School of Information Technologies
Dr. Ying Zhou

COMP5349: Cloud Computing **Sem. 1/2019**

Assignment 1: Data Analysis with MapReduce and Spark

Individual Work: 20% **22.03.2019**

1 Introduction

This assignment tests your ability to implement simple data analytic workload using basic features of `MapReduce` and `Spark` framework. In particular, you are encouraged to practice the skills of designing algorithms by organizing data as key value pairs, the concept that is central to both frameworks. The data set you will work on is adapted from Trending Youtube Video Statistics data from Kaggle. There are **two workloads** you should design and implement against the given data set. **You are required to implement one workload with `MapReduce` and the other workload with `Spark`.**

2 Input Data Set Description

The dataset contains several months' records of daily top trending YouTube video in the following ten countries: Canada,France, Germany, India,Japan, Mexico, Russia, South Korea, United Kingdom and United States of America. There are up to 200 trending videos listed per day.

In the original data set, each country's data is stored in a separate CSV file, with each row representing a trending video record. If a video is listed as trending in multiple days, each trending appearance has its own record. The record includes video id, title, trending date, publish time, number of views, and so on. The record also includes a `category_id` field. The categories are slightly different in each country. A JSON file defineing the mapping between category ID and category name is provided for each country.

The following preprocessing have been done to ensure that you can focus on the main workload design.

- Merge the 10 individual CSV files into a single CSV file;
- Add a column `category` to store the actual category name based on the mapping
- Add a column `country` to store the trending country, each country is represented by two capital letter code.

- Remove rows with invalid video id values
- Remove textual columns that are not relevant to the workloads and may cause encoding and parsing issue

The results is a CSV file `AllVideos_short.csv` with mostly numeric and date columns.

3 Analysis Workload Description

3.1 Category and Trending Correlation

Some videos are trending in multiple countries. We are interested to know if there is any correlation between video category and trending popularity among countries. For instance, we all know that "music has an universal appeal", in the context of Youtube videos, we may expect to see a common set of trending music videos among many countries. On the contrary, we may expect to see each country with a distinctive set of trending political videos.

In this workload, you are asked to find out the average country number for videos in each category. For instance, if in the data set there are five videos belonging to category `Sports`, their trending data are as follows:

video_id	category	trending_date	views	country
1	Sports	18.17.02	700	US
1	Sports	18.18.02	1500	US
2	Sports	18.11.03	3000	US
2	Sports	18.11.03	2000	CA
2	Sports	18.11.03	5000	IN
2	Sports	18.12.03	7000	IN
3	Sports	18.17.04	2000	JP
4	Sports	18.16.04	3000	KR
4	Sports	18.17.04	9000	KR
5	Sports	18.16.04	4000	RU

We can see that video 1 appears in 1 country; video 2 appears in 3 countries; video 3, 4 and 5 each appears in 1 country respectively. The average country number for videos in category `Sports` would be $\frac{(1+3+1+1+1)}{5} = 1.4$ The final result of this work load would look like the following:

```
Music: 1.31
News & Politics: 1.05
...
```

3.2 Controversial Trending Videos Identification

Listing a video as trending would help it attract more views. However, not all trending videos are liked by viewers. It is not unusual for a trending video to have more `dislikes` than `likes`; For some video, listing it as trending would increase its `dislikes` number more than the increase of its `likes` number. This workload aims to identify such videos.

