COMP5349 Cloud Computing-Assignment 1
Tutor: Chenhao Huang **Name**: Yili Wang **SID**: 470106260

1.Workload Allocation

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

2.Workload-Category and Trending Correlation

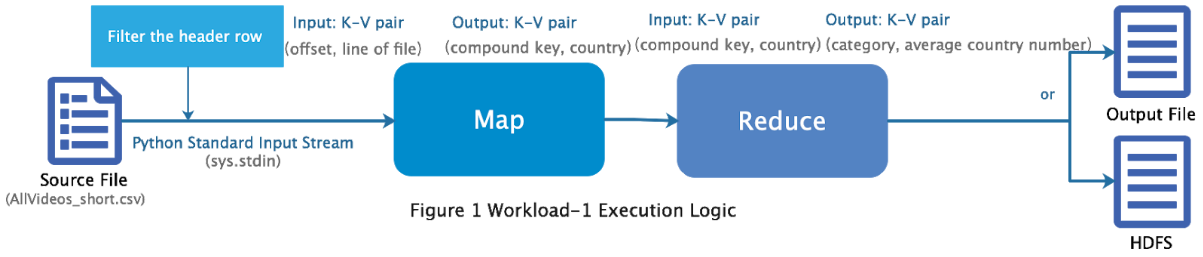

Figure 1 Workload-1 Execution Logic

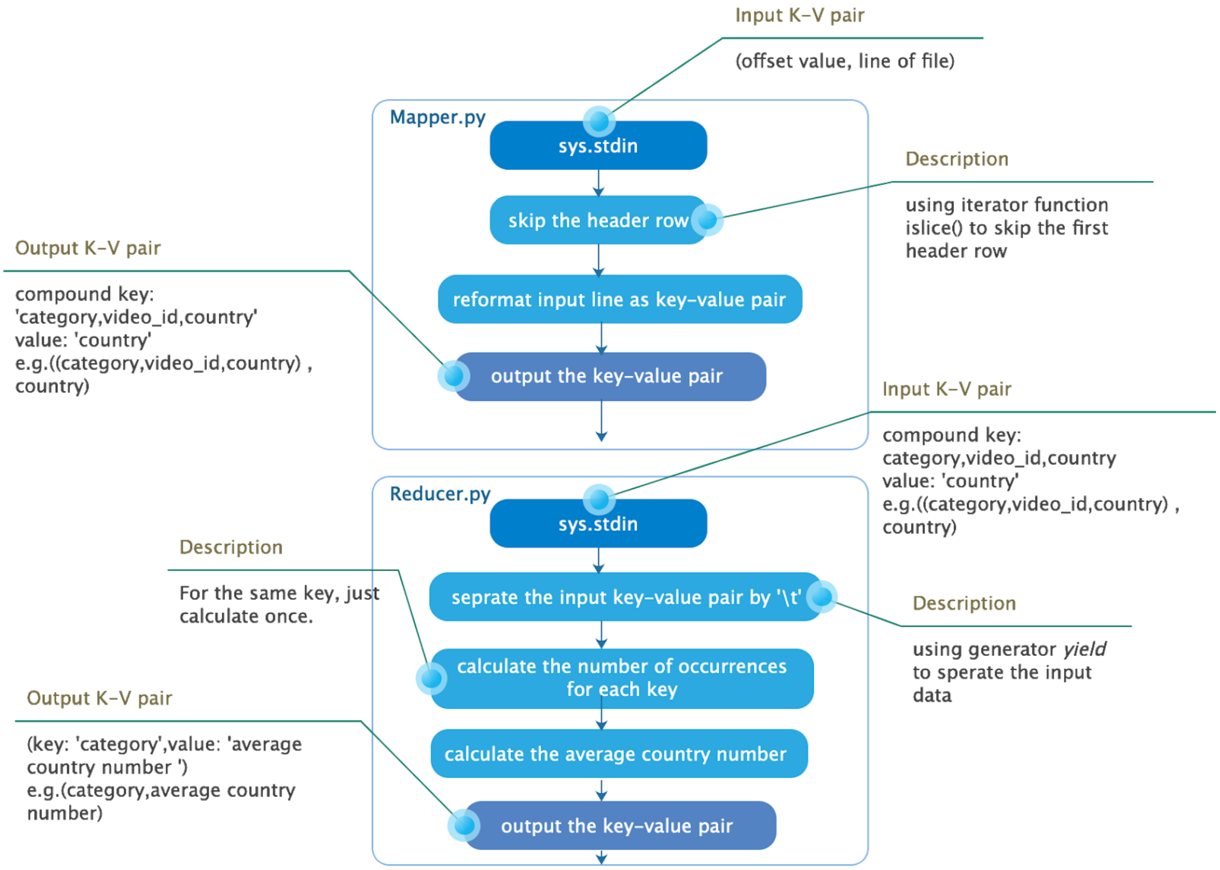
Input K-V pair
(offset value, line of file)
Mapper.py
sys.stdin
skip the header row
reformat input line as key-value pair
output the key-value pair
Description
using iterator function islice() to skip the first header row
Output K-V pair
compound key: 'category,video_id,country'
value: 'country'
e.g.((category,video_id,country) , country)
Input K-V pair
compound key: category,video_id,country
value: 'country'
e.g.((category,video_id,country) , country)
Reducer.py
sys.stdin
seprate the input key-value pair by '\t'
calculate the number of occurrences for each key
calculate the average country number
output the key-value pair
Description
For the same key, just calculate once.
Description
using generator yield to sperate the input data
Output K-V pair
(key: 'category',value: 'average country number ')
e.g.(category,average country number)

