UAM UNIVERSIDAD AUTONOMA DE MADRID		**Escuela Politécnica Superior Ingeniería Informática Prácticas de Sistemas Informáticos 2**			
Grupo	**2401**	**Práctica**	2	**Fecha**	05/04/2018
Alumno/a		Marcos Manchón, Pablo			
Alumno/a		Nevado Catalán, David			

Ejercicio 1

Seguimos los pasos especificados para generar el archivo **P2.jmx** que servirá para ejecutar las pruebas de rendimiento utilizando jmeter.

Configuramos **http request defaults** para que la dirección por defecto sea **10.1.9.2** y el puerto **80**, además desactivamos el modo debug. Nos definimos la variable **samples** con valor 1000. Creamos los threads de hilos correspondientes a P1-base, P1-ws y P1-ejb-remoto así como los contadores y variables aleatorias. Tras ello configuramos los parámetros de las peticiones http. El fichero **P2.jmx** se adjunta en la práctica, el cual hemos utilizado en los siguientes ejercicios para las pruebas de rendimiento.

Ejercicio 2

Desplegamos la máquinas según el esquema del enunciado (Figura 21), para ello asignamos la ip **10.1.9.1** a la máquina virtual 1 donde se ejecutará el servidor, la base de datos y el JMeter y la ip **10.1.9.2** a la máquina virtual 2 donde se desplegará el servidor de aplicación.

Tras iniciar los servidores en las máquinas virtuales y modificar las ips de los archivos build.properties y postgres.properties creamos el script **desplegar.bash** para desplegar y replegar todas las aplicaciones automáticamente. El script **desplegar.bash** se encuentra adjunto en la entrega.

A continuación procedemos a ver el estado en reposo de los dos ordenadores y sus respectivas máquinas virtuales con los comando free y nmon.

- **PC1**

```
             total       usado       libre     compart.     búffers      almac.
Mem:       7584160     7181480      402680     2140952       38484    5640828
-/+ buffers/cache:     1502168     6081992
Intercambio:     8191996        5400     8186596
```

```
nmon-14g ──[H for help]──Hostname=localhost──Refresh= 2secs ──16:58.35
 Memory Stats
              RAM     High      Low     Swap    Page Size=4 KB
 Total MB   7406.4     -0.0     -0.0   8000.0
 Free  MB    325.9     -0.0     -0.0   7994.7
 Free Percent  4.4%   100.0%   100.0%   99.9%
            MB                 MB                 MB
                 Cached=  5513.4     Active=  3044.1
 Buffers=   37.7 Swapcached=    0.4  Inactive =  3760.8
 Dirty  =    1.0 Writeback =    0.0  Mapped   =   732.8
 Slab   =  161.3 Commit_AS =  5881.2 PageTables=    28.5
```

- **PC2**

```
             total       usado       libre     compart.     búffers      almac.
Mem:       7584160     7377788      206372     2160148       17104    5105780
-/+ buffers/cache:     2254904     5329256
Intercambio:     8191996         472     8191524
```

```
nmon-14g ──[H for help]──Hostname=localhost──Refresh= 2secs ──16:54.57
 Memory Stats
              RAM     High      Low     Swap    Page Size=4 KB
 Total MB   7406.4     -0.0     -0.0   8000.0
 Free  MB    154.1     -0.0     -0.0   7999.5
 Free Percent  2.1%   100.0%   100.0%   100.0%
            MB                 MB                 MB
                 Cached=  4977.2     Active=  3172.4
 Buffers=   19.0 Swapcached=    0.1  Inactive =  3751.2
 Dirty  =    0.2 Writeback =    0.0  Mapped   =   809.3
 Slab   =  200.1 Commit_AS =  7953.4 PageTables=    39.5
```

