Group Assignment 1
CS 850 4 Big Data

Summer 2015

Coded By

naironics

Q1. On the cloudera image you find a folder called datasets that has a zip file in it called: median_income_CA.csv. It contains a couple of comma separated columns:

Data Location:
https://www.dropbox.com/s/ptuhcti6hn4dezo/median_income_CA.csv?dl=0
GEO.id,Zip,GEO-label,Median Total,Median Owner occupied,Median Renter occupied

We want to use Hadoop and MapReduce to analyze this file. You can do this in Java or python streaming API. Please hand in your results and your code (.java file, or the mapper and reducer .py file)

For each interval of 10 zip codes we want to know the minimum and maximum salary in the table for all those zip code ranges. Observe that there are zip codes like 999HH, please ignore all zip codes that are not integer numbers.

As submission please explain step by step how you accomplished this exercise and along with screenshots, queries and other aspects of analysis steps and code. Be as detail oriented as possible.

Solution :

What this MapReduce program does :

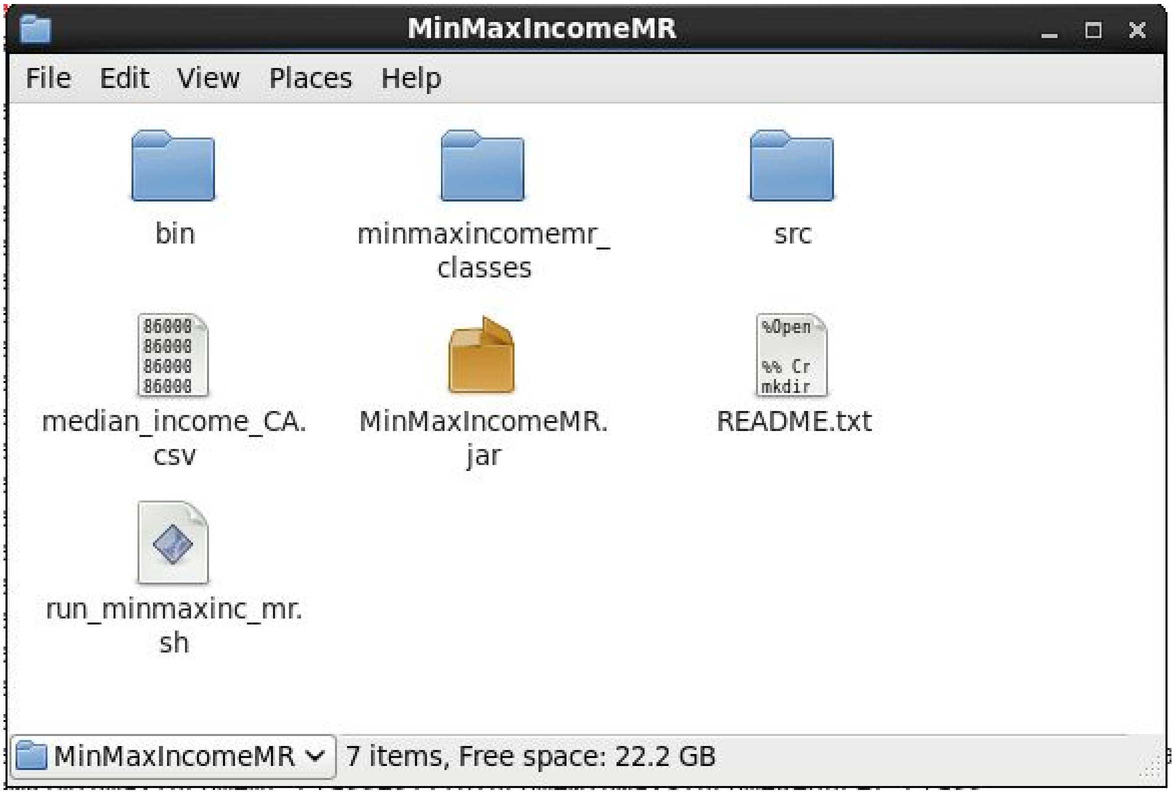
The main java file ZipIncomeMinMax.java, contains all the hadoop job specific logic, which includes the mapper, reducer, checking if a zip code is valid and driver etc.

A Custom Data Type MinMaxIncome.java is created to hold the minimum and maximum values of a zip code range, which is identified by the ten consecutive valid zip codes.

In the Map Stage, records are emitted one by one to the map() tasks, which are then split into fields and required field (zip code in our case) is taken as key, after checking that is a valid zip code, and its min and max values combo are saved as value in the custom data type MinMaxIncome.

In the Reduce Stage, all the values of the zip codes corresponding to the same range are aggregated to find the local minimum and maximum for that range and is written to the output file, which yields our final output part-r-0000 . This output can be later used for human inspection or execution of next job or altogether a new program.

