		Escuela Politécnica Superior Ingeniería Informática Prácticas de Sistemas Informáticos 2			
Grupo	2401	Práctica	2	Fecha	08/04/2018
Alumno		Alcover, Couso, Roberto			
Alumno		Aced, Fuentes, Emilio Samuel			

Práctica 2: Arquitectura JAVA EE (Segund Parte)

Ejercicio 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.

Ver en el archivo P1.jmx

Ejercicio 2:

Preparar los PCs con el esquema descrito en la Figura 21. Para ello:

Anote en la memoria de prácticas las direcciones IP asignadas a cada PC.

Detenga el servidor de GlassFish de los PCs físicos

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-clienteremoto, con el siguiente esquema:

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

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)

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)

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 los PCs físicos. Anote sus comentarios en la


```
nmon-12f-----[H for help]-----Hostname=si2srv01-----Refresh= 2secs -----08:42.34-----

#####
# # # # #
## # ## ## # # #
# # # # # # # # #
# # # # # # # # #
# ## # # # # #
# # # # # #####
#####

For help type H or ...
nmon -? - hint
nmon -h - full

To start the same way every time
set the NMON ksh variable

Use these keys to toggle statistics on/off:
c = CPU          l = CPU Long-term      - = Faster screen updates
m = Memory       j = Filesystems        + = Slower screen updates
d = Disks        n = Network            U = Virtual Memory
r = Resource     N = NFS                v = Verbose hints
k = kernel       t = Top-processes      . = only busy disks/procs
h = more options q = Quit
```

Free Pc2:
free

```
e321075@localhost:~/si2/Pla/P1-base$ free
              total        usado         libre       compart.      búffers      almac.
Mem:          7584160      7421700         162460       817088         55916       5657372
-/+ buffers/cache:    1708412      5875748
Intercambio:    8191996           0       8191996
```

Nmon Pc2:

```
nmon-14g-----Hostname=localhost-----Refresh= 2secs -----16:39.27-----

-----
#   #   #   #   #####   #   #
##  #  ##  ##  #   #  ##  #
#  #  #  ##  #   #   #  #  #
#  ##  #   #   #   #  #  #
#   ##  #   #   #   #  ##
#   #  #   #   #####  #   #
-----

For help type H or ...
nmon -? - hint
nmon -h - full

To start the same way every time
set the NMON ksh variable

Use these keys to toggle statistics on/off:
c = CPU          l = CPU Long-term    - = Faster screen updates
m = Memory       j = Filesystems       + = Slower screen updates
d = Disks        n = Network           V = Virtual Memory
r = Resource     N = NFS               v = Verbose hints
k = kernel       t = Top-processes     . = only busy disks/procs
h = more options q = Quit
```

Free MVPC2:

```
si2@si2srv02:~$ free
              total        used        free      shared    buffers     cached
Mem:          767168      596740      170428          0       17504     147588
-/+ buffers/cache:      431648      335520
Swap:        153592           0       153592
```

Nmon MVPC2:

```
nmon-12f——[H for help]——Hostname=si2srv02——Refresh= 2secs ——08:42.26

# # # # ##### # #
## # ## ## # # ## #
# # # # ## # # # # #
# # # # # # # # # #
# ## # # # # # # ##
# # # # # ##### # #

For help type H or ...
nmon -? - hint
nmon -h - full

To start the same way every time
set the NMON ksh variable

Use these keys to toggle statistics on/off:
c = CPU          l = CPU Long-term      - = Faster screen updates
m = Memory       j = Filesystems        + = Slower screen updates
d = Disks        n = Network            V = Virtual Memory
r = Resource     N = NFS                v = Verbose hints
k = kernel       t = Top-processes      . = only busy disks/procs
h = more options q = Quit
```

Pago de calentamiento con P1-ws:

 10.1.1.2:8080/P1-ws-cliente/procesapago

Pago con tarjeta

Pago realizado con éxito. A continuación se muestra el comprobante del mismo:

idTransaccion: 1
idComercio: 1
importe: 1.0
codRespuesta: 000
idAutorizacion: 1

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Pago de calentamiento con P1-ejb-cliente

