Universidad Autónoma

Prácticas de Sistemas Informáticos II

Práctica 2

Memoria

Autores:
Adrián Fernández
Santiago González-Carvajal

Pareja 1
Grupo 2401

8 de abril de 2019

Índice

1. Ejercicio 1

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.

1.1. Fichero generado P2.jmx

```
<?xml version="1.0" encoding="UTF-8"?>
<jmeterTestPlan version="1.2" properties="5.0" jmeter="5.1 r1853635">
  <hashTree>
    <TestPlan guiclass="TestPlanGui" testclass="TestPlan" testname="
        Test P2" enabled="true">
      <stringProp name="TestPlan.comments"></stringProp>
      <boolProp name="TestPlan.functional_mode">false</boolProp>
      <boolProp name="TestPlan.tearDown_on_shutdown">true</boolProp>
      <boolProp name="TestPlan.serialize_threadgroups">true</boolProp>
      <elementProp name="TestPlan.user_defined_variables" elementType="
          Arguments" guiclass="ArgumentsPanel" testclass="Arguments"
          testname="Variables definidas por el Usuario" enabled="true">
        <collectionProp name="Arguments.arguments"/>
      </elementProp>
      <stringProp name="TestPlan.user_define_classpath"></stringProp>
    </TestPlan>
.
.
.
```

Este fichero *jmx* describe el plan de pruebas de **Apache JMeter** sobre la prueba de rendimiento que vamos a realizar. El resto del fichero se entrega como adjunto en el zip de la práctica.

2. Ejercicio 2

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-ejb-cliente-remoto**, 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, a notar 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 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.

2.1. Evidencias

A continuación se muestran los resultados de la ejecución de comandos *free* y *nmon* en ambos PCs y ambas máquinas virtuales. Siendo PC1 el ordenador donde está desplegada la máquina virtual MV1 con IP 10.1.1.1 y PC2 el ordenador donde está desplegada la máquina virtual MV2 con IP 10.1.1.2.

Estado del PC1:

```
e341074@4-4-65-105:~/si2/p2/P1-ejb-cliente-remoto$ free
              total usado libre compartido búfer/caché disponible
Memoria:     7576260     1796160     1155784     1139704     4624316     4362104
Swap:        8191996           0     8191996
```

Figura 1: Resultado de free en el PC1.

```
nmon-16g                    Hostname=4-4-65-105   Refresh= 2secs   17:45.11
 Memory and Swap
 PageSize:4KB   RAM-Memory  Swap-Space       High-Memory     Low-Memory
 Total (MB)         7398.7      8000.0       - not in use   - not in use
 Free  (MB)         1005.1      8000.0
 Free Percent         13.6%      100.0%
 Linux Kernel Internal Memory (MB)
                       Cached=    4173.7     Active=    2903.0
 Buffers=     285.5 Swapcached=       0.0  Inactive =    3109.9
 Dirty  =       1.6 Writeback =       0.0  Mapped   =    1138.5
 Slab   =     219.2 Commit_AS =    8203.6 PageTables=      57.7
```

Figura 2: Resultado de nmon en el PC1.

Estado del PC2:

```
e340625@4-5-65-106:~/si2$ free
              total usado libre compartido búfer/caché disponible
Memoria:     7576260     2155992      218944     1139588     5201324     3980616
Swap:        8191996           0     8191996
```

Figura 3: Resultado de free en el PC2.

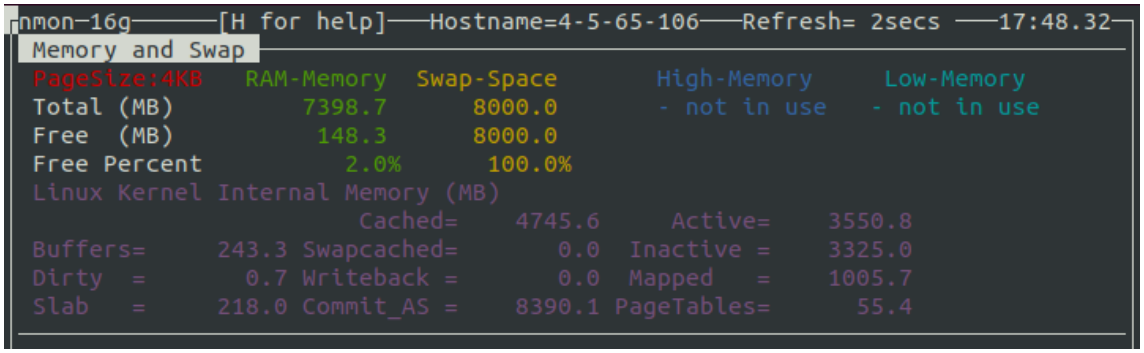

Figura 4: Resultado de nmon en el PC2.

