

Análisis de algoritmos de exclusión mutua: Red de Bibliotecas

Esteban Barrios, rol 201773530-9, esteban.barrios@sansano.usm.cl
Francisco Riveros, rol 20173581-3, francisco.riveros@sansano.usm.cl

1. Qué se hizo y cómo se hizo

Para el presente informe se analizará el caso hipotético de una red de bibliotecas que se encuentra en la búsqueda de experimentar con nuevos métodos de distribución de sus existencias digitales a sus suscriptores que no han podido acceder a sus instalaciones producto de la pandemia actual. Para ello, se solicita realizar un sistema de distribución de libros (archivos) que garantice el acceso de estos a sus clientes. Y, para tener una mejor distribución de recursos y accesos más rápidos, los encargados de la biblioteca decidieron fragmentar sus libros y distribuirlos en un conjunto de nodos. De esta manera, el sistema tiene que lograr coordinar el acceso de los nodos a un log en el que se registran los libros disponibles del sistema junto con las referencias de donde se encuentra cada fragmento de estos. Todo esto considerando que, en ningún caso, más de un nodo puede acceder a este registro simultáneamente.

Entonces, para realizar dicho sistema, se toma en consideración lo siguiente:

- 2 diferentes algoritmos de exclusión mutua para la coordinación de nodos: el algoritmo distribuido y el algoritmo centralizado.
- 3 estructuras diferentes de nodos: Cliente, DataNode y NameNode.
- 4 máquinas disponibles: 3 DataNodes y 1 NameNode
- Los clientes se pueden encontrar en cualquiera de estas máquinas

El Cliente tiene la opción, en primer lugar, de subir libros al sistema, ya sea a través del algoritmo centralizado o distribuido. Para ello, se divide el libro en chunks de 250 [KB] exceptuando el último, que puede ser de menor tamaño. Luego,

se envía dichos chunks a un DataNode en particular (proceso aleatorio de entre los 3 DataNode disponibles) indicando el algoritmo con el cual se accederá a la parte crítica del NameNode (Log). En segundo lugar, tiene las opciones de bajar y de revisar los libros que ya se encuentren en el sistema. En este caso, se comunica directamente con el NameNode y se consulta por el registro, de esta manera si se desea bajar un libro se piden los chunks correspondientes a los DataNodes y se reesmacla el archivo original, o se lee el Log para saber los libros en él. Cabe destacar que en caso de que el DataNode accedido se encuentre apagado, se repetirá el proceso hasta encontrar uno activo.

Los DataNode son los encargados, por un lado, de almacenar los diferentes fragmentos de los libros subidos por los clientes en disco y, por otro lado, de realizar las propuestas de división de sus fragmentos hacia el NameNode a través del algoritmo indicado por el cliente. Para el algoritmo centralizado, el encargado de aprobar dicha propuesta y de permitir el acceso al log, es el NameNode, por lo que se realiza una comunicación directamente con él y en caso de aprobar se distribuye los chunks de la manera propuesta por el DataNode, mientras que, en caso negativo, el NameNode responde con una propuesta modificada para que se distribuya de dicha forma. Por otro lado, para el algoritmo distribuido los encargados de aprobar la propuesta y de permitir el acceso al log, corresponden a los mismos DataNodes. Para ello, cada DataNode que quiera acceder al NameNode envía una propuesta a los demás y a sí mismo, indicando cual es el tiempo (clock) en que se realizó. La propuesta se acepta automáticamente si el DataNode que la recibe se encuentra activo, pero además se consulta si dicho DataNode también quiere acceder al NameNode.

Por lo cual, si se genera un conflicto (más de uno quiere acceder) se compara los clocks de cada uno y se da el acceso al que sea menor. Para los casos que sean mayores, estos esperarán a los que entraron antes para utilizar al NameNode. Cabe destacar que para controlar el acceso de los clientes a los DataNodes, se realizó de igual manera un algoritmo centralizado, en donde si el DataNode se encuentra ocupado con un cliente, estos esperarán a que se desocupe para acceder a él y responde al orden de llegada de los clientes ("semáforo").

