**Caso de Estudio 3 – Análisis de Desempeño Logística y Seguridad Aeroportuaria**

**Informe**

Sergio Madera, código 201417957

Andrés Zuleta, código 201413405

1. **Descripción de los cambios realizados para la medición de indicadores**

Las modificaciones realizadas sobre la aplicación para medir los tiempos solicitados consistieron de la adición de apróximadamente ocho líneas de código sobre el método que maneja el protocolo de comunicación con seguridad, además de la creación de dos clases, IndicadorTransaccion y ReporteTransaccion, que guardaran los tiempos medidos para una transacción específica y almacenaran los tiempos medidos para un cierto grupo de transacciones respectivamente. En adición a esto, se creó una función que manejara el protocolo de comunicación sin seguridad. Respecto a los cambios realizados en este y en el método que maneja el protocolo con seguridad, se crearon dos variables de tipo float para guardar dos momentos en el tiempo de ejecución diferentes. Posteriormente, se realizaron asignaciones a estas variables en los momentos de ejecución apropiados utilizando System.nanoTime(), de tal forma que al restar los valores de tiempo contenidos en ambas variables fuera posible conocer el tiempo que se tardó en ejecutar cierta parte del programa. No obstante, el valor obtenido está en nanosegundos, por lo que se realiza la conversión apropiada a segundos y se almacena dicho valor en una instancia de IndicadorTransaccion, el cual a su vez es guardado en un reporte, es decir, en una instancia de ReporteTransaccion.

1. **Identificación de la plataforma**

La máquina en la que se corrió el servidor y se realizaron todos los experimentos tiene las siguientes características:

* Arquitectura de 64 bits
* 8 núcleos (cores)
* Procesador de 2.40 GHz
* Memoria RAM de tamaño de 12GB
* 1GB de memoria asignado a la JVM

1. **Respuestas al punto 3**

La información recopilada mediante la realización de experimentos se encuentra registrada en un archivo de Microsoft Excel adjunto a la entrega. A continuación, se presentan las gráficas solicitadas, que fueron generadas a partir de los datos registrados.

1. Gráficas de # threads vs. tiempo de autenticación
2. Gráficas de # Threads vs. # transacciones perdidas. El número de transacciones perdidas es abreviado como “# transac\_perdidas”. En adición, se interpretó este indicador por el número de transacciones perdidas en relación al total de transacciones intentadas, es decir, un porcentaje.
3. Gráficas de # threads vs porcentaje de uso de la CPU
4. Gráfica de tiempo de autenticación vs porcentaje de uso de la CPU para todas las cargas con un número fijo de 2 threads.
5. Gráfica de tiempo de autenticación vs porcentaje de uso de la CPU para todas las cargas con un número fijo de 16 threads.

Se concluye a partir de los datos obtenidos mediante las pruebas que el tiempo de respuesta de las transacciones individuales se mantiene relativamente constante para diferentes tipos de carga, siempre que el número de threads sea el mismo. Conforme aumenta el número el threads este tiempo se reduce para ciertas configuraciones, pero para cada una hay un cierto número de threads en el cual su utilización le cuesta más al sistema de lo que su capacidad de atender solicitudes de manera concurrente compensa. Para todos los tipos de carga, esto sucede al menos cuando el número de threads es de 16 por las razones mencionadas anteriormente; a pesar de que 16 threads son capaces de atender una gran cantidad de transacciones concurrentes, coordinarlos y administrarlos también es costoso para el sistema, por lo que no es recomendable utilizar tantos threads si la carga que se espera no es lo suficientemente grande. Respecto al porcentaje de transacciones perdidas, también tendió a aumentar al tiempo que el número de threads, pero el promedio de esta no fue alarmantemente alto para ningún escenario. Algo similar sucede con el porcentaje de uso de la CPU relacionado al tiempo de autenticación. En adición, los tiempos de respuesta monitoreados no subieron hasta valores inaceptables en ningún momento, por lo que se considera que el sistema es capaz de responder a todas las configuraciones definidas de forma satisfactoria.

1. **Respuestas al punto 4**

Al comparar el comportamiento entre aplicaciones que implementan funciones de seguridad y aplicaciones que no las implementan, se espera que las aplicaciones que no utilizan funciones de seguridad tengan mejores tiempos de respuesta, puesto que, sin importar qué tan eficientes relativamente sean las funciones de seguridad de una aplicación, su costo temporal no se hace completamente insignificante. En adición, la diferencia entre los tiempos de respuesta será más notoria para aquellas transacciones que utilizan más funciones de seguridad. Para propósitos prácticos, sin embargo, las pocas decenas de milisegundos de costo extra que cargan las aplicaciones con seguridad son un intercambio razonable por el bienestar de la información que le confían al programa.

A continuación, se encuentran las mismas gráficas generadas en el punto 3, esta vez creadas a partir de datos obtenidos en pruebas sobre un servidor sin seguridad.

1. Gráficas de # threads vs. tiempo de autenticación con un protocolo sin seguridad

1. Gráficas de # Threads vs. # transacciones perdidas en un servidor sin seguridad
2. Gráficas de # threads vs porcentaje de uso de la CPU en un servidor sin seguridad
3. Gráfica de tiempo de autenticación vs porcentaje de uso de la CPU para todas las cargas con un número fijo de 2 threads en un servidor sin seguridad.
4. Gráfica de tiempo de autenticación vs porcentaje de uso de la CPU para todas las cargas con un número fijo de 16 threads en un servidor sin seguridad.

Se puede observar en las gráficas que el tiempo de respuesta para la solicitud de actualización disminuye de forma considerable en esta ocasión. Esto sucede porque dicha solicitud utiliza una gran cantidad de funciones de seguridad, y, al no ser implementadas, el tiempo que tarda en terminarse la transacción para el usuario final se comporta de esta manera. No está de más mencionar que el tiempo de respuesta de la transacción de autenticación, mucho menos pesada en seguridad, se mantuvo relativamente constante respecto a las primeras pruebas realizadas, documentadas con anterioridad en este escrito. En consecuencia, se concluye que el comportamiento observado en los experimentos realizados sobre un servidor sin seguridad es similar a aquél esperado en esta situación.