Estado de la MV1:

```
si2@si2srv01:~$ free
             total       used       free     shared    buffers     cached
Mem:        767168     605256     161912          0      25120     286660
-/+ buffers/cache:     293476     473692
Swap:       153592          0     153592
```

Figura 5: Resultado de free en la MV1.

```
nmon-12f                    Hostname=si2srv01      Refresh= 2secs    08:46.46
 Memory Stats
                RAM      High       Low      Swap
 Total MB       749.2       0.0     749.2     150.0
 Free  MB       156.5       0.0     156.5     150.0
 Free Percent    20.9%      0.0%     20.9%    100.0%
              MB                       MB                      MB
                     Cached=    280.1      Active=    346.2
 Buffers=     24.6 Swapcached=      0.0  Inactive =    221.8
 Dirty  =      0.1 Writeback =      0.0  Mapped   =     25.8
 Slab   =     15.6 Commit_AS =    964.8 PageTables=      1.5
```

Figura 6: Resultado de nmon en la MV1.

Estado del MV2:

```
si2@si2srv02:~$ free
             total       used       free     shared    buffers     cached
Mem:        767168     448244     318924          0      18296     255920
-/+ buffers/cache:     174028     593140
Swap:       153592          0     153592
```

Figura 7: Resultado de free en la MV2.

```
nmon-12f                    Hostname=si2srv02      Refresh= 2secs    08:48.14
 Memory Stats
                RAM      High       Low      Swap
 Total MB       749.2       0.0     749.2     150.0
 Free  MB       309.9       0.0     309.9     150.0
 Free Percent    41.4%      0.0%     41.4%    100.0%
              MB                       MB                      MB
                     Cached=    250.0      Active=    203.4
 Buffers=     17.9 Swapcached=      0.0  Inactive =    212.6
 Dirty  =      0.0 Writeback =      0.0  Mapped   =     22.5
 Slab   =     14.6 Commit_AS =    804.6 PageTables=      1.3
```

Figura 8: Resultado de nmon en la MV2.

Como era de esperar, el comando nmon nos muestra que la zona de swap está libre en su totalidad (ya que las máquinas no están paginando). También, podemos

apreciar, que la máquina que más RAM está utilizando es la MV2 ya que es la que tiene desplegado el cliente de la aplicación para realizar los pagos (mientras que la ocupación de la MV1 se debe al despliegue de la base de datos).

3. 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.** anteriores

3.1. Evidencias

A continuación se muestran las capturas de pantalla con los resultados de las pruebas completas y la prueba sobre *P1-ejb* repetida.

Pruebas completas:

#	count
1	3000

Figura 9: Resultado de la query SELECT COUNT(*) FROM PAGO.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
P1-base	1000	14	14	16	17	18	7	541	0,000%	68,9465	88,45	0
P1-ws	1000	56	54	67	75	88	38	609	0,000%	17,62549	22,77	0
P1-ejb	1000	34	33	43	46	51	14	436	0,000%	29,07568	38,02	0
Total	3000	34	33	57	63	80	7	609	0,000%	28,39807	36,75	0

Figura 10: Resultados de las pruebas con 0 % de error.

Prueba sobre P1-ejb con el cliente en local:

# ▲	count
1	1000

Figura 11: Resultado de la query SELECT COUNT(*) FROM PAGO.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
P1-ejb	1000	4	3	3	4	5	2	1910	0,000%	205,93081	266,23	0
Total	1000	4	3	3	4	5	2	1910	0,000%	205,93081	266,23	0

Figura 12: Resultados de la prueba.

3.2. Comparación de resultados

Al repetir la prueba en **P1-ejb** en local inhabilitando los "Thread Groups" de *P1-base* y *P1-ws* se ha obtenido una mejora del rendimiento (que es la columna que nos permite comparar de una manera más clara los resultados). Esto es lo esperado, ya que P1-ejb actúa con una interfaz local (luego en localidad es como P1-base), y además utiliza EJBs (lo que permite mejorar el rendimiento de P1-base).

3.3. Cuestiones

¿Cuál de los resultados le parece el mejor?

El mejor resultado de la primera parte es el de *P1-base*. Esto es lo esperado ya que P1-base es el que tiene un funcionamiento más local (no es un servicio web como P1-ws; tampoco tiene el cliente remoto como P1-ejb).