Pago con tarjeta

Pago realizado con éxito. A continuación se muestra el comprobante del mismo:

idTransaccion: 2
idComercio: 1
importe: 1.0
codRespuesta: 000
idAutorizacion: 2

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Pago de calentamiento con P1-base:

Pago con tarjeta

Pago realizado con éxito. A continuación se muestra el comprobante del mismo:

idTransaccion: 3
idComercio: 1
importe: 1.0
codRespuesta: 000
idAutorizacion: 1

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Borremos los pagos:

Pago con tarjeta

Lista de pagos del comercio 1

idTransaccion	Importe	codRespuesta	idAutorizacion
3	1.0	000	1
1	1.0	000	2
2	1.0	000	3

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Pago con tarjeta

Se han borrado 3 pagos correctamente para el comercio 1

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Pago con tarjeta

Lista de pagos del comercio 1

idTransaccion	Importe	codRespuesta	idAutorizacion
---------------	---------	--------------	----------------

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Ejercicio 3:

Ejercicio 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.

Vemos que se han realizado los 3000 pagos

```
visa=# select count(*) from pago;
count
3000
(1 row)
```

Vemos el informe agregado

Informe Agregado										
Nombre:		Informe Agregado								
Comentarios										
Escribir todos los datos a Archivo										
Nombre de archivo			Navegar...		Log/Mostrar sólo: <input type="checkbox"/> Escribir en Log Sólo Errores <input type="checkbox"/> Éxit					
Etiqueta	# Muestr...	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimie...
P1-base	1000	10	10	12	13	14	6	497	0,00%	94,1/sec
P1-ws-cli...	1000	50	49	59	65	77	35	1214	0,00%	19,6/sec
P1-ejb-cli...	1000	23	22	29	32	37	14	603	0,00%	42,1/sec
Total	3000	28	22	52	56	69	6	1214	0,00%	35,2/sec

También con Tora:

select count(*) from pago;	
#	count
1	3000

Ejercicio 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.xml1` . 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 del laboratorio con `scp`.

Revisar el script `si2-monitor.sh` e indicar los mandatos `asadmin2` que debemos ejecutar en el Host PC1 para averiguar los valores siguientes, mencionados en el Apéndice 1, del servidor P1VM1:

1. Max Queue Size del Servicio HTTP

```
asadmin -H 10.1.1.1 --user admin -W /opt/SI2/passwordfile get "*" | grep -i http | grep -i max | grep -i queue
```

```
configs.config.default-config.thread-pools.thread-pool.http-thread-pool.max-queue-size=4096
```

```
--->configs.config.server-config.thread-pools.thread-pool.http-thread-pool.max-queue-size=4096
```

2. Maximum Pool Size del Pool de conexiones a nuestra DB

```
si2@si2srv02:/opt/glassfish4/glassfish/domains/domain1$ asadmin -H 10.1.1.1 --user admin -W /opt/SI2/passwordfile get "*" | grep -i pool | grep -i max | grep -i visa
```

```
resources.jdbc-connection-pool.VisaPool.max-connection-usage-count=0
```

```
--->resources.jdbc-connection-pool.VisaPool.max-pool-size=32
```

```
resources.jdbc-connection-pool.VisaPool.max-wait-time-in-millis=60000
```

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

```
--->get -m server.http-service.server.request.errorcount-count
```

Ejercicio 5:

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

Ver hoja Excell entregada con la práctica

Ejercicio 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.