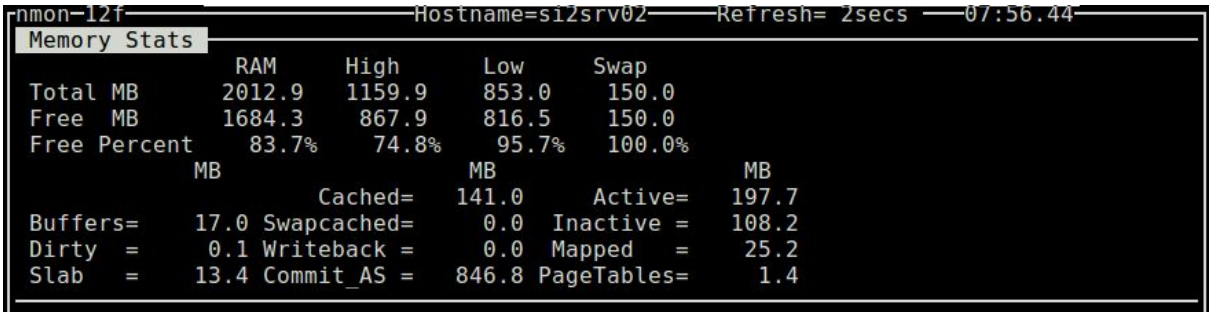
- VM1 (10.1.9.1)

```
si2@si2srv01:~$ free
             total       used       free     shared    buffers     cached
Mem:        767168     452532     314636          0      29348     164960
-/+ buffers/cache:     258224     508944
Swap:       153592          0     153592
```

```
nmon-12f ──────────────Hostname=si2srv01──Refresh= 2secs ──08:03.11
 Memory Stats
              RAM     High      Low     Swap
 Total MB    749.2      0.0    749.2    150.0
 Free  MB    305.7      0.0    305.7    150.0
 Free Percent  40.8%     0.0%    40.8%   100.0%
            MB                 MB                 MB
                 Cached=   161.2     Active=   299.7
 Buffers=   28.7 Swapcached=    0.0  Inactive =   120.6
 Dirty  =    0.0 Writeback =    0.0  Mapped   =    26.5
 Slab   =   14.1 Commit_AS =   908.3 PageTables=     1.6
```

- VM2 (10.1.9.2)

```
si2@si2srv02:~$ free
             total       used       free     shared    buffers     cached
Mem:       2061220     334976    1726244          0      17368     144240
-/+ buffers/cache:     173368    1887852
Swap:       153592          0     153592
```



Tras comprobar que no estamos saturando la RAM y que no estamos utilizando la swap realizamos manualmente un pago por cada método y comprobamos que funcionan correctamente. Tras ello procederemos a ejecutar el plan de pruebas.

Ejercicio 3

Tras ejecutar el plan de pruebas de P2.jmx comprobamos que efectivamente se han generado 3000 pagos.

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

visa=#
```

Resultado base de datos tras ejecución plan de pruebas

Ejecutamos varias veces el plan de pruebas completo, para eliminar interferencias en las mediciones producidas por el sistema. En todas ellas el porcentaje de error es de 0,00%. El log del servidor glassfish (**10.1.9.2**) correspondiente a una ejecución en limpio de las pruebas lo adjuntamos en la práctica en el archivo **server.log**. A continuación dos de los resultados de la ejecución, donde se comprueba que los datos apenas varían.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimien...	Kb/sec	Sent KB/sec
P1-base	1000	9	10	12	13	14	6	30	0,00%	99,7/sec	127,78	0,00
P1-ws	1000	48	47	56	61	71	34	330	0,00%	20,7/sec	26,72	0,00
P1-ejb-cli...	1000	24	25	29	31	34	15	40	0,00%	40,3/sec	51,28	0,00
Total	3000	27	25	50	54	65	6	330	0,00%	36,0/sec	46,21	0,00

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimien...	Kb/sec	Sent KB/sec
P1-base	1000	10	11	12	13	14	7	26	0,00%	91,9/sec	117,84	0,00
P1-ws	1000	47	47	56	60	67	34	254	0,00%	21,0/sec	27,16	0,00
P1-ejb-cli...	1000	24	25	31	32	36	14	46	0,00%	39,9/sec	50,77	0,00
Total	3000	27	25	50	54	63	7	254	0,00%	35,9/sec	46,04	0,00

Dos resultados ejecución plan de pruebas

A continuación una tabla con los valores medios obtenidos tras varias ejecuciones.

	P1-base	P1-ws	P1-ejb-cliente	Total
# Muestras	1000	1000	1000	3000
Media	9,7	47,4	24	27,03
Mediana	11	47	25	27
90% Line	12	56,2	31,1	33,1
95% Line	13,2	60,7	31,3	35,06
99% Line	14	68,5	35,2	39,23
Min	6	34	14	6
Máx	30	330	46	330
% Error	0,00%	0,00%	0,00%	0,00%
Rendimiento	96,8/secs	20,9/secs	40,1/secs	52,6/secs
Kb/secs	120,3	26,9	51,1	66,1
sent Kb/secs	0,00	0,00	0,00	0,00

A continuación deshabilitamos P1-base y P1-ws y ejecutamos la última prueba con el EJB local . Vemos que se generan correctamente 1000 pagos.