¿Por qué?

Porque, a parte de tener la menor media de latencia, tiene el percentil de 99 % notablemente más bajo que el resto. Esto significa que las peticiones que tardan mucho en ser respondidas son muy inusuales y que el tiempo de respuesta habitual es bajo. Además, es la que tiene el rendimiento más elevado.

¿Qué columna o columnas elegiría para decidir este resultado?

La media, el percentil de 99 % y sobre todo el rendimiento son las más significativas.

4. Ejercicio 4

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.xml` . 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 asadmin que debemos ejecutar en el Host PC1 para averiguar los valores siguientes, mencionados en el Apéndice 1, del servidor PC1VM1:

- **Max Queue Size del Servicio HTTP**
- **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.

4.1. Comandos asadmin requeridos

Todos los comandos siguientes suponen que el directorio en el que se ejecuta el comando contiene el archivo passwordfile.

Max Queue Size del Servicio HTTP

```
$asadmin --host 10.1.1.2 --user admin -W passwordfile get
server.thread-pools.thread-pool.http-thread-pool.max-queue-size
```

Maximum Pool Size del Pool de conexiones a nuestra DB

```
$asadmin --host 10.1.1.2 --user admin -W passwordfile get
resources.jdbc-connection-pool.VisaPool.max-pool-size
```

Número de errores en las peticiones al servidor web

```
$asadmin --host 10.1.1.2 --user admin -W passwordfile get
--monitor server.web.request.errorcount-count
```

5. Ejercicio 5

Ejercicio 5: Registrar en la hoja de cálculo de resultados los valores de configuración que tienen estos parámetros.

5.1. Evidencias

En el fichero **SI2-P2-curvaProductividad.ods**:

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 13: Tabla de valores de configuración.

6. Ejercicio 6

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. Responda a las siguientes cuestiones:

- 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 ...)
- ¿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.

6.1. Evidencias

A continuación se muestran las gráficas obtenidas de la salida de *nmon* en la máquina virtual del PC2:

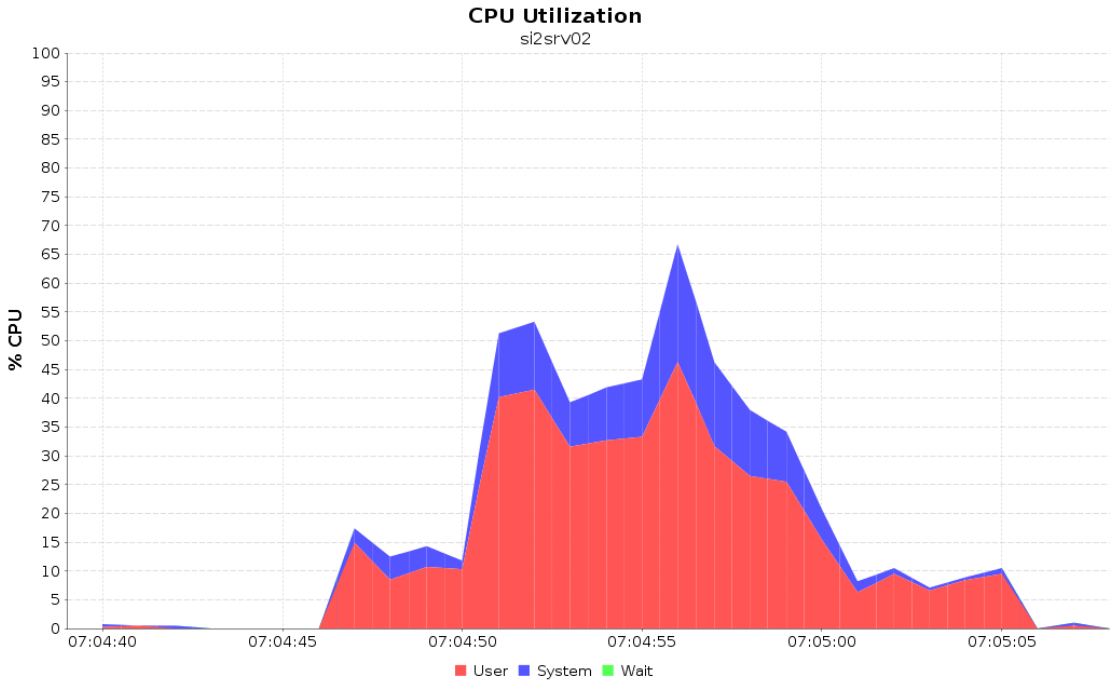

Figura 14: Gráfica de uso de CPU.

