PDP Assignment 1 – MRJob / Mapreduce

Name: Sjors Grooff

Student number: 634293

Email: 634293@student.inholland.nl

Github url: https://github.com/groofy98/PDP/blob/main/Assignment%201/pdp_assignment_1.py

Date: 9-9-2021

The assignment

I did the first two steps of this assignment. First counting the number of ratings per movie then sorting the list by the most number of ratings.

The script

```
1.  # Command: python pdp_assignment_1.py --hadoop-streaming-jar /usr/hdp/current/hadoop-
    mapreduce-client/hadoop-streaming.jar  -r hadoop hdfs:///user/maria_dev/ml-100k/u.data
2.
3.  # Import necessary libraries
4.  from mrjob.job import MRJob
5.  from mrjob.step import MRStep
6.
7.  class CountMovieRatings(MRJob):
8.
9.      # Define the steps that need to be executed.
10.     def steps(self):
11.         return [
12.             MRStep(mapper=self.mapper_get_ratings,
13.                 reducer=self.reducer_count_ratings_by_movie),
14.             MRStep(reducer=self.reducer_sort_movie_by_rating_count)
15.             ]
16.
17.     # Load the data by splitting the incoming lines.
18.     # We only keep the movie id and add a 1 to each tuple to be able to sum later.
19.     def mapper_get_ratings(self, _, line):
20.         (user_id, movie_id, rating, timestamp) = line.split('\t')
21.         yield movie_id, 1
22.
23.     # Sum all the "1" values we have put in the tuple grouped my movie
24.     def reducer_count_ratings_by_movie(self, movie_id, value):
25.         yield None, (sum(value), movie_id)
26.
27.     # Sorts the movies by number of ratings and reverse so we have the most rated movie
    first.
28.     def reducer_sort_movie_by_rating_count(self, _, rating_counts):
29.         for count, key in sorted(rating_counts, reverse=True):
30.                 yield (key, int(count))
31.
32. # Mandatory line to make the script run.
33. if __name__ == '__main__':
34.     CountMovieRatings.run()
```

How it works

First off, we define the necessary MRJob libraries. In this case MRJob and MRStep.

Then we define our class in this case CountMovieRatings.

After that I created the steps method that contains all the actions that need to be executed on the Hadoop cluster. The steps are as follows:

1. We use the mapper to split the lines and only keep the movie id's
2. We use a reducer to sum the number of times a movie_rating is present.
3. The last step sorts the list descending by the amount of rating per movie

How to run it

To run this script there are the following prerequisites:

- Running Hadoop cluster
- A recent python version with Pip
- Use Pip install to get the MRJob dependencies
- The Movielens dataset stored on hdfs

To run the script I used the following linux command while connected to Virtualbox via SSH:

```
python pdp_assignment_1.py --hadoop-streaming-jar /usr/hdp/current/hadoop-mapreduce-
client/hadoop-streaming.jar  -r hadoop hdfs:///user/maria_dev/ml-100k/u.data
```

The result

The picture below shows the most rated movies