P1-base

MV en reposo

```
e321075@localhost: ~/Desktop/apach... e321075@localhost: ~/Desktop/apach... e321075@localhost: ~/Desktop/apach... x
hmon-12f -----Hostname=s12srv02-----Refresh= 2secs -----08:04.09-----
CPU Utilisation
CPU  User%  Sys%  Wait%  Idle|0          |25          |50          |75          |100|
 1   0.0    0.5    0.0    99.5|>
 2   0.0    0.0    0.0   100.0|>
Avg  0.0    0.0    0.0   100.0|>

Memory Stats
RAM      High      Low      Swap
Total MB 749.2      0.0     749.2   150.0
Free MB   177.2      0.0     177.2   140.2
Free Percent 23.7%    0.0%    23.7%   93.5%
MB        MB        MB
Cached= 128.0      Active= 467.9
Buffers= 8.1 Swapcached= 2.1 Inactive = 83.0
Dirty = 0.0 Writeback = 0.0 Mapped = 23.3
Slab = 11.7 Commit_AS = 1314.9 PageTables= 1.7

Network I/O
I/F Name Recv=KB/s Trans=KB/s packin packout insize outsize Peak->Recv Trans
lo      0.0      0.0      0.0      0.0      0.0      0.0      0.4      0.4
eth0     0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
eth1     2.8      0.3     37.3      0.5     76.6    610.0     4.3      3.4

Top Processes Procs=93 mode=3 (1=Basic, 3=Perf 4=Size 5=I/O)
PID  %CPU  Size  Res  Res  Res  Res  Shared  Faults  Command
Used KB  Set  Text Data  Lib  KB  Min  Maj
 1   0.0   2788 1296 104  0   512  984  0  0 init
 2   0.0  0  0  0  0  0  0  0  0  0 kthreadd
 3   0.0  0  0  0  0  0  0  0  0  0 migration/0
 4   0.0  0  0  0  0  0  0  0  0  0 ksoftirqd/0
 5   0.0  0  0  0  0  0  0  0  0  0 watchdog/0
 6   0.0  0  0  0  0  0  0  0  0  0 migration/1
 7   0.0  0  0  0  0  0  0  0  0  0 ksoftirqd/1

Warning: Some Statistics may not shown
```

PC en reposo

```
e321075@localhost: ~/Desktop e321075@localhost: ~/Desktop/P2/P1-ejb-cliente-remoto x
hmon-14g -----[H for help]-----Hostname=localhost-----Refresh= 2secs -----16:05.14-----
CPU Utilisation
CPU  User%  Sys%  Wait%  Idle|0          |25          |50          |75          |100|
 1   8.5    3.5    0.0    87.9|UUUUU>
 2   7.5    3.0    0.0    89.4|UUUU>
 3   6.5    4.0    0.0    89.4|UUUU>
 4  12.1    4.5    0.0    83.4|UUUUUU>
Avg  8.8    3.9    0.0    87.4|UUUU>

Memory Stats
RAM      High      Low      Swap      Page Size=4 KB
Total MB 7406.4      -0.0     -0.0    8000.0
Free MB   531.0      -0.0     -0.0    8000.0
Free Percent 7.2%    100.0%    100.0%    100.0%
MB        MB        MB
Cached= 4363.4      Active= 2778.3
Buffers= 114.3 Swapcached= 0.0 Inactive = 3828.0
Dirty = 1.8 Writeback = 0.0 Mapped = 1127.6
Slab = 148.6 Commit_AS = 5973.6 PageTables= 34.5

Network I/O
I/F Name Recv=KB/s Trans=KB/s packin packout insize outsize Peak->Recv Trans
vmnet8  0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
eth0    11.4      0.0     81.9      0.5    142.3    66.0     20.2     5.3
lo       0.0      0.0      0.0      0.0      0.0      0.0      0.8      0.8
vmnet1  0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0

Top Processes Procs=226 mode=3 (1=Basic, 3=Perf 4=Size 5=I/O)
PID  %CPU  Size  Res  Res  Res  Res  Shared  Faults  Command
Used KB  Set  Text Data  Lib  KB  Min  Maj
2580 19.9 2095988 435028 12 1409552 0 41052 10 0 cinnamon
1487 11.4 799472 458800 2220 452920 0 39288 0 0 Xorg
1996 7.4 543308 9872 4 336068 0 5916 0 0 yate
3369 4.5 2631200 847204 13748 1620516 0 827828 10 0 vmware-vmx

Warning: Some Statistics may not shown
```

```

python3 -m psutil --help [H for help] --Hostname=si24srv02 --Refresh= 6secs --08.06.39
CPU Utilisation
+-----+
CPU  User%  Sys%  Wait%  Idle|0|25|50|75|100|
1  25.1  7.0  0.0  67.9|███████████|
2  9.3  0.0  0.0  90.7|███████████|
+-----+
Avg 16.4  2.8  0.0  80.8|███████████|
+-----+

Memory Stats
+-----+
Total MB  749.2  0.0  749.2  150.0
Free MB   172.5  0.0  172.5  140.2
Free Percent 23.0%  0.0%  23.0%  93.5%
+-----+
MB          MB          MB
Cached= 129.7  Active= 471.9
Buffers= 8.3  Swapcached= 2.1  Inactive = 84.2
Dirty = 0.4  Writeback = 0.0  Mapped = 23.3
Slab = 11.7  Commit_AS = 1317.7  PageTables= 1.7
+-----+

Network I/O
+-----+
I/F Name Recv=KB/s Trans=KB/s packin packetout insize outside Peak->Recv Trans
lo 0.0 0.0 0.0 0.0 0.0 0.0 0.4 0.4
eth0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
eth1 137.6 249.0 686.6 649.6 205.3 392.4 192.8 259.6
+-----+

Top Processes Procs=93 mode=3 (1=Basic, 3=Perf 4=Size 5=I/O)
+-----+
PID %CPU Size Res Res Res Res Shared Faults Command
Used KB Set Text Data Lib KB Min Maj
3372 44.9 978000 439800 4 0 954252 18572 0 0 java
1 0.0 2788 1296 104 0 512 984 0 0 init
2 0.0 0 0 0 0 0 0 0 0 kthreadd
3 0.0 0 0 0 0 0 0 0 0 migration/0
4 0.0 0 0 0 0 0 0 0 0 ksoftirqd/0
5 0.0 0 0 0 0 0 0 0 0 watchdog/0
6 0.0 0 0 0 0 0 0 0 0 migration/1
+-----+
Warning: Some Statistics may not show up