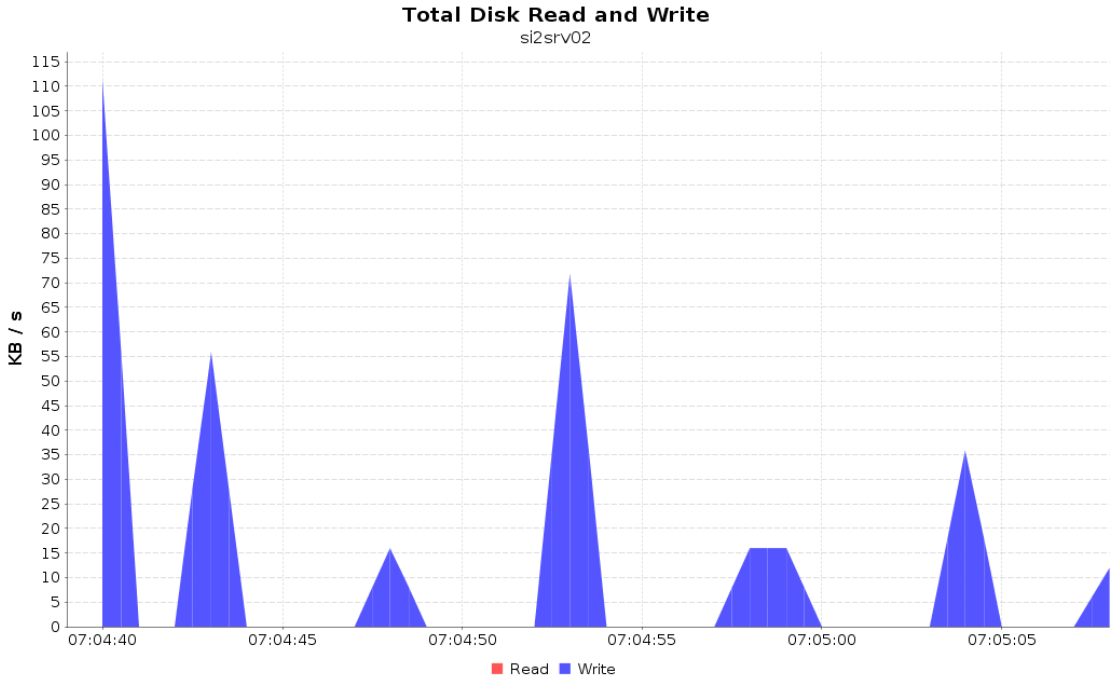

Figura 15: Gráfica de uso de disco.

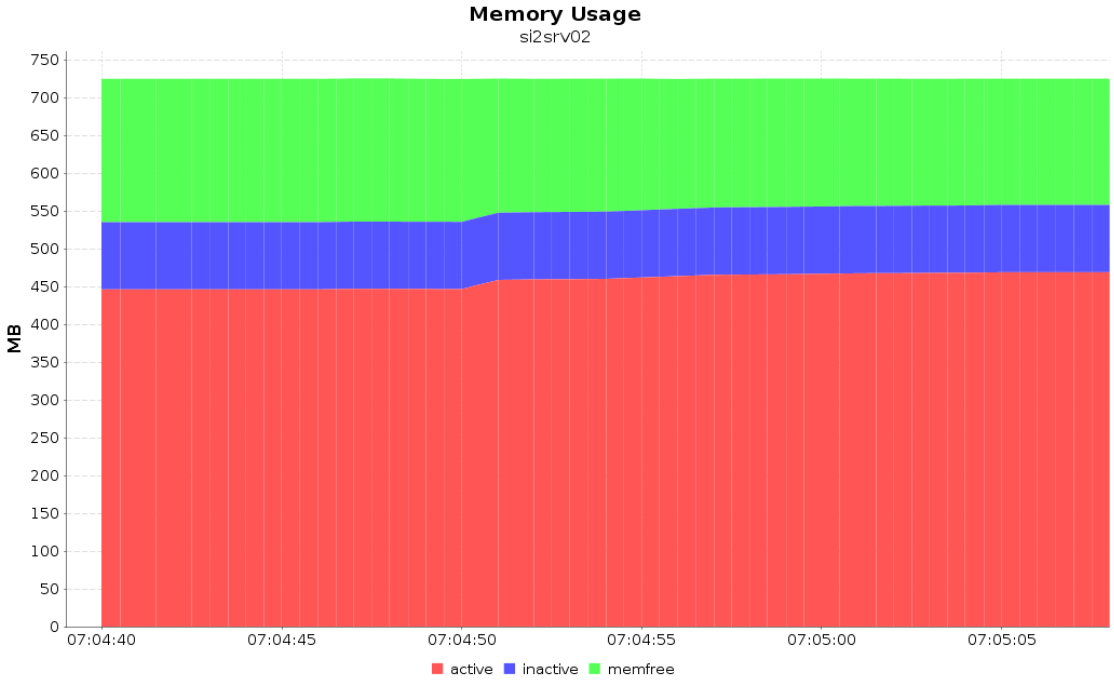

Figura 16: Gráfica de uso de memoria.

