Práctica 2

Germán Esquinazi Bachoer
Patricia Losana Ferrer

12 de abril de 2018

Ejercicio 1

Siguiendo todos los pasos que se indican en el guión, hemos generado el fichero P2.jmx (que puede encontrarse dentro de la ruta '/Entregables/Ejercicio1' localizada en el directorio de la práctica).

Observaciones: para ejecutar el .jmx es necesario el directorio apache-jmeter-4.0. Debido a que el tamaño del mismo es de 300MB y el tamaño máximo permitido en la entrega es de 10MB, no hemos podido incluirlo.

Ejercicio 2

Si analizamos el diagrama de despliegue facilitado en el guión de la práctica (y los datos del enunciado del ejercicio), las direcciones IP asignadas a cada PC dentro de los ficheros build.properties y postgresql.properties para cada aplicación son:

	base	ws	ejb-servidor-remoto	ejb-cliente-remoto
build.properties				
as.host:	10.5.7.2	-	-	10.5.7.2
as.host.client:	-	10.5.7.2	10.5.7.2	-
as.host.server:	-	10.5.7.1	10.5.7.2	-
postgresql.properties				
db.host:	10.5.7.1	10.5.7.1	10.5.7.1	10.5.7.1
db.client.host:	10.5.7.2	10.5.7.1	10.5.7.2	10.5.7.2

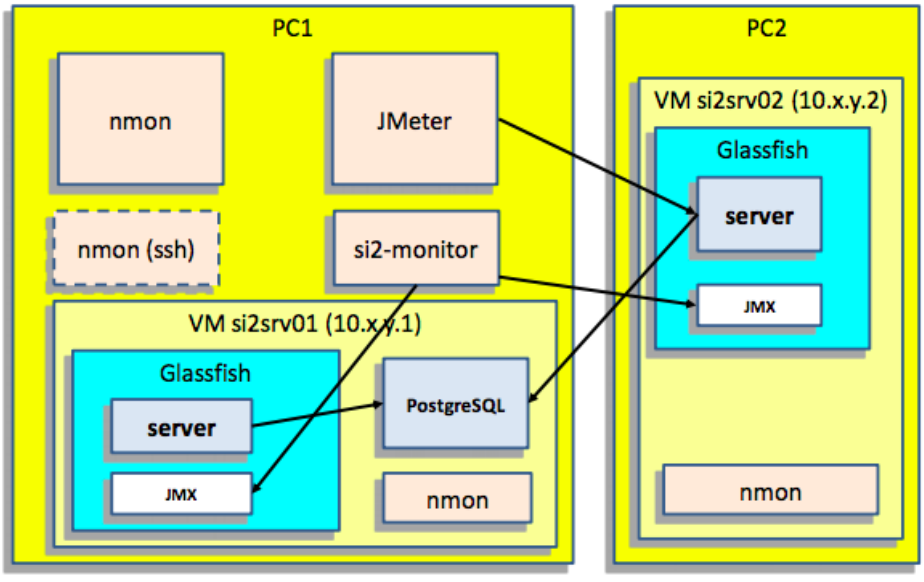


Figura 1: Diagrama de despliegue

Tras haber configurado e iniciado las máquinas virtuales, y haber desplegado los 3 servicios de pago, ejecutamos el comando *free* para indicar el estado de la memoria RAM en el momento previo a realizar las pruebas. Cabe destacar que, por error de lectura del enunciado, se le asignó a las maquinas virtuales 1GB de memoria RAM en vez de los 768MB pedidos (En las siguientes pruebas se subsanó este error). A continuación se muestran las salidas del comando *free* en los diferentes sistemas junto con la salida del comando *nmon -m*:

PC1

Físico

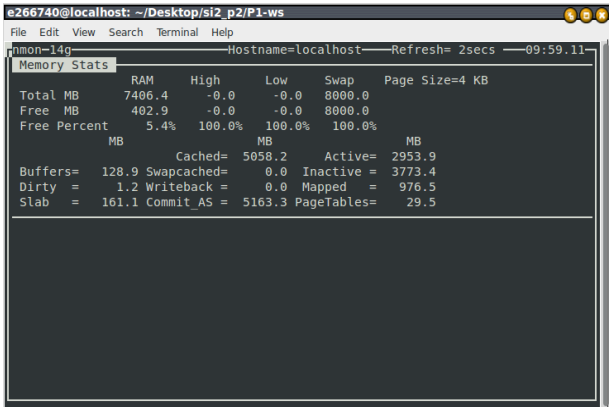


Figura 2: PC1 físico

	total	usado	libre	compartido	búffers	almacenado
Mem:	7584160	6858300	725860	1077552	130768	5117660
-/+ buffers/cache:	1609872	5974288	-	-	-	-
Intercambio:	8191996	0	8191996	-	-	-

