DBMS Models and implementation

Project - 3

Fall 2023

Group: 10

Sri Harsha Kolli – 1002063354

Abhinay Reddy Gurrala – 1002058438

Project:

A Map/Reduce program to list the names of people, who have directed and acted in the same IMDb title/category of any genre and output title, director, actor, genre, and year with the given parameters.

Overall Status:

Understood the problem statement and implemented all the necessary functions to give the desired output. We used 3 mappers to map the input from the given three input files. The mapper output is sent to the one or two reducers depending on the configuration. We tried running both 3M-1R and 3M-2R configurations to analyze and compare the performances.

Input and Output:

The job takes three input files ('title.basics.tsv', 'title.actors.tsv', 'title.crew.tsv') and produces output in the form of key-value pairs with the title ID as the key and relevant details (title, actorName, actor, director, genre, year) as the value.

Parameters:

startYear: 1960-1970

titleType: tvMovie

Approach:

1. **Mappers:** The three mappers that we implemented are as follows –
    a. **TitleBasicsMapper:** Maps the input from ‘title.basics.tsv’. The input classes to this mapper are LongWritable and Text. The key corresponds to the row number and value corresponds to the field values in the tsv file. The mapper first splits the input to separate and extract the column values. This mapper also filters out the rows based on the given parameters and eliminates null (\n) values. Finally, the title ID (tconst) is emitted as key and primarytitle, genre and startyear are emitted as values to the reducer(s).

    b. **TitleActorsMapper:** Similar to TitleBasicsMapper, this mapper processes the input from 'title.actors.tsv'. It extracts the actor's ID, name, and the title ID (tconst). The mapper filters out the header and emits key-value pairs with tconst as the key and actor details (Actor, actorid, actorName) as the value.

    c. **TitleCrewMapper:** Maps the input from 'title.crew.tsv'. It extracts the title ID (tconst) and the director(s) associated with the title. The mapper filters out null (\n) values and emits key-value pairs with tconst as the key and director details (Crew, directors) as the value.

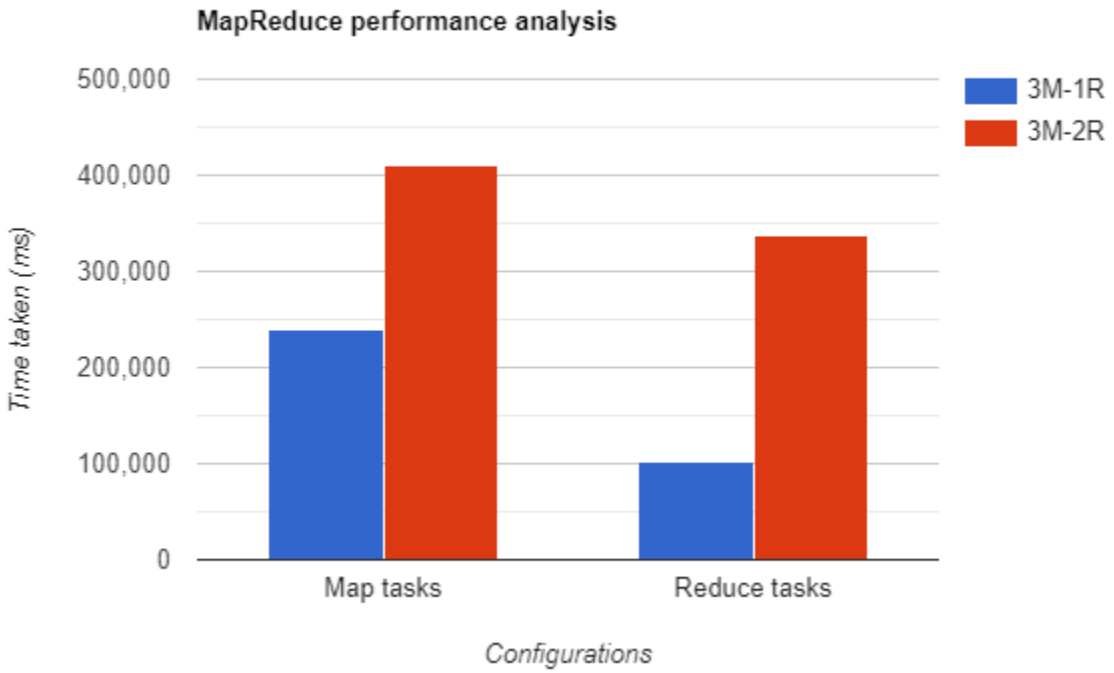
2. **Reducer:** The **MovieActorDirectorReducer** class consolidates the information from the three mappers based on the title ID (tconst). It iterates through the values associated with each key and extracts information such as title, genre, start year, director(s), actor(s), and actor names. It checks if the actor is also a director by comparing their names and, if they match, sets a flag (actorIsDirector) to true. For this program, after testing different configurations. We set the number of reducers to 1. For more than one reducer, the time taken by mapper and reducer tasks is significantly higher.

Hardware Configuration:

- Processor: Intel(R) Core (TM) i7-8750H CPU @ 2.20GHz (12 CPUS)
- Installed RAM: 16.0 GB
- System type 64-bit operating system, x64-based processor
- SSD: 512GB
- OS: Windows 11 Pro
- Display Card: NVIDIA GeForce RTX 2070 Max-Q Design
- Display Memory: 8GB

Analysis:

We have used different configurations i.e., 3M-1R and 3M-2R for our MapReduce task.



- As seen in the above figure, the job with 2 reducers took more time (410 seconds) compared to the job with 1 reducer (239 seconds). This is expected as the overhead of managing multiple reducers adds some latency.
- The job with 2 reducers resulted in a more distributed reduction of data, potentially leading to a more balanced workload across the reducers.
- Using a single reducer is preferable in this scenario, as it showed better performance in terms of total execution time.

File Descriptions:

1. imdb_mr.java: Contains the implementation for all the mappers and reducer including the main function for the MapReduce program.
2. imdb_mr.jar: The jar file which aggregates all the class files.
3. 3m-1r.log: The log file which contains the log of MapReduce program with three mappers and one reducer configuration.
4. 3m-2r.log: The log file which contains the log of MapReduce program with three mappers and two reducers configuration.
5. part-r-00000: The final output file of the MapReduce job.

Division of Labor:

- Sri Harrsha Kolli: Installed Hadoop in WSL, implemented the mappers and reducer for MapReduce program, analysis of different configurations.
  Time spent: 20 hrs
- Abhinay Reddy Gurrala: Debugging the code, wrote the SQL query that needs to be executed on Omega for output verification.
  Time spent: 15 hrs

M/R configuration details for multiple inputs and other details:

Used MultipleInputs from org.apache.hadoop.mapreduce.lib package to add multiple input paths to the program.