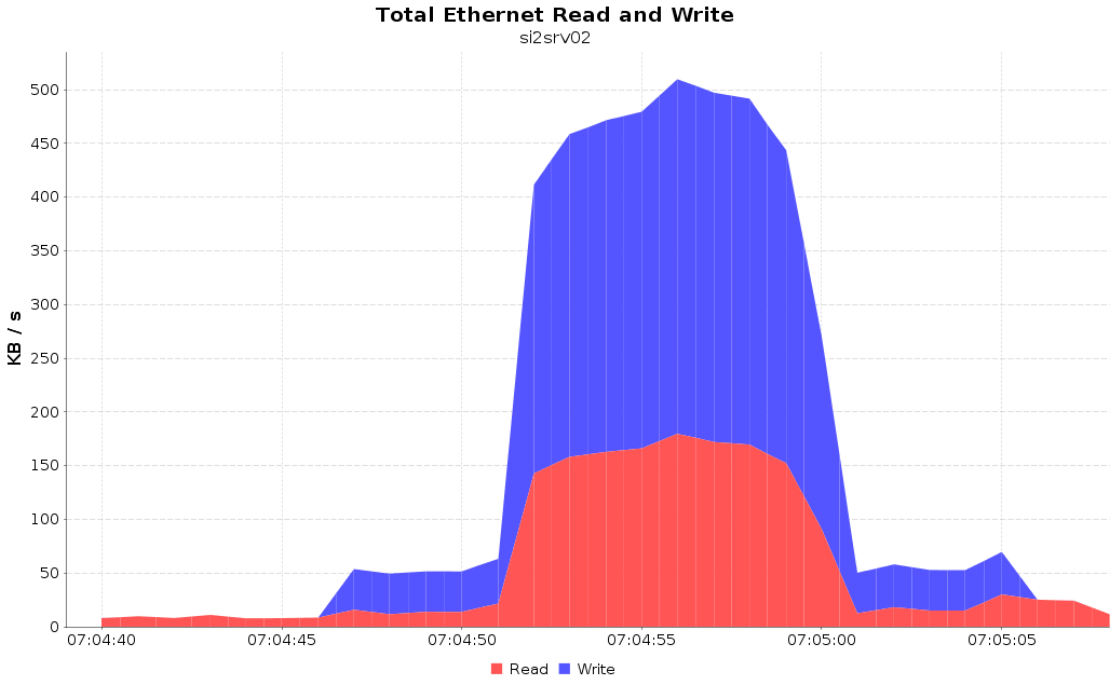

Figura 17: Gráfica de uso de red.

```
+e341074@4-14-65-115:~/si2/p2$ ./si2-monitor.sh 10.1.1.2
    #Muestra numJDBCCount numHTTPCount     numHTTPQ
           0           -1            0            0
           1           -1            0            0
           2           -1            0            0
           3           -1            0            0
           4           -1            0            0
           5            0            0            0
           6            0            1            0
           7            0            0            0
           8            0            1            0
           9            0            1            0
          10            0            1            0
          11            0            1            0
          12            0            1            0
          13            0            1            0
          14            0            0            0
          15            0            0            0
          16            0            0            0
          17            0            0            0
          18            0            0            0
^C
TOT.MUESTRAS        MEDIA:
          19    -0.263158     0.368421            0
```

Figura 18: Salida de si2-monitor-sh.

6.2. Cuestiones

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 ...)

Podemos observar que el recurso más utilizado es el envío de mensajes por red.

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

No, porque hemos configurado las pruebas de **JMeter** para que envíe las peticiones de una en una. Esto no se ajusta a la realidad, donde un servidor tiene que soportar la entrada de muchas peticiones de forma simultánea sin saturarse (comportamiento que estudiaremos en ejercicios posteriores).

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

La saturación de red se solucionaría desplegando tanto la base de datos como el servidor en una sola máquina, lo cual evitaría el uso de la red en la comunicación con la base de datos.