Por último, el NameNode es el encargado de registrar las propuestas aceptadas en un archivo en memoria llamado Log. Para el caso del algoritmo centralizado, este realiza de manera análoga el proceso de "semáforo" anterior pero para los mismos DataNodes, es decir, si el NameNode se encuentra en uso, los DataNodes esperarán a que se desocupe de acuerdo al orden de llegada. Por otro lado, para el algoritmo distribuido, simplemente da acceso a todos los que se logren conectar con él.

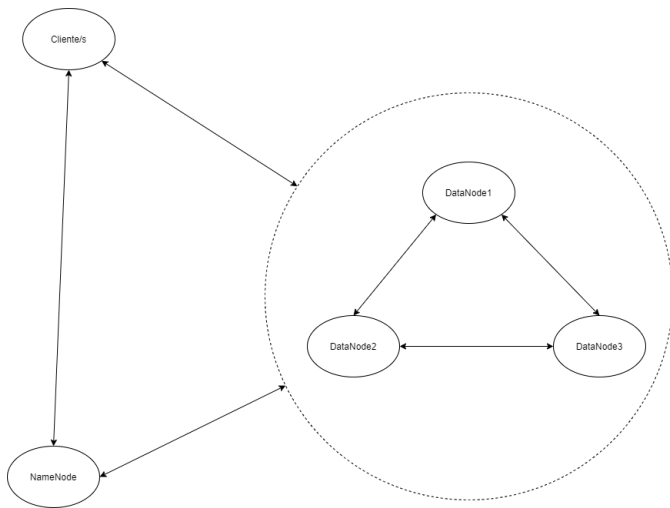


Figura 1: Diagrama de comunicación del sistema.

2. Resultados

Para analizar los algoritmos de exclusión mutua realizados en el sistema, se medirá el tiempo de subida del libro "Don Quijote de la Mancha" en cada caso, junto con el total de mensajes intercambiados durante el proceso. Notar que el subir un archivo conlleva a acceder al Log en el NameNode. También, se evaluará el impacto de realizar la subida al considerar: 3 DataNode activos, 2 DataNode activos y 1 DataNode activo. Y se repetirá el experimento 5 veces.

Entonces, para el caso del algoritmo centralizado se obtuvieron los siguientes resultados:

➤ 3 DataNodes activos

n°	Tiempo de subida [ms]	Mensajes intercambiados
1	130.59	17
2	129.23	17
3	117.25	17
4	152.10	17
5	162.40	17
Promedio	138.31	17

➤ 2 DataNodes activos

n°	Tiempo de subida [ms]	Mensajes intercambiados
1	40.51	14
2	75.62	14
3	103.31	14
4	57.88	14
5	81.88	14
Promedio	71.84	14

➤ 1 DataNode activo

n°	Tiempo de subida [ms]	Mensajes intercambiados
1	38.46	10
2	19.03	10
3	31.82	10
4	43.58	10
5	29.06	10
Promedio	32.39	10

Mientras que, para el algoritmo distribuido se obtuvo lo siguiente:

➤ 3 DataNodes activos

n°	Tiempo de subida [ms]	Mensajes intercambiados
1	152.88	19
2	256.33	19
3	111.46	19
4	124.07	19
5	170.15	19
Promedio	135.22	19

➤ 2 DataNodes activos

n°	Tiempo de subida [ms]	Mensajes intercambiados
1	128.48	16
2	140.09	16
3	94.13	16
4	116.66	16
5	124.81	16
Promedio	118.92	16

➤ 1 DataNode activo

n°	Tiempo de subida [ms]	Mensajes intercambiados
1	33.51	10
2	30.67	10
3	50.14	10
4	55.37	10
5	43.39	10
Promedio	42.61	10

3. Análisis

En cuanto a la subida de archivos con 3 DataNodes encendidos, se observó que el algoritmo de exclusión mutua posee un tiempo promedio inferior de envío por tan solo 3 milisegundos con un total de 135.22ms y 138.31ms para cada uno, en cuanto a la cantidad de envío de mensajes, el algoritmo centralizado posee una menor cantidad, con tan solo un total de 17 mensajes logró desarrollar el proceso completo en comparación a los 19 mensajes enviados por el algoritmos de exclusión mutua.