```
visa=# select count(*) from pago;
 count
-------
  1000
(1 row)
```

Base de datos tras prueba ejb-local

Los resultados son significativamente superiores, pues aumenta el rendimiento de **~40 peticiones/segundo** a **~284 peticiones/segundo**, el rendimiento de la ejecución en modo local es casi 8 veces superior.

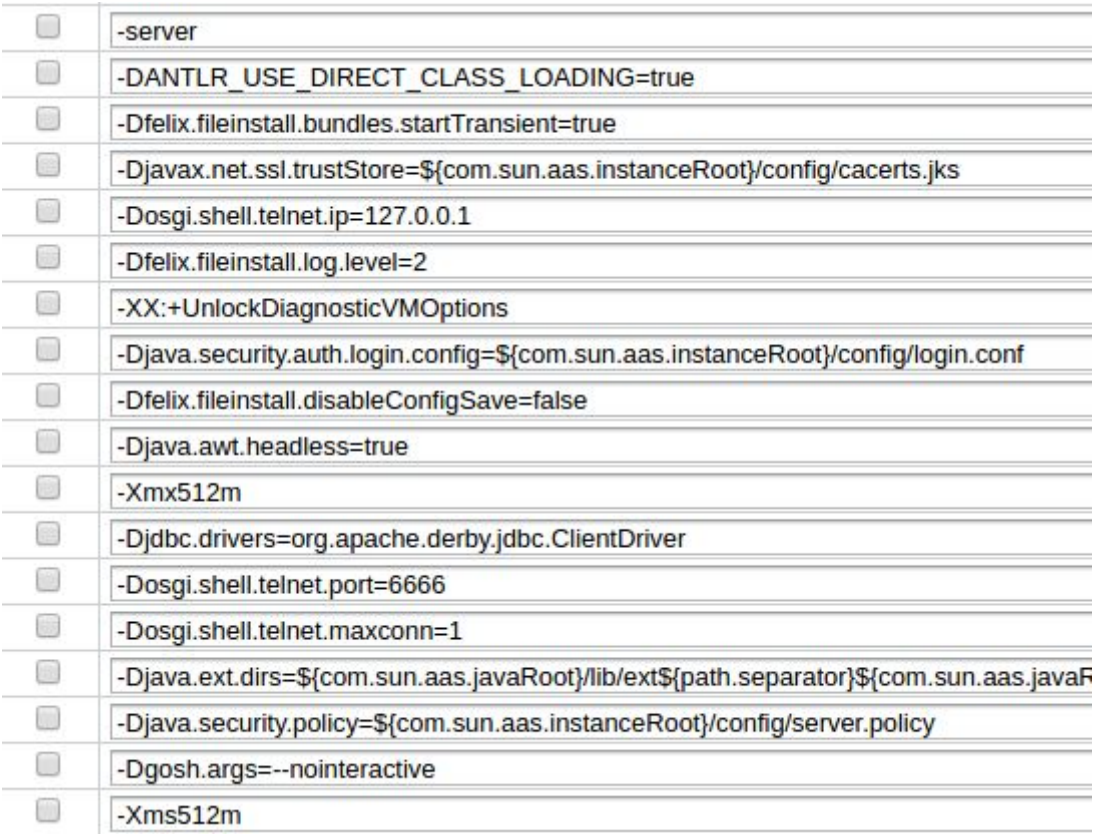
Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimien...	Kb/sec	Sent KB/sec
P1-ejb-cli...	1000	3	3	4	5	8	2	17	0,00%	284,1/sec	368,36	0,00
Total	1000	3	3	4	5	8	2	17	0,00%	284,1/sec	368,36	0,00

Resultado ejecución prueba ejb-local

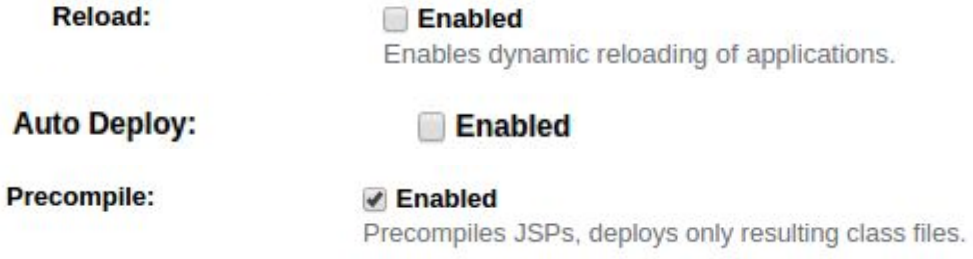
Esta diferencia es bastante lógica, pues las peticiones se mandan a la máquina virtual 1 (10.1.9.1) y esta resuelve los pagos directamente, pues la base de datos se encuentra en esta máquina. Además la maquina virtual 1 se encuentra en el ordenador en el que se está ejecutando jmeter por lo que las peticiones se realizan aún más rápido, pues no hay intercambio de paquetes fuera del PC1.

Ejercicio 4

Configuramos el servidor de aplicaciones (10.1.9.2) para simular el entorno de desarrollo



Valores de JVM options



Reload y auto deploy desactivados y autodeploy activado

Select	Module	Monitoring Level
☐	Jvm	HIGH
☐	Transaction Service	OFF
☐	Connector Service	OFF
☐	Jms Service	OFF
☐	Security	OFF
☐	Web Container	HIGH
☐	Jersey(RESTful Web Services)	OFF
☐	Web Services Container	OFF
☐	Java Persistence	OFF
☐	Jdbc Connection Pool	HIGH
☐	Thread Pool	HIGH
☐	Ejb Container	OFF
☐	ORB (Object Request Broker)	OFF
☐	Connector Connection Pool	OFF
☐	Deployment	OFF
☐	Http Service	HIGH

Detenemos el servidor de aplicaciones y copiamos el archivo de configuración con el comando