Virtual

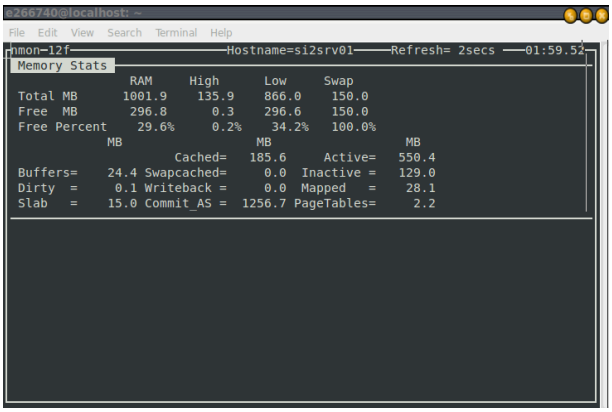


Figura 3: PC1 virtual

	total	usado	libre	compartido	buffers	almacenado
Mem:	1025976	720596	305380	0	24944	189904
-/+ buffers/cache:	505748	520228	-	-	-	-
Intercambio:	153592	0	153592	-	-	-

PC2

Físico

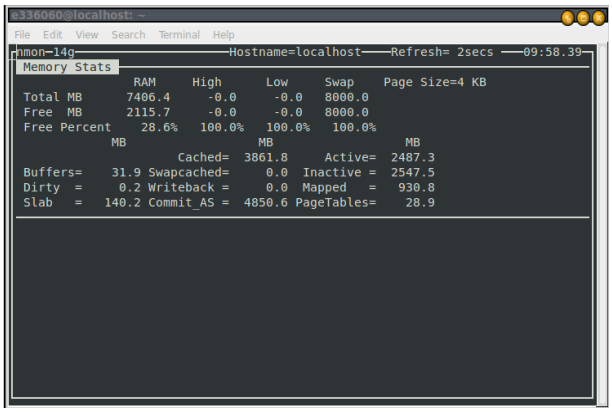


Figura 4: PC2 físico

	total	usado	libre	compartido	buffers	almacenado
Mem:	7584160	5390244	2193916	1089600	32440	3954704
-/+ buffers/cache:	1403100	6181060	-	-	-	-
Intercambio:	8191996	0	8191996	-	-	-

