

# Informe Laboratorio 2 Sistemas Distribuidos

Coordinación de componentes de un sistema distribuido mediante algoritmos con exclusión mutua centralizada/distribuida

**Gianni Stefano Carlini Blanco**

**201773105-2**

**02-12-2020**

## **1.- Qué se hizo y Cómo se hizo**

El sistema en general cuenta con 3 DataNode y 1 NameNode en los cuales se debe elegir el modo de funcionamiento (Centralizado o Distribuido) y a los DataNode se les debe informar si son o no el coordinador elegido. Por parte del Cliente este cuenta también con los 2 modos y debes elegir cual es el DataNode coordinador, dentro de sus funciones esta subir y bajar archivos

**1a.- Centralizado:** En el modo centralizado primero contamos con el modo subida, por parte del cliente este realiza la división del archivo en chunks de 256Kb y se establece una conexión mediante GRPC con el nodo coordinador, con la conexión ya establecida se envían los bytes, el nombre del archivo, el numero de partes y el nombre del libro. Por la otra parte en el nodo coordinador este recibe los parámetros y avisa que recibió, guarda los nombre en un arreglo y los bytes en otro arreglo y cuando llegan todas las partes genera la propuesta inicial, habiendo generado esta propuesta establece una comunicación GRPC con el NameNode para enviar esta propuesta, esta propuesta llega al NameNode junto con el nombre y la cantidad de partes del libro, en centralizado primero escribo el nombre y la cantidad de partes en el archivo log y paso a ver si la propuesta es aceptada, hay un 50% de que esta propuesta sea aceptada, si se acepta paso a escribir esta propuesta en el log y le envié la distribución al coordinador en caso contrario (rechazo de la propuesta inicial) entro a un loop donde genero una propuesta nueva y hay un 90% de que esta sea aceptada donde al aceptarla se hace lo mismo escribo en log y respondo. Ahora en el nodo coordinador teniendo la distribución en mano genero los 6 arreglos(3 de bytes y 3 de nombre) correspondientes a cada uno de los nodos, parto creando los archivos que necesito guardar en este Nodo, después establezco conexión con el nodo 2 y le envié la distribución (nombre y bites) donde este los recibe, crea sus archivos y responde con un "ok" para asegurar que esta correcto el proceso, repite esto con el nodo 3 y termina el proceso de subida.

Ahora para el proceso de descarga de archivos inicia desde el cliente haciendo una conexión GRPC con el NameNode para recibir el listado de libros disponibles la cual esta en un arreglo de NameNode, luego ingreso el nombre del libro que quiero y realizo otra conexión con el NameNode enviándole el nombre del libro para que me responda con los chunks que necesito, en el NameNode reviso la lista de data donde es un arreglo donde le primer espacio es el nombre del libro y el resto son strings con "nombre,ip" del libro, busco el que corresponda al nombre enviado y envié el arreglo de ubicaciones, de vuelta en el cliente desempaqueto este arreglo saltando el espacio 0 que es el nombre, hago un Split para separa nombre e ip y reviso por puertos, si el puerto corresponde al DataNode1 establezco una conexión GRPC con este y envié el nombre del archivo que necesito, en el DataNode todo este nombre de archivo y lo utilizo para abrirlo y

pasarlo a bytes para enviar esto de vuelta a cliente, teniendo de vuelta ya estos bytes creo el archivo y repito el proceso las veces necesarias y repito con el DataNode 2 y 3, ya con todos los archivos en poder del cliente se reconstruye el pdf del libro y se guarda en descargas.

**1b.-Distribuido:** Al igual que el modo centralizado cuenta con dos modos, primero el modo subida, aquí el cliente realiza una conexión con el nodo coordinador divide el archivo en chunks y se los envía, dentro del nodo coordinador genero 2 arreglos para los nombre y bytes de cada chunk, cuando llegan todas las partes genero una propuesta inicial y realizo 2 conexiones GRPC con los nodos restantes y espero las respuestas donde esta propuesta es recibida y hay un porcentaje (50%) de ser aceptada en el nodo que recibió la propuesta reviso primero la flag para saber si es una propuesta inicial o no, después reviso si es inicial y está ya fue aceptada anteriormente (ósea la propuesta inicial ya fue aceptada pero necesitaba un caso para identificar cuando crear los archivos) creo los archivos que me corresponden, ahora de vuelta al nodo coordinador reviso si ambas son aceptada es que la propuesta inicial fue aceptada por lo cual procedo envié las partes correspondientes a ambos nodos estos , creo mis archivos y realizo una conexión GRPC con el NameNode para infórmale como quedo la distribución, en caso de que la propuesta inicial fue rechazada por alguno de los nodos procedo a enviar una segunda propuesta con el flag = 2 donde los nodos que reciben esta aceptan la propuesta e informan al coordinador. Por parte del NameNode tengo una función en subida que es recibir las distribuciones donde recibe el nombre y el largo el cual lo escribe en le log y mediante un for a una lista con las distribuciones voy escribiendo el nombre de la parte y la ip donde se encuentra, en caso de conflicto con la escritura usamos ricart y agrawala que se programo de la siguiente manera cuando tu inicias una conexión para escribir el log pasas a estar en un estado true simbolizando que estas ocupando el log con lo cual te conectas con los otros nodos y le preguntas su estado, si alguno de estos está ocupando el log state = true te quedas en un loop esperando a que lo desocupen, al momento de recibir la respuesta por parte del log cambias tu estado a false lo que simboliza que no estas ocupando más el log.