```
scp si2@10.1.9.2:/opt/glassfish4/glassfish/domains/domain1/config/domain.xml ./
```

El fichero domain.xml se adjunta en la entrega.

Tras revisar el script **si2-monitor.sh**, vemos cómo se realizan las peticiones de información de recursos con asadmin. Los comandos y sus resultados son los siguientes:

1. Max Queue Size del Servicio HTTP

```
e321079@localhost ~/Desktop/si2/Practica2/P2-alumnos $
asadmin -H 10.1.9.1 --user admin -W ./passwordfile get
configs.config.server-config.thread-pools.thread-pool.h
ttp-thread-pool.max-queue-size
configs.config.server-config.thread-pools.thread-pool.h
ttp-thread-pool.max-queue-size=4096
Command get executed successfully.
```

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

```
e321079@localhost ~/Desktop/si2/Practica2/P2-alumnos $
asadmin -H 10.1.9.1 --user admin -W ./passwordfile get
resources.jdbc-connection-pool.VisaPool.max-pool-size
resources.jdbc-connection-pool.VisaPool.max-pool-size=32
Command get executed successfully.
```

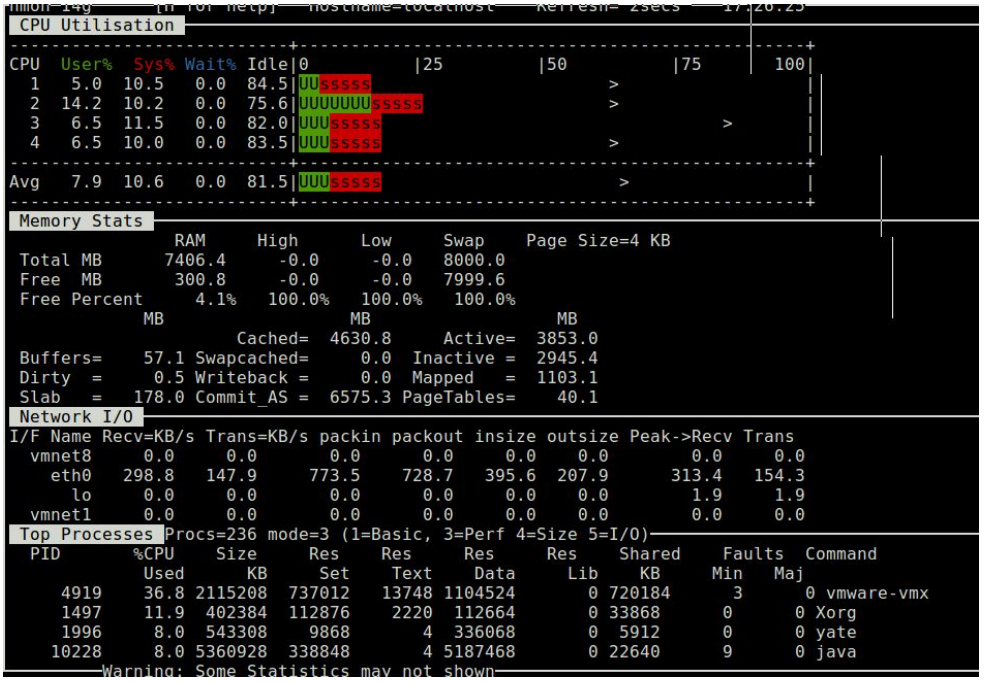
Ejercicio 5

Recurso	Valor
Máximo heap memoria JVM	512 Mb
Mínimo heap memoria JVM	512 Mb
Número máximo de hilos en la pool	5
Número máximo de cola de conexiones	4096
Número máximo de conexiones por contenedor Web	-1 (ilimitadas)
Número máximo de conexiones en pools JDBC	32

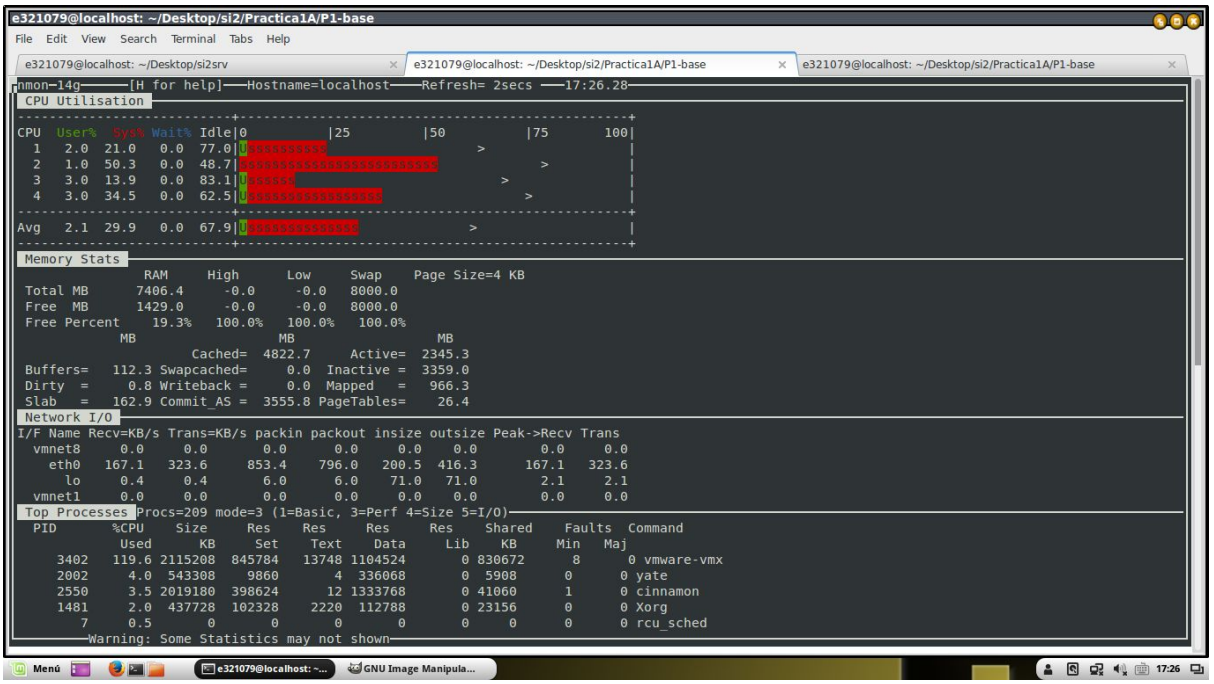
Ejercicio 6

Reiniciamos el servidor de la vm 2 y dejamos únicamente desplegada P1-base

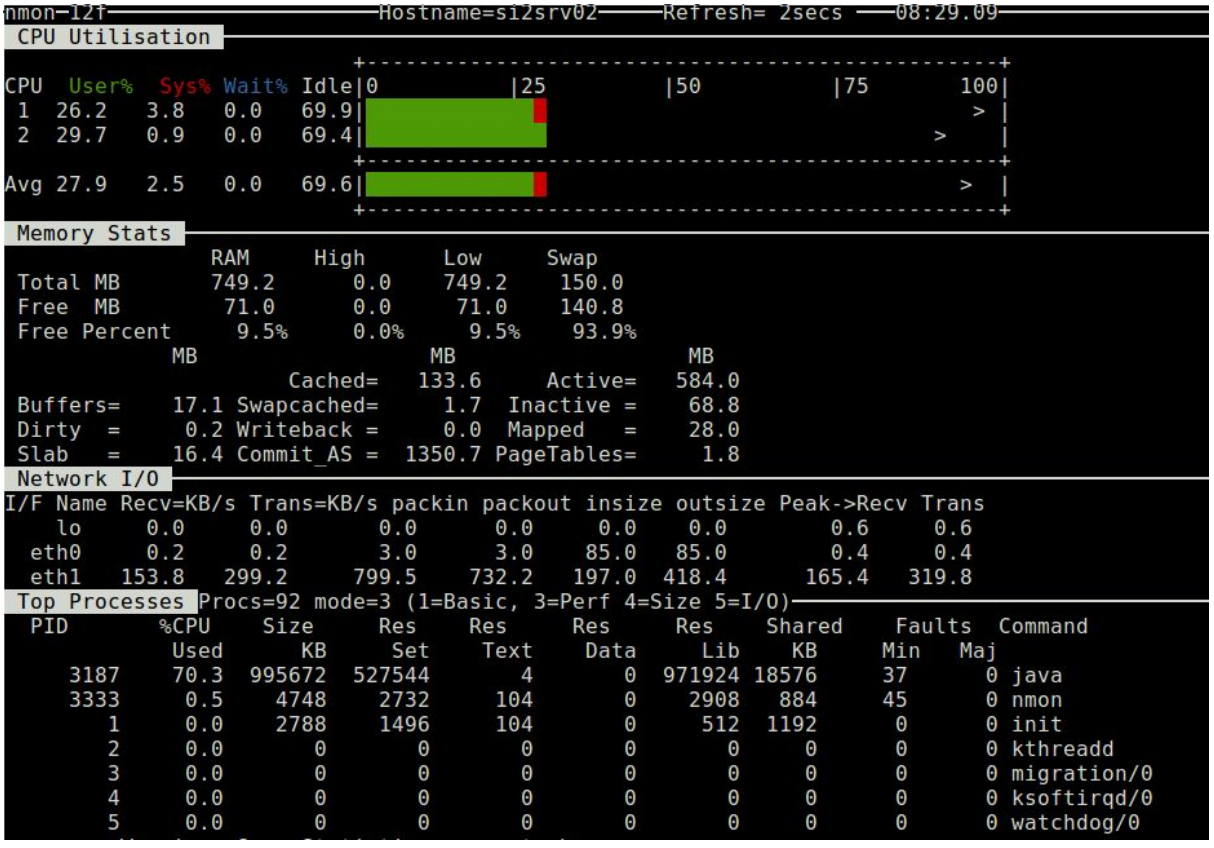
- Nmon de PC1



- Nmon desde PC2



- Nmon de VM2



- Resultado Script de monitorización

TOT.MUESTRAS MEDIA:

232 0.0517241 0.0603448 0

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
P1-base	1000	8	8	11	12	17	5	21	0,00%	114,3/sec	146,56	0,00
