



Para 100:

eps@eps: ~/Escritorio/Si2/P2-alumnos

eps@eps: ~/Escritorio/Si2/P2-alumnos

P2-curvaProductividad.jmx (/home/eps/Escritorio/Si2/P2-alumnos/P2-curvaProductividad.jmx) - Apache JMeter (5.2.1)

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Max	% Error	Rendimiento...	Kb/sec	Sent KB...
/P1-base/comien...	1000	3	3	5	6	13	2	41	0.00%	17.8/sec	34.55	0.00
/P1-base/proces...	1000	8	7	12	16	24	5	48	0.00%	17.8/sec	21.73	0.00
Total	2000	5	6	10	13	23	2	48	0.00%	34.2/sec	54.13	0.00

Para 250:

eps@eps: ~/Escritorio/Si2/P2-alumnos

eps@eps: ~/Escritorio/Si2/P2-alumnos

P2-curvaProductividad.jmx (/home/eps/Escritorio/Si2/P2-alumnos/P2-curvaProductividad.jmx) - Apache JMeter (5.2.1)

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Max	% Error	Rendimiento...	Kb/sec	Sent KB...
/P1-base/comien...	2500	3	3	6	9	22	1	55	0.00%	44.1/sec	85.63	0.00
/P1-base/proces...	2500	9	7	15	22	36	4	74	0.00%	43.8/sec	53.66	0.00
Total	5000	6	5	12	17	33	1	74	0.00%	84.6/sec	133.98	0.00

eps@eps: ~/Escriptorio/Si2/P2-alumnos

```

Archivo Editar Ver Buscar Terminal Ayuda
29      2      0      0
30      1      0      0
31      1      3      0
32      1      2      0
33      0      0      0
34      1      0      0
35      0      0      0
36      0      0      1
37      0      0      0
38      0      0      0
39      1      1      7
40      0      1      7
41      1      0      0
42      0      5      0
43      0      0      0
44      0      0      0
45      0      0      0
46      0      1      1
47      0      0      0
48      0      0      0
49      1      0      0

AC
TOT.MUESTRAS          MEDIA:
50                  0.5              0.78              0.48

eps@eps:~/Escriptorio/Si2/P2-alumnos$

```

eps@eps: ~/Escriptorio/Si2/P2-alumnos

```

Archivo Editar Ver Buscar Terminal Ayuda
1 0 0 18904 18232 185044 0 0 0 0 2453 2557 17 27 56
1 0 0 18748 18232 185092 0 0 0 0 2443 2502 15 24 61
3 0 0 18500 18232 185148 0 0 0 0 16 2642 2714 20 34 45
1 0 0 18160 18232 185196 0 0 0 0 0 2505 2365 22 30 48
1 0 0 17912 18240 185236 0 0 0 0 16 2717 2811 18 31
1 0 0 17664 18240 185296 0 0 0 0 0 2446 2591 18 28 54
^C 0 0 0 17292 18240 185344 0 0 0 0 0 2393 2497 17 25
1 0 0 17168 18240 185396 0 0 0 0 0 2526 2792 20 36 45
0 0 0 16672 18240 185448 0 0 0 0 0 2435 2443 20 28 51
0 0 0 16424 18248 185484 0 0 0 0 16 2558 2751 21 28 51
0 0 0 16268 18248 185540 0 0 0 0 0 2216 2383 15 24 62
0 0 0 15928 18248 185576 0 0 0 0 0 1953 1975 18 21 62
0 0 0 15772 18248 185612 0 0 0 0 0 1950 1858 24 35 40
0 0 0 15308 18248 185644 0 0 0 0 16 1515 1509 21 13 67
2 0 0 15276 18256 185656 0 0 0 0 48 1561 1495 13 16 71
0 0 0 15152 18256 185688 0 0 0 0 0 1366 1316 15 12 73
0 0 0 15028 18256 185700 0 0 0 0 0 823 887 9 5 85
0 0 0 14936 18256 185708 0 0 0 0 64 824 780 12 6 82
0 0 0 14564 18256 185716 0 0 0 0 0 567 455 16 3 82
0 0 0 14564 18256 185716 0 0 0 0 0 150 172 0 0 10
0 0 0 14564 18264 185716 0 0 0 0 12 105 141 0 0 10

MEDIA
NR: 67 CPU: 41.0462

sl2@sl2srv02:~$