Virtual

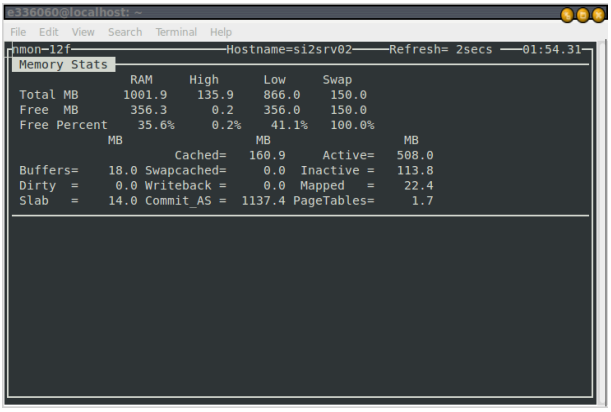


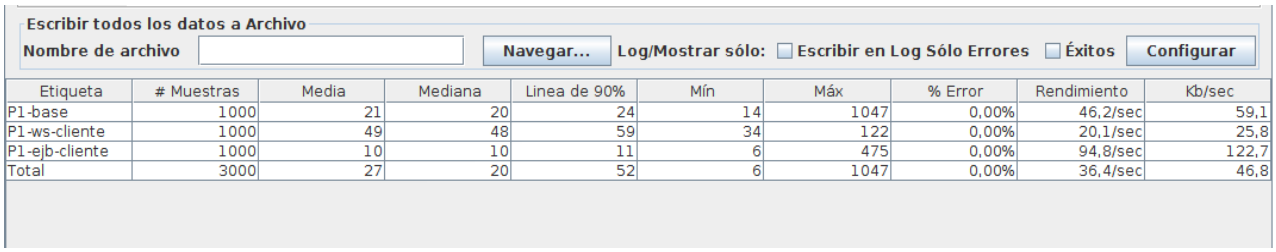
Figura 5: PC2 virtual

	total	usado	libre	compartido	buffers	almacenado
Mem:	1025976	659792	366184	0	18412	164644
-/+ buffers/cache:	476736	549240	-	-	-	-
Intercambio:	153592	0	153592	-	-	-

Observación: aunque en el enunciado se recomienda crear un Shell script que automatice el proceso de limpiar y desplegar los proyectos, al utilizarlo dejaba de funcionar la generación automática de stubs en el proyecto ws. Por tanto, todas las pruebas se han generado a mano con los comandos *ant* correspondientes.

Ejercicio 3

En este ejercicio se pide ejecutar el plan completo de pruebas sobre las 3 versiones de la práctica. Realizamos algunas pruebas para comprobar que todo funcione correctamente y aquí adjuntamos 2 de las mismas.

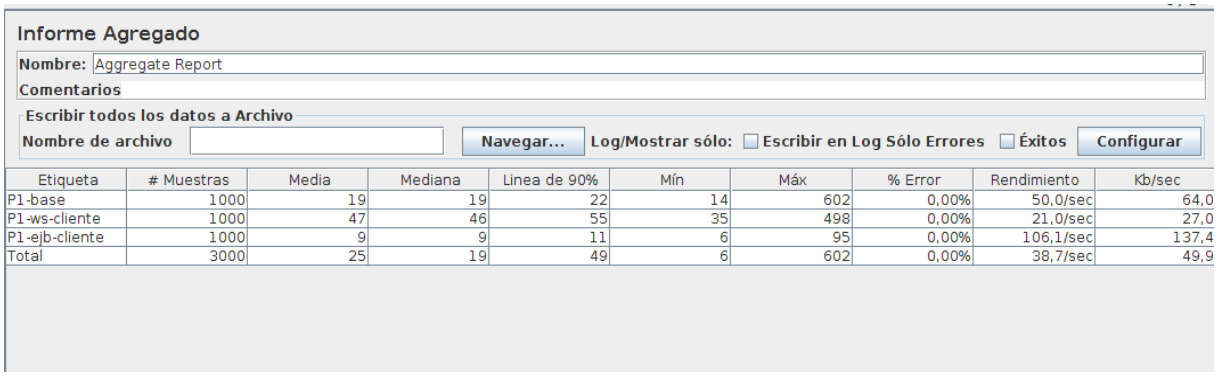


Escribir todos los datos a Archivo

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

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
P1-base	1000	21	20	24	14	1047	0,00%	46,2/sec	59,1
P1-ws-cliente	1000	49	48	59	34	122	0,00%	20,1/sec	25,8
P1-ejb-cliente	1000	10	10	11	6	475	0,00%	94,8/sec	122,7
Total	3000	27	20	52	6	1047	0,00%	36,4/sec	46,8

Figura 6: Plan de pruebas 1



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 | Configurar

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
P1-base	1000	19	19	22	14	602	0,00%	50,0/sec	64,0
P1-ws-cliente	1000	47	46	55	35	498	0,00%	21,0/sec	27,0
P1-ejb-cliente	1000	9	9	11	6	95	0,00%	106,1/sec	137,4
Total	3000	25	19	49	6	602	0,00%	38,7/sec	49,9