Total	1000	8	8	11	12	17	5	21	0,00%	114,3/sec	146,56	0,00

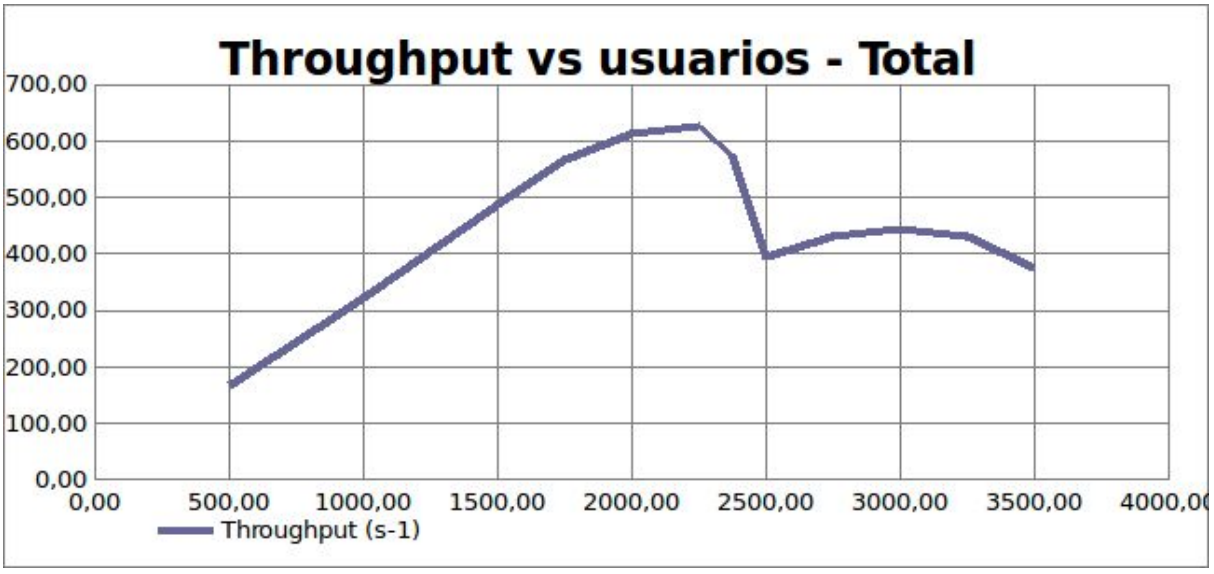
Ninguno de los recursos del servidor (CPU, Memoria, Red) se saturaron durante el plan de pruebas. De todas formas, el que que parecía más solicitado es la CPU.

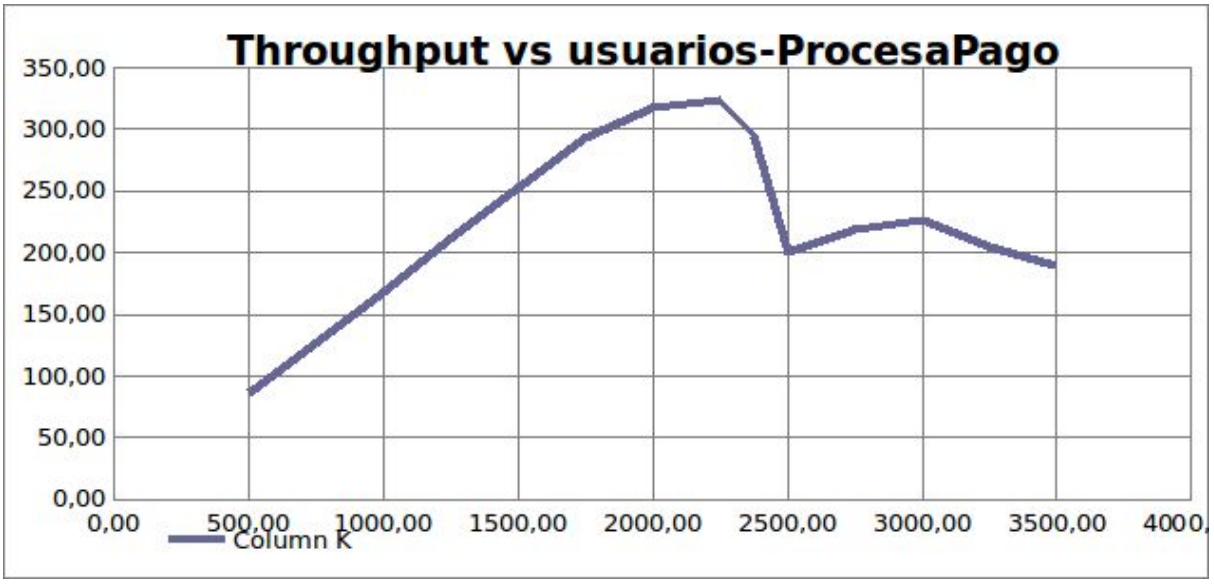
El situación simulada durante el ejercicio no parece muy realista. Principalmente la red: el entorno donde ejecutamos las pruebas es bastante controlado y carece de las variaciones impredecibles a las que podría estar sujeto un sistema distribuido conectado a través de Internet.

Un posible posibilidad para mejorar la capacidad de procesamiento del servidor sería no ejecutarlo en una MV, sino en un contenedor o directamente en una máquina real (aunque esto también tiene sus desventajas). Si estas alternativas no son posibles, se podría aumentar el número de cores destinados a la MV.