```

```
e321075@localhost: ~ /Desktop
* e321075@localhost: ~ /Desktop/P2/P1-ejb-cliente-remoto

jmon-14g Hostname=localhost Refresh= 2secs 16:07.32

CPU Utilisation
-----+-----+-----+-----+-----+
CPU  User%  Sys%  Wait%  Idle%  0 25 50 75 100
  1  13.1  10.6   0.0  76.3  UUUUUU 3536 >
  2  12.6  10.6   0.0  76.8  UUUUUU 3536 >
  3  12.2  10.2   0.0  77.6  UUUUUU 3536 >
  4  12.1  10.6   0.0  77.3  UUUUUU 3536 >
-----+-----+-----+-----+
Avg  12.5  10.6   0.0  76.9  UUUUUU 3536 >
-----+-----+-----+-----+

Memory Stats
-----+-----+-----+-----+
Total MB 7406.4 -0.0 -0.0 8000.0
Free MB 470.1 -0.0 -0.0 8000.0
Free Percent 6.3% 100.0% 100.0% 100.0%
-----+-----+-----+-----+
MB MB MB
Cached= 4376.9 Active= 2829.4
Buffers= 114.5 Swapcached= 0.0 Inactive= 3837.0
Dirty = 2.2 Writeback= 0.0 Mapped = 1130.6
Slab = 149.0 Commit_AS = 5986.9 PageTables= 34.5
-----+-----+-----+-----+

Network I/O
-----+-----+-----+-----+
I/F Name RecvKB/s TransKB/s packin packout insize outsize Peak->Recv Trans
vmnet8 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
eth0 247.3 130.0 692.2 626.4 365.8 212.6 323.0 218.1
lo 0.2 0.2 2.0 2.0 101.5 101.5 0.8 0.8
vmnet1 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
-----+-----+-----+-----+

Top Processes Procs=226 mode=3 (1=Basic, 3=Perf 4=Size 5=I/O)
-----+-----+-----+-----+
PID %CPU Size Res Res Res Res Res Res Res Res
Used KB Set Text Data Lib KB Min Maj
3369 35.4 2631200 847204 13748 1620516 0 827828 18 0 vmware-vmx
2580 25.4 2101844 435888 12 1415408 0 41052 6 0 cinnamon
1487 15.5 799472 461876 2220 452920 0 42364 0 0 Xorg
1996 8.0 543308 9872 4 336068 0 5916 0 0 yate
-----+-----+-----+-----+
Warning: Some statistics are not shown
```

Responda a las siguientes cuestiones: A la vista de los resultados, ¿qué elemento de proceso le parece más costosa? ¿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,...) ¿Le parece una situación realista la simulada en este ejercicio? ¿Por qué? Teniendo en cuenta cuál ha sido el elemento más saturado, proponga otro esquema de despliegue que resuelva esa situación.

