Name: Huyen Nguyen

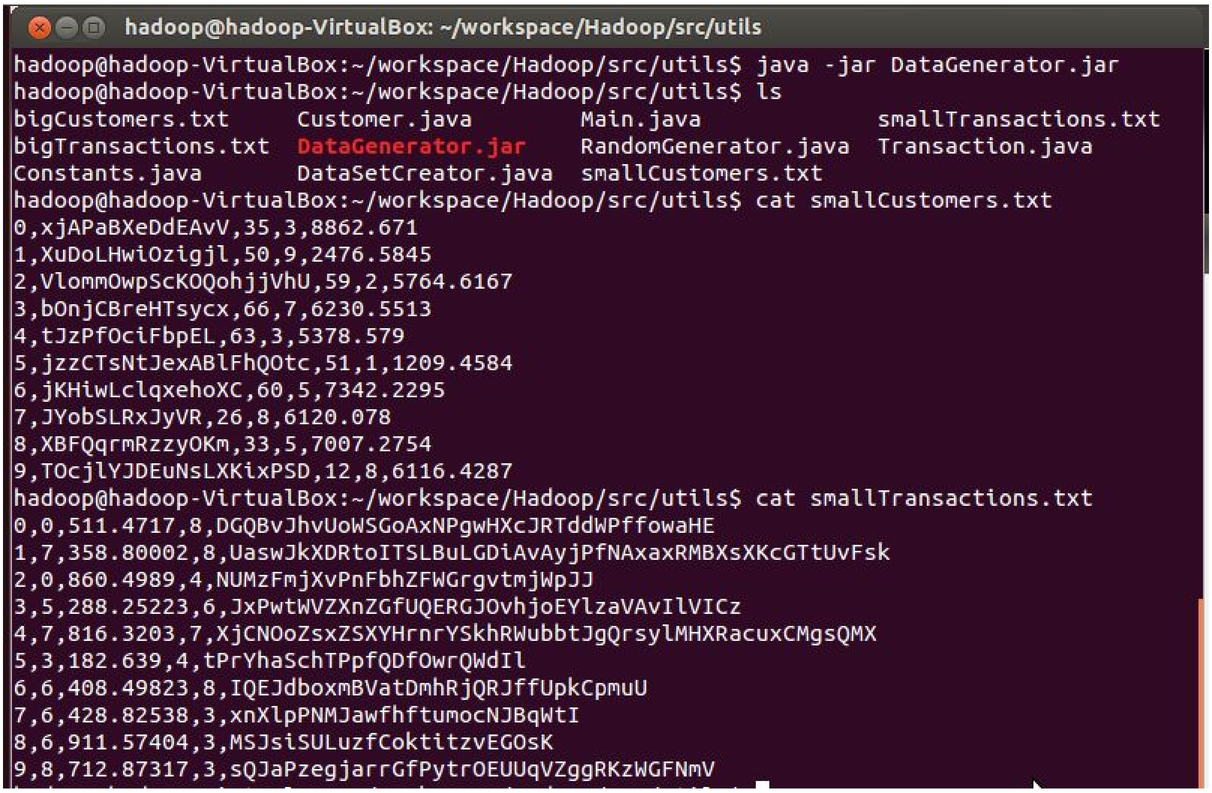
Hadoop

Create datasets programmatically

1.List all **commands** and **actions** needed to run your program and view the output respectively.
`$ java -jar DataGenerator.jar` # run data generator
`$ vi smallCustomers.txt` # view dataset with 10 customers
`$ vi smallTransactions.txt` # view dataset with 10 transactions associated with smallCustomers.txt
`$ vi bigCustomers.txt` # view dataset with 50,000 customers
`$ vi bigTransactions.txt` # view dataset with 10 transactions associated with bigCustomers.txt

