



Exclusión mutua centralizada y distribuida

Laboratorio 2

Introducción:

La comunicación es primordial en los sistemas distribuidos, a medida que estos se vuelven más complejos la coordinación va tomando más relevancia, los sistemas además de comunicarse deben poder trabajar en conjunto. La tarea de este laboratorio consiste en la distribución de libros a través de nodos, en donde existen clientes de tipo Uploader, quienes entregan los libros a un nodo para su distribución, y clientes de tipo Downloader, quienes pueden descargar los libros que están distribuidos, el registro donde están almacenados los chunks de los libros se mantiene en un nodo de tipo "Name Node".

Qué se hizo y cómo:

Lo primero que hizo fue definir el comportamiento general del programa, para ello se pide por consola que tipo de algoritmo se va a utilizar (Distribuido/Centralizado). Después se pide por consola el tipo de cliente, ya sea Uploader o Downloader. Si el cliente es Uploader se le pide el nombre del libro que va a subir a la plataforma. Luego se asume que tenemos los 3 Data Node disponibles y se selecciona de forma aleatoria uno de ellos. Se divide el archivo en chunks de tamaño 250 kB y mediante gRPC se pasan al Data Node. Luego este tiene dos caminos disponibles:

- Si al principio se definió el comportamiento general del programa como "Distribuido", entonces el Data Node lo que hace es generar una propuesta asumiendo en principio que los 3 Data Node están disponibles, si la cantidad de partes del archivo es mayor o igual que 3 entonces este plantea en la propuesta que los 3 primeros chunks sean almacenados uno en cada Data Node, el resto se almacena de forma aleatoria. Si la cantidad de partes del archivo es menor o igual a 2, entonces el Data Node plantea en la propuesta que el se quede con 1 chunk y el otro sea repartido de forma aleatoria entre los candidatos posibles. Luego de tener una propuesta consolidada, se intenta contactar con cada participante de esta propuesta, si uno de ellos no responde entonces lo saca de los candidatos posibles y genera nuevamente otra propuesta hasta converger. Finalmente cuando la propuesta es aceptada, esta se registra en el LOG del Name Node y el Data Node distribuye los chunks acorde a la propuesta. Cabe destacar



INF 343 - Sistemas Distribuidos
Profesora: Erika Rosas

que en este caso no se resolvió de ninguna manera el conflicto que se produce si más de un Data Node intenta escribir en el registro del Name Node.

- Si al principio se definió el comportamiento general del programa como “Centralizado” entonces los primeros pasos son iguales al anterior, cuando el Data Node genera la primera propuesta, esta se le manda al Name Node para que se encargue de contactar a los participantes de la propuesta, si alguno no responde entonces el Name Node genera una nueva propuesta hasta converger. Finalmente cuando la propuesta es aceptada, esta se registra en el LOG del Name Node y se le informa al Data Node para que distribuya los chunks acorde a la propuesta. Al igual que el caso anterior no se resolvió de ninguna manera el conflicto que se produce si más de un Data Node intenta escribir en el registro del Name Node.

Si el cliente es Downloader, este puede pedir al Name Node el listado de libros disponibles y descargar cualquier libro que se encuentre en ese listado, para ello se consulta al Name Node la ubicación de los chunks que componen el libro mediante Protocol Buffers, después se guardan los chunks que necesita y en que Data Node se encuentran en dos slices de GO. Posteriormente se recorren los slices y se contacta a cada uno de ellos para pedir el chunk correspondiente, estos se guardan en la carpeta “chunks” y finalmente se reconstruye el libro a partir de estos chunks. El nombre del archivo descargado tiene la siguiente estructura: “Descargado_de_Vuamos_nombre-libro.pdf”.

Resultados:

Para medir el tiempo se utiliza un Timestamp desde que el cliente sube un libro hasta que se escribe en el Log y los mensajes corresponden a cada conexión que realiza el Data Node que recibe estos chunks (Por simplicidad, se considera que Data Node 1 va a ser el que recibe los chunks del cliente).

Para el desarrollo del experimento se considera el siguiente libro, el cual se encuentra en la carpeta Libros:

- “La_dama_de_Monsoreau-Alejandro_Dumas_Padre.pdf”



INF 343 - Sistemas Distribuidos
Profesora: Erika Rosas

Este experimento cuenta con 4 test de prueba, los que se detallan a continuación:

- 1) Los 3 Data Node disponibles
- 2) Solo Data Node 1 y 2 disponibles
- 3) Solo Data Node 1 y 3 disponibles
- 4) Solo Data Node 1 disponible

Cada vez que se intenta contactar a un Data Node se suma uno a la cantidad de mensajes, al escribir en el LOG también se suma uno, y en el caso del algoritmo “Centralizado” se contabiliza de la misma forma pero considerando que se necesitan dos mensajes extra, uno para pasar la propuesta inicial al Name Node y otro para recibir la propuesta de distribución.

Los resultados obtenidos se muestran a continuación:

Test	Algoritmo	Número de mensajes enviados	Tiempo que demora en escribir en el log
1	Distribuido	15	6.29 [s]
	Centralizado	14	5.25 [s]
2	Distribuido	15	6.66 [s]
	Centralizado	14	6.78 [s]
3	Distribuido	12	7.75 [s]
	Centralizado	13	7.88 [s]
4	Distribuido	3	8.39 [s]
	Centralizado	5	10.45 [s]



Tabla 1: Valores obtenidos.

Análisis y Discusión:

Los valores obtenidos se deben a la forma en que está implementado el programa, en donde debe existir comunicación entre los nodos para la distribución de chunks de un libro y para la posible descarga. El tiempo de demora se considera razonable pues no es excesivo y deja el tiempo para que la consulta al log esté actualizada. A pesar de no poseer gran variación un algoritmo respecto al otro esto se debe a que es un programa a pequeña escala, aún así es necesario destacar que el algoritmo distribuido resultó tener un menor tiempo de demora en promedio al escribir en el Log, aunque el tiempo de ejecución del algoritmo depende en gran parte del random que se hace, ya que si la propuesta incluye mucho al datanode 0 entonces no habrá un paso de mensajes, ya que este se encarga de distribuir los chunks. Para el caso del número de mensajes enviados es similar, pues varía harto según el random realizado, sin embargo ambos algoritmos presentan resultados similares.

Conclusión:

En general, es posible concluir que el algoritmo de exclusión mutua distribuido resulta tener un mejor desempeño en aquellos casos en que solo se encuentra disponible el Data Node 1 y en los demás casos hay que tener en consideración que cada algoritmo genera la propuesta de forma random, es por ello que la cantidad de mensajes depende mucho de este, dado que si existe mayor participación del Data Node 1 en la propuesta, entonces habrá una menor cantidad de mensajes. Además, las pruebas son muy pequeñas en comparación con la realidad, donde los archivos son mucho más grandes que los que tenemos y eventualmente los nodos se encuentran a una distancia no menor, que puede perjudicar levemente el desempeño del algoritmo.