7. Ejercicio 8

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.

- 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:

  - 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).
  - 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).
  - 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.

- 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.

7.1. Evidencias

Los resultados obtenidos han sido estudiados utilizando el comando vmstat en la MV2, y el si2-monitor-sh en el PC1 con la dirección IP de la MV2 también. Esto nos ha permitido obtener la evidencia correspondiente que adjuntamos en la entrega (carpeta evidence), donde los ficheros siguen la nomenclatura monSH_nusers.txt (correspondientes a la salida de si2-monitor-sh) y vs_nusers.txt (correspondientes a la salida de vmstat).

Resultados obtenidos con las pruebas **JMeter**:

	Sistema	Monitores			Total			ProcesaPago		
Usuarios	CPU, average (%)	Visa Pool used Conns	HTTP Current Threads Busy	Conn queued, instant	Average (ms)	90% line (ms)	Throughput (s^{-1})	Average (ms)	90%line (ms)	Throughput (s^{-1})
1,00	6,63	0,00	0,00	0,00	9,00	15,00	0,42	14,00	16,00	0,22
251,00	25,23	0,42	0,58	0,00	9,00	16,00	82,70	14,00	17,00	43,00
501,00	31,28	0,40	0,66	0,03	6,00	13,00	166,40	10,00	14,00	86,60
751,00	30,80	1,42	1,32	0,05	6,00	11,00	246,50	9,00	13,00	128,50
1001,00	29,10	1,97	2,47	0,21	6,00	13,00	326,90	10,00	15,00	169,90
1251,00	29,26	2,95	3,41	0,97	8,00	16,00	409,70	13,00	19,00	212,20
1501,00	30,24	3,32	4,38	4,65	11,00	21,00	489,10	16,00	25,00	253,70
1751,00	36,80	3,97	4,74	13,05	17,00	38,00	570,70	22,00	42,00	295,80
2001,00	37,53	4,26	4,92	57,13	49,00	108,00	640,40	54,00	112,00	330,90
2251,00	37,79	4,18	5,00	231,15	202,00	344,00	691,80	208,00	351,00	357,90
2501,00	73,98	4,69	5,00	555,56	604,00	954,00	607,20	617,00	985,00	311,50
2751,00	79,34	4,49	5,00	560,74	847,00	1489,00	439,70	866,00	1486,00	223,50
3001,00	77,10	4,67	5,00	559,50	1089,00	1872,00	363,00	1096,00	1910,00	183,30
3501,00	78,98	4,57	5,00	550,57	1418,00	2037,00	429,30	1413,00	2045,00	217,60

Figura 19: Salida de las puebas con JMeter.

A modo de evidencia de que estos datos han sido obtenidos mediante las pruebas pertinentes, se han incluido en la entrega (carpeta evidence) los ficheros *csv* resultantes de las distintas pruebas (la nomenclatura seguida es tabla_nunsers.csv).

Se puede observar que el número de hilos alcanza su valor máximo (5) cerca de los 2.000 usuarios, lo cual coincide con la caída en rendimiento en las posteriores gráficas de *thoughput*.

Gráficas de ProcesaPago:

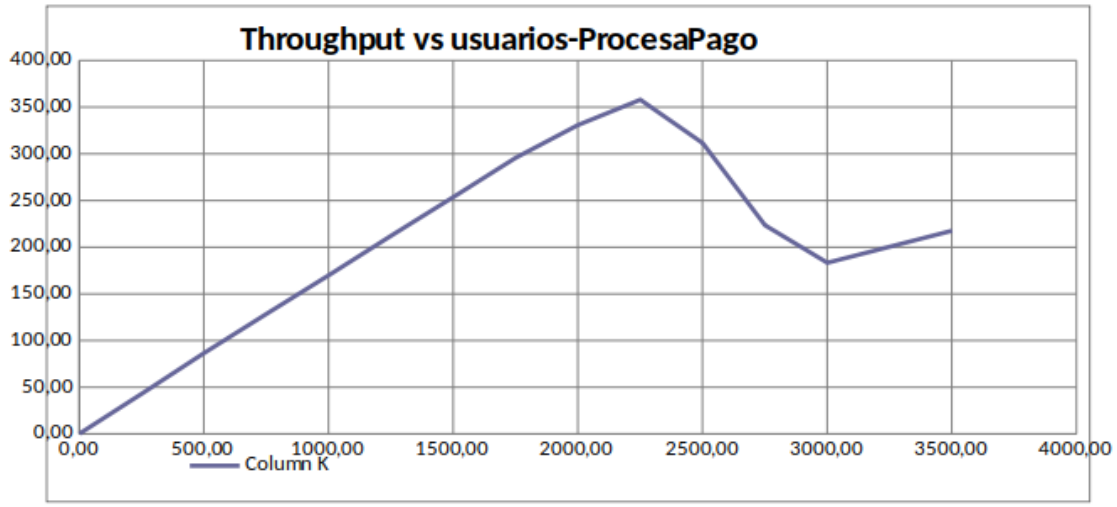

Figura 20: Throughput de ProcesaPago.

