COMP5349 Assignment 1 Tutor Name: Shizhe Zang
Name: Zhuoyang Li SID: 480164337

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

Workload1: Category and Trending Correlation

The Computation Graph is illustrated in Figure 1.

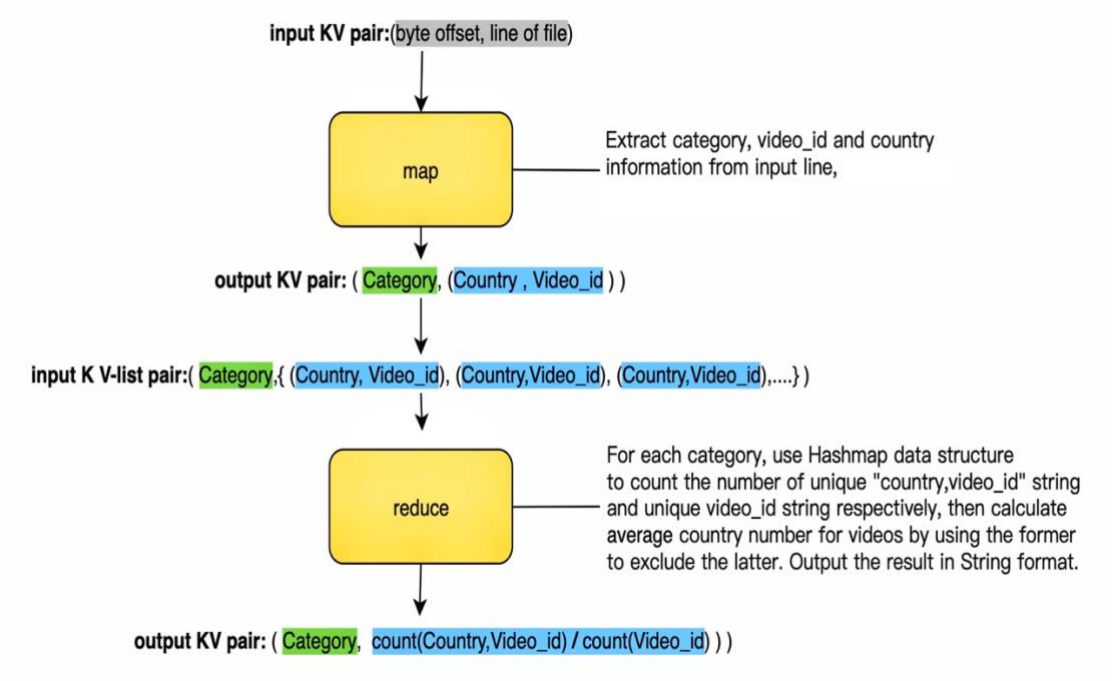


Figure 1

The sequence of transformations and actions are illustrated in Figure 1.

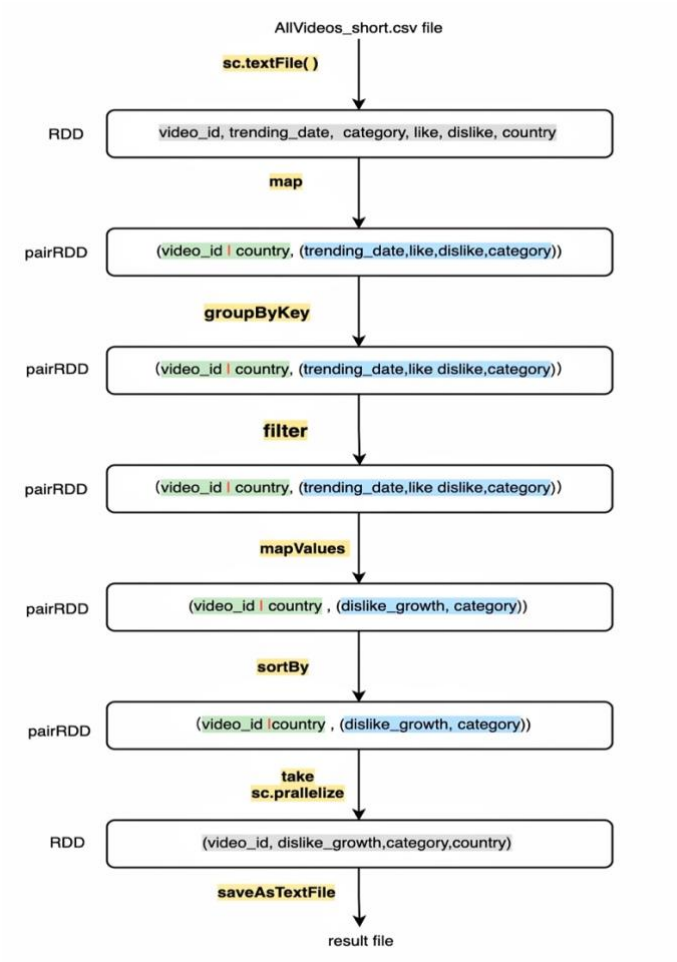
Map function: Split the data into array, extract category, video_id, country and convert to string, combine the country and video_id together by adding ","inside, using .set() function to set category as key and (country, video_id) as value.

Reduce function: For each category, use two HashMap function, one for unique (country, video_id) string and other for unique video_id string, set two counter for them respectively, use containsKey() to check if there already has the same string, and if not same string, put the string and count 1 in the HashMap data structure , and counter++ for calculate how many string has been put into it. Finally use the unique (country, video_id) string's counter exclude the unique video_id string's counter, get the average country number for videos.

COMP5349 Assignment 1 Tutor Name: Shizhe Zang
Name: Zhuoyang Li SID: 480164337

Workload2: Controversial Video Identification

The Computation Graph is illustrated in Figure 2.



The sequence of transformations and actions are illustrated in Figure 2.

Read in the AllVideos_short.csv file and map to the create(video_id|country,(trending_date,like,dislike,category))RDD pair, groupByKey transformation is then applied for group the RDD pair by (video_id|country), filter the grouped RDD pair by len(trending_date,like,dislike,category) to filter out the "video_id|country" with only one trend, because if there is only one trend, there will be no increments like and dislike. Then use mapValues to do the sort_and_calculate function on the filtered value of RDD pair. the function of sort_and_calculate sort the value (trending_date,like,dislike,category) by date order , and then use dislike and like in second date subtract dislike and like in first date respectively to get dislike_increase and like_increase, use the former to subtract the later to calculate the dislike_growth. Output key and calculated values (RDD pair).

With sortby, sort the RDD pair in descending order by the dislike_growth in the value. With take, get the top 10, here the RDD pair convert to list format. So use the sc.prallelize to convert the list to become the RDD. Finally, saveAsTextFile.