Figura 7: Plan de pruebas 2

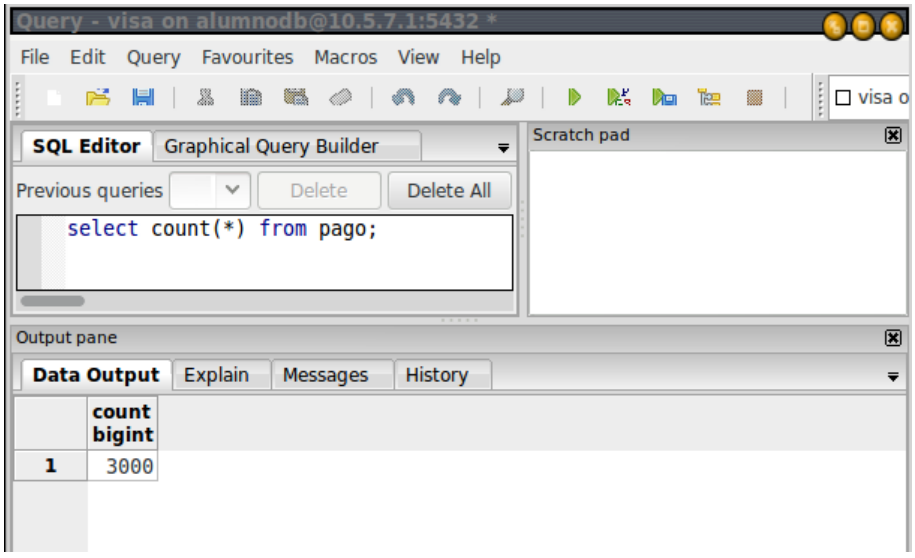


Figura 8: Base de Datos

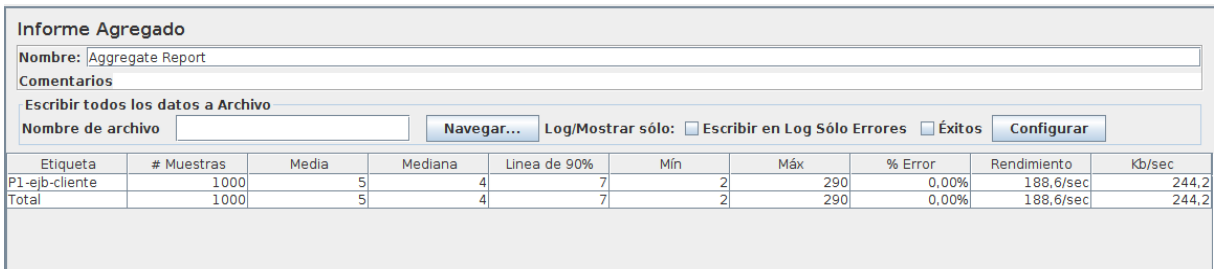
Tras realizar varias pruebas obteniendo datos muy similares y haber comprobado su correcta inserción en la base de datos podemos asegurar que el plan de pruebas está bien configurado.

Se ha adjuntado el fichero server.log solicitado dentro de la ruta '/Entregables/Ejercicio 3' localizada

en el directorio de la práctica.

Para decidir que método es mejor entre los proyectos base, ws o ejb, se ha comparado la columna del rendimiento. Esto se debe a que la columna de rendimiento mide toda la transacción en conjunto y es, al final, una de las cosas que más importan en un sistema informático. Si analizamos los resultados podemos ver claramente que EJB es muy superior con casi el doble de eficiencia que los otros 2 proyectos.

Luego, realizamos una batería de pagos para P1-ejb con los cambios propuestos en el ejercicio.



Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
P1-ejb-cliente	1000	5	4	7	2	290	0,00%	188,6/sec	244,2
Total	1000	5	4	7	2	290	0,00%	188,6/sec	244,2

Figura 9: Plan de pruebas EJB

Comparando estos resultados con los de el anterior plan de pruebas vemos una clara mejoría en rendimiento. Esto se debe principalmente a que ejecutamos en local por lo que no se ve afectado por las variaciones de la red.

Ejercicio 4

Tras la correcta configuración de Glassfish, hemos guardado la configuración en el fichero *domain.xml* solicitado dentro de la ruta '/Entregables/Ejercicio 4' localizada en el directorio de la práctica.