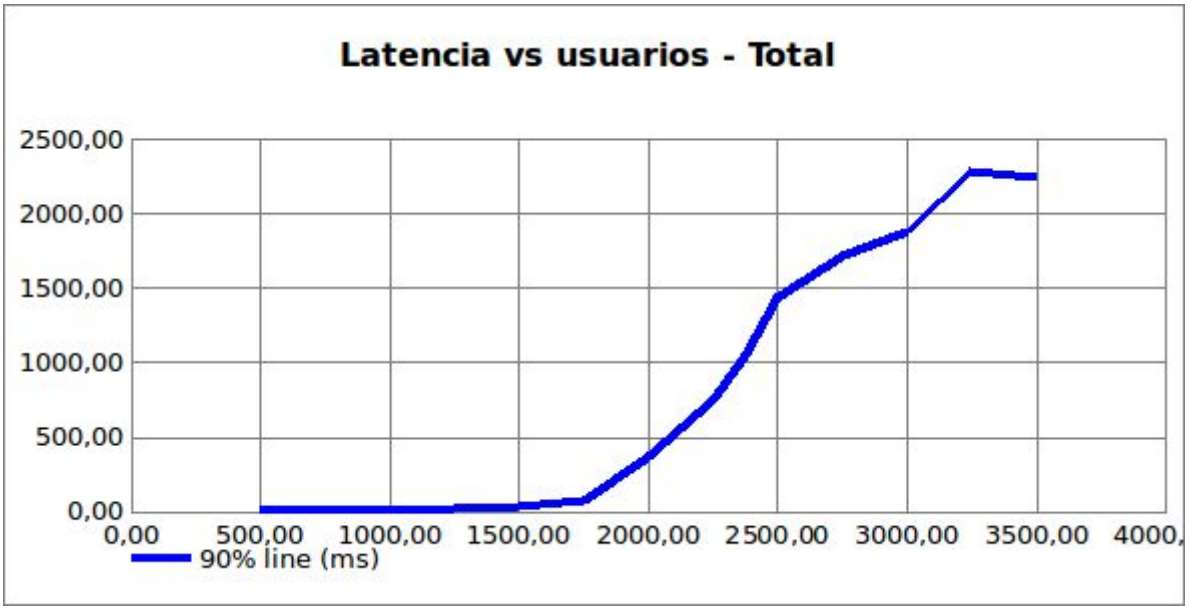
Ejercicio 8

Ejecutamos las pruebas y obtenemos la siguiente curva de rendimiento



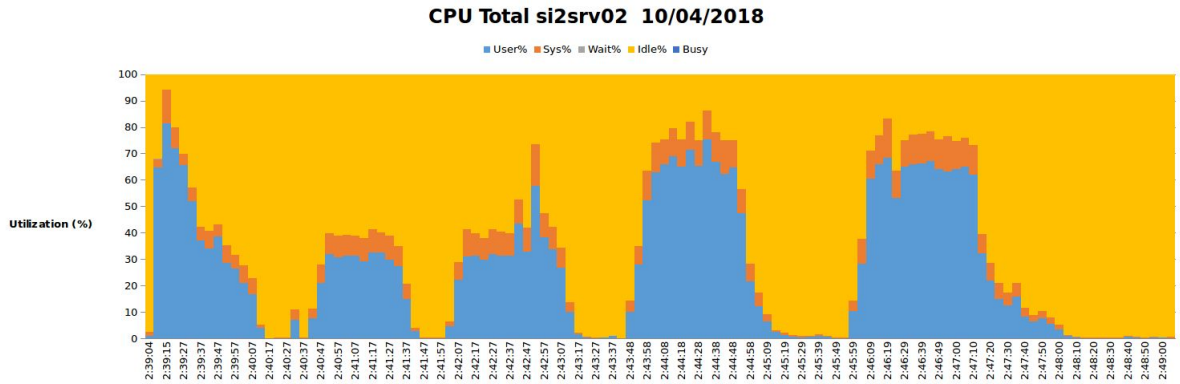


Vemos como el rendimiento decae completamente al llegar a 2500 hilos. De forma análoga aumenta la latencia de de los usuarios y aumenta significativamente cuando llega al punto de saturación.



Para recolectar muestras cada 5 segundos durante 10 minutos que incluya los top processes en un fichero llamado log-file.nmon ejecutamos:

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



Gráfica correspondientes al porcentaje de CPU usada, en ella se ven los picos de procesamiento correspondientes a las pruebas de P1-Base correspondientes con una ejecución errónea con 1000 hilos, 1000, 2000, 2500 y 3000 hilos correctas.

Ejercicio 9

Los datos obtenidos sugieren que el punto de saturación se encuentra alrededor de los 2250 usuarios. Con esta cantidad se obtiene el throughput máximo, en este caso de 628. A partir de este número de usuarios se entra en la zona de saturación, donde el throughput es más bajo. En esta zona el máximo fue de 445 y se consiguió con 3000 usuarios.

Hemos aumentado el número de máximas conexiones en el pool de hilos del servidor de aplicaciones, de 5 a 500 conexiones. Reejecutamos la prueba con 2375 hilos del jmeter, punto donde el rendimiento había decaído.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
/P1-base/c...	23750	82	4	295	1004	1035	1	1157	0,00%	378,6/sec	734,86	0,00
/P1-base/p...	23750	94	13	322	1014	1056	4	1168	0,00%	374,2/sec	457,92	0,00
Total	47500	88	10	304	1006	1046	1	1168	0,00%	725,0/sec	1147,14	0,00

Efectivamente ha aumentado el rendimiento, de 572/sec a 725/sec superando incluso el rendimiento máximo alcanzado en las pruebas anteriores.