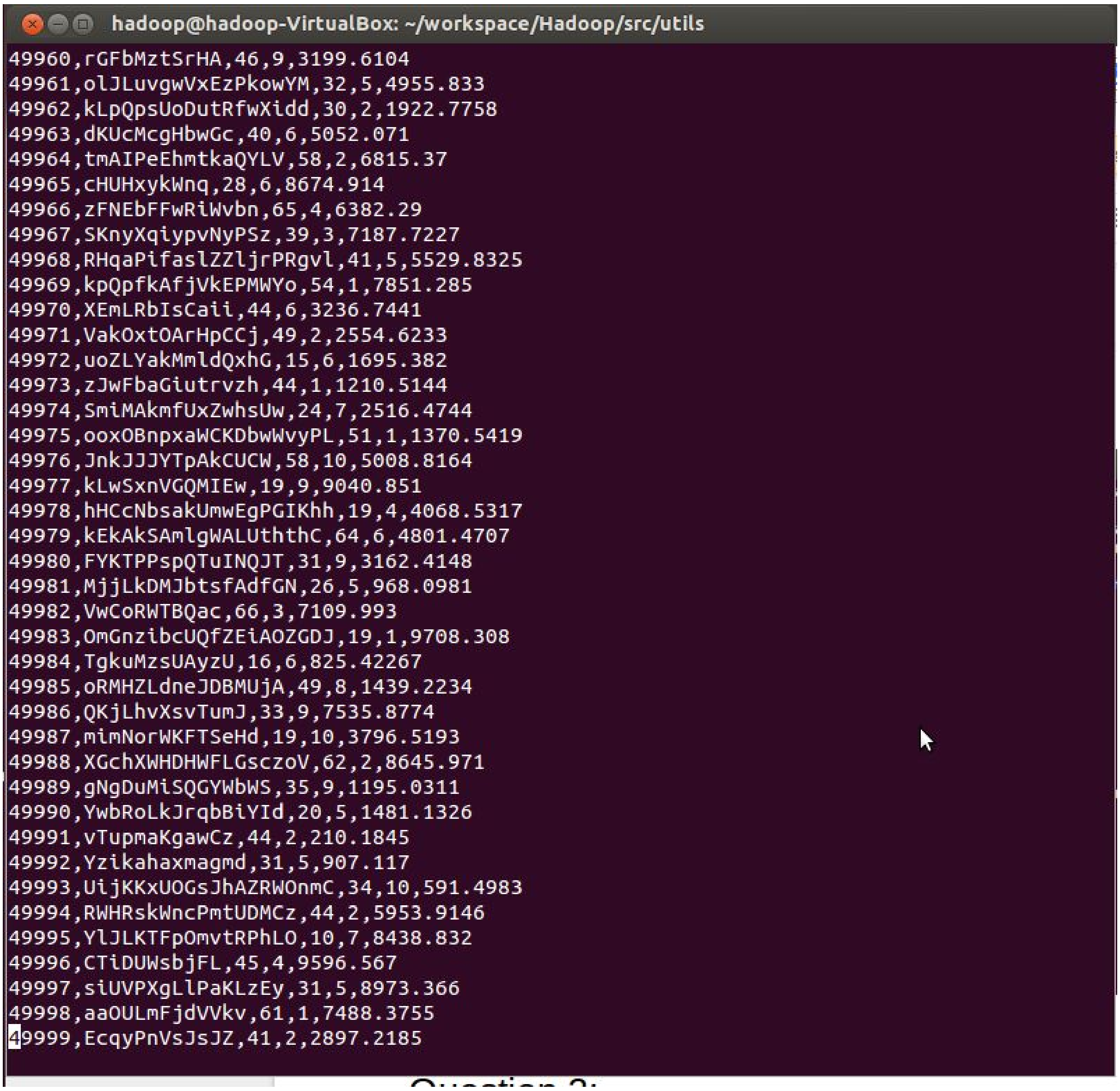
2.If there are any issues with a program, please list them.
3.Include a screen clip of the output and list any observations.

Contents of smallCustomers.txt and smallTransactions.txt



```
hadoop@hadoop-VirtualBox:~/workspace/Hadoop/src/utils$ java -jar DataGenerator.jar
hadoop@hadoop-VirtualBox:~/workspace/Hadoop/src/utils$ ls
bigCustomers.txt     Customer.java        Main.java            smallTransactions.txt
bigTransactions.txt  DataGenerator.jar    RandomGenerator.java  Transaction.java
Constants.java       DataSetCreator.java  smallCustomers.txt
hadoop@hadoop-VirtualBox:~/workspace/Hadoop/src/utils$ cat smallCustomers.txt
0,xjAPaBXeDdEAvV,35,3,8862.671
1,XuDoLHwiOzigjl,50,9,2476.5845
2,VlommOwpScKOQohjjVhU,59,2,5764.6167
3,bOnjCBreHTsycx,66,7,6230.5513
4,tJzPfOciFbpEL,63,3,5378.579
5,jzzCTsNtJexABlFhQOtc,51,1,1209.4584
6,jKHiwLclqxehoXC,60,5,7342.2295
7,JYobSLRxJyVR,26,8,6120.078
8,XBFQqrmRzzyOKm,33,5,7007.2754
9,TOcjlYJDEuNsLXKixPSD,12,8,6116.4287
hadoop@hadoop-VirtualBox:~/workspace/Hadoop/src/utils$ cat smallTransactions.txt
0,0,511.4717,8,DGQBvJhvUoWSGoAxNPgwHXcJRTddWPffowaHE
1,7,358.80002,8,UaswJkXDRtoITSLBuLGDiAvAyjPfNAxaxRMBXsXKcGTtUvFsk
2,0,860.4989,4,NUMzFmjXvPnFbhZFWGrgvtmjWpJJ
3,5,288.25223,6,JxPwtWVZXnZGfUQERGJOvhjoEYlzaVAvIlVICz
4,7,816.3203,7,XjCNOoZsxZSXYHrnrYSkhRWubbtJgQrsylMHXRacuxCMgsQMX
5,3,182.639,4,tPrYhaSchTPpfQDfOwrQWdIl
6,6,408.49823,8,IQEJdboxmBVatDmhRjQRJffUpkCpmuU
7,6,428.82538,3,xnXlpPNMJawfhftumocNJBqWtI
8,6,911.57404,3,MSJsiSULuzfCoktitzvEGOsK
9,8,712.87317,3,sQJaPzegjarrGfPytrOEUUqVZggRKzWGFNmV
```