Para poder copiar el archivo desde la MV1 al PC1 hemos ejecutado el siguiente comando: *scp domain.xml e266740@10.10.66.17:home*, siendo 10.10.66.17 la dirección IP asignada al PC físico tras la ejecución del comando:

```
sudo /opt/si2/virtualip.sh eth0
```

Al analizar el script si2-monitor.sh podemos ver que los mandatos asadmin que debemos ejecutar en el Host PC1 para averiguar los valores solicitados son:

Max Queue Size del Servicio HTTP:

```
asadmin --type connectionqueue --user adminuser httpservice
```

Maximum Pool Size del Pool de conexiones a la Base de Datos:

```
asadmin --type threadpool --user adminuser server
```

Ejercicio 5

Los parámetros de configuración del servidor de cara al rendimiento son los mostrados en la siguiente Figura:

Parámetros de configuración			
Elemento	**Parámetro**		**Valor**
JVM Settings	Heap Máx. (MB)		512
JVM Settings	Heap Mín. (MB)		512
HTTP Service	Max.Thread Count		5
HTTP Service	Queue size		4096
Web Container	Max.Sessions		-1
Visa Pool	Max.Pool Size		32

Figura 10: Configuración de los parámetros

El documento de curva de productividad donde se encuentran estos datos está incluido en la raíz del directorio de la práctica.

Ejercicio 6

Tras la configuración del entorno de nuevo, ejecutamos el plan de pruebas y comprobamos con *nmon* algunos parámetros del ordenador (cpu, memoria, disco y red).

A la vista de los resultados, ¿qué elemento de proceso le parece más costosa?
La CPU experimenta una subida de 0 % a 50 % durante la ejecución. La memoria RAM se mantiene a un 80 % de uso (su elevado uso es debido a la máquina virtual). El acceso a disco es de apenas unos 100 KB/s y la carga en la red también es muy baja. Con estos datos podemos concluir que, de estos parámetros, lo que más consume es CPU.

¿Le parece una situación realista la simulada en este ejercicio?
La simulación es todo lo realista que se puede esperar en un entorno de estudio, pero los datos obtenidos pueden no ser los más objetivos o válidos. Esto es debido, para empezar, a la ejecución en segundo plano de una máquina virtual en el mismo PC (como hemos podido observar en el alto uso de RAM) y además, el entorno en el que se prueba es un Sistema Operativo de propósito generalista (Ubuntu), el cual no es el más adecuado para este tipo de problemas.

Teniendo en cuenta cuál ha sido el elemento más saturado, proponga otro esquema de despliegue que resuelva esa situación.
Una solución válida puede ser cambiar de entorno a uno que consuma menos, o a uno que esté especializado en este problema. Además, sería conveniente que, en lugar de simular el servidor desde ese ordenador, existiera uno de verdad con su Hardware correspondiente.

Ejercicio 7

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

Ejercicio 8

En este ejercicio obtenemos la curva de productividad siguiendo los pasos detallados en el enunciado. Utilizando los parámetros de configuración obtenidos en el ejercicio 5, medimos:

- Mediante el comando nmon, la media de utilización de la CPU a nivel usuario.
- Mediante el programa de monitorización si2-monitor.sh, el número de Pools utilizados, los hilos HTTP ocupados y las peticiones encoladas.
- Mediante jmeter, los valores Average, 90 % line y Throughput para las peticiones totales y las de procesaPago.

Para llevar a cabo la prueba empezamos con 1 cliente y fuimos incrementando de 250 en 250 hasta llegar a la saturación en 1500. A partir de ahí seguimos hasta 2000 peticiones, que es cuando llegó a la zona de muerte. Para cada valor realizamos 10 pruebas, y los resultados medios pueden observarse en las figuras 11 y 12.

Prueba de rendimiento

	Sistema	Monitores		
Usuarios	CPU, average (%)	Visa Pool used Conns	HTTP Current Threads Busy	Conn queued, instant
1,00	10,00	0,00	0,00	0,00
250,00	34,00	0,00	0,82	0,02
500,00	46,00	0,00	2,88	1,13
750,00	52,00	0,00	4,96	84,17
1000,00	55,00	0,00	4,95	394,01
1250,00	65,00	0,00	4,80	400,98
1500,00	76,00	0,00	5,00	497,55
1750,00	88,00	0,00	4,72	486,52
2000,00	92,00	0,00	4,09	365,26

