**UNIVERSIDAD DE LOS ANDES**

**FACULTAD DE INGENIERÍA**

**DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN**

**INFRAESTRUCTURA DE COMUNICACIONES**

**Profesor: Harold Enrique Castro Barrera**

[**hcastro@uniandes.edu.co**](mailto:hcastro@uniandes.edu.co)

**Asist. De laboratorio: Laura María Ruiz Gómez**

[**lm.ruizg@uniandes.edu.co**](mailto:lm.ruizg@uniandes.edu.co)

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **INFORME DE LABORATORIO 5 – ANÁLISIS DE COMUNICACIÓN POR SOCKETS TCP** | |  | |

**Nombres: -Ana María Espinosa Chaparro Códigos: -201425031**

**-Fabio Andrés López Corredor -201423782**

1. **Describa el proceso para determinar el número máximo de conexiones.**

Para determinar el número máximo de conexiones se deben tener en cuenta el uso de recursos físicos, el tiempo de respuesta y el tamaño de los archivos que se van a enviar por comunicación. Si no se establece correctamente este parámetro puede ocurrir que una sola conexión utilice la mayoría de los recursos, limitando el desempeño de las otras conexiones. En este caso no se tienen que cumplir con un número específico de conexiones simultáneas por lo que se pueden realizar pruebas para determinar cuál es el más conveniente para la aplicación.

1. **Cambie el tamaño del buffer a un 10% de su tamaño inicial y realice la transferencia de un archivo, vuelva a modificar el tamaño del buffer a un 500% del tamaño inicial y** **realice la transferencia de un archivo. Describa los cambios observados, haga el análisis comparativo de las dos situaciones y concluya determinando cual es el valor adecuado para su diseño.**

En principio se estableció el tamaño del buffer en 45 y se cambió a 50 y a 225. Se

observó que en los dos primeros casos los tiempos de ejecución fueron muy similares, cambiando solamentes unos pocos milisegundos. En el tercer caso se observa un aumento de masomenos 3 segundos en comparación con los otros dos casos. Esto se debe a que hay mayor cantidad de peticiones a ser procesadas por el servidor y se van a consumir mayor cantidad de recursos, volviendo un más lento el tiempo de respuesta. Para determinar los tiempos mencionados anteriormente, dentro del servidor se crearon unas variables tipo long que tomaban el tiempo, “currentTimeMillis” que indicaban tiempo inicial y tiempo final, y después se restaban estas. Por lo anterior es posible decir que un valor óptimo para el buffer en esta aplicación es de 50.

1. **Modifique el tamaño de los mensajes a un 10% del valor inicial y realice la transferencia de un archivo, luego a un 200% del valor inicial y realice la transferencia de un archivo, describa los cambios observados, realice el análisis comparativo de las dos situaciones y concluya determinando cual es el valor adecuado para su diseño.**

El tamaño de los mensajes inicialmente se configura con un número de 1024 bytes. Al modificar este tamaño al valor de 102 bytes se esperaba un decremento en el tiempo de descarga del archivo. Esto no solo ocurre dado que cliente y servidor deben imprimir el valor del paquete que están transfiriendo. También, se ve este deterioro en la velocidad ya que en teoría lo que hace el paquete, al agrupar los bytes de datos, es añadir un encabezado con el puerto destino y en general con información para controlar su envío.

Al usar un 200% del valor inicial, es decir 2048 bytes, se ve un incremento en la velocidad de envío de los archivos. Resulta que al aumentar el tamaño de los paquetes, en teoría, se empieza a aumentar el tiempo de envío dado que el tiempo de transmisión aumenta con el tamaño de cada paquete.Esto no se da de esta forma gracias a las dificultades de paquetes muy pequeños, mencionadas anteriormente.

Tras experimentación e investigación se encuentra que alrededor de 42KB es el tamaño adecuado para cada paquete. Entre los valores sugeridos el que más se acerca a este es el de 2 KB, razón por la cual vimos una mejor eficiencia para este caso y lo tomamos como el valor más adecuado para el diseño realizado.

Es necesario aclarar que al tratarse de un protocolo orientado a conexión esta modificación del tamaño de los paquetes es más una simulación antes que otra cosa. El tamaño de los paquetes suele estar predeterminado para los sockets y no suele ser cambiado precisamente para poder contar y manipular byte por byte.

1. **Cambie los valores de las preferencias iniciales, describa los cambios percibidos, realice el análisis comparativo de las diferentes situaciones y concluya determinando cual es el valor adecuado para su diseño. Para esta prueba use la transferencia del archivo de tamaño grande.**

Al modificar los tres valores de preferencias se observan al principio cambios desfavorables en la aplicación. Al modificar el tiempo de conexión lo que se hace es disminuir el tiempo en que se puede perder una conexión por ausencia de peticiones hechas. Esto es desfavorable desde el punto de vista del cliente final pues debe estar realizando peticiones de forma mucho más rápida. Para el servidor esto es muy favorable dado que no tiene que reservar espacio de memoria para un cliente que no realiza peticiones de forma frecuente, y puede atender otras conexiones más importantes (que sí requieren envío de archivos frecuentemente). Si este tiempo se aumenta se verán resultados inversos para cliente y servidor, o al menos así sería de no ser porque llegado un punto se empezarían a bloquear las conexiones hechas por nuevos clientes.

Al generar una latencia alta y un menor ancho de banda la calidad de las transacciones se ve afectada fuertemente. Con un ancho de banda menor los bytes enviados por unidad de tiempo son menores y los archivos demoran mucho más tiempo en ser enviados. Ocurre exactamente lo mismo cuando la latencia soportada por el servidor es mayor, pues permite un tiempo mayor de respuesta para cada cliente, haciendo más ineficiente al sistema y ocupando por más tiempo sus recursos.

