UAM UNIVERSIDAD AUTONOMA DE MADRID		**Escuela Politécnica Superior Ingeniería Informática Prácticas de Sistemas Informáticos 2**			
Grupo	**2321**	**Práctica**	2	**Fecha**	14/04/2024
Alumno/a		García, Santa, Carlos			
Alumno/a		Junoy, Ortega, Eduardo			

Práctica 2: Rendimiento

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.

Se ha establecido un plan de pruebas utilizando JMeter. Este proceso implicó la configuración de varios elementos clave dentro de la herramienta siguiendo el tutorial, tales como grupos de hilos, samplers HTTP, extractores de expresiones regulares y listeners. De esta forma hemos podido simular cargas y medir con precisión el rendimiento de las aplicaciones P1-base, P1-ws y P1-ejb.

El archivo .jmx queda adjuntado en el entregable de la práctica.

Ejercicio 2:

Preparar el PC con el esquema descrito en la Figura 30. Para ello:

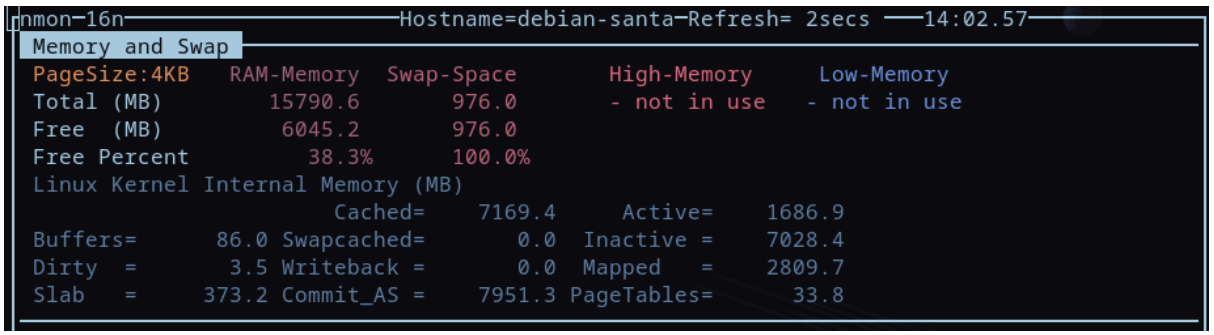
- **Anote en la memoria de prácticas las direcciones IP asignadas a las máquinas virtuales y al PC**
- **Inicie los servidores GlassFish en las máquinas virtuales**
- **Repliegue todas las aplicaciones o pruebas anteriores (P1-base, P1-ws, etc), para limpiar posibles versiones incorrectas.**
- **Revise 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, los ficheros web.xml y glassfish-web.xml,de los proyectos P1-ws y P1-ejb respectivamente, para modificar las IPs de despliegue de los servidores.**
- **Despliegue las siguientes prácticas: P1-base, P1-ws, P1-ejb, 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 será PC2VM con IP 10.X.Y.2 (as.host.client de P1-ws y as.host.client de P1-ejb)**
  - **El destino del despliegue de la parte servidor de P1-ws y de P1-ejb será PC1VM con IP 10.X.Y.1 (as.host.server de P1-ws y as.host.server de P1-ejb)**
  - **La base de datos en todos ellos será la de PC1VM con IP 10.X.Y.1 (db.host)**

Se han realizado todos los puntos mencionados, las direcciones IP asignadas a las máquinas virtuales y al PC son:

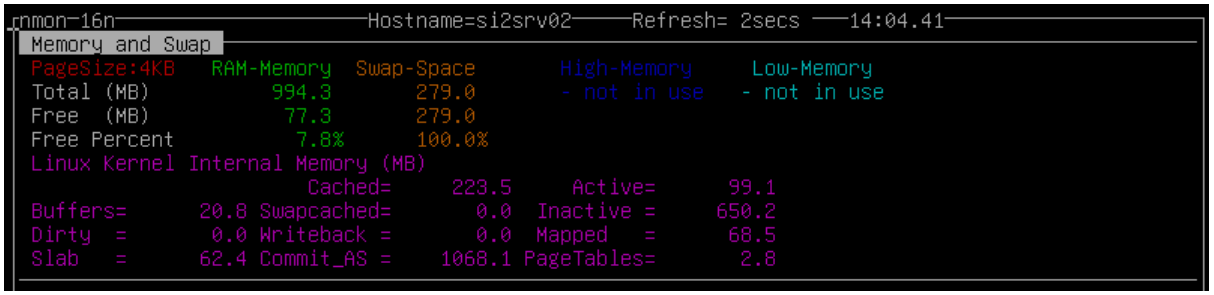
- Servidor: 10.4.9.1
- Cliente: 10.4.9.2
- PC: 10.10.0.18

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 voto "de calentamiento" por cada uno de los métodos anteriores y verifique que funciona a través de la página testbd.xhtml. (comprobado)

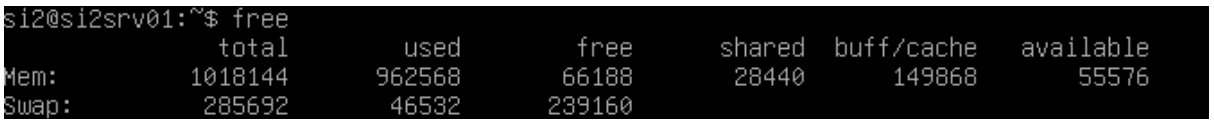
Comando *nmon* en el PC host:

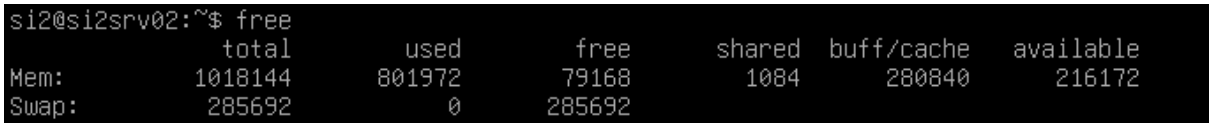


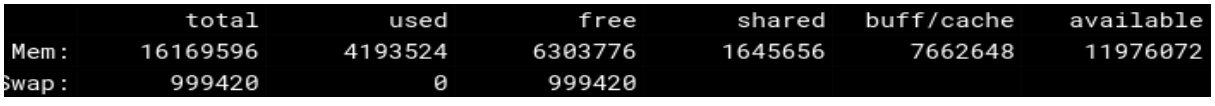
Comando *nmon* en las máquinas virtuales:



Comando free en las máquinas virtuales y host:







El PC host cuenta con suficiente memoria libre, aproximadamente 6 GB de 15.7 GB, para manejar operaciones de carga intensiva, lo cual es favorable para las pruebas que vamos a realizar. No se está utilizando la memoria swap, así que memoria física disponible es adecuada.
En las máquinas virtuales, la memoria disponible es bastante limitada. En este caso solo 77 MB libres de 994.3 MB. Esto ha resultado en el uso de la memoria swap, que asciende a 279 MB. Este uso de swap es un indicativo de que la memoria física no es adecuada, lo cual podría afectar negativamente el rendimiento de las pruebas.

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 votos. Es decir, la siguiente consulta deberá devolver "3000":**

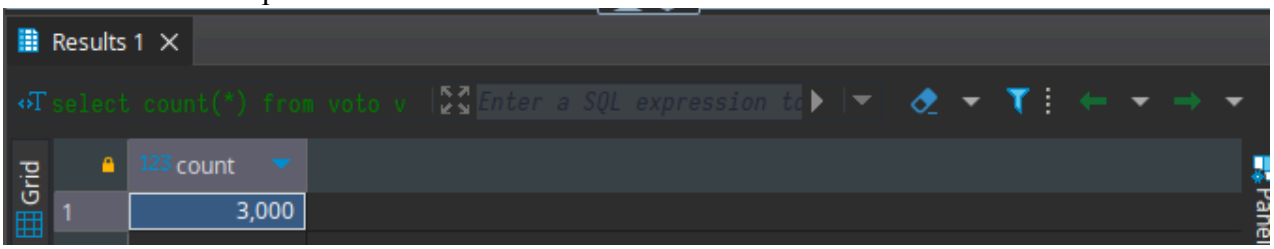
  SELECT COUNT(*) FROM votos;
- **Compruebe que ninguna de las peticiones ha producido un error. Para ello revise que la columna %Error indique 0% en todos los casos.**

