

## Fahad Fiaz – (303141) – G2

### System Info:

Processor	i7-5500U , 2.40GHz
Cores	4
Operating system	Windows 64 Bit
Ram	8GB
Programming Language	Python 3.7.7

### Q1:

Making directory in HDFS

```
PS E:\hadoop-2.7.0\hadoop-2.7.0> hadoop fs -mkdir /wordcount_ex/input
```

Copying text file to HDFS

```
PS E:\hadoop-2.7.0\hadoop-2.7.0> hadoop fs -put -f E:\world_count\input_data\words.txt /wordcount_ex/input
PS E:\hadoop-2.7.0\hadoop-2.7.0>
```

Running the prebuild World count program to count occurrence of words.

```
PS E:\hadoop-2.7.0\hadoop-2.7.0> bin\yarn jar E:\hadoop-2.7.0\hadoop-2.7.0\share\hadoop\mapreduce\hadoop-mapreduce-examples-2.7.0.jar wordcount /wordcount_ex/input/words.txt wordcount_less_text
```

```

PS E:\hadoop-2.7.0\hadoop-2.7.0> hdfs dfs -cat wordCount_less
Author: 1
Brothers 2
EBook 1
Edgar 1
Edwardes 1
Fairy 2
Grimm 2
Grimms 1
Gutenberg 2
License 1
Marian 1
Project 2
Tales 1
Tales, 1
Taylor 1
The 3
This 1
Title: 1
Translator: 1
You 1
almost 1
and 2
anyone 1
anywhere 1
at 2
away 1
by 1
copy 1
cost 1
eBook 2
for 1
give 1
included 1
is 1
it 2
it, 1
may 1
no 2
of 3
online 1

```

/ Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	fahad	supergroup	0 B	6/15/2020, 5:19:51 AM	0	0 B	<a href="#">sample</a>
drwxr-xr-x	fahad	supergroup	0 B	6/15/2020, 5:55:33 AM	0	0 B	<a href="#">user</a>
drwxr-xr-x	fahad	supergroup	0 B	6/15/2020, 5:42:32 AM	0	0 B	<a href="#">wordcount_ex</a>

## **Q2:**

### **Mapper:**

It will read data from STDIN, Split rows and output specific rows to STDOUT

```
for line in sys.stdin:
    line = line.split(',')
    try:
        print((line[3], line[6], line[8]))
    except Exception as e:
        print("Error: ", e)
```

### **Reducer:**

It will read the results of mapper.py from STDIN (the output format of mapper.py and the expected input format of reducer.py must match).

Since the value received from STDIN is in string format so first we need to convert this string tuple to tuple.

```
line = literal_eval(line) # convert string-tuple to tuple
try:
    line = (line[0], float(line[1]), float(line[2]))
except Exception as e:
    line = None
```

Following lines check whether if we are parsing a specific airport for 1<sup>st</sup> time or Not. If we are parsing it for first time it simply saves the airport data in “final” dictionary with airport name as key. If the airport name in new row is already in final dictionary then we compare current row data with already previous save data in final dictionary to find maximum, minimum, and average departure delay for each airport.

In “final” dictionary, key is airport names and values are tuple. In each tuple first column represent airport name, 2<sup>nd</sup> represent minimum, 3<sup>rd</sup> represent maximum, 4<sup>th</sup> represent count of specific airport in dataset and 5<sup>th</sup> represent total arrival delay for specific airport.

```
if line:
    if line[0] in final.keys():
        old_val = final[line[0]]
        airport = old_val[0]
        low = old_val[1]
        high = old_val[2]
        total = old_val[3] + line[1]
        count = old_val[4] + 1
        total_arrival_delay = old_val[5] + line[2]
```

```

        if line[1] < old_val[1]:
            low = line[1]
        if line[1] > old_val[2]:
            high = line[1]

    final[line[0]] = (airport, low, high, total, count, total_arrival_delay)

else:
    # airport_Names.append(line[0])
    final[line[0]] = (line[0], 9999, -9999, 0, 1, 0)

```

## Show the final results:

```

for k,v in final.items():
    print("Airport Name:{},maximum departure delay:{},maximum departure delay:{},
    averaga departure delay:{}, averaga arrival
    delay:{}".format(v[0],v[1],v[2],v[3]/v[4],v[5]/v[4]))

