



Laboratorio 2: Algoritmos de exclusión mutua aplicados en un sistema distribuido de una simulación de biblioteca digital.

Macarena Hidalgo 201473608-8

David Medel 201573548-4

1. ¿Qué se hizo?

En este laboratorio se elaboró una red de bibliotecas para el proceso de distribución de sus libros de manera online, permitiendo a múltiples clientes subir y descargar archivos (libros) según corresponda.

Además de esto y con el fin de permitir una mejor distribución de libros en términos de calidad y rapidez se decidió fragmentar los libros en *chunks*, los cuales son repartidos entre un conjunto de nodos que hacen uso de algoritmos de exclusión mutua para la coordinación de acceso a un LOG que lleva registro de la distribución de los *chunks*.

2. ¿Cómo se hizo?

Para este sistema de distribución se implementaron 3 tipos de nodos principales en el sistema: Cliente (Uploader/ Downloader), Name node (Quien lleva el registro de los libros) y Data nodes (encargados de almacenar los chunks), además se implementaron 2 diferentes protobuffers uno para el envío de chunks y otro para la comunicación de propuestas mediante Protocol buffers.

Para la creación de los nodos mencionados se utilizó el lenguaje de programación **GO** corriendo los archivos en cuestión en 4 maquinas diferentes de forma que:

- **Maquina dist69:** Se ejecuta el archivo *Namenode.go*
- **Maquina dist70:** Se ejecuta el archivo *Datanode.go*
- **Maquina dist71:** Se ejecuta el archivo *Datanode.go*
- **Maquina dist72:** Se ejecuta el archivo *Datanode.go*

Mientras que el archivo *Cliente.go* se puede ejecutar en cualquiera de las maquinas en cuestión.

Entre las consideraciones asociadas a la implementación de los algoritmos se tienen:

1. El *Data node* encargado de la distribución de los chunks es seleccionado de manera aleatoria al comienzo de la ejecución del SD, en caso de que la máquina del Data node elegido esté caída, el algoritmo seguirá buscando conectarse a alguna que esté operativa y disponible.
2. El *Data node* seleccionado divide cada libro en n chunks, los cuales se almacenen en el siguiente formato:



- '*nombre libro*' + '-' + '*idchunk*', donde '*idchunk*' va de 0 a $n - 1$.
 - Cada chunk es enviado para ser almacenado en el disco de cada maquina según la distribución señalada en el LOG (*Name node*), luego de ser aceptada o rechazada la propuesta.
3. La propuesta inicial generada por los data nodes tienen el siguiente formato:
- [*maquina1, maquina2, maquina3, maquina1, ...*], donde el tamaño de la lista es la cantidad de chunks del libro fragmentado y el indice de la lista donde se encuentra la máquina corresponde al número de chunk. Para facilitar la comunicación entre nodos la propuesta es enviada al *Name node* en forma de string, es decir: *maquina1, maquina2, maquina3, maquina1, ...*
4. Una vez generada la propuesta dependiendo el tipo de exclusión mutua seleccionado por el cliente ocurre lo siguiente:
- a) **Centralizada:** El rechazo de una propuesta por parte del Name node se produce cuando una de las maquinas consideradas en la propuesta se encuentra caída, en este caso el Name node eliminará dicha maquina de la propuesta inicial y en su lugar introducirá una de las otras maquinas que esten operando correctamente de manera aleatoria.
 - b) **Distribuida:**
5. Cuando más de un Data node quiere enviar propuesta al Name node, se crea una variable booleana "*ocupado*" que indica que si este se encuentra ocupado procesando otra propuesta de distribución, así los otros Data nodes quedarán esperando hasta que el Name node vuelva a estar disponible.
6. Para descargar un libro lo primero que hace el cliente es solicitar un listado de los libros disponibles en la biblioteca al *Namenode* el cual lee su *LOG.txt* y envía el listado. En el cliente se selecciona el libro que se quiere descargar y el *Namenode* envía la ubicación, es decir, en que Data nodes se encuentran los chunks correspondientes.

3. Resultados

A continuación se presenta una tabla comparativa entre los algoritmos de exclusión mutua, donde se muestra según los experimentos realizados los tiempos de ejecución y cantidad de mensajes intercambiados promedio obtenidos, los cuales fueron tomados definiendo variables dentro de los códigos, que luego fueron mostradas por consola.