Contents of bigCustomers.txt



```
49960,rGFbMztSrHA,46,9,3199.6104
49961,olJLuvgwVxEzPkowYM,32,5,4955.833
49962,kLpQpsUoDutRfwXidd,30,2,1922.7758
49963,dKUcMcgHbwGc,40,6,5052.071
49964,tmAIPeEhmtkaQYLV,58,2,6815.37
49965,cHUHxykWnq,28,6,8674.914
49966,zFNEbFFwRiWvbn,65,4,6382.29
49967,SKnyXqiypvNyPSz,39,3,7187.7227
49968,RHqaPifaslZZljrPRgvl,41,5,5529.8325
49969,kpQpfkAfjVkEPMWYo,54,1,7851.285
49970,XEmLRbIsCaii,44,6,3236.7441
49971,VakOxtOArHpCCj,49,2,2554.6233
49972,uoZLYakMmldQxhG,15,6,1695.382
49973,zJwFbaGiutrvzh,44,1,1210.5144
49974,SmiMAkmfUxZwhsUw,24,7,2516.4744
49975,ooxOBnpxaWCKDbwWvyPL,51,1,1370.5419
49976,JnkJJJYTpAkCUCW,58,10,5008.8164
49977,kLwSxnVGQMIEw,19,9,9040.851
49978,hHCcNbsakUmwEgPGIKhh,19,4,4068.5317
49979,kEkAkSAmlgWALUththC,64,6,4801.4707
49980,FYKTPPspQTuINQJT,31,9,3162.4148
49981,MjjLkDMJbtsfAdfGN,26,5,968.0981
49982,VwCoRWTBQac,66,3,7109.993
49983,OmGnzibcUQfZEiAOZGDJ,19,1,9708.308
49984,TgkuMzsUAyzU,16,6,825.42267
49985,oRMHZLdneJDBMUjA,49,8,1439.2234
49986,QKjLhvXsvTumJ,33,9,7535.8774
49987,mimNorWKFTSeHd,19,10,3796.5193
49988,XGchXWHDHWFLGsczoV,62,2,8645.971
49989,gNgDuMiSQGYWbWS,35,9,1195.0311
49990,YwbRoLkJrqbBiYId,20,5,1481.1326
49991,vTupmaKgawCz,44,2,210.1845
49992,Yzikahaxmagmd,31,5,907.117
49993,UijKKxUOGsJhAZRWOnmC,34,10,591.4983
49994,RWHRskWncPmtUDMCz,44,2,5953.9146
49995,YlJLKTFpOmvtRPhLO,10,7,8438.832
49996,CTiDUWsbjFL,45,4,9596.567
49997,siUVPXgLlPaKLzEy,31,5,8973.366
49998,aaOULmFjdVVkv,61,1,7488.3755
49999,EcqyPnVsJsJZ,41,2,2897.2185
```

Contents of bigTransactions.txt

Upload the datasets into the Hadoop File System

1.List all **commands** and **actions** needed to run your program and view the output respectively.

```
$ hadoop fs -put smallCustomers.txt /user/hadoop/input/smallCustomers.txt
```

# upload smallCustomers.txt to HDFS

```
$ hadoop fs -put smallTransactions.txt
/user/hadoop/input/smallTransactions.txt
$ hadoop fs -put bigCustomers.txt /user/hadoop/input/bigCustomers.txt
$ hadoop fs -put bigTransactions.txt
/user/hadoop/input/bigTransactions.txt
```

2.If there are any issues with a program, please list them.

3.Include a screen clip of the output and list any observations.

