Assignment 8

Map-Reduce

Computing Lab (II)
3rd Mar 2020

This assignment is on map-reduce, which is a distributed and scalable way of extracting/mining required information from multiple datasets stored on multiple servers. Follow the tutorial to understand how you can design mapper and reducer for specific queries/operations.

Tutorial on map-reduce
You can start with a simple word count problem. Say, we have a text file and we want to count the frequency of occurrence of each word. The tutorial below explains how to solve this problem using a map-reduce algorithm.
Tutorial References :
http://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/
Next you can also look into the following tutorial for slightly harder query (tf-idf scores)
https://www.tutorialspoint.com/map_reduce/map_reduce_tutorial.pdf

Tasks

1. Study HDFS and MapReduce.
2. Write the necessary Mapper and Reducer routines to implement the queries mentioned in later sections. Write the code for these queries in python.
3. Write the routine to print results of queries to a text file (ensure the asked output-format).

(Questions will be asked in viva-voce. Prepare well !!!.)

Dataset
We will be using Amazon ratings dataset. Please download the dataset from the below link.
https://drive.google.com/drive/folders/1FmMtF_XfBLM82sz0OkrYtTGuQU22XFcM?usp=sharing
There are 2 different files. You have to extract required information from those files. Below are the sample data **(python dictionaries)** available in the files. (You can use the **eval()** to easily read data from the files)

1. Reviews file (**reviews.json.gz**)


```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays the piano.  He is having a wonderful time playing these old hymns.  The music  is at times hard to read because we think the book was published for singing from more than playing from.  Great purchase though!",
```

```
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```


where

- reviewerID - ID of the reviewer, e.g. A2SUAM1J3GNN3B
- asin - ID of the product, e.g. 0000013714
- reviewerName - name of the reviewer
- helpful - helpfulness rating of the review, e.g. ⅔
- reviewText - text of the review
- overall - rating of the product
- summary - summary of the review
- unixReviewTime - time of the review (unix time)
- reviewTime - time of the review (raw)

2.Items file (**items.json.gz**)


```
{
  "asin": "0000031852",
  "title": "Girls Ballet Tutu Zebra Hot Pink",
  "description":"This is real vanilla extract made with only 3 premium ingredients. GMO
free,.....",
  "price": 3.17,
  "imUrl": "http://ecx.images-amazon.com/images/I/51fAmVkTbyL._SY300_.jpg",
  "related":
  {
    "also_bought": ["B00JHONN1S"],
    "also_viewed": ["B002BZX8Z6"],
    "bought_together": ["B002BZX8Z6"]
  },
  "salesRank": {"Toys & Games": 211836},
  "brand": "Coxlures",
  "categories": [["Sports & Outdoors", "Other Sports", "Dance"]]
}
```


where

- asin - ID of the product, e.g. 0000031852
- title - name of the product
- price - price in US dollars (at time of crawl)
- imUrl - url of the product image
- related - related products (also bought, also viewed, bought together, buy after viewing)
- salesRank - sales rank information

- brand - brand name
- categories - list of categories the product belongs to

Sample code to read the files

```
import gzip
fp=gzip.open("reviews.json.gz")
for line in fp:
   review=eval(line)
```

Queries

The queries are to be implemented in the Mapper and Reducer phases. Some of them may give empty results. You need to implement these following queries in this assignment.

1. Find all the users (user-ids) who have rated at least 10 products. **[10]**
2. Find the item (item-id) with the highest number of 4 or 5 stars ratings. **[10]**
3. Find all the items (item-ids) which cost more than 3 USD and have appeared in "related"-->"also_viewed" section of at least 5 of the items. **[10]**
4. Find top-10 users (user-ids) who have the highest number of reviews with length more than 20 characters. **[10]**
5. Find all the items (item-ids) which cost more than 2 USD and have at least 5 reviews with 5 star rating. (Hint: You will have to use both the input files. In <key,value> pairs, the key can also be a tuple too.) **[20]**

How to run and test your code:

Since we do not have access to a Hadoop cluster, we will be testing our codes on a Linux system as follows:

```
cat input.json | python mapper.py | sort | python reducer.py
```

Or just `python mapper.py | sort | python reducer.py`

Explanation on the above commands:

1. "cat" is a linux command to print the contents of a file on console.
2. The pipe operator (|) directs the output of the previous command to the next command.]
3. "sort" is a linux command to sort the input lexicographically.

Deliverables: Python codes (mapper and reducer), result.txt, readme+makefile

Evaluation Scheme

Results: 60 marks
Coding Style: 10 marks
Viva voce: 30 marks

Important Instructions

1. Submission Rule: Python code for above functionalities must be compressed as **.gz** (gzip) and named "**A8_<RollNo>.gz"**. For each of the query, make a directory named "**Query<no.>"**. Your files must be inside the respective directories. Strictly adhere to this naming convention. Submissions not following the above guidelines will attract penalties.
2. Plagiarism Rule: If your code matches (more than 50%) with another student's code, all those students whose codes match will be awarded with zero marks (may be with -ve marks too depending on the situation) without any evaluation. Therefore, it is your responsibility to ensure that neither you copy anyone's code nor anyone is able to copy yours.
3. Code error: If your code doesn't run or gives error while running, you will be awarded with zero mark.