```

## Running commands:

```

Windows PowerShell
PS E:\hadoop-2.7.0\hadoop-2.7.0\share\hadoop\tools\lib> hadoop fs -mkdir /lab5
PS E:\hadoop-2.7.0\hadoop-2.7.0\share\hadoop\tools\lib> hadoop fs -put -f E:\Lab5_data\airline_Data.csv /lab5
PS E:\hadoop-2.7.0\hadoop-2.7.0\share\hadoop\tools\lib> hadoop fs -ls /lab5
Found 1 items
-rw-r--r-- 3 fahad supergroup 27646481 2020-06-15 09:47 /lab5/airline_Data.csv
PS E:\hadoop-2.7.0\hadoop-2.7.0\share\hadoop\tools\lib> hadoop jar E:\hadoop-2.7.0\hadoop-2.7.0\share\hadoop\tools\lib\hadoop-streaming-
2.7.0.jar -file E:\programs\mapper.py -mapper "python mapper.py" -file E:\programs\reducer.py -reducer "python reducer.py" -input /l
ab5/airline_Data.csv -output /lab5/output1.txt
20/06/15 09:49:46 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [E:\programs\mapper.py, E:\programs\reducer.py] [] C:\Users\fahad\AppData\Local\Temp\streamjob7111810228542283141.jar tm
pdir=null
20/06/15 09:49:47 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id
20/06/15 09:49:47 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=
20/06/15 09:49:47 INFO jvm.JvmMetrics: Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
20/06/15 09:49:48 INFO mapred.FileInputFormat: Total input paths to process : 1
20/06/15 09:49:48 INFO mapreduce.JobSubmitter: number of splits:1
20/06/15 09:49:48 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1058155937_0001
20/06/15 09:49:48 INFO mapred.LocalDistributedCacheManager: Localized file:/E:/programs/mapper.py as file:/tmp/hadoop-fahad/mapred/loca
l/1592239788344/mapper.py
20/06/15 09:49:48 INFO mapred.LocalDistributedCacheManager: Localized file:/E:/programs/reducer.py as file:/tmp/hadoop-fahad/mapred/loca
l/1592239788345/reducer.py
20/06/15 09:49:48 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
20/06/15 09:49:48 INFO mapred.LocalJobRunner: OutputCommitter set in config null
20/06/15 09:49:48 INFO mapreduce.Job: Running job: job_local1058155937_0001
20/06/15 09:49:48 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapred.FileOutputCommitter
20/06/15 09:49:48 INFO output.FileOutputCommitter: File output committer Algorithm version is 1
20/06/15 09:49:48 INFO mapred.LocalJobRunner: Waiting for map tasks
20/06/15 09:49:48 INFO mapred.LocalJobRunner: Starting task: attempt_local1058155937_0001_m_000000_0
20/06/15 09:49:48 INFO output.FileOutputCommitter: File output committer Algorithm version is 1
20/06/15 09:49:48 INFO util.ProcfsBasedProcessTree: ProcfsBasedProcessTree currently is supported only on Linux.
20/06/15 09:49:48 INFO mapred.Task: Using ResourceCalculatorProcessTree : org.apache.hadoop.yarn.util.WindowsBasedProcessTree@4de72242
20/06/15 09:49:49 INFO mapred.MapTask: Processing split: hdfs://0.0.0.0:9000/lab5/airline_Data.csv:0+27646481
20/06/15 09:49:49 INFO mapred.MapTask: numReduceTasks: 1
20/06/15 09:49:49 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
20/06/15 09:49:49 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
20/06/15 09:49:49 INFO mapred.MapTask: soft limit at 83886080
20/06/15 09:49:49 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
20/06/15 09:49:49 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
20/06/15 09:49:49 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
20/06/15 09:49:49 INFO streaming.PipeMapRed: PipeMapRed exec [python, mapper.py]

```

Windows PowerShell

```
File System Counters
  FILE: Number of bytes read=27878696
  FILE: Number of bytes written=42382486
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=55292962
  HDFS: Number of bytes written=49052
  HDFS: Number of read operations=13
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=4
Map-Reduce Framework
  Map input records=450018
  Map output records=450018
  Map output bytes=13036676
  Map output materialized bytes=13936718
  Input split bytes=93
  Combine input records=0
  Combine output records=0
  Reduce input groups=184440
  Reduce shuffle bytes=13936718
  Reduce input records=450018
  Reduce output records=297
  Spilled Records=900036
  Shuffled Maps =1
  Failed Shuffles=0
  Merged Map outputs=1
  GC time elapsed (ms)=6
  Total committed heap usage (bytes)=578289664
shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=27646481
File Output Format Counters
  Bytes Written=49052
20/06/15 09:50:05 INFO streaming.StreamJob: output directory: /lab5/output1.txt
```

