Large Scale Data Management: Assignment 1

Tsirmpas Dimitris f3352315

As discussed in-class, the code and output logs are included as separate java source and txt files respectively. This report will include all links and references to these files.

Part 1

The document we selected is the 1st Harry Potter book in .txt format, available through this link.

We include the script used to get the file, deploy it to the docker container, deleting the previous output directory if it exists, executing the map-reduce job, getting the output, and saving the execution logs to a .txt file. Note that this script assumes a running, healthy vagrant instance.

```
vagrant ssh -c "
wget \"https://raw.githubusercontent.com/amephraim/nlp/master/texts/J.%20K.%20Rowling%20-%20
docker exec namenode hdfs dfs -put harry_potter.txt /user/hdfs/input/;
docker cp \"harry_potter.txt\" namenode:/;
cd /vagrant/hadoop-mapreduce-examples/;
mvn clean install;
docker cp /vagrant/hadoop-mapreduce-examples/target/hadoop-map-reduce-examples-1.0-SNAPSHOT-
docker exec namenode hdfs dfs -rm -r /user/hdfs/output/
docker exec namenode hadoop jar /hadoop-map-reduce-examples-1.0-SNAPSHOT-jar-with-dependenci
docker exec namenode hdfs dfs -text /vagrant/hdfs/output/part-r-00000 | head -100;
" | tee output.txt
```

(Script also available at `part1/deploy.sh`).

The execution logs can be found in `part1/output.txt`.

Part 2

We implement our own classes `Driver.java` and `SpotifyStats.java` at `part2/map-reduce-spotify/src/main/java/gr/aueb/dimits/mapreduce/spotifystats/`. Since we changed the execution parameters and slightly modified the directory structure, we use a modified `pom.xml` file. For the sake of reproducability, we include the classes, pom file, deploy script, and output file in their *original* directory structure.

Implementation details and design decisions are available in the form of javadoc strings and comments in said source files.

The script used to execute the job in a manner similar to the equivalent script in Part 1 can be found below. Note that this script assumes a running, healthy vagrant instance.

```
vagrant ssh -c "
cd /vagrant/hadoop-spotify;
docker cp universal_top_spotify_songs.csv namenode:/;
docker exec namenode hdfs dfs -put universal_top_spotify_songs.csv /user/hdfs/input/;
cd map-reduce-spotify;
mvn clean install;
docker cp /vagrant/hadoop-spotify/map-reduce-spotify/target/hadoop-spotify-1.0-SNAPSHOT-jar-
docker exec namenode hdfs dfs -rm -r /user/hdfs/output/;
docker exec namenode hadoop jar /hadoop-spotify-1.0-SNAPSHOT-jar-with-dependencies.jar namen
docker exec namenode hdfs dfs -text /user/hdfs/output/part-r-00000 | head -100;
" | tee output.txt
```

(Script also available at `part2/deploy_spotify.sh`)

The results of the script's execution can be found in `part2/output.txt`.