Trabajo Práctico Especial 2

72.42 - “Programación de Objetos Distribuidos”

2023 - 1Q

Realizado por:
Cornidez, Milagros - 61432

1 de Junio de 2023

Diseño de componentes MapReduce

Query 1: Total de viajes iniciados por miembros por estación

En la primera query, se pide listar el nombre de la estación y la cantidad total de viajes iniciados en esa estación por usuarios miembros.

Para el Map, primero se verifica que la estación dada por el BikeRent esté en el mapa de estaciones que contiene las estaciones listadas en stations.csv. De esta manera sólo tenemos en cuenta las estaciones presentes en el archivo. Luego la función emite el nombre de la estación y el valor de is_member el cual es 0 en el caso de no ser miembro, y 1 si lo es.

El Reduce se encarga de sumar por cada estación el valor de is_member y devuelve la suma, dando así el resultado pedido por la query.

También se implementó un combiner que contiene la misma función que el Reduce y un Collator que se encarga de ordenar el resultado del reduce de mayor cantidad de miembros a menor cantidad y, en el caso de empate, por orden alfabético.

Query 2: Top N viajes más rápidos de cada estación de inicio

En la segunda query, se pide listar la estación de inicio del viaje, la estación de destino del viaje, la fecha y hora de inicio del viaje, la fecha y hora del fin del viaje, la distancia aproximada de ese viaje y la velocidad aproximada de ese viaje.

Se implementó la clase Journey la cual consiste en un viaje y contiene:

- startDate
- endDate
- emplacement_pk_start
- emplacement_pk_end
- distance
- speed

Esto se hizo para poder devolver un par <clave, valor> en el cual la clave es el nombre de la estación y el valor es un journey.

Para el Map, primero se verifica que la estación dada por el BikeRent esté en el mapa de estaciones que contiene las estaciones listadas en stations.csv. De esta manera sólo tenemos en cuenta las estaciones presentes en el archivo. Luego la función emite el nombre de la estación y un nuevo Journey con los datos presentes en el BikeRent.

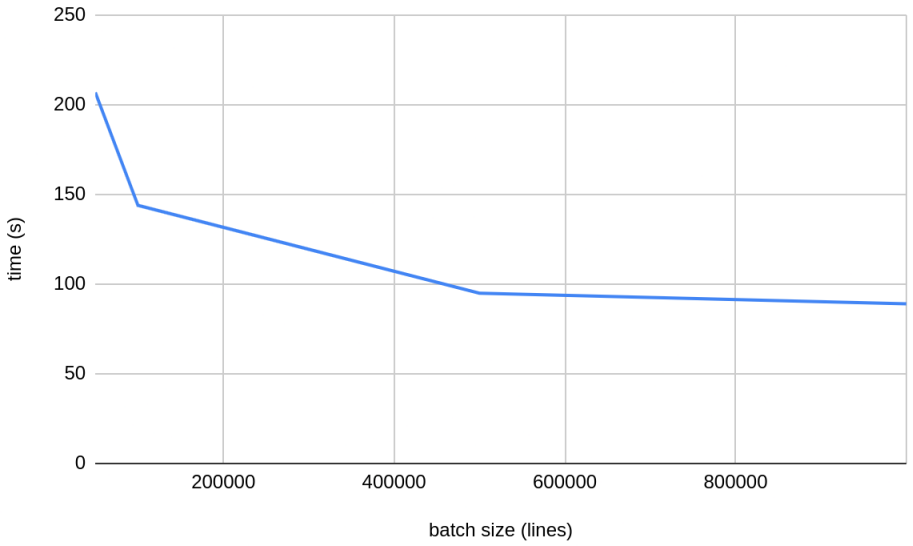
El Reduce se encarga de encontrar el Journey con mayor velocidad y devolverlo.

También se implementó un combiner que realiza la misma función que el Reduce y un Collator que se encarga de ordenar el resultado del reduce de mayor velocidad a menor velocidad y, en el caso de empate, por orden alfabético.

Análisis de tiempo de resolución

Para poder subir la totalidad del archivo de bikes.csv a Hazelcast, se implementó el método fillBikesIList que se encarga de leer de a 500.000 líneas el archivo e ir subiendolas de a batches. Se probó con batches más grandes pero Java arrojaba un error de heap space.

Debajo se puede observar como varían los tiempos en base al batch size. Con valores mayores a 1.000.000 hubieron errores de heap por lo tanto se decidió tomar un valor menor para asegurar el correcto funcionamiento.



Ambas queries tienen tiempos prácticamente iguales de MapReduce ya que en ambos se necesitan mapear y reducir todas las estaciones y bicicletas. Este tiempo dio un promedio de 74s luego de 10 corridas cada uno.

Con respecto al combiner, el código fue probado con un solo nodo, por lo cual no tiene sentido evaluar con y sin combiner debido a que su función principal es reducir la cantidad de datos que se transmiten desde los nodos Map al nodo Reduce. Como fue probado con un solo nodo, los resultados no varían. Pero debería verse una mejora en el tiempo de ejecución usando un combiner y corriendo en más de un nodo.

Potenciales puntos de mejora y/o expansión

Uno de los potenciales puntos de mejora es implementar test unitarios, particularmente para las clases mapper y reducer para asegurar su correcto funcionamiento.

Otro punto de mejora es hacer un mejor análisis para la reducción del uso de la memoria ya que por como corre el server, es necesario que la computadora disponga de 8GB de memoria para correr (debido al flag -Xmx8192m).

Por último, se podrían analizar todos los parámetros uno por uno para ver que tengan el formato correcto. Al día de hoy, no se filtran y se asume que se están enviando correctamente.