3116 – Lab Distributed Data Analytics – Group 2

Exercise Sheet 9

Yuvaraj Prem Kumar

303384, premyu@uni-hildsheim.de

Part 1: Preparing your Hadoop infrastructure

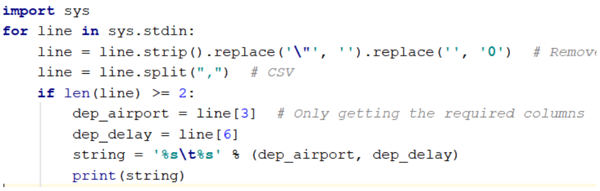
The text details a long tutorial for setting up Hadoop in a personal computing environment. It will be a simulated HDFS. However, I was not able to install Hadoop even after trying many times, due to system conflicts and various errors. Hence for this lab, I am using a VM provided by Cloudera on the Cloudera Hadoop Distribution[3]. I had no choice so for this last resort method.

Part 2: Analysis of airport efficiency with Map Reduce

We can download the dataset as per the instructions, into a csv file. This is a fairly straightforward task, to calculate the average, maximum, and minimum departure delay.

Mapper step:

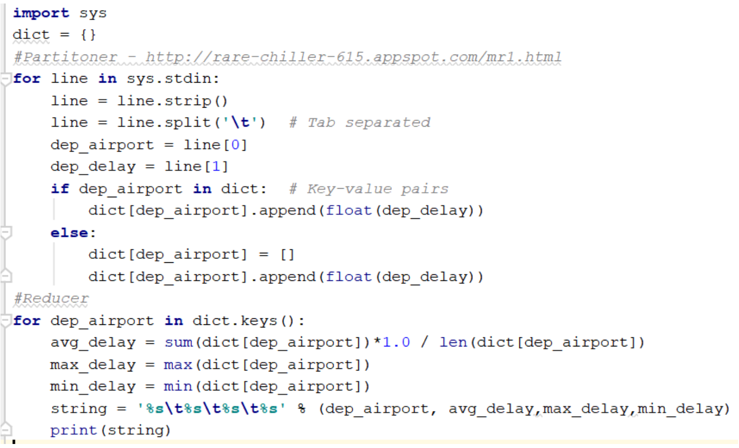
In the file mapper.py, we read line by line via system standard input. Then we have line operations for some pre-processing. The main point here is to extract the required columns {departure airport} & {departure delay}.



```
import sys
for line in sys.stdin:
    line = line.strip().replace('\"', '').replace('', '0')  # Remov
    line = line.split(",")  # CSV
    if len(line) >= 2:
        dep_airport = line[3]  # Only getting the required columns
        dep_delay = line[6]
        string = '%s\t%s' % (dep_airport, dep_delay)
        print(string)
```

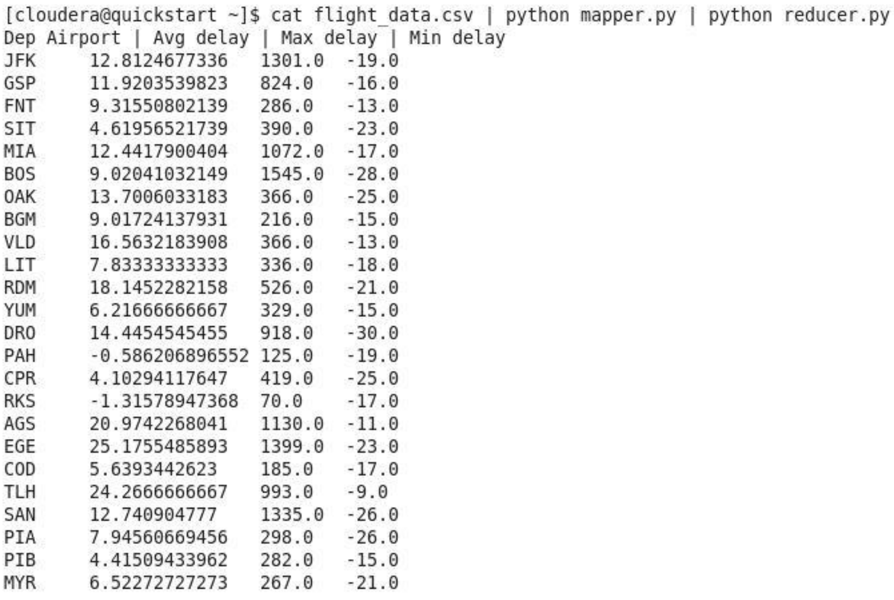
Reducer step:

Here we read the output of mapper.py into a python dictionary. This is for the key-value pairs of {departure airport,departure delay}. The we can simply calculate the required values.



```python
import sys
dict = {}
#Partitoner - http://rare-chiller-615.appspot.com/mr1.html
for line in sys.stdin:
    line = line.strip()
    line = line.split('\t')   # Tab separated
    dep_airport = line[0]
    dep_delay = line[1]
    if dep_airport in dict:   # Key-value pairs
        dict[dep_airport].append(float(dep_delay))
    else:
        dict[dep_airport] = []
        dict[dep_airport].append(float(dep_delay))
#Reducer
for dep_airport in dict.keys():
    avg_delay = sum(dict[dep_airport])*1.0 / len(dict[dep_airport])
    max_delay = max(dict[dep_airport])
    min_delay = min(dict[dep_airport])
    string = '%s\t%s\t%s\t%s' % (dep_airport, avg_delay,max_delay,min_delay)
    print(string)
```

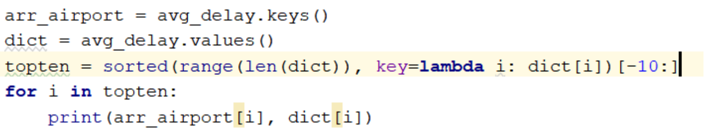
Sample output:



```
[cloudera@quickstart ~]$ cat flight_data.csv | python mapper.py | python reducer.py
Dep Airport | Avg delay | Max delay | Min delay
JFK     12.8124677336   1301.0  -19.0
GSP     11.9203539823   824.0   -16.0
FNT     9.31550802139   286.0   -13.0
SIT     4.61956521739   390.0   -23.0
MIA     12.4417900404   1072.0  -17.0
BOS     9.02041032149   1545.0  -28.0
OAK     13.7006033183   366.0   -25.0
BGM     9.01724137931   216.0   -15.0
VLD     16.5632183908   366.0   -13.0
LIT     7.83333333333   336.0   -18.0
RDM     18.1452282158   526.0   -21.0
YUM     6.21666666667   329.0   -15.0
DRO     14.4454545455   918.0   -30.0
PAH     -0.586206896552 125.0   -19.0
CPR     4.10294117647   419.0   -25.0
RKS     -1.31578947368  70.0    -17.0
AGS     20.9742268041   1130.0  -11.0
EGE     25.1755485893   1399.0  -23.0
COD     5.6393442623    185.0   -17.0
TLH     24.2666666667   993.0   -9.0
SAN     12.740904777    1335.0  -26.0
PIA     7.94560669456   298.0   -26.0
PIB     4.41509433962   282.0   -15.0
MYR     6.52272727273   267.0   -21.0
```

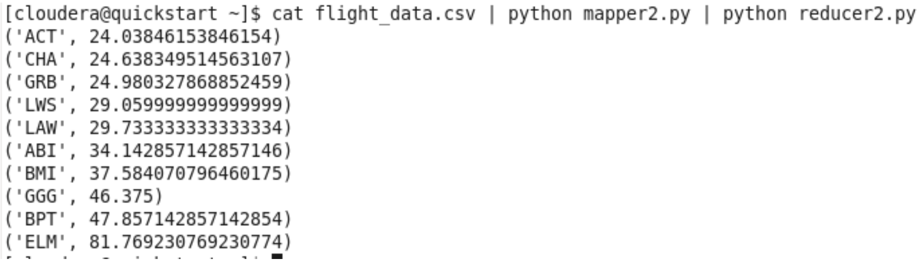
To get the ranking list that contains top 10 airports by their average Arrival delay, we use the same mapper.py; except here we just take columns 'arrival airport' and 'arrival delay'. Reducer.py is mostly the same as the first part, firstly calculate the average delay using same formula as above.

However now we should 'order by' as per SQL. Since it's a list, we can use python sort. Apply lambda function, to get the top 10 rows – negative count of 10 from the dictionary index.



```python
arr_airport = avg_delay.keys()
dict = avg_delay.values()
topten = sorted(range(len(dict)), key=lambda i: dict[i])[-10:]
for i in topten:
    print(arr_airport[i], dict[i])
```

Then we get the output as below:



```
[cloudera@quickstart ~]$ cat flight_data.csv | python mapper2.py | python reducer2.py
('ACT', 24.03846153846154)
('CHA', 24.638349514563107)
('GRB', 24.980327868852459)
('LWS', 29.059999999999999)
('LAW', 29.733333333333334)
('ABI', 34.142857142857146)
('BMI', 37.584070796460175)
('GGG', 46.375)
('BPT', 47.857142857142854)
('ELM', 81.769230769230774)
```

References:

1. http://rare-chiller-615.appspot.com/mr1.html
2. https://stackoverflow.com/questions/30043212/mapreduce-python-how-to-sort-reducer-output-for-top-n-list
3. https://www.cloudera.com/downloads/quickstart_vms/5-13.html).