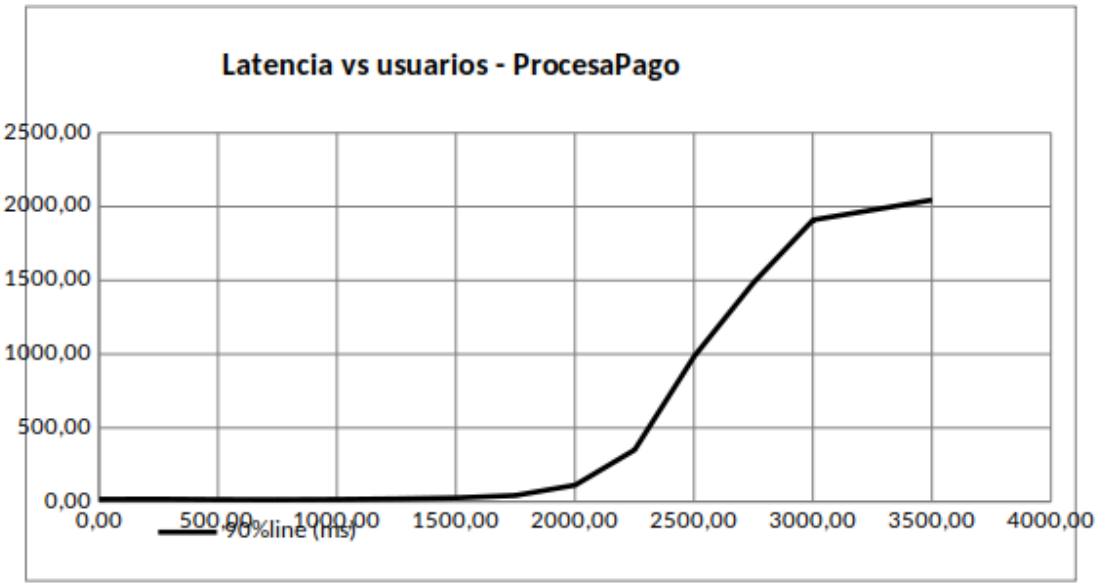

Figura 21: Latencia de ProcesaPago.

Gráficas del total:

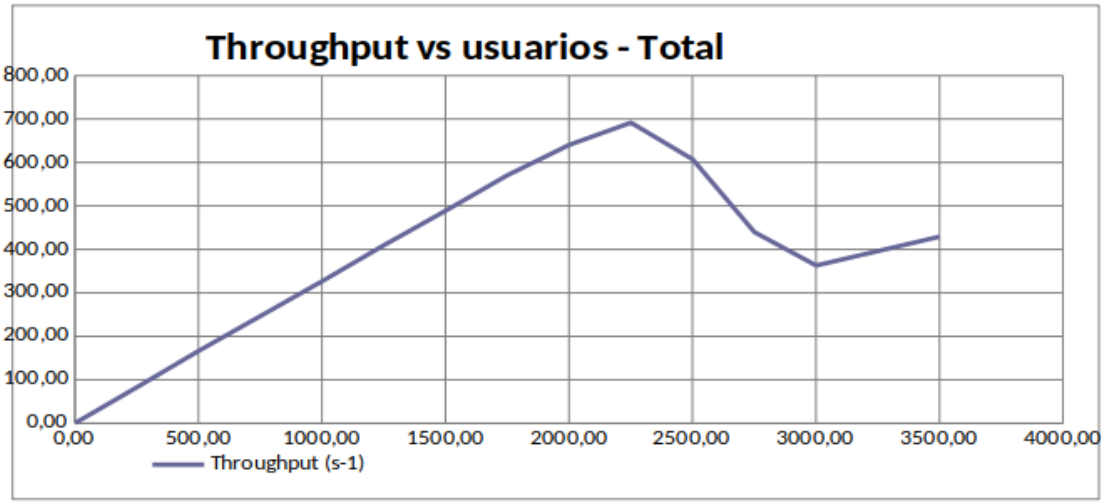

Figura 22: Throughput total.

