THE UNIVERSITY OF SYDNEY

School of Information Technologies
Faculty of Engineering & IT

ASSIGNMENT/PROJECT COVERSHEET - INDIVIDUAL ASSESSMENT

Unit of Study: COMP5349 Cloud Computing

Assignment name: Assignment 1

Tutorial time: R16C Thursday 4:00PM - 6:00PM

Tutor name: Shizhe Zang

DECLARATION

I declare that I have read and understood the *University of Sydney Academic Dishonesty and Plagiarism in Coursework Policy*, and except where specifically acknowledged, the work contained in this assignment/project is my own work, and has not been copied from other sources or been previously submitted for award or assessment.

I understand that failure to comply with the the *Academic Dishonesty and Plagiarism in Coursework Policy*, can lead to severe penalties as outlined under Chapter 8 of the *University of Sydney By-Law 1999* (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgement from other sources, including published works, the internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

I realise that I may be asked to identify those portions of the work contributed by me and required to demonstrate my knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

Student ID: 470349281

Student name: Wenjing Deng

Signed Wenjing Deng **Date** 2019.04.10

SIT Building, J12
The University of Sydney
NSW 2006 Australia

T +61 2 9351 3423
F +61 2 9351 3838
E sit.info@sydney.edu.au
sydney.edu.au/it

ABN 15 211 513 464
CRICOS 00026A

COMP5349 Assignment 1
Name: Wenjing Deng

Tutor Name: Shizhe Zang
Name SID: 470349281

Workload	Implementation	Programming Language
Category and Trending Correlation	Map Reduce	Python
Controversial Video Identification	Spark	Python

Workload: Category and Trending Correlation

There is one MapReduce job is used for this workload. The Figure 1 below illustrates the implementation of this job. It consists of two phases: map and reduce, which is separately implemented in mapper.py and reducer.py.

The sequence of the operations in this workload is:

Step 1: Read the file named "AllVideos_short.csv" line by line and feed the lines to the stdin of the process.

Step 2: For each line, split it by ",". Do the map which means keeping three variables which is category, country, and video_id, and converting them to the key/value pair as map out. In this process, the category is key, and the country and video_id are values.

Step 3: The reducer received the key/value output from mapper which sorted by key: category, and then filter out the headers which is the first line. A dictionary is set which named is id_category. It is used to store the correspondence between video_id and country.

Step 4: In the process of reduce, when the category is same, put video_id and its corresponding country into the id_category dictionary. When the category change, count the number of the video and the country they appeared each to get the total average country number for videos in certain category and then print output. After that, clear the dictionary for next category.

Step 5: Finally output the data in the form like: current_category: countCountry/countID.

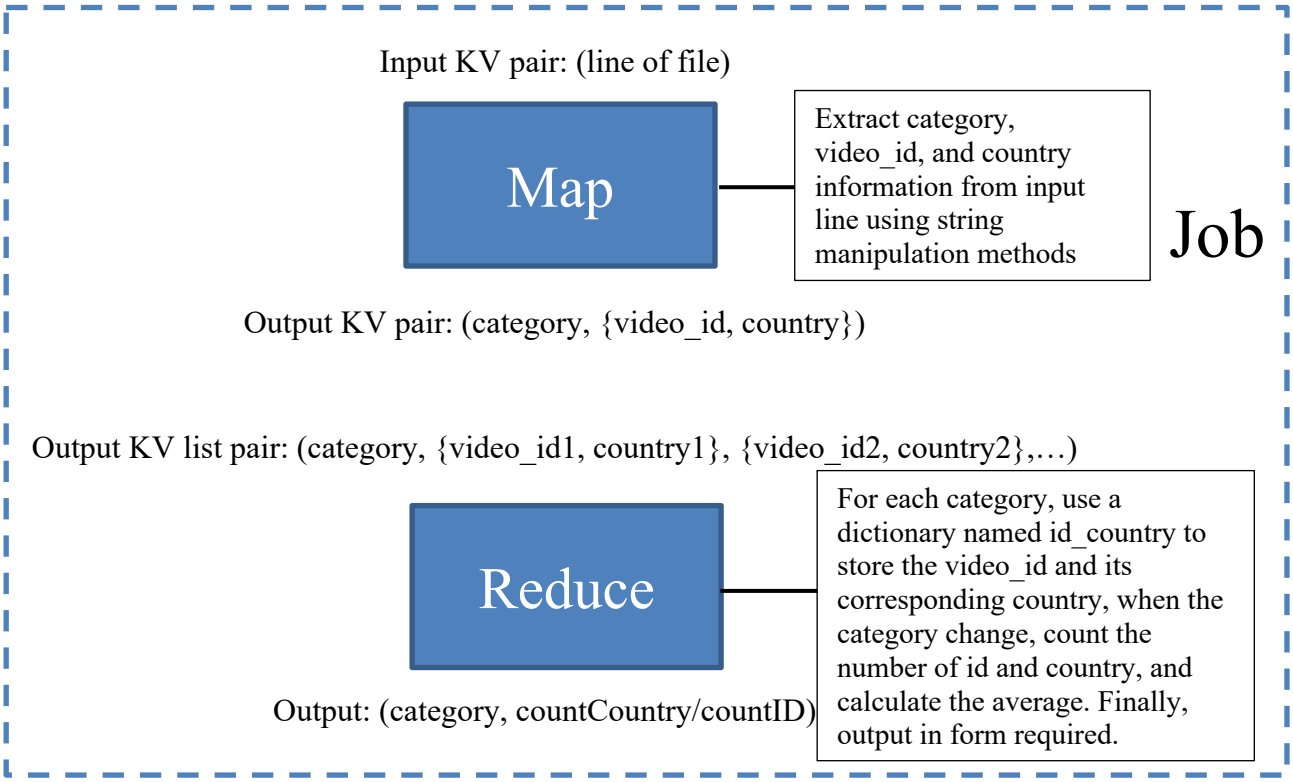

Figure 1: The MapReduce Computation Graph for workload 1.

