Universidad Técnica Federico Santa María
Sistemas Distribuidos
Segundo Semestre 2020

Ayudantía Laboratorio 2
Recopilación de preguntas y respuestas

Profesora: Erika Rosas
Ayudantes de Tarea: Sebastián Alvarado **sebastian.alvaradoa@sansano.usm.cl**
Sebastián Ávalos **sebastian.avalosh@sansano.usm.cl**
Franco Zalavari **franco.zalavari@sansano.usm.cl**

16 de noviembre de 2020

Aclaracion (Enunciado)

- Cuando se dice que el cliente Uploader envia los chunks a un DataNode particular, no necesariamente va a ser el mismo siempre, la eleccion debe ser aleatoria, puesto que si siempre es el mismo nodo, nunca va a existir condicion de carrera en el log del NameNode
- Para el algoritmo centralizado, si el NameNode rechaza la propuesta, es este quien genera otra propuesta

A considerar

- Las respuestas están modificadas para dar información más precisa.
- Preguntas repetidas (considerando ambas ayudantías) serán omitidas en este documento, así como también preguntas que no tengan relación directa con el laboratorio.
- Cuando se habla de input, se refiere a una seleccion mediante un menu, como muchos hicieron para elegir el tipo de cliente en el Lab1

1. Ayudantía Miércoles 11 de Noviembre

¿Debemos crear una implementación (código) para cada tipo de cliente?
T: Minuto **3:15** del video
R: No necesariamente, puede ser una donde se tengan tanto el cliente uploader y downloader. Podrían seleccionarlo mediante un input, por ejemplo.

¿Cómo se hacía la comunicación?
T: Minuto **4:58** del video
R: La tecnología que tienen que utilizar los componentes para llevar a cabo la comunicación es gRPC.

¿En qué máquina deben implementarse los clientes?

T: Minuto **5:27** del video
R: Los clientes pueden estar distribuidos en las 4 máquinas, de la manera que quieran. Por ejemplo, pueden poner todos los clientes en la máquina del NameNode, o bien, un cliente en cada máquina. Deben registrarlo en su readme para saber en que maquina ejecutarlo.

¿Las implementación de los algoritmos de exclusión mutua deben hacerse en archivos distintos?
T: Minuto **8:04** del video
R: Se llegó a la conclusión de que lo ideal sería implementar ambos algoritmos en un mismo código, utilizando inputs para saber cuál se utilizará en la ejecución.

¿Cómo funciona la propuesta de distribución en la implementación centralizada?
T: Minuto **9:10** del video
R: El DataNode que recibe los chunks del archivo hará una propuesta de como se van a distribuir los chunks entre los DataNode, la cual debe ser aprobada por el NameNode. La idea es: Si tengo 3 chunks, entonces debo dejar 1 chunk en cada DataNode. Si hay más de 3, se deja 1 chunk en cada DataNode, y el resto puede ser distribuido aleatoriamente. Si hay menos de 3, se distribuyen en 2 DataNodes distintos.

¿En que caso se rechaza una propuesta de distribución de chunks?
T: Minuto **10:40** del video
R: Cuando se haya caído algún nodo básicamente. Por ejemplo, cuando evaluemos la tarea podemos apretar Ctrl+C para terminar la ejecución de un DataNode. En dicho caso, si la propuesta de distribución incluía el nodo caído, entonces debe ser rechazada y debe crearse una nueva que no lo considere.
Ademas de ello, ustedes pueden producir sus propios criterios para determinar la factibilidad de una propuesta, las cuales deben documentar en su readme para dejar constancia.

¿Los archivos son reconstruidos en los DataNodes?
T: Minuto **25:35** del video
R: No. Los Clientes Downloader rescatan la ubicación de los chunks desde el NameNode para luego pedir los chunks a los DataNodes que correspondan. Una vez reúna todos los chunks, el cliente procede a reconstruirlo.

¿Cómo se reconstruyen los archivos?
T: Minuto **28:00** del video
R: En el enunciado se dejo un link donde se explica el proceso de fragmentación y reconstrucción de archivos.

¿Si un chunk queda de menor tamaño, se rellena hasta alcanzar los 250kB?
T: Minuto **28:12** del video
R: No. El último chunk puede tener un tamaño menor a 250kB.

2. Ayudantía Jueves 12 de Noviembre

¿Cómo se decide quién accede al log del NameNode?
T: Minuto **3:45** del video
R: Pueden utilizar el ID del DataNode para tomar esta decisión. La idea es que asignen un id para cada DataNode, y el DataNode con el Id más grande tendrá prioridad. Por ejemplo, si un DataNode con ID=1 y un DataNode con ID=3 quieren acceder al log, entonces el DataNode con ID=3 tendrá prioridad de acceso al log.

¿Cómo funciona la descarga de archivos?
T: Minuto **5:00** del video
R: El cliente solicitará la ubicación de los chunks del archivo al NameNode. Una vez recuperada esta información, el cliente se comunicará directamente con cada uno de los DataNodes que contengan los chunks, los cuales descargará. Una vez tenga todos los chunks, el cliente procederá a reconstruir el archivo.

¿Como funciona la división de archivos en chunks?
T: Minuto **7:41** del video
R: La idea es que el archivo que se vaya a subir sea dividido en chunks de 250kB (ver enlace en enunciado) para ser subido al DataNode. Cuando el archivo sea descargado por un cliente, este debe juntar los chunks y reconstruir el archivo original.

¿Es necesario el Log?
T: Minuto **12:20** del video
R: Si. Cuando los DataNodes distribuyen los chunks, deben dejar registro de la ubicación de cada uno de estos, para que así el cliente pueda recuperarlos posteriormente para reconstruir el archivo.

¿Si se trata de recuperar un archivo que contenga algún chunk en una máquina caída, entonces se debe mostrar un error?
T: Minuto **27:39** del video
R: Efectivamente, deben imprimir en pantalla un mensaje de error indicando que no se ha podido recuperar el archivo exitosamente.

¿Cuál es la diferencia entre las dos implementaciones?
T: Minuto **28:40** del video
R: Básicamente la diferencia radica en cómo se aprueban las propuesta de distribución de los chunks y en cómo se controla el acceso al log (recurso compartido).