Por otro lado, en cuanto a 2 DataNodes encendidos, se tiene que el tiempo promedio de envio disminuye en ambos casos, pero esta vez es el algoritmo centralizado quien posee un tiempo inferior, con un total de 71.84ms en comparación a los 118.92ms del algoritmo de exclusión mutua, en cuanto al envío de mensajes el algoritmo centralizado también posee una menor cantidad, con tan solo 14 mensajes logró finalizar el proceso correctamente, en comparación a los 16 mensajes enviados utilizando el algoritmo de exclusión mutua.

Por último, en cuanto a 1 solo DataNode disponible, se observa que el promedio de tiempo de subida del algoritmo de exclusión mutua centralizado tiene un valor equivalente a 32,39ms siendo menor que el obtenido para el caso del algoritmo distribuido, siendo de 43,61 ms. Por lo que en este caso existe una diferencia en promedio 10,22ms en los cuales el algoritmo centralizado trabaja con mayor velocidad.

4. Discusión

En base a los resultados obtenidos, se observa que a medida que se va disminuyendo los DataNodes disponibles a utilizar por ambos algoritmos, es el algoritmo centralizado el que genera mejores resultados, puede deberse a que como está basado en el uso de un coordinador en este caso el NameNode, el cual se encarga de asignar las distribuciones respectivas según sea solicitado. Por lo que a mayor cantidad de nodos debe poseer una mejor coordinación y asignarle un tiempo de atención a cada uno para resolver sus solicitudes, siendo poco conveniente en un sistema con mayor cantidad de DataNodes.

En cambio el algoritmo distribuido, a medida de que aumenta la cantidad de nodos, le tomará más tiempo contactar con todos los nodos vecinos asignados al reparto, no obstante el crecimiento del tiempo promedio de envío es inferior al del algoritmo centralizado, esto se puede deber a que como quien resuelve el conflicto de la solicitud es entre los mismos DataNodes, no necesitan de un coordinador en conjunto para tomar la decisión de cómo asignar cada paquete, por lo que escala de una mejor manera al aumentar los recursos disponibles, ya que no existe el cuello de botella del NameNode para poder realizarla tarea, no obstante la cantidad de mensajes sigue siendo superior al algoritmo centralizado, dado que debe comunicarse con todos sus vecinos respectivos en búsqueda de conflictos, hacer el envío al NameNode para el registro en el Log y finalmente realizar el reparto de chunks.

5. Conclusión

Con todo lo anteriormente analizado, en esta experiencia se puede concluir que a partir de los diferentes recursos que presentan los sistemas distribuidos, un algoritmo tiene ventajas sobre el otro. Es decir, cuando se tiene sistemas de menor escala, el algoritmo centralizado tiene un mejor rendimiento en comparación al distribuido en cuanto a la latencia de la subida de archivos, sin embargo, presenta un solo punto de fallo y al tener que procesar todo en un solo punto, entonces en caso de tener un gran número

de solicitudes dirigidas a él, se puede generar deadlock en algunos procesos. Por otro lado, para casos de sistemas de mayor escala, se tiene que el algoritmo distribuido tiene una menor latencia en cuanto a subida de archivos que el algoritmo centralizado, aun cuando existe una mayor cantidad de mensajes. Esto se debe a que no existe un “cuello de botella” para acceder al log del NameNode. En otras palabras, simplemente lo utiliza.