Figure 2 Computing Graph for MapReduce for Workload-1

Description:

The picture above shows the whole execution process for the workload-1 (*Category and Trending Correlation*). The steps are the following statement.

step1: read the data from the source file (*AllVideo_short.csv*) by python standard input stream.

step2: skip the header row by using python iterator *islice()* (which is more efficient than using *if* clause in the loop)(Amos, 2018), such as:

```
#ignore the first row which is the headers
for line in islice(file, 1, None):
```

step3: the mapper is responsible for reformat the input lines from the csv file as the key-value pair. In this stage, the unwanted attributes will be removed, and the output pair will be formatted as: (key:(category,video_id,country),value:country).
step4: the reducer receive the Key-Value pair formed from mapper. Then, calculating the average country number for each category. And outputting the computation result into HDFS or target file.

3.Workload-Controversial Video Identification

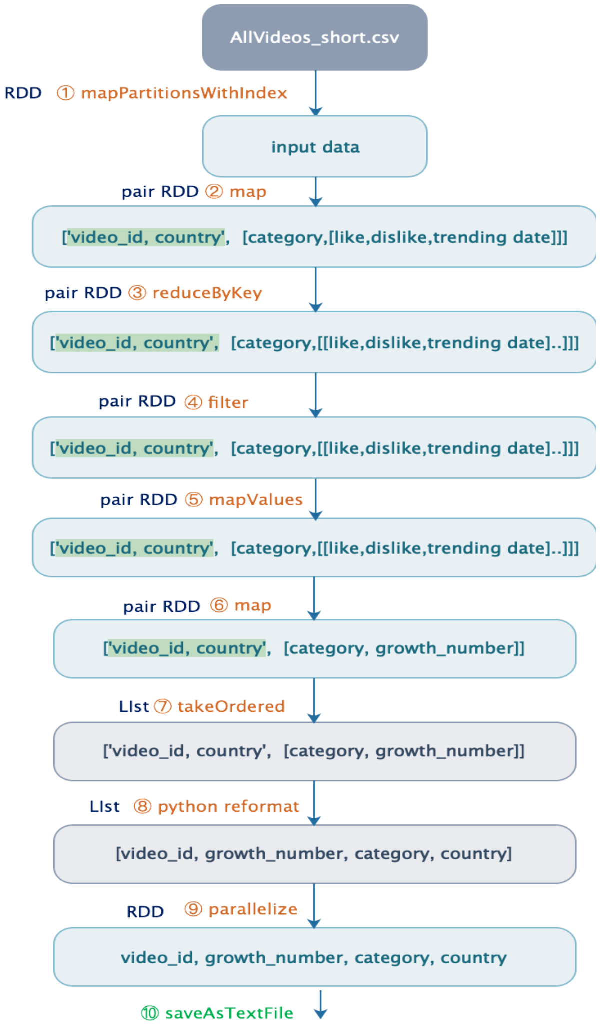
AllVideos_short.csv
RDD ① mapPartitionsWithIndex
input data
pair RDD ② map
['video_id, country', [category,[like,dislike,trending date]]]
pair RDD ③ reduceByKey
['video_id, country', [category,[[like,dislike,trending date]..]]]
pair RDD ④ filter
['video_id, country', [category,[[like,dislike,trending date]..]]]
pair RDD ⑤ mapValues
['video_id, country', [category,[[like,dislike,trending date]..]]]
pair RDD ⑥ map
['video_id, country', [category, growth_number]]
LIst ⑦ takeOrdered
['video_id, country', [category, growth_number]]
LIst ⑧ python reformat
[video_id, growth_number, category, country]
RDD ⑨ parallelize
video_id, growth_number, category, country
⑩ saveAsTextFile

Figure 3 Computing Graph for Spark for Workload-2

Description:
stage 1: after reading the content from csv file, using *mapPartitionsWithIndex* to remove the first row which shows headers and return RDD containing all attributes from the input file.
stage 2: using *map* to organize the selected data as the key-value pair, this stage will remove the unwanted attributes. Then return the RDD pair.
stage 3: using *reduceByKey* to define the first category of a video as its finally category and merge the number of like and dislike for the same key.
stage 4: in order to further calculate the growth number for each key, in this stage the videos that have less than two records(like and dislike records) are removed.
stage 5: using *mapValues* combines *sorted()* to sort the trending date of with ascending order.
stage 6: using map to calculate the each key's growth number which is the difference of like and dislike records between the first and second records.
stage 7: select the result with the largest 10 growth number, and return a list.
stage 8: reorganize the format for the result list as the outpout format.
stage 9: transforming the data type of the result from list to RDD.
stage 10:this action will save the result RDD to file system,such as HDFS.

References

Amos, D. (2018). *Itertools in Python 3, By Example – Real Python*. Retrieved from https://realpython.com/python-itertools/
Rathbone, M. (2019). *Hadoop MapReduce Advanced Python Join Tutorial with Example Code.* Retrieved from https://blog.matthewrathbone.com/2016/02/09/python-tutorial.html