Laboratorio 2: Sistemas Distribuidos
Distribución vs Centralización

Universidad Técnica Federico Santa María
Campus Santiago San Joaquín
Departamento de Informática
Segundo Semestre 2020

Gonzalo Larraín 201673516-K
Daniel Toro 201673595-K

1. Qué se hizo y Cómo se hizo

En este informe se exploran las diferencias de rendimiento entre una implementación de un sistema de biblioteca online de forma centralizada y distribuida. Para evaluar estas diferencias se considero el tiempo que se demora en escribir en un archivo log y la cantidad de mensajes enviados.

A grandes rasgos el funcionamiento del sistema es el siguiente:

En primer lugar un cliente selecciona qué libro desea cargar a la biblioteca, luego los divide en chunks de 250[kB] y los envía a uno de los DataNode al azar. Este último crea una propuesta de distribución que debe ser aprobada. Con esta propuesta distribuye los chunks a los otros DataNodes y manda su ubicación al NameNode el cual la registra en un archivo log. Finalmente, un cliente consulta al NameNode los libros disponibles y rescata la ubicación de cada uno de los chunks del libro de interés y se los pide a los DataNode respectivos y reconstruye el libro original.
A continuación se explican las diferencias entre ambas versiones.

1.1. Lógica Centralizada

- DataNode: En esta versión la propuesta generada es enviada solamente al NameNode.
- NameNode: Aquí el NameNode se encarga de aprobar la propuesta de distribución. En caso que detecte que uno de los DataNodes no este disponible genera una nueva propuesta no considerándolo. Además, es el encargado de dar acceso de escritura al log por medio de una variable de estado. Cuando el data node quiere escribir ve el valor de la variable, si el valor es 0 quiere decir que esta libre y puede escribir en el log, en el caso contrario espera un tiempo y vuelve a preguntar por el estado de la variable hasta que pueda escribir.

1.2. Lógica Distribuida

- DataNode: En esta versión la propuesta generada es enviada a cada uno de los otro DataNodes, esperando que la acepten. En particular, para esta implementación, si aceptan o no la propuesta depende de una probabilidad, en donde se acepta con un 75 % de probabilidad. Si todos sus pares aceptan la propuesta este la envía al NameNode para ser escrita en el log, de no ser así genera propuestas hasta tener la aprobación de todos. En el caso que uno de los DataNodes no este disponible no lo considera para la nueva propuesta. Para el acceso de escritura al log pregunta a los otros nodos si están ocupando el recurso, sin embargo esto no se implementó.
- NameNode: En este implementación solo es encargado de escribir en el log la ubicación de los chunks

Para probar el rendimiento se hicieron los siguiente experimentos:

1. Cargar un libro con todos los DataNode activos
2. Cargar un libro con dos DataNode activos
3. Cargar un libro con un DataNode activo

El tiempo se calcula con la entrada del log del cliente cuando indica ”Tiempo de inicio” y la entrada del log del NameNode cuando indica ”Escribí en el log”. De la misma forma, para el número de mensajes se cuentan en los logs al cargar un libro de los DataNode, NameNode y cliente las entradas que dicen "Mensaje Enviado".

2. Resultados Obtenidos

2.1. Caso Centralizado

En el caso centralizado, utilizaremos distintos libros en cada prueba pruebas, utilizando los 3 Data nodes live, 2 Data nodes y solamente 1 Data node disponible. Los resultados de este experimento fueron (El tiempo de este experimento puede variar dependiendo de la latencia en las maquinas, por lo que nos apoyamos también de pruebas locales):

- 3 Data nodes: Menos de 1 segundo, 21 mensajes en total, Dracula.
- 2 Data nodes: 10 segundos, 17 mensajes en total, Frankestein.
- 1 Data node: 20 segundos, 8 mensajes en total, Peter Pan.

2.2. Caso Distribuido

En el caso distribuido, utilizamos los mismos libros en las pruebas anteriores para obtener resultados acordes. Al igual que el centralizado, utilizamos las posibles variaciones de maquinas apagadas y encendidas para ver los tiempos que toman. (Al igual que el centralizado, nos apoyamos en pruebas locales para omitir la intermitencia de las maquinas virtuales):

- 3 Data nodes: Menos de 1 segundo, 36 mensajes en total, Dracula.
- 2 Data nodes: 22 segundos, 17 mensajes en total, Frankestein.
- 1 Data node: 38 segundos, 8 mensajes en total, Peter pan.

3. Análisis

Podemos ver que cuando los 3 Data nodes están activados, los tiempos son bastante parecidos, pero esto puede cambiar debido al porcentaje de fallo que tiene implementado el sistema distribuido, lo cual incrementaría aun mas la diferencia de mensajes. Podemos decir que si no hay fallas ni rechazos en los sistemas, se parecen bastante en la cantidad de mensajes. Podemos ver que en general, el tiempo centralizado es aproximadamente 2 veces mas rápido para el envío de un solo libro, pero, como fue mencionado, esto puede cambiar debido al random implementado en la probabilidad de fallo.

Podemos analizar también el tamaño de la muestra, como tenemos una instancia de ejecución pequeña, por lo que los datos a mayor escala son distintos. Por ultimo, podemos decir que el algoritmo centralizado en esta instancia pequeña tiene mejor rendimiento general. El sistema distribuido pasa a ser mas seguro y fuerte en una instancia mayor o cantidades masivas de acciones.

4. Discusión

Los resultados obtenidos son esperados en cuanto a la implementación, respectivo a los tiempos y los mensajes. Si realizamos una cantidad grande de pruebas y de intentos, podremos ver que los mensajes y el tiempo en centralizado aumentara, debido a la probabilidad de fallo y el loop que se puede generar en esta interacción. Además, hay un tiempo que no esta presenciado en el experimento, y es el de la exclusión mutua. La forma básica deberá sumar tiempo en la ejecución centralizada, mientras que en la versión distribuida con la implementación del algoritmo de Ricart y Agrawala, con su respectivo reloj lógico, sumara en la cantidad de mensajes y el tiempo.

Podemos ver también el funcionamiento de los nodos en si. El algoritmo centralizado deja la carga de la propuesta a manos del Name Node, diciéndole a el que se encargue de conectarse a los otros Data Node para asegurarse de que estén operativos, mientras que en el distribuido, esta carga es netamente de los propios Data node, haciendo que el Name node sea solamente el que recibe los datos y la escritura.

Pese a esto, consideramos que el tiempo en el algoritmo distribuido es un poco alto, esto lo podemos atribuir a fallas de código y mala optimización de este mismo, al igual que la baja experiencia utilizando gRPC.

5. Conclusión

En conclusión, podemos ver mediante la evidencia que en una instancia menor y acotada, el algoritmo centralizado tiene un mejor rendimiento general que el algoritmo distribuido, sin embargo bajo teoría afirmamos que un sistema distribuido tiene distintos beneficios a largo plazo y en una instancia mas masiva del problema. Respecto a las instancias de caídas, el algoritmo centralizado tiene una forma "bruta" de solucionarlo, que es simplemente no utilizar aquellas maquinas que no respondan a tiempo, mientras que el distribuido se asegura de que las maquinas estén en "sintonia", que acepten la propuesta y estén operativas, por lo que saca una ventaja en ese ámbito.