En el caso de descarga el cliente se conecta con el nameNode para pedir la lista de libros disponibles mediante GRPC donde se envía una petición o string para que el NameNode responda con la lista, luego ingresa el nombre y se conecta con el NameNode nuevamente para solicitarle las ubicaciones del archivo enviándole el nombre, acá el nameNode busca en un array de array libros (donde el primer lugar es el nombre y el resto de espacios son las ubicaciones) y se lo envía a cliente para que sepa dónde conectarse, ya con el array en cliente divido por casos, omito el primer valor ya que es el nombre del libro, después separo el nombre del archivo y la ip y filtro por ips, genero las conexiones GRPC correspondientes, pongamos el caso de conexión con data 1, filtre la ip de data 1 con lo cual me conecto con este, le envié el nombre del archivo y espero el chunk, datanode lo que hace es recibir este nombre abrir el archivo con ese nombre y enviar el arreglo de bytes correspondiente, se repite el proceso con todos los chunks que necesite, ahora el ultimo paso con todos los chunks en mano es reconstruir el archivo y generar el .pdf el cual lo guarda en la carpeta descargas.

## 2.- Resultados

	Centralizado	Distribuido
Cliente Subida	1.2493042 s	1.3905016 s

Data Subida	1.6388026 s	1.6735706 s
Total	2.8881068 s	3.0640722 s

	Centralizado	Distribuido
Cliente Bajada	1.683681s	1.6278481s
Total	1.683681s	1.6278481s

Cantidad Mensajes	Centralizado	Distribuido
Subida	20	28
Bajada	11	11

Se utilizó de referencia un libro con 7 partes con lo cual son 7 mensajes de envío de chunks y 7 de respuestas. Para los tiempos se realizó un time stamp en el inicio y final de cada función involucrada y se tomó la diferencia de estos. Para el caso de los mensajes se mapeó las funciones por cantidad de conexiones GRPC que tenían cada método sabiendo que por cada conexión hay un request y un reply.

### 3.- Análisis

Dado los resultados anteriormente presentados es consecuente decir que a primera vista el tiempo y la cantidad de mensajes varía a consecuencia del tamaño del archivo de origen, ahora la cantidad de mensajes en el sistema distribuido es mayor y su demora no es mucho mayor a el sistema centralizado. Los tiempos de descarga son iguales lo que guarda sentido ya que solo necesito hacer la misma consulta al log y los data nodes para la recuperación.

### 4.- Discusión

Los resultados obtenidos guardan concordancia con lo esperado, en el sistema distribuido se esperaba una mayor cantidad de mensajes ya que al implementar el algoritmo de Ricart-Agrawala se espera un mínimo de 4 mensajes más (2 de ida y 2 de vuelta por nodo) para la comprobación del ingreso a la sección crítica, en cuanto a los tiempos se esperaba un menor tiempo en el sistema centralizado lo que guardo lógica con los resultados y ya que al tener una implementación de menor complejidad se necesitaban una menor cantidad de mensajes necesarios por cada acceso.

### 5.- Conclusión

Como conclusión queda visto que cada sistema tiene sus ventajas y desventajas, por parte del centralizado es un poco más rápido en el sistema probado y es más simple de implementar, cuenta con una menor cantidad de mensajes al tener solo un coordinador, pero esto mismo nos supone una gran falla dado a que si se produce algún problema en este es crítico ya que solo este se encarga de enviar los datos al NameNode, en cambio su contraparte el sistema distribuido es un poco más lento cosa que en sistemas mayores se puede ver con mejor claridad pero garantiza lidiar con problemas de inanición teniendo  $n$  nodos para responder, pero a su vez  $n$  puntos de fallo lo cual en sistemas de muchos nodos aumentaría los tiempos, cosa que igualmente se puede resolver con mayor lógica por detrás.