Workload	Implementation	Programming Language
Category and Trending Correlation	Spark	Python
Impact of Trending on View Number	Map Reduce	Python

Workload: Category and Trending Correlation

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

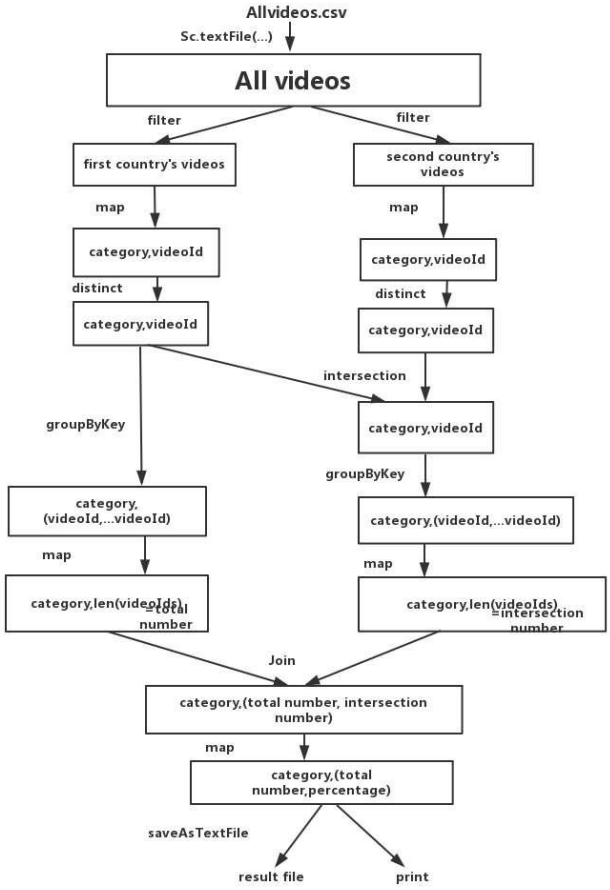

Figure 1: Spark phase for the workload

Allvideos.csv file is read in and is filtered with the country field to get first country and second country's videos records respectively. Then first country's records are mapped to create(category, videoId) RDD pair, and then get distinct(category, videoId) RDD pair using distinct function. Similarly, get the second country's distinct(category, videoId) RDD pair . Using intersection function for the first country's distinct(category, videoId) RDD pair and second country's distinct(category, videoId) RDD pair to get intersection.

Apply the groupByKey for the first country's distinct (category,videoId) RDD pair to get (category,(videoId,…)) RDD pair, and then it is mapped to get (category, total number)RDD pair, wherein, total number is the length of videoIds in above (category,(videoId,…)) RDD pair. Similarly, get (category, intersection number)RDD pair.

Then join the above mentioned (category, total number) RDD pair and (category, intersection number) RDD pair to get (category,(total number, intersection number))RDD pair, and then it is mapped to get (category, (total number, ratio of intersection number and total number)) RDD pair.

Finally, the result is printed and an action to save the result as text file is called to conclude the program.

Parallelization

The filter and map operations can run in parallel on different partitions of the videos contents.The distinct operations and intersection operation can run in parallel, but shuffling is required during these operations.The groupByKey operations and join operations can run in parallel and pipeline with the next map operation, but shuffling is required.

Workload: Impact of Trending on View Number

One MapReduce job is used for this workload. It consists of only two phases: Map then Reduce. For each input row described in figure2, the mapper extracts videoID, country, trendingDate and view numbers, and the mapper transforms trendingDate to a timestamp. Each row yield (country+**":"**+videoID, timestamp+**":"**+view number) key, value pair.

The reducer received output from mapper, thus get the key, value pair. Split the value to get timestamp and view number, and then load it into a list A, each item in the list A is like this: **country+":"+videoID:[(timestamp, view number)…(timestamp, view number)]**, and for each item, sort it and then calculate the ratio of the second view number and the first view number, if the ratio larger or equal to 10, then load **(country+":"+videoID: the ratio)** into an array B. For each item of array B, split the key into country and videoID, and load it into the third list C, each item in list C is like this: **country:[(ratio, videoID)…(ratio, videoID)]**, and for each item, reverse sort it and then traverse the item, print the country, videoID and ratio.

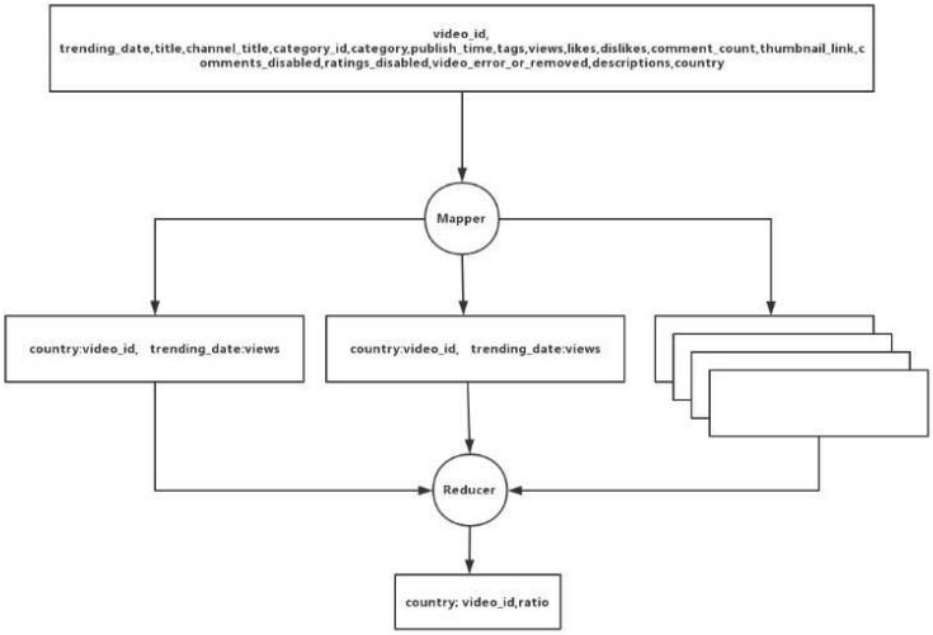

Figure 2: MapReduce phase for the workload.

Parallelization

Both mapper and reducer phases can run in parallel. Mappers run in parallel on different partitions of the input data. 3 reducers are set to use, and they run in parallel on different partitions of the intermediate results.