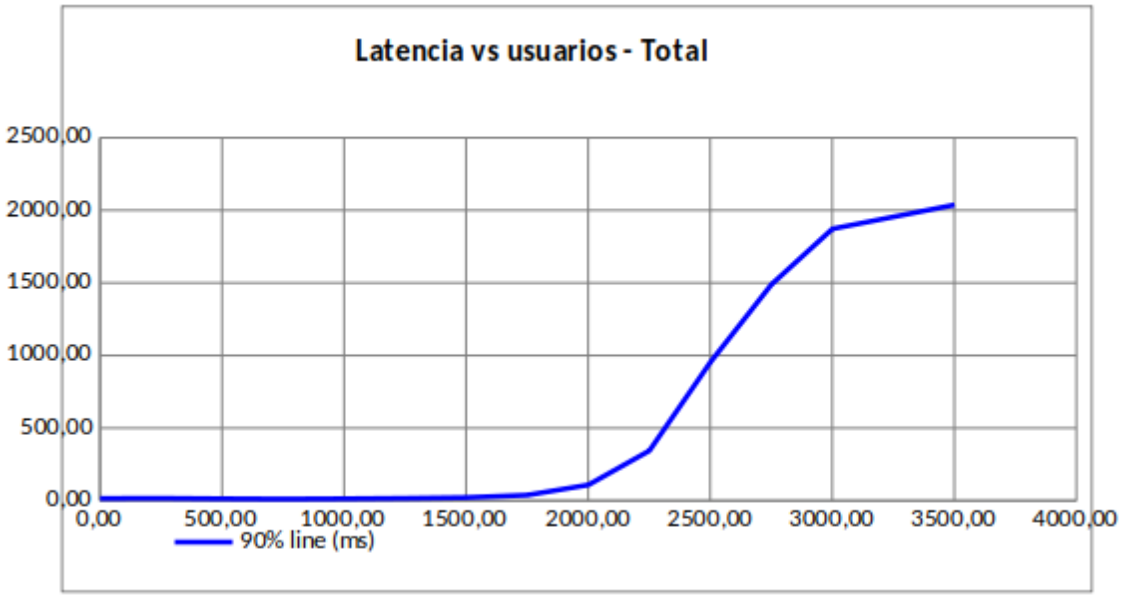

Figura 23: Latencia total.

8. Ejercicio 9

Ejercicio 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 *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.

8.1. Cuestiones

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

- El punto de saturación se alcanza entre los 2000 y 2251 usuarios.
- El thoughput del punto de saturación es de 600.
- El thoughput máximo es de 700.

Lo que pasa es que en el punto previo al punto de saturación, el servidor alcanza su máximo rendimiento (cuando el número de hilos es 4.92 ¡5 que es el máximo). Y cuando alcanza el 5, se satura y el rendimiento baja debido a ello.

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.

Aumentar el número máximo de hilos a 10 aumentaría la capacidad del servidor, ya que, como se puede observar en la figura 18, alcanza su valor máximo (5) entre 2000 y 2251 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.

Evidencias obtenidas mediante **JMeter**:

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
/P1-base/comienzapago	27510	827	609	1494	2446	4892	1	64552	0,000%	224,49080	436,18	0,00
/P1-base/procesapago	27510	866	624	1486	2439	5137	5	64927	0,000%	223,48411	274,25	0,00
Total	55020	847	617	1489	2444	5011	1	64927	0,000%	439,65352	696,88	0,00

Figura 24: Antes de la mejora.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
/P1-base/comienzapago	27510	28	5	108	144	174	1	221	0,000%	466,33442	906,07	0,00
/P1-base/procesapago	27510	41	18	124	158	189	6	240	0,000%	462,38403	567,41	0,00
Total	55020	35	15	116	151	183	1	240	0,000%	892,38505	1414,48	0,00

Figura 25: Después de la mejora.

Monitorización mostrada por la aplicación **si2-monitor**:

```
TOT.MUESTRAS          MEDIA:
          39     7.25641                9        57.0769
```

Monitorización mostrada por la herramienta **nmon**:

```
MEDIA
NR:   42 CPU:   72.175
```

Como podemos apreciar en los resultados, hemos repetido la prueba para 2751 usuarios (punto en el que el sistema estaba saturado cuando el máximo de hilos era 5), y los resultados que hemos obtenido han sido muy satisfactorios: el rendimiento que hemos obtenido es alrededor del doble que el que habíamos obtenido cuando el sistema estaba saturado.