Si vas para Chile: Análisis Histórico

Fabio Pazos A., rol 201773403-1, fabio.pazos@sansano.usm.cl
Sebastián Rojas G., rol 201773598-8, sebastian.rojasgue@sansano.usm.cl

10 de noviembre de 2020

Introducción

En el presente documento se describirá el razonamiento utilizado para resolver el problema planteado y la implementación utilizada. Se presentarán resultados de ejecución, los cuales serán analizados para posteriormente ser discutidos y finalmente, concluir qué algoritmo es mejor.

Contexto – La tarea consiste en desarrollar un sistema distribuido, el cual simule una biblioteca virtual. Dicho sistema está formado por cuatro nodos, de los cuales tres (DataNodes) almacenan partes de los libros (llamados **chunks)** en lugar de almacenar el libro entero. Cada chunk tiene un largo fijo de 250 [kB]. El cuarto nodo (NameNode) almacena un archivo llamado LOG, el cual contiene la información de cómo están distribuidos los chunks de los libros entre los nodos.

Desarrollo – Para la tarea se utilizaron cuatro máquinas virtuales entregadas por el departamento de informática de la universidad. El sistema debía utilizar tres máquinas como DataNodes y una como NameNode. El cliente encargado de realizar las solicitudes puede ejecutarse desde cualquier máquina virtual. A continuación se detallada el **qué hicimos** para cada nodo.

Cliente: El cliente es el encargado de realizar las solicitudes al sistema distribuido: puede cargar libros (distribuyendo los chunks en las máquinas de la propuesta utilizada) o puede descargar libros (utilizando la información de distribución localizada en el NameNode para ensamblar de manera efectiva el libro).

Data Node: Los data nodes son ejecutados como servidor, es decir, en su función *main* solamente se dedican a escuchar las solicitudes del cliente. Éste se encarga de ensamblar y desensamblar los chunks de los libros solicitados o subidos (respectivamente) además de almacenarlos.

Para el caso Centralizado, el data node genera una propuesta de distribución, la cual contiene solamente a aquellos DataNodes que lograron ser comunicados. La propuesta generada no supera un largo máximo, el cual es entregado como parámetro. Además, los nodos que son partícipe de la propuesta son seleccionados de manera aleatoria. La propuesta es enviada utilizando **protocol buffers**[1] hacia el NameNode. Éste, utilizando una probabilidad, aprobará o rechazará la propuesta. En caso de rechazarla, genera una nueva propuesta, y la envía al DataNode. En cualquier caso, el DataNode envía la información completa de la propuesta al NameNode, el cual escribe en su LOG la distribución correspondiente. Finalmente, distribuye los chunks de cada libro al resto de los DataNodes de la propuesta, los cuales almacenarán cada parte dentro de la carpeta **/partes**.

Para el caso Distribuido, el DataNode genera una propuesta (utilizando el mismo método anterior), pero esta vez son el resto de los nodos los que la aceptan o rechazan en base a una probabilidad. Si la propuesta es rechazada por alguno de estos nodos mientras se les pregunta, entonces se genera automáticamente una nueva propuesta, reduciendo la cantidad de nodos posibles en ésta. Si la propuesta es aceptada por todos, entonces se distribuyen los chunks y se realiza la escritura en el NameNode con la información correspondiente.

En caso de que la propuesta sea rechazada suficientes veces como para forzar a que la propuesta solamente tenga un nodo posible, ésta es aceptada automáticamente.

Name Node: Es el encargado de almacenar en un archivo la distribución de las partes de los libros. El archivo se llama **LOG.txt** y sigue el siguiente formato:

[1] A partir de este punto, se asume que cuando un nodo envía información a otro, es mediante protocol buffers.

LOG.txt

```
Nombre_Libro_1 Cantidad_Partes_1
parte_1_1 ip_maquina
...
parte_1_n ip_maquina
Nombre_Libro_2 Cantidad_Partes_2
parte_2_1 ip_maquina
...
parte_2_n
...
```

El NameNode recibe la información de la distribución mediante un DataNode.

Para el caso Centralizado, el NameNode está encargado de aceptar o rechazar la propuesta recibida por el DataNode. Si la rechaza, generará una nueva propuesta, la cual será enviada al DataNode para su posterior utilización.

El DataNode se comunica con el cliente cuando éste desea ver los libros disponibles o descargar un libro. El mensaje enviado contiene cuatro arreglos: uno con todos los títulos de los libros, otro con la cantidad de partes en la que fue dividido cada título, otro con los subtítulos (parte_x_y, con x e y números enteros) y otro con las direcciones de las máquinas. Los dos primeros arreglos están relacionados de manera que si se accede a la posición *i* de ambos, se está rescatando la información del mismo libro. Lo mismo ocurre con los otros dos arreglos: la *i*-ésima posición relaciona la dirección de la máquina con su subtítulo correspondiente.

Resultados – A continuación se presentan los resultados obtenidos en 3 ejecuciones del sistema para los algoritmos distribuidos y centralizados, subiendo 5 libros: Crimen y Castigo de Dostoyevski Fiodor, Dracula de Stoker Bram, Edipo Rey de Sofocles, Frankenstein de Mary Shelley y Los Miserables de Hugo Victor, los cuales están enumerados del 1 al 5 respectivamente en las tablas presentadas a continuación. Además, para cada ejecución se utilizó la misma propuesta de distribución.

La tabla presentada a continuación corresponde a la cantidad de tiempo que tardó en subirse cada libro en cada ejecución para el **algoritmo de exclusión mutua centralizada**:

Libro	T1[ms]	T2[ms]	T3[ms]
1	141.18	57.67	133.76
2	103.06	124.91	36.54
3	34.43	27.78	33.99
4	70.72	73.94	56.34
5	48.12	55.48	109.54

La siguiente tabla presenta el tiempo que tardó en subirse cada libro en cada ejecución para el **algoritmo de exclusión mutua distribuida**:

Libro	T1[ms]	T2[ms]	T3[ms]
1	166.69	57.67	254.77
2	143.12	106.59	53.84
3	23.31	21.47	24.49
4	117.48	32.45	75.88
5	107.19	45.35	62.12

A continuación, se presenta el promedio del tiempo que tardó cada libro en ser subido para cada algoritmo:

Libro	Distribuido [ms]	Centralizado [ms]
1	159.10	110.87
2	101.18	88.17
3	23.09	32.07
4	75.27	67.00
5	71.55	71.05

Además, se presentan la cantidad de mensajes intercambió cada algoritmo para crear, aceptar y/o rechazar la propuesta de distribución:

Análisis y Discusión – A partir de la tabla de promedios, que el algoritmo centralizado en promedio tiene menores tiempos de ejecución que el distribuido a excepción del libro 3, el cual fue más veloz para el algoritmo distribuido. Esto puede deberse principalmente a que el algoritmo de exclusión mutua distribuida requiere preguntarle a cada nodo si es que acepta la propuesta, mientras que en el algoritmo centralizado, la propuesta es rechazada o aceptada solamente por el NameNode.

La teoría indica que el algoritmo de exclusión mutua distribuida es más rápido que el centralizado. Sin embargo, los resultados obtenidos empíricamente demuestran lo contrario. Esto puede deberse al tamaño del sistema distribuido: al ser un sistema a baja a escala (o con pocos nodos), el impacto de utilizar dicho algoritmo no alcanza a apreciarse.

Conclusiones – Se puede concluir que para sistemas distribuidos con pocos nodos, el algoritmo de exclusión mutua centralizada posee una menor latencia en comparación al algoritmo de exclusión mutua distribuida.