## Output: Only part of output shown

Windows PowerShell

```
Airport Name:"SUN",maximum departure delay:-18.0 ,maximum departure delay:350.0, averaga departure delay:22.378048780487806, averaga arrival delay:16.9390243902439
Airport Name:"SWF",maximum departure delay:-20.0 ,maximum departure delay:137.0, averaga departure delay:4.372881355932203, averaga arrival delay:-5.271186440677966
Airport Name:"SYR",maximum departure delay:-19.0 ,maximum departure delay:1062.0, averaga departure delay:14.76027397260274, averaga arrival delay:2.7739726027397262
Airport Name:"TLH",maximum departure delay:-9.0 ,maximum departure delay:993.0, averaga departure delay:25.18867924528302, averaga arrival delay:19.251572327044027
Airport Name:"TPA",maximum departure delay:-27.0 ,maximum departure delay:830.0, averaga departure delay:8.712937542896363, averaga arrival delay:4.587336993822924
Airport Name:"TRI",maximum departure delay:-14.0 ,maximum departure delay:415.0, averaga departure delay:13.352201257861635, averaga arrival delay:1.9371069182389937
Airport Name:"TTN",maximum departure delay:-20.0 ,maximum departure delay:613.0, averaga departure delay:17.089385474860336, averaga arrival delay:6.201117318435754
Airport Name:"TUL",maximum departure delay:-17.0 ,maximum departure delay:381.0, averaga departure delay:4.634446397188049, averaga arrival delay:2.731985940246046
Airport Name:"TUS",maximum departure delay:-21.0 ,maximum departure delay:373.0, averaga departure delay:8.998590556730091, averaga arrival delay:4.603946441155744
Airport Name:"TVC",maximum departure delay:-19.0 ,maximum departure delay:1121.0, averaga departure delay:24.893939393939394, averaga arrival delay:20.97979797979798
Airport Name:"TWF",maximum departure delay:-13.0 ,maximum departure delay:460.0, averaga departure delay:26.170731707317074, averaga arrival delay:27.609756097560975
Airport Name:"TXK",maximum departure delay:-21.0 ,maximum departure delay:215.0, averaga departure delay:10.5625, averaga arrival delay:18.4375
Airport Name:"TYR",maximum departure delay:-7.0 ,maximum departure delay:58.0, averaga departure delay:9.5, averaga arrival delay:9.666666666666666
Airport Name:"TYS",maximum departure delay:-18.0 ,maximum departure delay:477.0, averaga departure delay:11.388785046728971, averaga arrival delay:5.319626168224299
Airport Name:"VLD",maximum departure delay:-13.0 ,maximum departure delay:366.0, averaga departure delay:15.292682926829269, averaga arrival delay:14.902439024390244
Airport Name:"VPS",maximum departure delay:-14.0 ,maximum departure delay:1244.0, averaga departure delay:22.08467741935484, averaga arrival delay:16.149193548387096
Airport Name:"WRG",maximum departure delay:-37.0 ,maximum departure delay:295.0, averaga departure delay:0.4915254237288136, averaga arrival delay:3.2711864406779663
Airport Name:"XNA",maximum departure delay:-20.0 ,maximum departure delay:840.0, averaga departure delay:18.746305418719214, averaga arrival delay:10.300492610837438
Airport Name:"YAK",maximum departure delay:-48.0 ,maximum departure delay:79.0, averaga departure delay:-17.041666666666668, averaga arrival delay:-19.145833333333332
Airport Name:"YUM",maximum departure delay:-15.0 ,maximum departure delay:329.0, averaga departure delay:6.277310924369748, averaga arrival delay:4.647058823529412
PS E:\hadoop-2.7.0\hadoop-2.7.0\share\hadoop\tools\lib>
```

To calculate the list of top 10 airports by their average Arrival delay we divide total arrival delay with total occurrence of specific airport in dataset. Then find the list of top 10 airports by their average Arrival delay.

```
Arrival_Delay_Average={}
for k,v in final.items():
    Arrival_Delay_Average[v[0]]=v[5]/v[4]

d = Counter(Arrival_Delay_Average)
for k, v in d.most_common(10):
    print("Airport Name:{}, averaga arrival delay:{}".format(k,v))
```

```
PS E:\hadoop-2.7.0\hadoop-2.7.0\share\hadoop\tools\lib> hadoop dfs -cat /lab5/output5.txt/*
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
Airport Name:"LAW", averaga arrival delay:85.25925925925925
Airport Name:"GGG", averaga arrival delay:64.0
Airport Name:"EKO", averaga arrival delay:62.95918367346939
Airport Name:"BPT", averaga arrival delay:46.0
Airport Name:"LWS", averaga arrival delay:42.744680851063826
Airport Name:"ABR", averaga arrival delay:34.95
Airport Name:"ASE", averaga arrival delay:32.87064676616915
Airport Name:"HDN", averaga arrival delay:31.818181818181817
Airport Name:"JAC", averaga arrival delay:29.457575757575757
Airport Name:"ABI", averaga arrival delay:29.40740740740741
PS E:\hadoop-2.7.0\hadoop-2.7.0\share\hadoop\tools\lib>
```