```

P2-curvaProductividad.jmx (/home/eps/Escriptorio/Si2/P2-alumnos/P2-curvaProductividad.jmx) - Apache JMeter (5.2.1)

Archivo Editar Search Lanzar Opciones Tools Ayuda

00:01:03 0 0/500

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: ☐ Escribir en Log Sólo Errores ☐ Éxitos ☐ Configurar

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Máx	% Error	Rendimie...	Kb/sec	Sent KB...
/P1-base/comien...	5000	7	3	12	23	79	1	431	0.00%	85.9/sec	166.93	0.00
/P1-base/proces...	5000	13	8	26	39	76	4	410	0.00%	85.6/sec	104.82	0.00
Total	10000	10	6	20	33	77	1	431	0.00%	164.5/sec	260.38	0.00

eps@eps: ~/Escritorio/SI2/P2-alumnos

Archivo	Editar	Ver	Buscar	Terminal	Ayuda
29	3	0		35	
30	0	5		1	
31	2	2		0	
32	1	1		3	
33	0	2		0	
34	4	5		13	
35	1	1		5	
36	1	1		2	
37	1	5		0	
38	0	5		48	
39	0	5		63	
40	4	5		48	
41	2	5		0	
42	2	2		0	
43	0	1		1	
44	1	0		0	
45	0	0		0	
46	1	0		0	
47	0	0		0	
48	3	0		0	
49	0	0		0	

TOT. MUESTRAS MEDIA:

50 1.6 2.84 9.88

eps@eps:~/Escritorio/SI2/P2-alumnos\$

P2-curvaProductividad.jmx (/home/eps/Escritorio/SI2/P2-alumnos/P2-curvaProductividad.jmx) - Apache JMeter (5.2.1)

eps@eps: ~/Escritorio/SI2/P2-alumnos

Archivo	Editar	Ver	Buscar	Terminal	Ayuda	
3	0	0	8080	12916	186172	
2	0	0	7708	12916	186248	
2	0	0	8360	12916	185136	
1	0	0	8020	12924	185212	
3	0	0	7864	12924	185288	
^C	4	0	0	7492	12924	185364
4	0	0	8452	12924	183668	
5	0	0	8204	12924	183748	
5	0	0	7796	12932	183808	
1	0	0	7488	12932	183876	
1	0	0	8608	12932	182220	
2	0	0	8484	12932	182272	
1	0	0	8236	12932	182328	
1	0	0	7988	12940	182360	
1	0	0	7740	12940	182396	
0	0	0	8484	12940	180744	
1	0	0	7988	12940	180760	
0	0	0	7764	12940	180768	
1	0	0	7724	12948	180760	
1	0	0	7724	12948	180760	
0	0	0	7724	12948	180772	

MEDIA
NR: 68 CPU: 59.1818

si2@si2srv02:~\$

Para 850:

eps@eps: ~/Escritorio/Si2/P2-alumnos

eps@eps: ~/Escritorio/Si2/P2-alumnos

TOT. MUESTRAS: 47, MEDIA: 2.51064, 3.61702, 14.9362

eps@eps: ~/Escritorio/Si2/P2-alumnos\$

si2@si2srv02: ~\$

P2-curvaProductividad.jmx (/home/eps/Escritorio/Si2/P2-alumnos/P2-curvaProductividad.jmx) - Apache JMeter (5.2.1)