Como se puede apreciar en las estadísticas de la mv2 tras la ejecución de las 3000 peticiones de pago la media de la ejecución ha llegado hasta el 55% que comparado con el 0% antes de empezar a enviar pagos representa un gran carga para la CPU también podemos apreciar en las estadísticas de red un pico en eth1 de 192 por tanto la red también es un elemento importante a la hora de ver el rendimiento de esta aplicación. La memoria en cambio no tiene grandes variaciones.

No. Para empezar todos los equipos están en la misma red cosa que en un sistema distribuido no tiene porqué ocurrir y además las condiciones de la red en los laboratorios están muy controladas.

Aunque la CPU de los ordenadores no esté totalmente saturada, como esta es una parte importante de la aplicación a mayor cantidad de CPU o procesadores más fluida irá la aplicación. También cabe mencionar que el aumento del ancho de banda de la red aumentará la cantidad de peticiones que se pueden realizar simultáneamente sin saturar la red.

Ejercicio 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 las prácticas 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.

No requería reflejarlo en la memoria

Ejercicio 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. 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.

Detenerlo cuando esté a punto de terminar la ejecución de la prueba. 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, mediante inspección del programa de monitorización nmon que se ejecuta en ambas máquinas.

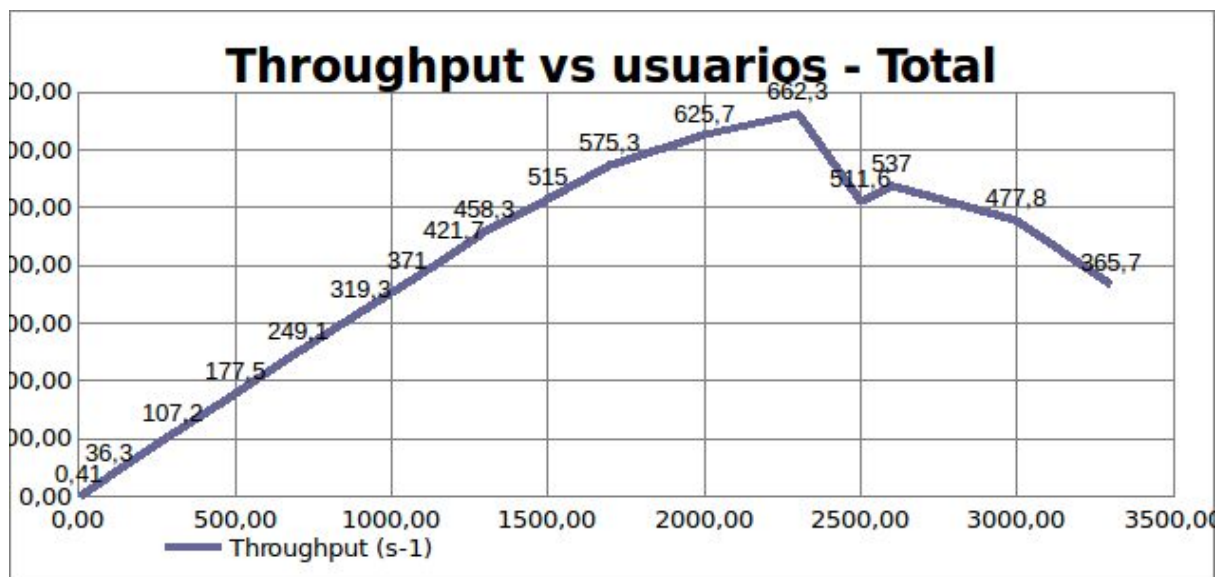
Finalizada la prueba, salvar el resultado de la ejecución del Aggregate Report en un archivo, y registrar en la hoja de cálculo de resultados los valores Average, 90% line y Throughput para las siguientes peticiones:

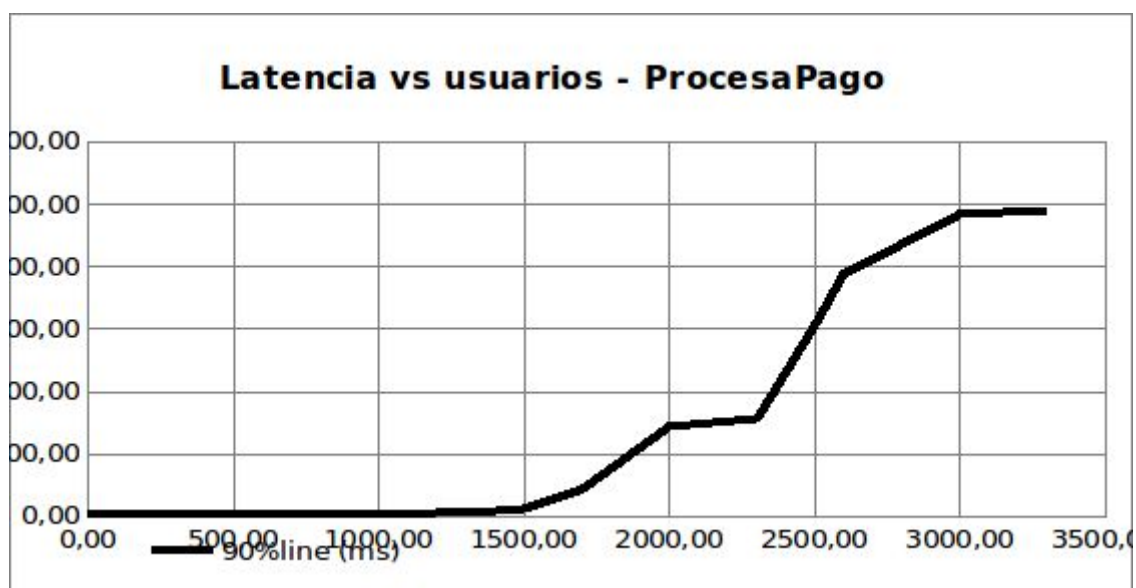
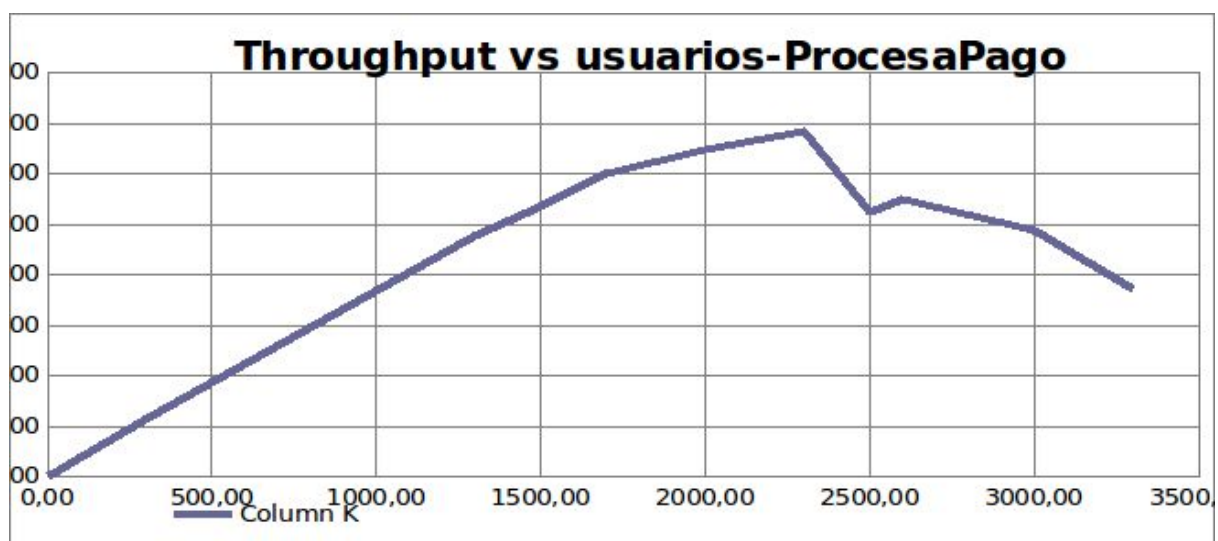
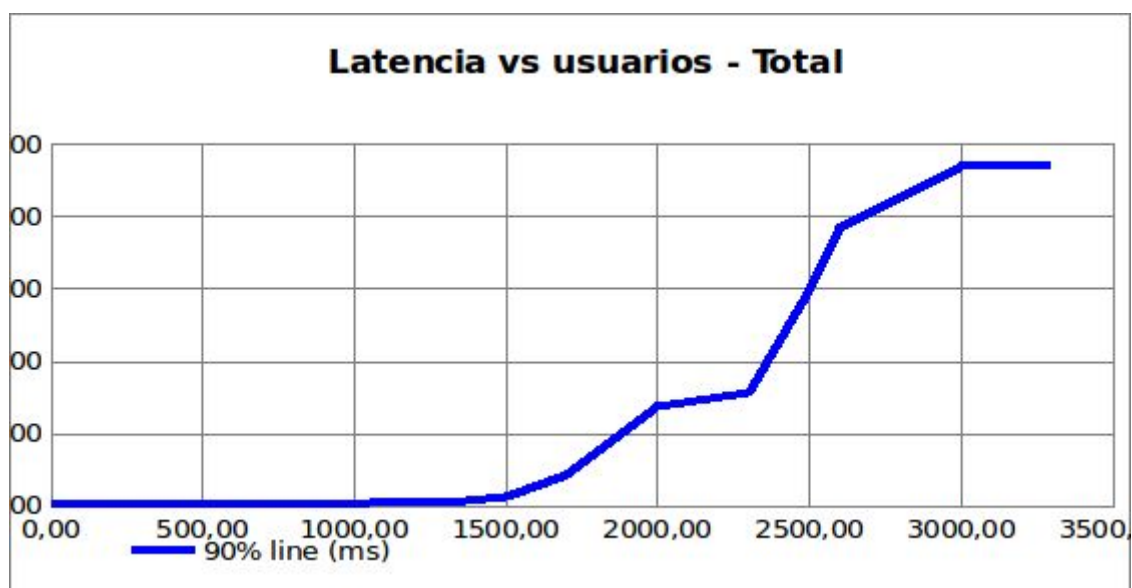
ProcesaPago.

