THE UNIVERSITY OF SYDNEY

School of Information Technologies
Dr. Ying Zhou

COMP5349: Cloud Computing **Sem. 1/2018**

Assignment 1: Simple Data Analysis with MapReduce and Spark

Individual Work: 20% **29.03.2018**

1 Introduction

This assignment tests your ability to implement simple data analytic workload using basic features of MapReduce and Spark framework. The data set you will work on is the **Trending Youtube Video Statistics** data from Kaggle (https://www.kaggle.com/datasnaek/youtube-new). There are two workloads you should design and implement against this data set. You are required to implement one with MapReduce and the other with Spark. You can choose which framework you want to use on which workload.

2 Input Data Set Description

The dataset contains several months' records of daily top trending YouTube video in the following five countries: Canada, France, Germany, UK and USA. There are up to 200 trending videos listed per day.

Each country's data is saved in a separate CSV file. Each row of the CSV file represents 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 video record also includes a category_id field. The categories are slightly different in each country. A JSON file is provided for each country. The JSON file defines the mapping between category ID and category name.

[Update in 05/04/2018] Another version is also provided. This version contains only a single CSV file. The single CSV file combines all rows in the original five CSV files. It also add two extra columns: category and country. The country column contains country code extracted from the original file name. The category column contains the actual category name obtained from the mapping JSON file. Both versions can be found in repository lab_commons.

A few kernels have been uploaded by Kaggle users, providing various analysis on the data set. You may find this one helpful in exploring the data set in general.
https://www.kaggle.com/hoonkeng/deep-analysis-on-youtube-trending-videos-eda

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 category and overlapping trending. For instance, if UK and CA users have common interests in music, but very different interest in sports, we might see 3% trending music videos in UK that also appear in the trending list of CA; while only 0.5% of trending sports videos in UK appears in CA's trending list.

In this workload, you are asked to find out, for a given pair of countries A and B, for each category in country A, the total number of videos trending in country A and the percentage of them that are also trending in country B. For any video with multiple trending appearances in a country, it should be counted as one video in that country.

The result would look like, suppose the country is GB and US

```
Entertainment; total: 617; 31.4% in US
Sports; total:152; 17.1% in US
...
```

It means that there are 617 videos from Entertainment category in UK's trending list. 31.4% of the 617 videos also appear in US's trending list; There are 152 videos from Sports category in UK's trending list. 17.1% of the 100 videos also appear in US's trending list.

3.2 Impact of Trending on View Number

Listing a video as trending would help it attract more views. The view number may quickly increase after a video is listed as trending for the first time. In fact it is not unusual for the view number to double between a video's first and second trending appearance.

Below are a few records of a particular video:

videoID	Trending Date	Publish Time	Views	Country
xYtsL9znopI	18.17.02	2018-02-16T14:00:09.000Z	960453	CA
xYtsL9znopI	18.18.02	2018-02-16T14:00:09.000Z	2109193	CA
xYtsL9znopI	18.19.02	2018-02-16T14:00:09.000Z	2768767	CA
xYtsL9znopI	18.20.02	2018-02-16T14:00:09.000Z	3213410	CA

The video has four trending appearances in CA between February 17 of 2018 and February 20 of 2018. The view number in its first appearance (2018/02/17) is 960,453; the view number in its second appearance (2018/02/18) is 2,109,193. There is a 119.6% increase between the second and first appearance. In contrast the increase between the third and the second appearance is only 31.2%.

In this workload, you are asked to find out, for each country, all videos that have greater than or equal to ~~100%~~ **1,000%** increase[1] in viewing number between its second and first trending appearance. The result should be grouped by country and sorted discerningly by percent increase.

The result would look like

```
DE; V1zTJIfGKaA, 19501.9%
DE; RIgNyiGttog, 12346.5%
...
CA; _I_D_8Z4sJE, 8438.1%
CA; -K9ujx8vO_A, 8298.3%
...
```

[update in 05/04/2018] Your output may slightly variate from the above example. For instance, you may choose different delimiter character, or your may show fold increase instead of percentage increase.

4 Coding and Execution Requirement

Your implementation should utilize features provided by the respective framework. In particular, you should parallelize most of the operations. The Hadoop implementation should run in a pseudo-distributed mode. The Spark implementation should run in a standalone cluster or YARN cluster on a single machine.

5 Deliverable

There are two deliverables: **source code** and **brief report** (up to 2 pages). Both are due on ~~**Wednesday 18th of April 23:59 (Week 6)**~~ **Wednesday 25th of April 23:59 (Week 7)** . Please submit the source code and a soft copy of the report as a zip or tar file in Canvas. You need to **demo** your implementation in week 7 during tutorial time. Please also submit a hard copy of your report together with signed cover sheet during the demo.

The report should describe the design of both workloads. In particular, you should describe the sequence of operations/actions taken to obtain the final result, and highlight the part that can be executed in parallel. You can use diagrams to help explaining the sequence.

[1] the original threshold 100% would generate too many results. We change it to 10-fold increase for easy marking. This should not affect your programming structure