Queries

Query 1

1.List all **commands** and **actions** needed to run your program and view the output respectively.

```
$ hadoop jar Query1.jar /user/hadoop/input/bigCustomers.txt
/user/hadoop/output/query1Big
$ hadoop fs -cat /user/hadoop/output/query1Big/part-m-00000 >
query1Output.txt
$ vi query1Output.txt
```

2.If there are any issues with a program, please list them.

3.Include a screen clip of the output and list any observations.

Note: the result consists of customer IDs whose country code is between 2 and 6 (inclusive)

Query 2

Without Combiner

1.List all **commands** and **actions** needed to run your program and view the output respectively.

```
$ hadoop jar Query2NoCombiner.jar /user/hadoop/input/bigTransactions.txt
/user/hadoop/output/query2BigNoCombiner
$ hadoop fs -cat /user/hadoop/output/query2BigNoCombiner/part-r-00000 >
query2BigNoCombinerOutput.txt
$ vi query2BigNoCombinerOutput.txt
```

2.If there are any issues with a program, please list them.

See discussion on page 11

3.Include a screen clip of the output and list any observations.

With Combiner

1.List all **commands** and **actions** needed to run your program and view the output respectively.

```
$ hadoop jar Query2WithCombiner.jar /user/hadoop/input/bigTransactions.txt
/user/hadoop/output/query2BigWithCombiner
$ hadoop fs -cat /user/hadoop/output/query2BigWithCombiner/part-r-00000 >
query2BigWithCombinerOutput.txt
$ vi query2BigWithCombinerOutput.txt
```

2.If there are any issues with a program, please list them.
See discussion on page 11
3.Include a screen clip of the output and list any observations.

Compare jobs with and without combiners

Without Combiner

Total execution time is 0.828s

```
$ time hadoop jar Query2NoCombiner.jar
/user/hadoop/input/bigTransactions.txt
/user/hadoop/output/query2BigNoCombiner
real    0m46.010s
user    0m0.752s
sys    0m0.076s
```

Total time = 0.752+0.076= 0.828 (see https://stackoverflow.com/questions/556405/what-do-real-user-and-sys-mean-in-the-output-of-time1 for the reason)

Number of records sent to reducer is 5,000,000

Execution times for Map tasks is 0.760s

Total time = 0.656+0.104=0.760

With Combiner

Total execution time is 0.796s

```
$ time hadoop jar Query2WithCombiner.jar
/user/hadoop/input/bigTransactions.txt
/user/hadoop/output/query2BigWithCombiner
real    0m34.971s
user    0m0.728s
sys    0m0.068s
```

Total time = 0.728+0.068=0.796

Number of records sent to reducer is 250,000

Execution times for Map tasks is 0.776s

Total time = 0.668 + 0.108 = 0.776

Discussion

- The total execution time when a combiner is used (0.796s) is less than when a combiner is not used (0.828s). The reason is the combiner processes subsets of data entries, so that at the end, the reducer does not have to handle as much as data entries as what the mapper produces.
- Despite this, the time difference is not that much since we have a single-node cluster, in which the network communication is negligible. However, for a multi-node cluster, the time difference between a job that uses a combiner is expected to be significantly less than the job that does not.
- The number of Reduce Input Records is 5,000,000 without the combiner. This makes sense since the number of Map Output Records is 5,000,000. In contrast, the number of Reduce Input Records is only 250,000 with the combiner. This makes sense since the combiner already does some work on aggregating the results for each worker node.
- The execution time for Map Tasks without the combiner (0.760s) is less than when a combiner is used (0.776s). The reason is besides the basic map tasks, the combiner adds some extra time to the execution time when it is used for aggregating data.
- The total values column for the 2 MapReduce jobs (the screenshots on pages 7 and 8, last columns) are a little different due to the rounding errors in Java when dealing with floating-point numbers.

Query 3

1.List all **commands** and **actions** needed to run your program and view the output respectively.

```
$ hadoop jar Query3.jar /user/hadoop/input/bigCustomers.txt
/user/hadoop/input/bigTransactions.txt /user/hadoop/output/query3Big
$ hadoop fs -cat /user/hadoop/output/query3Big/part-r-00000 >
query3Big.txt
$ vi query3Big.txt
```

2.If there are any issues with a program, please list them.

3.Include a screen clip of the output and list any observations.

Query 4

1.List all **commands** and **actions** needed to run your program and view the output respectively.

```
$ hadoop jar Query4.jar /user/hadoop/input/bigCustomers.txt
/user/hadoop/input/bigTransactions.txt /user/hadoop/output/query4Big
$ hadoop fs -cat /user/hadoop/output/query4Big/part-r-00000 >
query4Big.txt
$ vi query4Big.txt
```

2.If there are any issues with a program, please list them.

3.Include a screen clip of the output and list any observations.