Sistemas de Bases de Datos
2021

Centro Asociado de la UNED en Bizkaia

Martín Romera Sobrado
Bilbao

31 de diciembre de 2020

1. Introducción

El objetivo de la práctica es realizar una serie de tareas sobre una base de datos de libros importada de los archivos `authors.csv` y `datasets.csv` utilizando el entorno de Hadoop.

De las opciones que se ofrecían en el enunciado para resolver los ejercicios, mi desarrollo ha sido utilizando la **maquina virtual de cloudera**.

Con la entrega de mi práctica adjunto el notebook de jupyter tal y como se pide en los requerimientos, esta memoria explicando brevemente el desarrollo de la práctica y conclusiones de este, el código fuente de los *scripts* utilizados en el desarrollo (aunque se incluyan dentro del notebook en forma de comentarios), en forma de repositorio de git con los cambios que he realizado a lo largo del desarrollo registrados.

2. Desarrollo

2.1. Preparación del entorno

La preparación del entorno es un paso muy simple, ya que solo consiste en crear un directorio en el sistema de ficheros de hadoop para guardar todos los archivos relacionados con la base de datos, e inicializar la base de datos haciendo uso del fichero `db.hql` dentro del directorio `db/`.

2.2. Ejercicio 1

En este ejercicio creamos las dos tablas necesarias para el resto de ejercicios: `authors` y `dataset`. Las tablas serán externas ya que pretendemos diseñar en un futuro unas función externa (el mapper del ejercicio 5) que interactue con los datos de esta.

2.3. Ejercicio 2

En este ejercicio importamos los datos de `authors.csv` y `datasets.csv` en las tablas que acabamos de crear. Para ello primero se meten ambos archivos dentro del sistema de ficheros de hadoop, y haciendo uso del script `ej2.hql` importamos los datos en ambas tamblas.

2.4. Ejercicio 3

Para este ejercicio tenemos que crear una vista combinando datos de ambas tablas. La orden para hacer esto es muty similar a como serían usando SQL. Esta se recoge en el script `ej3.hql` dentro del directorio `db/`.

2.5. Ejercicio 4

Este ejercicio consiste en realizar operaciones dentro de la base de datos utilizando el propio lenguaje de consultas de Hadoop, *HiveQL*. Se divide en tres apartados:

a) Encontrar el título del libro más alto en el ranking de bestsellers.

b) Encontrar el título del libro con mejor valoración de Ana María Spagna.

c) Encontrar los 5 autores que han escrito más libros.

El mayor problema que puede haber en el desarrollo de estos ejercicios es los empates que puede haber a la hora de resolver la consulta.

En el apartado *a* no ocurre esto ya que al ser un ranking, suponemos que todos los libros (en el ranking) tienen un valor único.

En el apartado *b* para evitar que la consulta de 2 resultados, la he basado en ordenar los libros de la autora según su valoración, y mostrar el que está el inicio de de la tabla resultante de la ordenación con `LIMIT 1`.

En el apartado *c* de hago de forma similar al apartado *b*, ordeno a los autores según el número de libros que han escrito y muestro el "Top 5" con `LIMIT 5`.

2.6. Ejercicio 5

Finalmente, para este ejercicio hay que implementar las funciones `map` y `reduce` para resolver el apartado *a* del ejercicico anterior. De este ejercicio tengo más que hablar, ya que ha estado dandome errores durante casi un mes y medio, y no era capaz de encontrar el problema. Mi problema estaba en que no estaba considerando que hadoop, guardaba los ficheros *csv* partidos en diferentes racks. Esto daba error en mi implementación del *mapper* porque tenía un sistema para "autoencontrar" (para permitir que el programa funcionara para casos más generales) las columnas que necesitaba para sus operaciones de la siguiente manera:

```
if first_line:
        # Buscamos en que columna se encuentran los datos de title y bestseller-rank
        column_count = 0
        for column in line:
                if column == "title":
                        title = column_count
                if column == "bestsellers-rank":
                        rank = column_count
                column_count += 1
                if title == -1 or rank == -1:
                        raise Exception("title or rank column not found")
        first_line = False
```

Este condicional se ejecutaba con la primera linea de cada *csv* que recibía como entrada, y al no encontrar ninguna columna que se llamara o "title" o "bestsellers-rank", hacía saltar el error de la linea `raise Exception("title or rank column not found")`.

Este error, aunque era un error bastante básico, me ha sido dificil encontrarlo, ya que al dirigirme al panel de *Hue*, y observaba el trabajo fallido del mapreduce, si buscaba la salida estándar de errores (`stderr`), esta no se mostraba por hacerle falta la activación del componente log4j (supongo que un error de esa compilación de Hadoop).

Al final encuentro el problema al ver que en el trabajo se realizan dos "map"s, uno exitoso y el otro fallido. De lo que deduzco que Hadoop ha dividido el csv en 2 archivos, con uno de ellos sin la cabecera.

El problema lo resuelvo en el commit `45be2d` del git local, por si se quiere analizar el cambio.

Por lo demás he utilizado el patrón "Top Ten" de [MS12], solo que adaptado para que en vez de ser un top 10, sea un top 1.

Referencias

[MS12] Donald Miner y Adam Shook. "MapReduce Design Patterns". En: O'Rielly, 2012, págs. 58-65.