Workload: Controversial Video Identification

The sequence of the transformations and actions are illustrated in Figure 2. The file named AllVideos_short.csv is read in the format of RDD. Because the first row in it is the headings, filter out the headers after mapping to create ((video_id+country), (trending_date, likes, dislikes, category))RDD pair. GroupByKey transformation is used here. So the values of pairRDD are grouped to the same key(video_id+country). MapValues is then applied. It involves the function of calGrowth which sorting by time sequence and calculating the dislikes growth of each video in each country (by (2nd dislike - 1st dislike) – (2nd like - 1st like)). During calculating, we set the growth is 0 whose videos' trending date are less than two date. After that, change to a new RDD pair which the key is growth and value is (video_id, country, category) through a map transformation. The next step is, using SortByKey transformation to sort RDD pair in descending order by the key: growth and then get the top 10 of the dislikes growth. Becasuse after the function take(10), the data is the form of list, a parallelize transformation is used to convert list to RDD. Finally, the map transformation is used to change the format to suitable for requirements of assignment and use an action named saveAsTextFile to save the result.

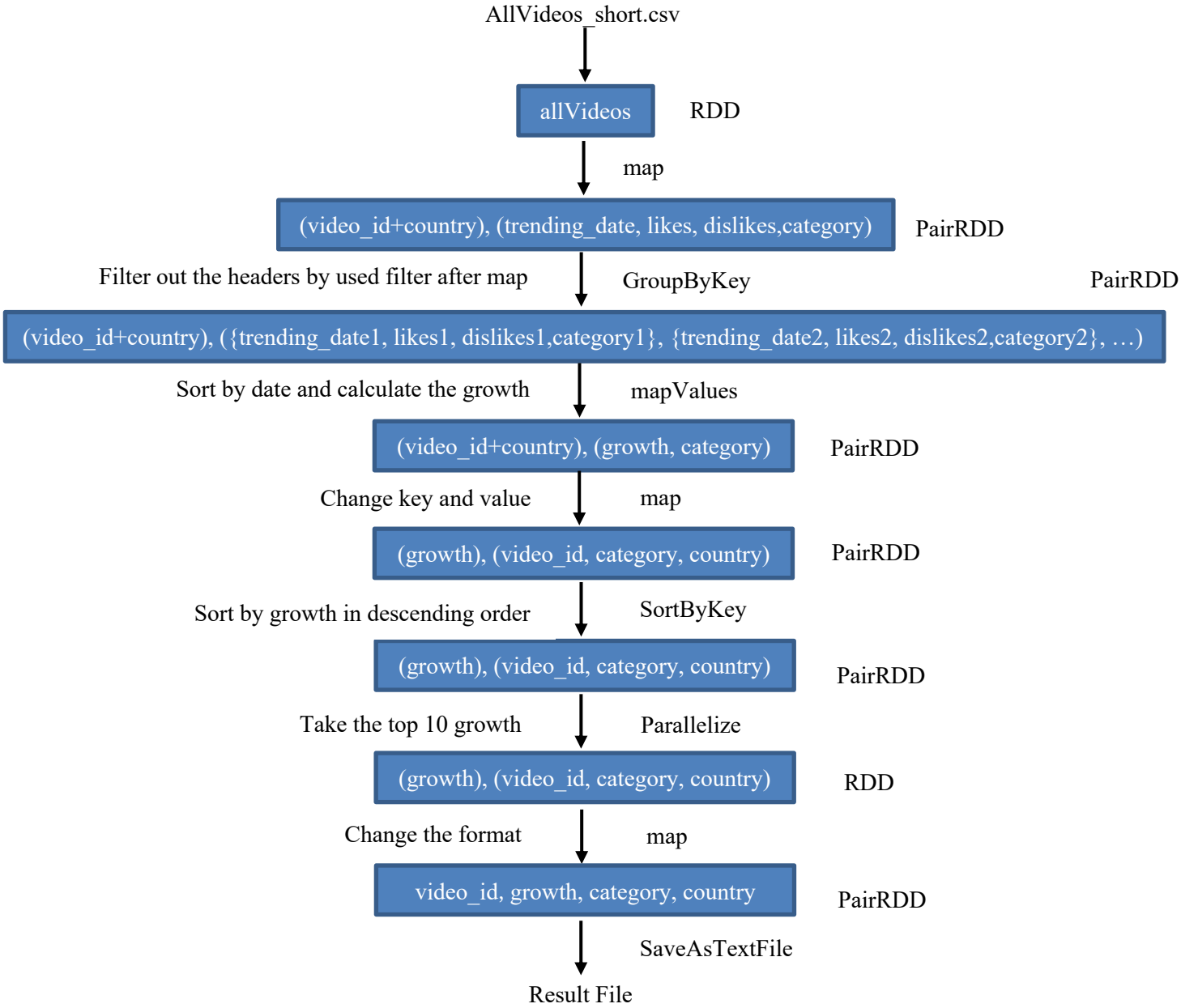

Figure 2: The Spark Computation Graph for workload 2.