

Comparación entre los algoritmos de exclusión centralizados y distribuidos en el contexto de un repositorio de archivos distribuidos.

Andres Herrera Salgado -- 201473593-6

¿Qué se hizo y cómo se hizo?

En este experimento se implementó un sistema repositorio de archivos distribuido para el contexto de una biblioteca buscando permitir que sus usuarios puedan almacenar y descargar libros (dividiéndolos en trozos de 250 [kB] y repartiéndolos entre los *Datanode*) en el repositorio. Este sistema se compone de tres instancias que simulan el comportamiento de un *Datanode*, el que es encargado de almacenar trozos de los archivos subidos al repositorio, además de una instancia que simula el comportamiento de un *Namenode*, el que es el encargado de llevar un registro de los libros presentes en el sistema, en conjunto a la ubicación de los trozos de este. Toda la comunicación presente entre los componentes del sistema se realiza mediante el uso de *Protocol Buffers*.

Dentro del sistema existe una implementación para cada uno de los algoritmos de exclusión mutua, el centralizado (donde el *Namenode* se encarga de encontrar y verificar una distribución viable para los trozos de los archivos) y el distribuido (donde la tarea de encontrar y verificar una distribución viable para los trozos recae en *Datanode* que recibe los archivos desde el cliente). El sistema tiene dos funcionalidades principales, las cuales serán detalladas a continuación:

- **Subir Archivo (caso algoritmo centralizado):** Con los tres *Datanode* y el *Namenode* ya en ejecución, el cliente se conecta a uno de los *Datanode* para enviarle a él los trozos del archivo a subir. Al finalizar de recibir los trozos, el *Datanode* procede a generar la primera propuesta de distribución de los trozos del archivo, la que contempla que cada uno de los *Datanode* almacenará un tercio de los trozos de los archivos (con el resto de la división de los trozos, distribuidos de forma equitativa entre los nodos). Luego se procede a enviar la propuesta al *Namenode*, el que posteriormente le consultará a cada *Datanode* si ellos se encuentran disponibles para recibir el archivo. Los *Datanode* pueden presentar dos razones por las cuales estos no se encuentran disponibles, la primera corresponde a si el *Namenode* no recibe una respuesta dentro de 100 segundos y la segunda corresponde a un error aleatorio que presenta una probabilidad del 10% para aparecer. En caso de que uno de los *Datanode* no se encuentre o retorna error, el *Namenode* crea una nueva propuesta considerando solamente a los *Datanode* que se encuentran disponibles, para después realizar una consulta a los mismos *Datanode* pertenecientes a la propuesta.

Ya con una propuesta válida para la distribución de los trozos, el namenode la escribe en su archivo *log.txt* y retorna la propuesta al *Datanode* para que este se encargue de conectarse con cada uno de los *Datanode* presentes en la propuesta y entregarles los trozos que deben almacenar.

- **Subir Archivo (caso algoritmo distribuido):** Con los tres *Datanode* y el *Namenode* ya en ejecución, el cliente se conecta a uno de los *Datanode* para enviarle a él los trozos del archivo a subir. Al finalizar de recibir los trozos, el *Datanode* procede a generar la primera propuesta de distribución de los trozos del archivo, la que contempla que cada uno de los *Datanode* almacenará un tercio de los trozos de los archivos (con el resto de la división de los trozos, distribuidos de forma equitativa entre los nodos).
Con la propuesta generada, el *Datanode* procede a comunicarse con el resto de los *Datanode* para verificar si se encuentran disponibles para almacenar los trozos que les corresponden. Basado en las mismas razones usadas en el algoritmo anterior, cada *Datanode* responde si es que se encuentra disponible para recibir los trozos. Tomando en cuenta las respuestas de cada *Datanode* se crea una nueva propuesta válida, la cual es mandada al *Namenode* para que este la guarde en su archivo *log.txt*. Posteriormente, el *Datanode* reparte los trozos a los otros *Datanode* y almacena en disco los trozos que les corresponden basado en la propuesta válida.
- **Descargar Archivo:** Esta funcionalidad es igual para ambos tipos de algoritmos. El cliente le hace una petición al *Namenode* para que este le envíe los nombres de los libros presentes, en conjunto a la locación de sus trozos, en su registro en *log.txt*. Basado en esta lista, el cliente procede a descargar todos los trozos necesarios desde cada una de los *Datanode* para poder generar el archivo. En caso de que falle la comunicación con uno de ellos, se cancela la descarga de los trozos. Con los trozos descargados, se procede a unirlos para poder generar el archivo buscado.