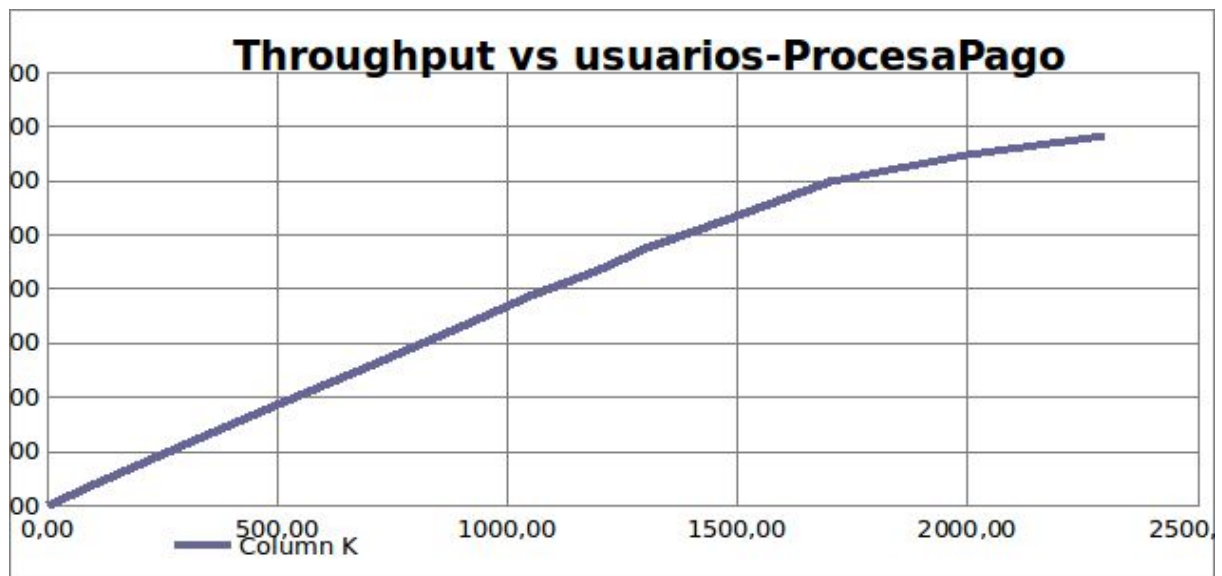
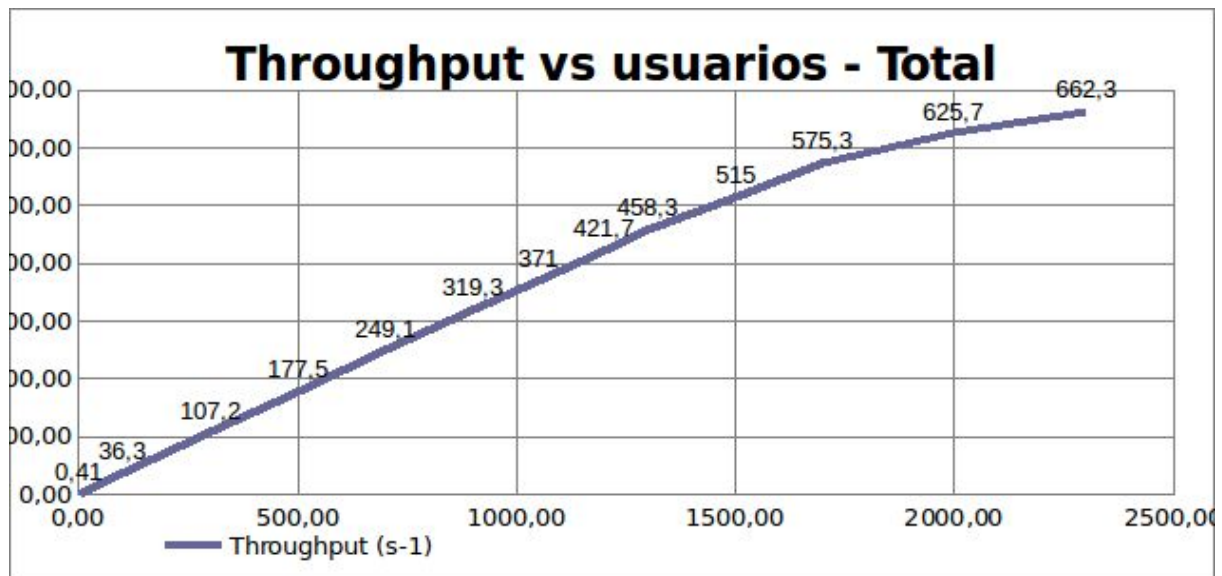
Total.