Figura 11: Rendimiento de Sistema y Monitores

Prueba de rendimiento

	Total			ProcesaPago		
Usuarios	Average (ms)	90% line (ms)	Throughput (s^{-1})	Average (ms)	90%line (ms)	Throughput (s^{-1})
1,00	60,00	31,00	0,40	48,00	31,00	0,21
250,00	13,00	29,00	83,10	23,00	32,00	43,40
500,00	20,00	44,00	163,90	34,00	51,00	85,50
750,00	256,00	370,00	227,90	273,00	379,00	118,40
1000,00	1128,00	1384,00	241,40	1158,00	1399,00	124,40
1250,00	1127,00	1735,00	292,80	1169,00	1796,00	152,30
1500,00	1220,00	1906,00	344,70	1284,00	1978,00	180,10
1750,00	1695,00	2531,00	326,20	1669,00	2410,00	166,80
2000,00	2257,00	3602,00	305,20	2348,00	3957,00	155,50

Figura 12: Rendimiento de las peticiones Totales y de ProcesaPago

Una vez realizadas las iteraciones necesarias para alcanzar la saturación podemos representar la curva de Throughput versus usuarios:

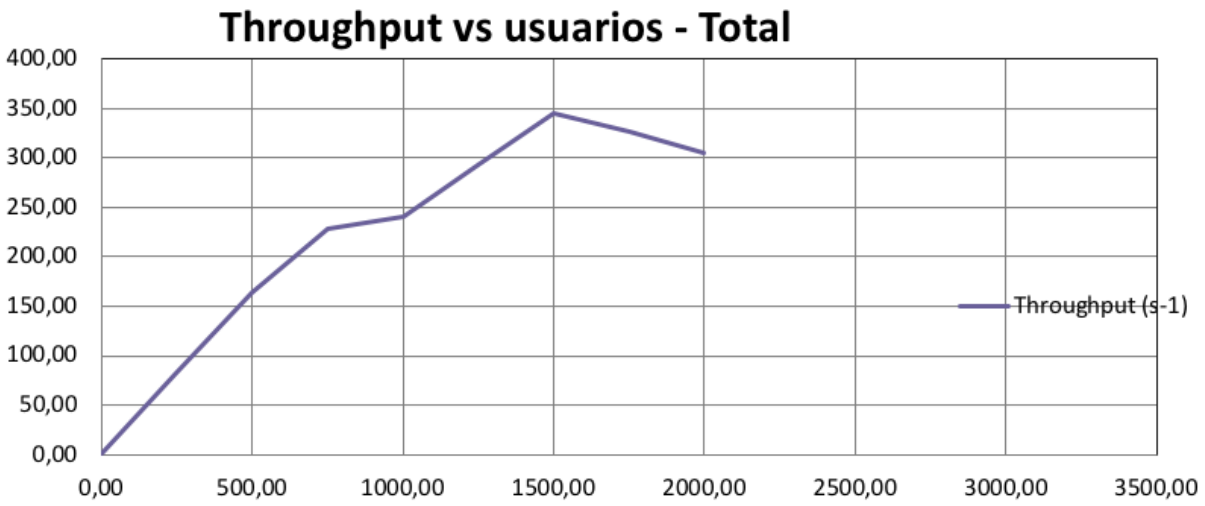


Figura 13: Throughput vs usuarios - Total

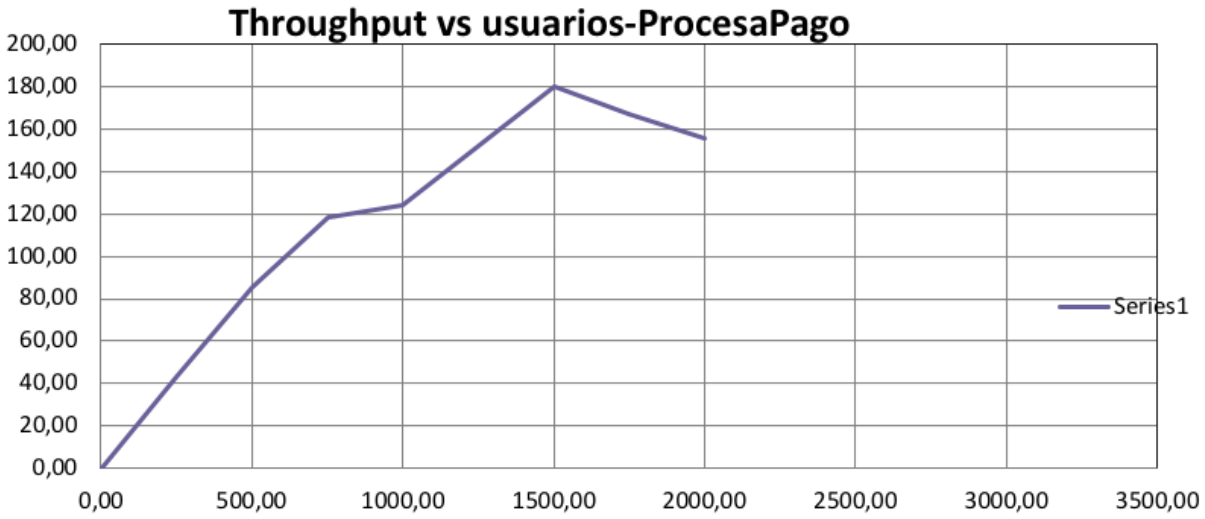


Figura 14: Throughput vs usuarios - ProcesaPago

También podemos observar que la latencia aumenta de manera exponencial a medida que el servidor llega a saturación:

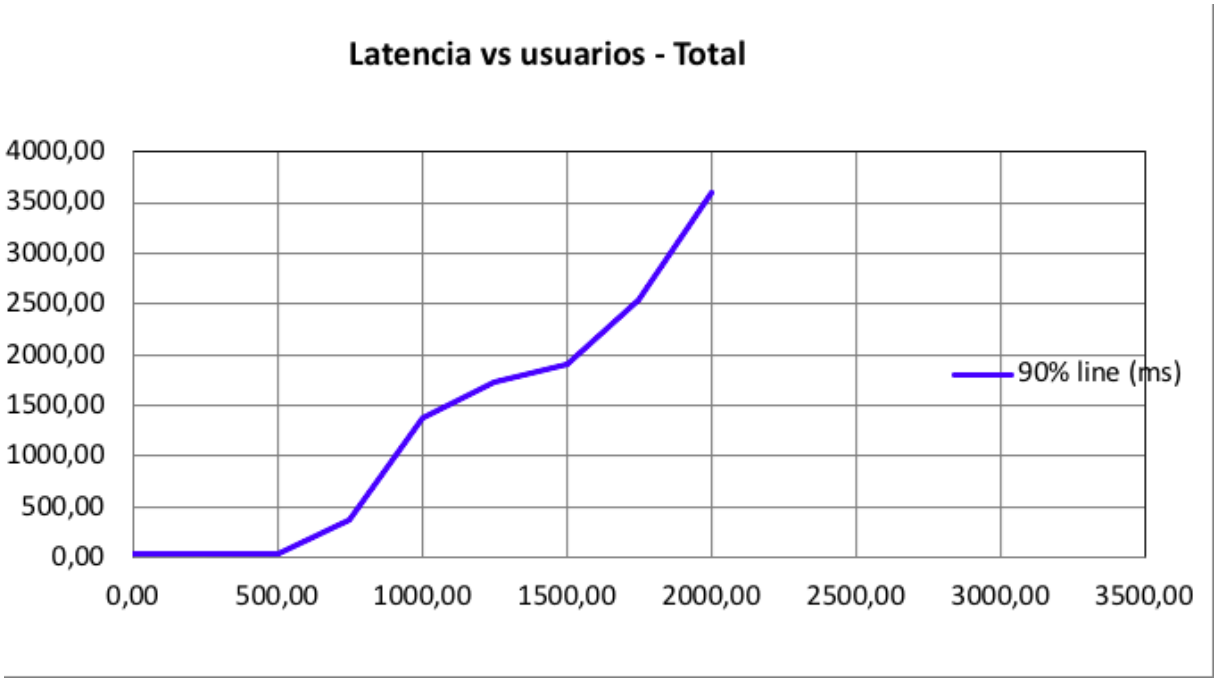


Figura 15: Latencia vs usuarios - Total

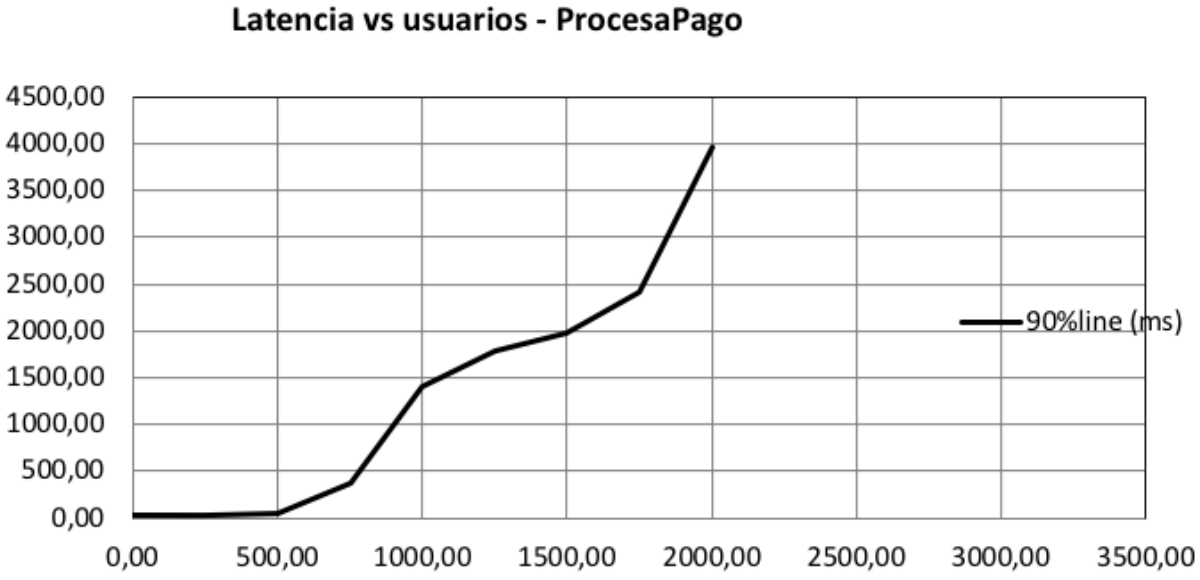


Figura 16: Latencia vs usuarios - ProcesaPago

El fichero P2-curvaProductividad.jmx utilizado para ejecutar los datos y el fichero SI2-P2-curvaProductividad.odt utilizado para la representación gráfica están incluidos en la raíz del directorio de la práctica.

¿Cómo debemos invocar a nmon para recolectar muestras cada 5 segundos durante 10 minutos que incluyan los 'top processes' en el fichero log-file.nmon?
El comando a utilizar es:

```
nmon -s5 -c120 -f
```

Donde el valor de 's' indica con qué frecuencia (en segundos) debe realizar capturas y 'c' indica el número de capturas que se van a realizar. Si queremos capturas cada 5 segundos en un intervalo de 10 minutos (es decir, 600 segundos), tendremos que llevar a cabo 600/5 = 120 capturas. Además, para guardar los resultados en un fichero es necesario utilizar 'f'.

Ejercicio 9

A partir de la curva obtenida, determinar para cuántos usuarios conectados se produce el punto de saturación, cuál es el máximo throughput que se alcanza en el mismo, y el throughput máximo que se obtiene en zona de saturación.
