

Laboratorio 3 Actividad 1

Sección 2 – Grupo 1

Camila Gomez

Diego Camelo

Juliana Galeano

William Mendez

Enlaces importantes:

Video: <https://youtu.be/ny23KAnETJo>

Repositorio: <https://github.com/InfraCom-G1/Lab3-1.git>

Drive con capturas de Wireshark y datos de 250MB:

https://drive.google.com/drive/folders/17pg4aoH1kCBs_fS0UK5nzf-oLmNzjNCR?usp=sharing

1. Configuración del ambiente de trabajo

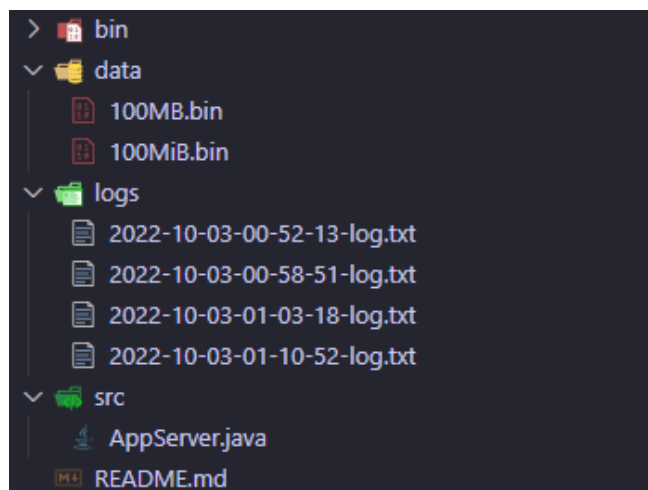
Para el desarrollo del laboratorio se usó como máquina cliente un computador personal con SO Windows 10 y como máquina servidor se usó una MV con Ubuntu 20.04 haciendo uso de VMware y la imagen dada como recurso.

2. Servidor de transferencia de archivos

Este se realizó una aplicación en java, esta aplicación recibe como parámetros:

- El tamaño de archivo que se va a enviar cero para un archivo de 100 megas, uno para un archivo de 250 megas.
- El puerto que será utilizado que por defecto debería ser 5555.
- La cantidad de clientes que se conectarán a la aplicación.

Una vez ingresan estos parámetros el servidor crea un delegado por cada solicitud y cuando se conectan esta cantidad de clientes la aplicación espera que los clientes realicen una solicitud de archivo para responder a cada cliente una confirmación de que se recibió el mensaje, luego espera que el cliente solicite el tamaño del archivo para enviarlo, una vez hecho esto el servidor espera que todos los clientes manden una confirmación de que están listos para recibir, esta espera se realiza con una barrera. en este punto el servidor envía el archivo y espera a que el cliente le solicite el hash calculado por el servidor para compararlo con el local si el cliente confirmó que el hash es correcto se termina la conexión, si no se reenvía el archivo hasta que este proceso sea exitoso, si en cualquier punto de la comunicación se envía un mensaje incorrecto por parte del cliente la conexión se termina.



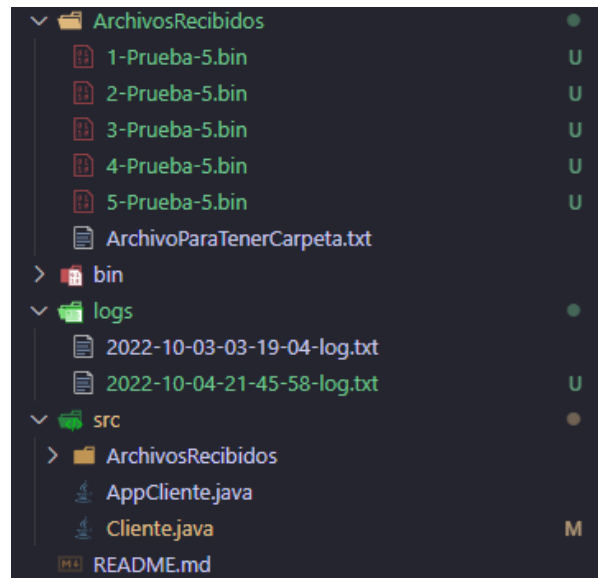
Durante todo el proceso de comunicación la aplicación guarda un log en la carpeta de logs con información detallada sobre la ejecución y los cálculos de velocidad.