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.







¿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?

nmon -F log-file.nmon -t -s 5 -c 120

Explicamos cada parámetro a continuación:

- -F <filename> fichero en el que se guardan los datos
- -t Indica que se incluyan los "top processes"
- -s <seconds> intervalos entre cada muestra
- -c <times> número de muestras :

120 muestras * 5 segundos = 10 minutos

9)

Responda a las siguientes cuestiones:

- 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.

El punto de saturación se obtiene pasando los 2000 usuarios, en este punto se obtiene un throughput de unas 625 p/s.

El throughput máximo obtenido en la zona de saturación es de 662p/s.

- **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.**

Podemos apreciar que si aumentamos la capacidad del sistema los usuarios pasarían menos tiempo en la cola de espera. Esto es si aumentamos el número de peticiones que realiza el servidor la probabilidad de que un usuario espera disminuye, por lo tanto disminuye el tiempo medio. Otra opción es aumentar el pool connection.

- **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.**

Hemos decidido realizar ambos cambios obteniendo los siguientes resultados:

Etiqueta	# Muestr...	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimie...
/Pl-base...	20000	78	60	147	228	558	1	1209	0,00%	332,0/sec
/Pl-base...	20000	87	72	157	240	401	5	1235	0,00%	330,3/sec
Total	40000	82	66	153	232	475	1	1235	0,00%	633,3/sec

Pool Settings

Initial and Minimum Pool Size:	<input type="text" value="20"/>	Connections	Minimum and initial number of connections maintained in the pool
Maximum Pool Size:	<input type="text" value="100"/>	Connections	Maximum number of connections that can be created to satisfy client request
Pool Resize Quantity:	<input type="text" value="5"/>	Connections	Number of connections to be removed when pool idle timeout expires
Idle Timeout:	<input type="text" value="300"/>	Seconds	Maximum time that connection can remain idle in the pool
Max Wait Time:	<input type="text" value="60000"/>	Milliseconds	Amount of time caller waits before connection timeout is sent

