



Rendimiento de un algoritmo centralizado y de un algoritmo distribuido para la subida y descarga de libros.

Camilo Farah E. , rol 201773534-1, camilo.farah@sansano.usm.cl
Christian Sepúlveda V. , rol 201791003-8, christian.sepulveda.13@sansano.usm.cl

1. Descripción del experimento

Para el siguiente experimento, se utilizarán cuatro máquinas virtuales para simular un sistema distribuido, las cuales actuarán como los siguientes actores: Cliente, tres DataNodes y un NameNode.

El experimento consiste en la distribución de libros digitales, mediante la subida y descarga de archivos. Para ello, se realizará la implementación del sistema de dos maneras: centralizado y distribuido. El objetivo de ello es obtener el rendimiento de ambas implementaciones y discutir las ventajas y desventajas de cada uno.

El **Cliente** se encarga de subir y descargar archivos, donde este último muestra los libros disponibles para descargar. Se describe cada proceso a continuación:

- Para subir archivos, en primer lugar divide el archivo en chunks de 250 kB y luego, los envía a un DataNode en específico. El DataNode elegido se hace por orden: si no funciona el primero, lo envía al segundo. Si no funciona el segundo, lo envía al tercero. En otro caso, no se puede enviar nada.

- Para descargar archivos, se comunica con el NameNode para que este le entregue la lista de libros, según la información registrada en el LOG y luego de elegido un archivo, obtiene la ubicación de los chunks que necesita para reconstruir el archivo. Luego de obtenidas las ubicaciones, se comunica con cada DataNode necesario y descarga los archivos, para finalmente reconstruirlo. Puesto que en el LOG las partes del archivo están ordenadas, el cliente pide cada chunk en ese mismo orden, por lo que después solo une las partes del archivo.

El **DataNode** se encargará principalmente de almacenar los chunks enviados por el cliente. A su vez, tendrá un comportamiento diferente dependiendo de la implementación del algoritmo.

- Para el algoritmo de exclusión mutua centralizada, el DataNode envía una propuesta hacia el NameNode, la cual será aceptada o rechazada. Cuando se tenga una propuesta

ya definida, el DataNode inicial distribuye los chunks en el resto de DataNodes.

- Para el algoritmo de exclusión mutua distribuida, el DataNode envía una propuesta hacia el resto de DataNodes, la cual será aceptada o rechazada. Si es rechazada, el DataNode inicial creará nuevas propuestas hasta que alguna sea aceptada. Luego de que se acepte una propuesta, se envía la información sobre la distribución al NameNode. Posterior a esto, se comienza la distribución de chunks en el resto de los DataNodes incluidos en la propuesta válida.

El **NameNode** se encargará de mantener y actualizar el LOG en el cual se registrarán los metadatos de los chunks de cada archivo. Además, mostrará un listado de libros disponibles hacia el cliente cuando éste lo requiera.

Adicionalmente y solo para el algoritmo de exclusión mutua centralizada, el NameNode se encargará de aceptar o rechazar propuestas por el DataNode y en el caso de rechazar, el NameNode creará propuestas para los DataNodes.

La estructura del LOG mencionado anteriormente es la siguiente:

```
Nombre_Libro_1 Cantidad_Part_1
parte_1_1 ip_maquina
...
parte_1_n ip_maquina
Nombre_Libro_2 Cantidad_Part_2
parte_2_1 ip_maquina
...
parte_2_n
...
```

2. Resultados

Para evaluar el rendimiento de los algoritmos, se subió el libro “Frankenstein” y se midió el tiempo de subida del archivo. El libro tiene un peso de 1.023.707 bytes, por lo que, según las reglas especificadas, el archivo se dividió en 5 partes (chunks).



Primero se comenzó usando el **algoritmo de exclusión mutua centralizado** para subir el archivo utilizando tres DataNodes activos, repitiendo el proceso seis veces. Los resultados son los siguientes:

Intento	Tiempo de subida (Centralizado, 3 nodos) [ms]
1	88.93
2	60.83
3	72.55
4	68.56
5	62.35
6	84.99
Promedio	73.04

Luego, se repitió el experimento con el mismo algoritmo, pero con dos nodos activos.

Intento	Tiempo de subida (Centralizado, 2 nodos) [ms]
1	65.14
2	96.90
3	53.66
4	63.10
5	47.80
6	47.81
Promedio	62.40

Por último, se realizó el mismo experimento con un solo DataNode activo:

Intento	Tiempo de subida (Centralizado, 1 nodo) [ms]
1	63.12
2	40.29
3	37.64
4	41.64
5	38.13
6	38.30
Promedio	43.19

Luego, se subió el mismo libro, pero usando el **algoritmo de exclusión mutua distribuido** la misma cantidad de veces y variando la cantidad de DataNodes activos.

Los resultados obtenidos con 3 nodos activos en el experimento son los siguientes:

Intento	Tiempo de subida (Distribuido, 3 nodos) [ms]
1	65.26
2	57.05
3	74.59
4	49.29
5	55.06
6	59.23
Promedio	60.08



Para 1 nodo = 6.55 %

Los resultados obtenidos con 2 DataNodes vienen a continuación:

Intento	Tiempo de subida (Distribuido, 2 nodos) [ms]
1	61.40
2	55.62
3	58.25
4	53.60
5	58.52
6	58.49
Promedio	57.65

Para terminar con el experimento, se presentan los resultados obtenidos al subir el archivo con un solo nodo:

Intento	Tiempo de subida (Distribuido, 2 nodos) [ms]
1	41.94
2	38.11
3	40.82
4	38.10
5	35.64
6	47.53
Promedio	40.36

3. Análisis

Observando los resultados, se puede notar claramente que el **algoritmo de exclusión mutua distribuido** presenta un mejor rendimiento que el centralizado. El porcentaje de mejora se presenta a continuación:

Para 3 nodos = 17.74 %

Para 2 nodos = 7.61 %

Además, se puede notar que a medida que la cantidad de nodos activos disminuye, la diferencia entre ambos algoritmos también disminuye.

4. Discusión

Como fue dicho anteriormente, el tiempo de subida para el **algoritmo de exclusión mutua distribuido** es mejor que el centralizado y disminuye su diferencia con este último a medida que disminuye la cantidad de DataNodes involucrados en la subida.

Esto puede deberse a que el **algoritmo de exclusión mutua centralizado** implica que el NameNode actúe como un intermediario, por lo que la comunicación de un DataNode con el resto de nodos requiere una comunicación previa con el NameNode, resultando en una mayor cantidad de mensajes y por tanto, también resulta en un mayor *delay*.

Por otro lado, el **algoritmo de exclusión mutua distribuido**, no necesita comunicación previa con el NameNode, es decir, se comunica directamente con los otros DataNodes, reduciendo el número de mensajes que el sistema necesita para coordinarse.

5. Conclusión

Podemos concluir que las ventajas del **algoritmo de exclusión mutua distribuido** son una menor latencia en la subida de archivos, además de que el rendimiento será mayor en comparación al **algoritmo de exclusión mutua centralizado** a medida que el sistema escale, o en otras palabras, el sistema presente una mayor cantidad de nodos involucrados en su arquitectura. De esta manera, se comprueba lo explicado en la teoría.