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Max	% Error	Rendimiento	Kb/sec	Sent KB...
/P1-base/comien...	8500	63	25	192	256	377	1	1208	0.00%	142.3/sec	276.34	0.00
/P1-base/proces...	8500	78	42	217	278	418	4	1615	0.00%	140.5/sec	172.14	0.00
Total	17000	71	33	205	269	405	1	1615	0.00%	270.9/sec	429.13	0.00

Para 1000:

eps@eps: ~/Escritorio/Si2/P2-alumnos

eps@eps: ~/Escritorio/Si2/P2-alumnos

TOT. MUESTRAS: 51, MEDIA: 2.47059, 4.62745, 73.4314

eps@eps: ~/Escritorio/Si2/P2-alumnos\$

si2@si2srv02: ~\$

P2-curvaProductividad.jmx (/home/eps/Escritorio/Si2/P2-alumnos/P2-curvaProductividad.jmx) - Apache JMeter (5.2.1)

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Max	% Error	Rendimiento	Kb/sec	Sent KB...
/P1-base/comien...	10000	172	145	371	439	760	1	1467	0.00%	158.7/sec	308.16	0.00
/P1-base/proces...	10000	191	165	386	455	873	4	1502	0.00%	157.7/sec	193.29	0.00
Total	20000	182	155	379	446	795	1	1502	0.00%	305.4/sec	483.65	0.00

Para 1250:

eps@eps: ~/Escritorio/Si2/P2-alumnos

Archivo Editar Ver Buscar Terminal Ayuda

40 2 5 300

41 5 5 257

42 3 5 201

43 5 5 305

44 4 5 432

45 5 5 455

46 3 5 379

47 5 5 156

48 3 5 272

49 2 5 318

50 3 5 322

51 4 5 231

52 3 0 3

53 3 5 21

54 0 1 0

55 0 0 0

56 2 2 0

57 1 1 0

58 0 3 0

59 1 0 0

60 0 0 0

AC
TOT. MUESTRAS 61
MEDIA: 2.67213 4.44262 281.246

eps@eps:~/Escritorio/Si2/P2-alumnos\$

eps@eps: ~/Escritorio/Si2/P2-alumnos

Archivo Editar Ver Buscar Terminal Ayuda

2 0 19164 7544 3112 107528 20 0 20 0 3076 4019 35 45 19

5 0 19160 8284 2768 106952 0 0 0 0 3230 3796 40 44 14

5 0 19156 7496 2768 107012 16 0 16 0 2301 2437 34 59 7

6 0 19228 8376 2636 105512 28 92 28 92 3436 3803 41 41 15

6 0 19224 7584 2644 105616 0 0 0 0 32 3150 4163 38 45 17

5 0 19260 8788 2644 104640 0 36 0 0 36 3518 4546 38 46 9

5 0 19260 8624 2644 104748 0 0 0 0 0 2842 3984 37 46 17

4 0 19252 8304 2644 104832 0 0 0 0 0 2553 3084 40 49 10

6 0 19252 7636 2644 104936 0 0 0 0 0 2870 4033 40 49 11

3 0 19288 8448 2504 103416 0 36 0 48 3292 4301 37 46 11

6 0 19284 8540 2504 103512 0 0 0 0 0 2994 4642 39 46 15

3 0 19284 8684 2504 103620 0 0 0 0 0 2955 4585 26 51 22

4 0 19284 8876 2504 103708 0 0 0 0 0 2908 4001 25 45 30

5 0 19280 7900 2504 103784 0 0 0 0 0 3438 3482 41 47 12

2 1 19300 7660 2512 102844 156 24 160 52 3164 3425 39 43 15

1 0 19296 7692 2480 102156 0 0 0 1976 2648 2466 33 34 29

3 0 19296 7444 2480 102204 0 0 0 0 2742 2681 22 36 42

2 0 19380 7940 2208 101220 0 84 0 84 1921 1660 17 20 58

1 0 19376 7692 2208 101240 0 0 0 0 1466 1406 18 13 69

0 0 19376 7888 2216 101236 0 0 0 52 739 691 7 7 86

0 0 19376 7888 2216 101248 0 0 0 0 120 163 0 0 10

MEDIA
NR: 78 CPU: 72.5

si2@si2srv02:~\$

P2-curvaProductividad.jmx (/home/eps/Escritorio/Si2/P2-alumnos/P2-curvaProductividad.jmx) - Apache JMeter (5.2.1)