Below are a few records of a particular video demonstrating the change of various numbers over time:

video_id	trending_date	views	likes	dislikes	country
QwZT7T-TXT0	2018-01-03	13305605	**835378**	629120	US
QwZT7T-TXT0	2018-01-04	23389090	1082422	1065772	US
QwZT7T-TXT0	2018-01-05	28407744	1204072	1278887	US
QwZT7T-TXT0	...	...	...	...	US
QwZT7T-TXT0	2018-01-09	37539570	1402578	**1674420**	US
QwZT7T-TXT0	2018-01-03	13305605	**835382**	629123	GB
QwZT7T-TXT0	2018-01-04	23389090	1082426	1065772	GB
QwZT7T-TXT0	2018-01-05	728407744	1204074	1278889	GB
QwZT7T-TXT0	...	...	...	...	GB
QwZT7T-TXT0	2018-01-18	45349447	1572111	**1944971**	GB

The video has multiple trending appearances in US and GB. In both countries, its `views`, `likes` and `dislikes` all increase over time with each trending appearance. As highlighted in the table above, the `dislikes` number grows much faster than the `likes` numbers. In both countries, the video ended with higher number of `dislikes` than `likes` albeit starting with higher `likes` number.

In this workload, you are asked to find out the top 10 videos with fastest growth of `dislikes` number between its first and second trending appearances. Here we measure the growth of `dislikes` number by the gap of `dislikes` increase and `likes` increase between the first two trending appearances in the same country.

For instance, the `dislikes` growth of video `QwZT7T-TXT0` in US is computed as follows: $(1065772 - 629120) - (1082422 - 835378) = 189608$

Where the first component is the increase of `dislikes` and the second component is the increase of `likes` between the first and second trending appearances .

The result of this workload should show a few details of the top 10 videos, including the video id, category, dislike growth value and country code. Below is a few sample results:

```
"BEePFpC9qG8", 366556, "Film & Animation",  "DE"
"RmZ3DPJQo2k", 334594, "Music",             "KR"
"1Aoc-cd9eYs", 192222, "Entertainment",     "GB"
"QwZT7T-TXT0", 189608, "Entertainment",     "US"
"QwZT7T-TXT0", 189605, "Entertainment",     "GB"
```

If a video has changed its category name over time, you can use the category of the first appearance. It is possible to include the same video multiple times in top 10 list if it has large `dislikes` growth in multiple countries. Video `QwZT7T-TXT0` is such an example.

4 Coding and Execution Requirement

Below are requirements on coding and Execution:

- You can implement the workloads in either Java or Python. **No other language is allowed**.
- You must use MapReduce and Spark framework in your implementation. Implementation using plain language features will not achieve any point. Implementation "pretends" to use the framework will not achieve any point. A typical example of "pretending" to use MapReduce framework is to design the workload as one job, with an identity mapper and a bloated reducer; The identity mapper just pass the input as is to the reducer, which has all the implemented everything in one function.
- Your code must take two parameters: an `input_path` and an `output_path`. The output should be written to a file.
- Your code must execute on AWS EMR with `emr-5.21.0` software release.
- For Java implementation, a script (e.g. ant build file) must be provided to allow easy creation of the executable jar file. The job submission command must be provided in a `read.me` file.
- For Python implementation, a shell script must be provided with job submission command.

5 Deliverable

There are two deliverables: **source code** and **brief report** (up to 2 pages). Both are due on **Wednesday 10th of April 23:59 (Week 7)**.

JAVA submission should be organized in the following folder structure and packed as a zip file:

```
/workload1
    /src
    buildfile
    read.me
/workload2
    /src
    buildfile
    read.me
/report
    <uniKey>-report.pdf
```

Python submission should be organized in the following folder structure and packaged as a zip file:

```
/workload1
    xxx.py (multiple files)
    run.sh
/workload2
    xxx.py (multiple files)
    run.sh
/report
    <uniKey>-report.pdf
```

The submitted zip file should be called `<labCode>-<uniKey>-<firstNae>-<lastName>.zip`.

There will be a **demo** in week 7 during tutorial time. The tutor will upload all submissions on AWS and run your code on EMR cluster during the demo. You are expected to answer design and implementation related questions duing the demo.

Submit a hard copy of your report together with signed cover sheet during the demo.

The report must contain a computation graph for each workload, with very brief descriptions. A sample report will be uploaded as reference.