Para la realización de este experimento se subió el mismo libro para ambos algoritmos (en este caso al no tener el algoritmo distribuido no se realiza esta parte del experimento con dicho algoritmo): *Aventuras_de_tres_rusos_y_tres_ingleses_en_el_Africa_austral* Verne_Julio.pdf de modo que la cantidad de chunks a distribuir fuera la misma en ambos casos.

- **Tabla 1 :** comparación de tiempos de ejecución y cantidad de mensajes intercambiados para los algoritmos de exclusión mutua centralizado y distribuido.



	Tiempo de ejecución [ms]	Cantidad de mensajes intercambiados
Centralizado	263.320123	14
Distribuido	-	-

4. Análisis

Según los resultados obtenidos en el proceso de experimentación, se observa que para el **algoritmo de exclusión mutua centralizado**, los tiempos según la literatura son mayores a los tiempos que se tendrían en el **algoritmo de exclusión mutua distribuido**, esto se debe a que en el **centralizado** al haber solo un nodo que permite aceptar las propuestas de distribución (*Name node*), se genera una demora o retraso en los tiempos de respuesta, dado que el nodo se siente sobrecargado o debido a algún eventual cuello de botella generado al recibir demasiadas propuestas a la vez. Al contrario de esto en el **algoritmo de exclusión mutua distribuido** no se interactúa con el Name node para procesar la propuesta, ya que son los mismos Data nodes quienes la generan y definen la distribución que tendrán los chunks.

Ahora en relación a la cantidad de mensajes intercambiados se espera que el **algoritmo de exclusión mutua distribuido** genere una mayor cantidad de intercambios de mensajes, ya que cuando un Data node quiere enviar una la propuesta inicial, esta se envía a todos los otros Data nodes, incluyéndose a si mismo, además de esto se espera que se generen mensajes en respuesta desde todos los Data nodes a este nodo, todo esto a diferencia de lo que sucede con el **algoritmo de exclusión mutua centralizados**, donde existe solo una interacción de intercambio de propuesta que se genera entre el Data node que propone y el Name node que acepta o rechaza.

5. Discusión

Según lo expuesto en la sección anterior y tomado en cuenta que el algoritmo distribuido no se alcanzó a implementar, podemos decir que basándonos en diversos artículos y lo expuesto en la literatura relacionada a los algoritmos de exclusión mutua, efectivamente existe una concordancia entre las siguientes relaciones:

- **Tiempos de ejecución:** *distribuidos < centralizados*
- **Mensajes intercambiados:** *distribuidos > centralizados*

Donde cada algoritmo posee sus pro y contras, y donde la conveniencia de cada uno depende de lo que se desee lograr, es decir si se requiere un sistema que sea rápido en el proceso de distribución, la implementación del sistema se debe efectuar con un algoritmo de exclusión mutua distribuido, pero en cambio si se desea un sistema en el que el intercambio de mensajes sea mínimo, entonces el sistema se debe implementar con un algoritmo de exclusión mutua centralizado.

En relación al caso expuesto donde se pide que el sistema sea eficaz en términos de calidad y rapidez, se observa que el algoritmo que más se ajusta a estos requerimientos es el distribuido, ya que cumple con ser más rápido, todo esto a pesar de que el número de mensajes que intercambie sea mayor y que posea poca tolerancia a fallos, ya que al fallar cualquier nodo en el proceso de distribución de la propuesta se impide que el algoritmo pueda continuar.



6. Conclusión

Tomando en cuenta lo discutido en el ítem anterior, se considera que según los requerimientos expuestos del sistema (eficaz en términos de calidad y rapidez) el algoritmo que más se ajusta a estos sería el distribuido, ya que en comparación al centralizado debiese ser más rápido.

En relación a las desventajas que posee el algoritmo distribuido también se puede ver que estas son menos influyentes que las desventajas que posee el algoritmo centralizado, el cual depende netamente del Name node, lo cual lo vuelve muy propenso a fallas debido a que si llegase a caerse este nodo el sistema completo dejaría de funcionar.