Archivo Editar Search Lanzar Opciones Tools Ayuda

00:01:14 0 0/1250

Curva de rendimiento aplicación Vi

Consultas

Aggregate Report

View Results Tree

User Defined Variables

IdTransaccion

Importe compra

Etiqueta # Muestras Media Mediana 90% Line 95% Line 99% Line Min Máx % Error Rendimie... Kb/sec Sent KB...

/P1-base/comien... 12500 569 591 924 1039 1245 1 1791 0.00% 180.3/sec 350.28 0.00

/P1-base/proces... 12500 586 608 941 1043 1261 4 1763 0.00% 178.5/sec 218.81 0.00

Total 25000 577 599 933 1042 1254 1 1791 0.00% 346.8/sec 549.37 0.00

¿Incluir el nombre del grupo en la etiqueta? Guardar la tabla de datos Guardar la cabecera de la tabla

Para 1300:

eps@eps: ~/Escritorio/Si2/P2-alumnos

Archivo Editar Ver Buscar Terminal Ayuda

45 4 5 443

46 4 5 356

47 4 5 307

48 1 5 346

49 5 5 205

50 5 5 216

51 2 5 359

52 4 5 525

53 5 5 549

54 4 5 494

55 3 5 252

56 2 5 64

57 4 5 22

58 0 1 1

59 0 1 0

60 2 2 0

61 1 0 0

62 1 0 0

63 0 0 0

64 0 0 0

65 0 0 0

AC
TOT. MUESTRAS 66
MEDIA: 2.65152 4.45455 338.348

eps@eps:~/Escritorio/Si2/P2-alumnos\$

eps@eps: ~/Escritorio/Si2/P2-alumnos

Archivo Editar Ver Buscar Terminal Ayuda

7 0 51492 7828 424 23928 0 0 128 0 3160 4461 38 48 14

5 0 51484 7908 424 23992 0 0 0 0 2851 4099 39 49 12

4 0 51328 7568 424 24120 156 0 156 0 2810 3674 41 48 11

2 0 51324 7600 424 24184 0 0 0 0 2919 4121 39 49 13

4 0 51852 8552 416 23272 0 528 0 560 4385 5212 26 47 22

6 0 51828 8888 416 23288 24 0 24 0 2902 4072 52 37 11

3 0 51796 8680 416 23448 88 0 88 0 2562 2903 42 53 5

7 0 51752 7940 416 23888 32 0 360 0 2317 2606 37 54 8

4 0 51640 7540 416 23896 188 0 188 0 2946 3174 46 44 10

1 0 51608 7688 424 24036 52 0 52 16 3214 4426 40 44 17

4 0 51588 8460 424 24088 0 0 0 0 3127 4804 37 46 18

1 0 51544 8672 424 24156 24 0 24 0 2731 3440 36 52 13

3 0 51532 8976 424 24248 0 0 0 1956 2874 3641 36 45 20

0 0 51488 8888 424 24356 4 0 4 0 2844 2959 24 38 37

2 0 51464 8920 432 24388 0 0 0 20 2601 2612 20 24 56

0 0 51456 9192 432 24424 0 0 0 0 2281 2471 29 38 33

0 0 51444 9100 432 24432 32 0 32 0 1510 1521 13 13 73

0 0 51424 9132 432 24492 0 0 0 0 930 881 11 8 82

0 0 51396 9132 432 24496 16 0 16 0 492 429 12 2 86

0 0 51396 9132 432 24484 0 0 0 0 370 279 8 2 90

0 0 51376 9212 440 24496 0 0 0 12 421 303 10 2 88

MEDIA
NR: 82 CPU: 72.65

si2@si2srv02:~\$

P2-curvaProductividad.jmx (/home/eps/Escritorio/Si2/P2-alumnos/P2-curvaProductividad.jmx) - Apache JMeter (5.2.1)