Una vez que los resultados han sido satisfactorios:

- **Anote los resultados del informe agregado en la memoria de la práctica.**
- **Salve los ficheros server.log que se encuentra en la ruta glassfish/domains/domain1/logs de Glassfish de las MVs y adjúntelo con la práctica (puedes usar el comando scp para copiar los ficheros de las MVs al PC host).**
- **Añada a la memoria de prácticas la siguiente información: ¿Cuál de los resultados le parece el mejor? ¿Por qué crees que ese proyecto obtiene el mejor resultado? ¿Qué columna o columnas elegiría para decidir este resultado?**

Incluir el directorio P2 en la entrega.

La consulta da la respuesta correcta:



En ninguna de las peticiones se ha producido un error:

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
P1-base Get	1000	1	1	1	2	2	0	39	0.00%	304.1/sec	1959.13	40.10
P1-base Post	1000	2	2	2	3	4	1	20	0.00%	305.6/sec	151.36	216.53
P1-ws-cliente Get	1000	2	1	3	3	6	1	706	0.00%	31.8/sec	205.86	4.38
P1-ws Post	1000	28	11	21	27	552	7	4601	0.00%	31.8/sec	15.77	22.74
P1-ejb-cliente Get	1000	1	1	2	2	6	1	56	0.00%	66.6/sec	431.04	9.24
P1-ejb-cliente P...	1000	13	11	14	17	29	9	584	0.00%	66.7/sec	33.02	47.69
TOTAL	6000	8	2	13	15	29	0	4601	0.00%	120.6/sec	419.49	51.20

El fichero .log se adjunta en el entregable.

P1-base muestra el mayor throughput, lo cual significa que pueden manejar más solicitudes por segundo en comparación con el resto, menores tiempos de respuesta y cero errores, lo que hace que sea el de mejor rendimiento entre las opciones dadas.

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 con scp.

Revisar el script si2-monitor.sh e indicar los mandatos asadmin 2 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.

1. Para averiguar el valor Max Queue Size del Servicio HTTP indicamos:

carlos@debian-santa:~/Code/SI2/P2$ /opt/glassfish7/glassfish/bin/asadmin --host 10.4.9.1 --user admin --passwordfile ./passwordfile get configs.config.server-config.thread-pools.thread-pool.ht

tp-thread-pool.max-queue-size

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

Command get executed successfully.

2. Para conseguir el Maximum Pool Size del Pool de conexiones a nuestra DB escribimos:

carlos@debian-santa:~/Code/SI2/P2$ /opt/glassfish7/glassfish/bin/asadmin --host 10.4.9.1 --u

ser admin --passwordfile ./passwordfile get resources.jdbc-connection-pool.VotoPool.max-pool

-size

resources.jdbc-connection-pool.VotoPool.max-pool-size=**32**

Command get executed successfully.

3. Para monitorizar el número de errores:

carlos@debian-santa:~/Code/SI2/P2$ /opt/glassfish7/glassfish/bin/asadmin --host 10.4.9.1 --user admin --passwordfile ./passwordfile get server.network.http-listener-1.http-service.virtual-server.server.request.error-count

server.network.http-listener-1.http-service.virtual-server.server.request.error-count=**0**

Command get executed successfully.

Ejercicio 5:

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

Se adjunta la hoja de cálculo a la entrega.