1. **Para esta parte de la práctica, se deberá hacer una descripción del diseño e implementación de cada conjunto de pruebas. Así mismo, se solicita la muestra de resultados, análisis y conclusiones para el grupo de pruebas. Apoyarse en las gráficas necesarias para sustentar lo anterior.**

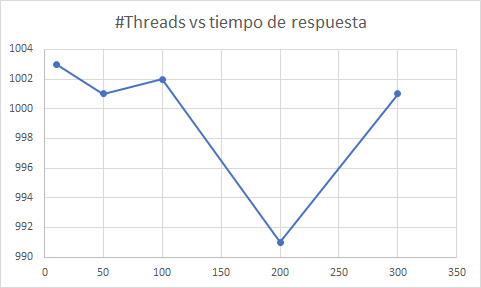
Para realizar las pruebas se crea un conjunto de threads que realicen conexiones y pidan archivos al servidor TCP. Cada thread representa un usuario, por lo que los threads ejecutados que realizan cada petición se aumentan como es indicado: 10, 50, 100, 200 y 300 (dos grupos de threads más que los sugeridos). Se diseñó para seguir aumentando este número hasta que el servidor empiece a presentar transacciones no realizadas y así comprobar la cantidad de conexiones que se deben soportar.

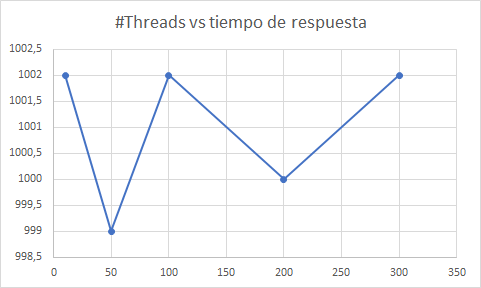
Las pruebas se realizan con un ramp-up de 1 segundo, lo que quiere decir que todos los threads ejecutados inician la conexión en este periodo de tiempo. Esto se hace de este modo para probar un escenario de usuarios concurrentes para grandes volúmenes de usuarios. Solo se hace una iteración en cada experimento, pues el número de usuarios ya se simula con la cantidad de threads y no tiene sentido hacer más de una conexión por thread para este caso.

Dentro de cada transacción se determina la IP y el puerto del servidor, junto al mensaje que se envía por el flujo al servidor.

Resultados:

Las siguientes son las gráficas de latencia para las dos iteraciones realizadas de los experimentos:





En ambas iteraciones de las pruebas se ve un tiempo de respuesta constante de la aplicación, lo cual indica un comportamiento favorable dada la cantidad de usuarios con las que debe lidiar el servidor. También se ve constantemente un error en la lectura de la información enviada por jmeter a la aplicación, dado que el servidor solo reconoce información con el formato UTF, el cual se logra enviar desde el cliente en java pero al hacer la conexión directa desde JMeter no se puede conseguir.

Se recomienda repetir la prueba evaluando este detalle para poder así avanzar en número de threads y ver un incremento en el tiempo de latencia para mayor número de usuarios.

1. **Analizar las aplicaciones Waze[[1]](#footnote-0), Netflix[[2]](#footnote-1), y Skype[[3]](#footnote-2). De acuerdo a los análisis efectuados en la práctica y a referencias adicionales, argumentar cuál es el tipo de comunicación empleada por la aplicación.**

Después de analizar las aplicaciones se puede decir que:

* Waze es una aplicación que muestra el tráfico en las ciudades y además muestra la ruta más óptima para llegar a algún lugar. En principio para mostrar el tráfico en la ciudad, Waze utiliza UDP para reportar que hay trancones o accidentes, esto funciona bien así debido que las personas quieren conocer lo más rápido posible como se encuentra una vía en específico, y en caso de que ese accidente ya no se encuentre en esa zona no afectará a los usuarios, pero sí que sepan al instante si ocurre uno. Para definir una ruta a los usuarios Waze deberá utilizar TCP debido a que no puede ocurrir que muestre un camino y este tenga saltos entre una calle y otra, tendrá que mostrarle la ruta exacta al usuario sin ninguna pérdida. Es por esto que a la hora de calcular una nueva ruta el usuario tiene que esperar un momento.
* Netflix es una aplicación en la que se encuentra un catalogo de peliculas y series que los usuarios pueden ver. Este utiliza TCP debido a que garantiza a los usuarios que si eligen una película específica esta sea la que van a poder ver, y además que mientras el usuario este visualizando el contenido, este no vaya a tener saltos en las escenas o vean alguna parte del contenido se pierdan detalles como los subtítulos o rasgos de los personajes. Es por esta razón que a la hora de reproducir una película se da un pequeño tiempo de antes de poder verla.
* En Skype se ofrece el servicio de realizar llamadas y videollamadas. Para que las personas no sientan el retardo que se da al transferirse la voz y las imágenes estos datos son enviados por UDP, y a pesar que se de la pérdida de algún paquete los usuarios repetirán lo que están diciendo y seguirán con la comunicación, en cambio a si se utiliza TCP en donde se demoraria mas tiempo en transmitir los mensajes y los usuarios se sentirían molestos con estos retrasos. A pesar de esto, Skype utiliza TCP cuando un usuario A indica que quiere comunicarse con el usuario B, para garantizar que la comunicación si se de entre estos dos usuarios.

1. Página oficial: <https://www.waze.com/es/> [↑](#footnote-ref-0)
2. Página oficial: <https://www.netflix.com/co/> [↑](#footnote-ref-1)
3. Página oficial: <http://www.skype.com/en/> [↑](#footnote-ref-2)