Si se observan las figuras 13 y 14, el punto de saturación se obtiene en 1500 peticiones de clientes, con un throughput de casi 350 s-1.

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.
En vista de los resultados obtenidos durante la elaboración de la curva, una posible solución para obtener el punto de saturación en un número mayor de usuarios sería aumentar el número de procesadores del servidor. Si en lugar de 1 procesador utilizase 2, la saturación se alcanzaría con un número de clientes más elevado. Otra alternativa sería incrementar el tamaño de la RAM.

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.
Siguiendo con las propuestas que indicamos en el anterior apartado, reconfiguramos el servidor asignándole 2 cores y 1 GB de RAM. Tomamos muestras para los valores de 1500, 1750 y 2000 peticiones (los .csv correspondientes a sus resultados están adjuntos en la ruta 'Entregables/Ejercicio 9' localizada en el directorio de la práctica). Para el caso de 1500 peticiones podemos ver:

	% CPU	Throughput total	Throughput procesaPago
1 core + 750 MB RAM:	76 %	344,70 s-1	180,10 s-1
2 cores + 1 GB RAM:	50 %	411,90 s-1	212,73 s-1

Estos resultados confirman nuestra hipótesis inicial de que incrementar el número de cores y la memoria RAM del sistema mejoraría el resultado final.