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, red, ...)**
- **Teniendo en cuenta los resultados de monitorización obtenidos, ¿le parece una situación realista la simulada en este ejercicio? ¿Por qué?**
- **Describa la cadena de procesamiento asociada a la aplicación. Esto es, qué componentes: red, servidor de aplicaciones, base de datos, etc., son utilizados en cada solicitud de un cliente.**
- **Teniendo en cuenta cuál ha sido el elemento más saturado, proponga otro esquema de despliegue que resuelva esa situación y repita el plan de pruebas para ver hay mejora del rendimiento.**

Durante la monitorización del sistema con *nmon* en el entorno virtual, se observó que el elemento más utilizado fue la CPU. Esta alcanzó un pico de utilización del 81.5%

No es una situación realista, ya que es un único usuario realizando 1000 peticiones, además el usuario realiza las peticiones instantáneamente. Además, al tener postgresql desactivado en el máquina virtual cliente no se realizan operaciones escritura/lectura sobre la base de datos obteniendo una actividad nula en disco.

La cadena de procesamiento asociada a la aplicación tiene los siguientes componentes:

- Red: La solicitud del cliente llega al servidor de aplicaciones a través de la red. Aquí, la latencia y el ancho de banda pueden afectar el tiempo de respuesta.
- Servidor de Aplicaciones (GlassFish): Procesa las solicitudes HTTP recibidas. Dentro del servidor de aplicaciones, diferentes hilos gestionan las peticiones, ejecutan la lógica de negocio y formulan cualquier solicitud necesaria a la base de datos.
- Base de Datos (PostgreSQL): Si la solicitud requiere operaciones de lectura/escritura, el servidor de aplicaciones se comunica con la base de datos, la cual gestiona las transacciones y devuelve los resultados al servidor de aplicaciones.
- Disco Duro: Donde reside la base de datos y sobre el que se realizan operaciones de lectura y escritura de datos. Aunque en el escenario dado está inactivo debido a que PostgreSQL se desactivó en la máquina virtual cliente.

Como posibles mejoras del servidor podemos:

- Implementar un balanceador de carga que distribuya las solicitudes entre múltiples instancias del servidor de aplicaciones podría repartir la carga de trabajo de la CPU.
- Añadir un escalado horizontal del servidor de aplicaciones porque más servidores de aplicaciones y configurarlos en un clúster puede ayudar a manejar más peticiones simultáneamente, reduciendo la carga sobre un solo servidor y mejorando la utilización de la CPU.
- Añadir o mejorar la velocidad de los núcleos, que proporciona mayor paralelismo y procesamiento más rápido de las tareas.

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 fichero.csv, ya que, de dicho archivo, al igual que en los ejercicios anteriores, se obtienen los datos necesarios para simular el voto.
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.

Para realizarlo hemos cambiado la IP a la de nuestro servidor de aplicaciones, después hemos cambiado el thinkTimeMin y el thinkTimeMax y, por último, hemos comprobado que los pagos se realizan correctamente observando los ticks.

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 votos en la base de datos voto.**
- **Ejecutar la herramienta de monitorización nmon en ambas máquinas.**
- **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 %). 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 3.**
- **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:**
    - **Post censo.xhtml.**
    - **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.

Todos los resultados están incluidos en las carpetas de la entrega.

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.**

La latencia se mantiene relativamente estable al principio, pero comienza a aumentar de manera más pronunciada a medida que el número de usuarios se acerca a 1500. Este aumento en la latencia puede ser un indicador de que el sistema está alcanzando su límite de capacidad de manejo, provocando tiempos de respuesta más lentos. El punto de saturación se alcanza entre los 1750 y 2000 usuarios.

Para lograr obtener un punto de saturación en un número mayor de usuarios se puede aumentar la cantidad de procesadores, ya que la máquina virtual únicamente utiliza uno, o utilizar las técnicas descritas anteriormente: implementar un balanceador de carga y añadir un escalado horizontal o vertical.

Tras aumentar el número de CPUs, el rendimiento ha mejorado con creces. Ha mejorado el uso medio de la CPU, las tareas y los valores del monitor.