El archivo 250 megabytes no se pudo agregar en el repositorio sin embargo está subido en un drive junto con las capturas de wireshark del análisis.

3. Aplicación de cliente

Ésta se realizó con java y su funcionamiento es similar al de la del servidor, esta recibe la cantidad de clientes que se van a ejecutar y concurrentemente los crea para conectarse al servidor por medio de la ruta y el puerto indicado anteriormente, una vez los clientes se conectan al servidor mandan una solicitud de petición de archivo para que el servidor les conteste, luego piden el tamaño del archivo y con ese tamaño crean un buffer para recibir el archivo. Con esto preparado cada cliente le indica al servidor que está listo para recibir y esperan la transferencia del archivo, luego recibirlo le solicitará a el servidor el hash generado para ser comparado con un hash generado localmente y en caso de que este sea incorrecto informárselo al servidor para que este reenvíe el archivo, si no se confirma el envío correcto y se termina la conexión.

Al igual que el servidor, esta aplicación guarda logs de las conexiones y guarda los archivos recibidos en la carpeta archivos recibidos en la carpeta ArchivosRecibidos.



4. Análisis de desempeño del servicio:

Prueba	Transferencia Exitosa	Bytes recibidos x cada cliente	Tiempo transferencia x cada cliente (sg)	Taza transferencia a cada cliente(MB/sg)	Puerto usado por el cliente	Bytes transmitidos por el servidor al cliente	Tiempo transferencia x cada cliente desde el servidor(sg)	Puerto de conexión a clientes (servidor)
1-100	Si	100000000	1	97.65	52269	100000000	1.004	5555
5-100	Si	100000000	1.8	58.5936	50277	100000000	2.2286	5555
10-100	Si	100000000	3.4	31.1917	59031	100000000	3.7623	5555
1-250	Si	250000000	2	122.07	62959	250000000	2.854	5555
5-250	Si	250000000	5.6	43.945	58822	250000000	6.0964	5555
10-250	Si	250000000	7.8	31.5343	53362	250000000	8.5132	5555

3)

- * Al solo tener 1 cliente, hay una transmisión de archivos muy rápida, por lo que tenemos el mejor tiempo de transferencia desde el cliente y el servidor.
- * Aumentando el número de clientes, la maquina empieza a tardar más, haciendo que la taza de transferencia se reduzca a poco más de la mitad. Sin embargo, los tiempos siguen siendo bastante ágiles.

* Al llegar al máximo número de clientes solicitados para el laboratorio. Se ve como la tasa llega a su punto más bajo en las pruebas del archivo de 100 MB. Sin embargo, los tiempos siguen siendo muy cercanos a los de las pruebas anteriores, por lo que se concluye un buen desempeño.

* Al comparar la tasa ahora con una prueba de 250 MB, se ve como la tasa es mejor que la de la prueba anterior. Sin embargo, el tiempo de transmisión es mayor que las primeras 2 pruebas.

* Para la prueba con 5 usuarios, hubo una diferencia importante entre esta prueba y la de los 100 MB, pues tarda cerca de 3 veces más, además de que su tasa de transferencia bajo con respecto a lo anterior.

* Finalmente, la última prueba muestra una tasa de transferencia mejor que la de la prueba de 100 MB, sin embargo, su tiempo es el peor de todas las pruebas siendo más del doble de la anterior y teniendo una diferencia importante entre la transmisión y recepción.

4)

Podemos ver todas las pruebas que el comportamiento para 5 usuarios suele ser el más mediocre, pues sus tiempos, a pesar de no ser tan altos como los de 10 usuarios, tienen una tasa de transferencia muy baja. Sin embargo, la tasa de los 10 usuarios a 250 MB tiene una tasa mejor que la misma a 100 MB aunque sus tiempos sean considerablemente menores