Archivo Editar Search Lanzar Opciones Tools Ayuda

00:01:16 0 0/1300

Curva de rendimiento aplicación Vi

Consultas

Aggregate Report

View Results Tree

User Defined Variables

IdTransaccion

Importe compra

Etiqueta # Muestras Media Mediana 90% Line 95% Line 99% Line Min Máx % Error Rendimie... Kb/sec Sent KB...

/P1-base/comien... 13000 682 755 1136 1297 1416 1 2400 0.00% 180.4/sec 350.41 0.00

/P1-base/proces... 13000 706 769 1185 1313 1424 4 2261 0.00% 180.0/sec 220.76 0.00

Total 26000 694 762 1158 1305 1419 1 2400 0.00% 348.8/sec 552.78 0.00

¿Incluir el nombre del grupo en la etiqueta? Guardar la tabla de datos Guardar la cabecera de la tabla

Para 1400:

eps@eps: ~/Escritorio/Si2/P2-alumnos

eps@eps: ~/Escritorio/Si2/P2-alumnos

TOT. MUESTRAS: 67 MEDIA: 2.65672 4.67164 455.03

eps@eps: ~/Escritorio/Si2/P2-alumnos\$

si2@si2srv02: ~\$

P2-curvaProductividad.jmx (/home/eps/Escritorio/Si2/P2-alumnos/P2-curvaProductividad.jmx) - Apache JMeter (5.2.1)

00:01:27 0/1400

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Max	% Error	Rendimie...	Kb/sec	Sent KB...
/P1-base/comien...	14000	1092	1153	1676	1951	3552	1	7409	0.00%	168.3/sec	326.96	0.00
/P1-base/proces...	14000	1130	1174	1675	1967	3744	5	7478	0.00%	167.3/sec	205.18	0.00
Total	28000	1111	1163	1675	1955	3603	1	7478	0.00%	326.6/sec	517.52	0.00

Para 1500:

eps@eps: ~/Escritorio/Si2/P2-alumnos

eps@eps: ~/Escritorio/Si2/P2-alumnos

TOT. MUESTRAS: 47 MEDIA: 2.82979 4.87234 503.383

eps@eps: ~/Escritorio/Si2/P2-alumnos\$

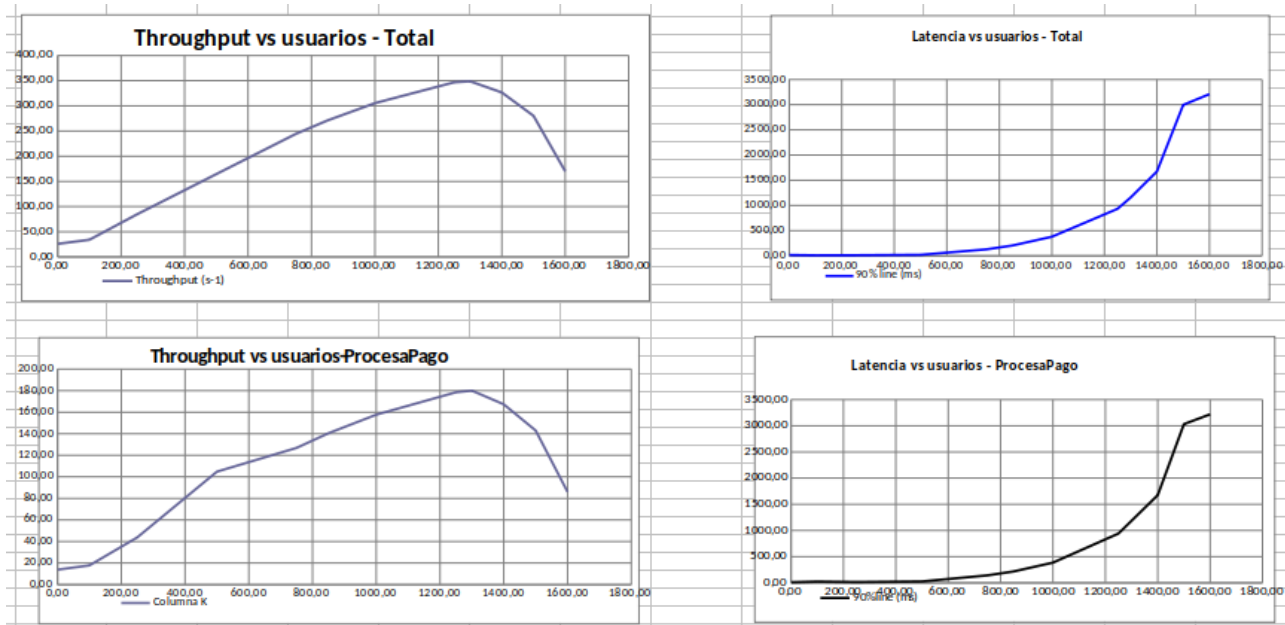
si2@si2srv02: ~\$

P2-curvaProductividad.jmx (/home/eps/Escritorio/Si2/P2-alumnos/P2-curvaProductividad.jmx) - Apache JMeter (5.2.1)

00:01:49 0/1500

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Max	% Error	Rendimie...	Kb/sec	Sent KB...
/P1-base/comien...	15000	1529	1332	2970	3701	6438	1	31729	0.00%	143.2/sec	278.27	0.00
/P1-base/proces...	15000	1585	1350	3032	3711	6214	4	31807	0.00%	143.0/sec	175.34	0.00
Total	30000	1557	1342	2996	3709	6369	1	31807	0.00%	280.2/sec	443.97	0.00

Como resultado de estas mediciones obtenemos las siguientes gráficas:

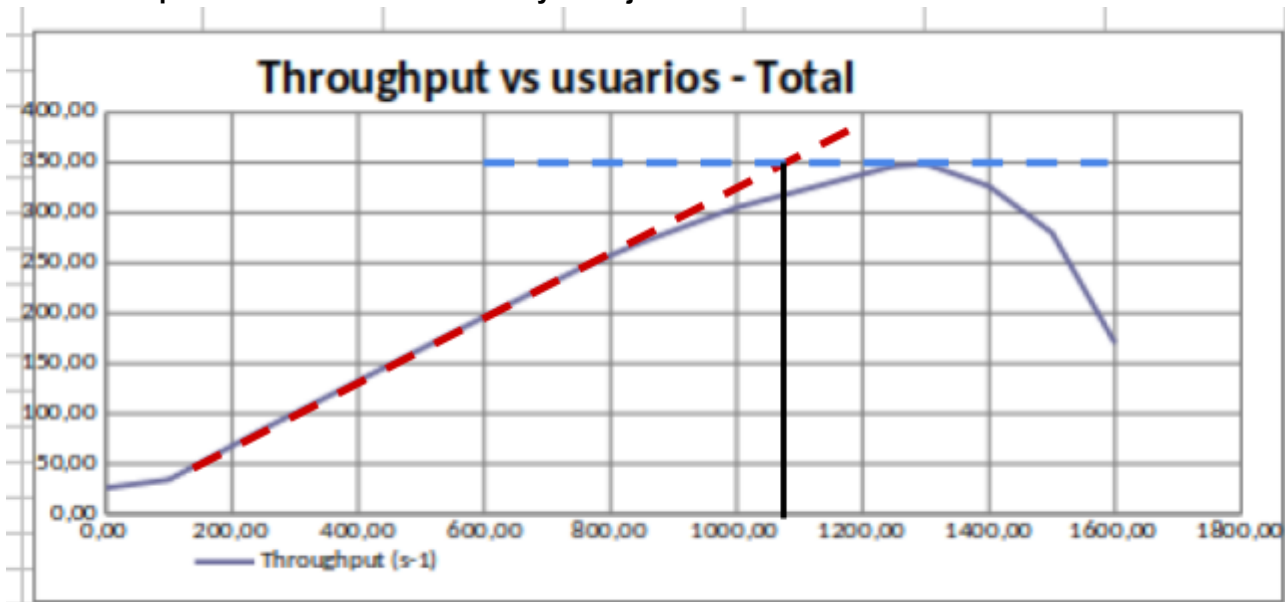


Ejercicio número 9:

A partir de la curva obtenida, determinar para cuántos usuarios conectados se produce el punto de saturación, cuál es el throughput que se alcanza en ese punto, y cuál el throughput máximo que se obtiene en zona de saturación.

Analizando los valores de monitorización que se han ido obteniendo durante la elaboración de la curva, sugerir el parámetro del servidor de aplicaciones que se cambiaría para obtener el punto de saturación en un número mayor de usuarios.

Realizar el ajuste correspondiente en el servidor de aplicaciones, reiniciarlo y tomar una nueva muestra cercana al punto de saturación. ¿Ha mejorado el rendimiento del sistema? Documente en la memoria de prácticas el cambio realizado y la mejora obtenida.



Por los datos que muestra la gráfica, se infiere que el punto de saturación se encuentra en torno a los 1300 usuarios, punto en el que la latencia se dispara y el throughput empieza a decaer.

El throughput máximo es por tanto 348,8, a partir de ese valor entramos en zona de saturación dentro del cual el rendimiento va en declive, estando el punto de saturación sobre entre 1000 y 1100 alrededor de los 1075

Como ya vimos en puntos anteriores, la CPU es el punto que más costo genera en el procesamiento, es por ello que se decide aumentar el pool de hilos y volver a tomar medidas.

Edit Thread Pool

Save Cancel

Modify an existing thread pool.

Load Defaults

Configuration Name: server-config

Name: http-thread-pool

Class Name:
The name of the class that implements the thread pool

Max Queue Size:
The maximum number of threads in the queue. A value of -1 indicates that there is no limit to the queue size.

Max Thread Pool Size:
The maximum number of threads in the thread pool

Min Thread Pool Size:
The minimum number of threads in the thread pool

Idle Thread Timeout: Seconds
The maximum amount of time that a thread can remain idle in the pool. After this time expires, the thread is removed from the pool.

Repetimos la prueba para el valor 1000 al ser un punto próxima al de saturación y en el cual realizamos pruebas sin los cambios para observar las diferencias, podemos ver como el rendimiento ha sido superior en esta ocasión al pasar de los 305/sec a los 330/sec en rendimiento lo que implica una mejora cercana al 8,197% en el nuevo rendimiento tras aplicar los cambios con respecto a los valores originales, lo cual demuestra que ha habido una mejora en el rendimiento al administrar las peticiones más rápido mejorando la eficiencia.

Archivo Editar Search Lanzar Opciones Tools Ayuda

Curva de rendimiento aplicación Visa

Consultas

- Aggregate Report
- View Results Tree
- User Defined Variables
- IdTransaccion
- Importe compra
- Configuración del CSV Data Set
- HTTP Request Defaults
- HTTP Cookie Manager
- HTTP Cache Manager
- Pago

Informe Agregado

Nombre: Aggregate Report

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar...

Log/Mostrar sólo: ☐ Escribir en Log Sólo Errores ☐ Éxitos

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Máx	% Error	Rendimie...	Kb/sec	Sent KB...
/P1-base/comien...	10000	12	5	27	49	113	1	1043	0,00%	171,6/sec	333,33	0,00
/P1-base/proces...	10000	24	16	50	76	138	4	1016	0,00%	171,6/sec	210,43	0,00
Total	20000	18	10	40	64	128	1	1043	0,00%	329,9/sec	522,75	0,00