Project Folder Structure :

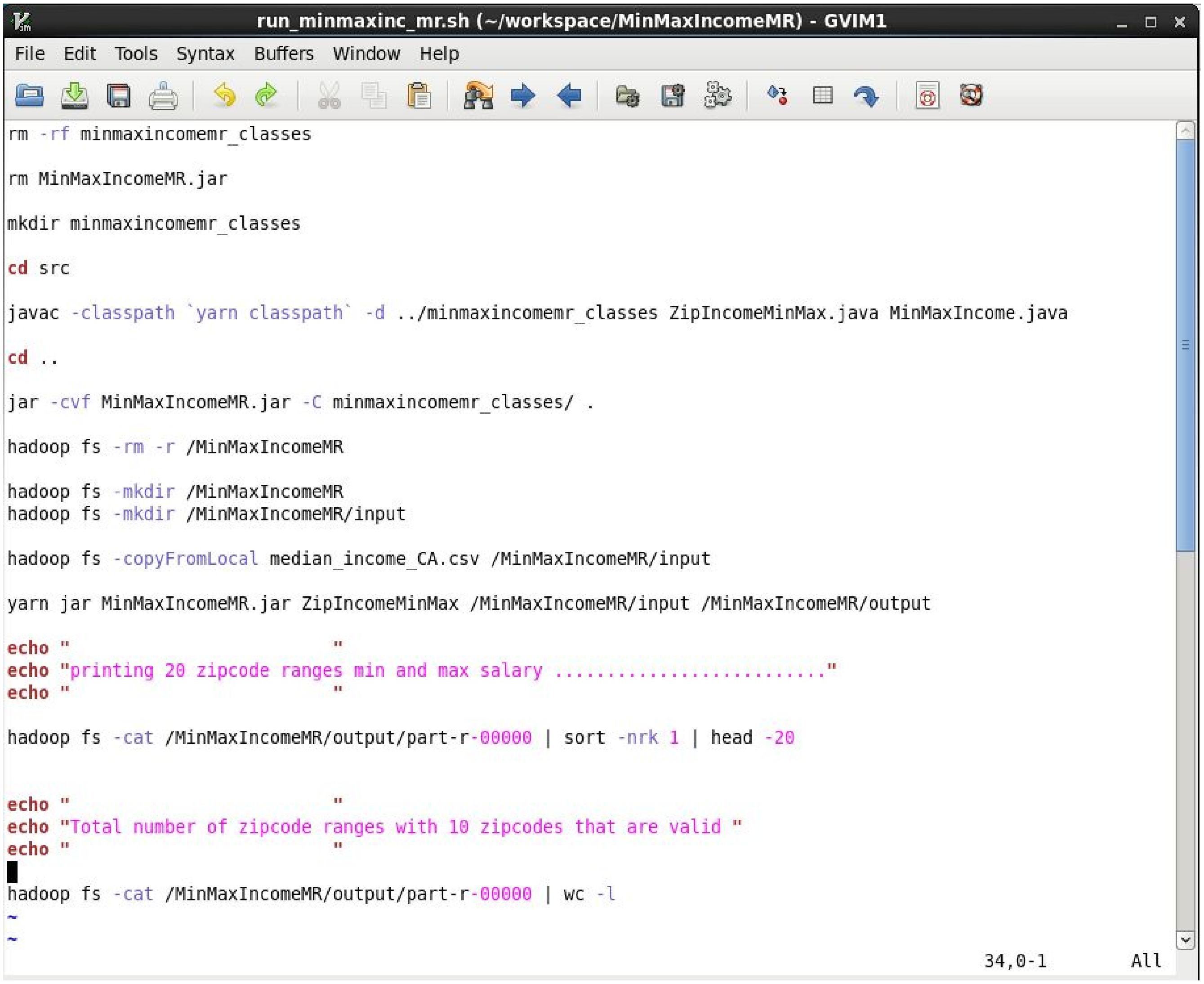


Project Source Folder Structure and Contents Explanation :

The source folder contains following directories and files :

bin	This folder contains all the binaries
src	This folder contains the java files ZipIncomeMinMax.java and MinMaxIncome.java
minmaxincomemr_classes	All the compiled Java Classes are in this folder
median_income_CA.csv	Input data for the MapReduce job
MinMaxIncomeMR.jar	The Executable JAR generated for the MapReduce job
run_minmaxinc_mr.sh	The shell script that automates compilation of classes, generation of JAR, creating input directory in HDFS , copying input data to the input folder in HDFS , running the map reduce job and saving the output in an output directory in HDFS , printing out necessary output from the output file.
README.txt	This file has step by step procedure of how a JAR is created given a Java Project with just src and bin folders.and how to execute it .

Shell Script :


run_minmaxinc_mr.sh (~/workspace/MinMaxIncomeMR) - GVIM1
File Edit Tools Syntax Buffers Window Help
34,0-1 All

```
rm -rf minmaxincomemr_classes

rm MinMaxIncomeMR.jar

mkdir minmaxincomemr_classes

cd src

javac -classpath `yarn classpath` -d ../minmaxincomemr_classes ZipIncomeMinMax.java MinMaxIncome.java

cd ..

jar -cvf MinMaxIncomeMR.jar -C minmaxincomemr_classes/ .

hadoop fs -rm -r /MinMaxIncomeMR

hadoop fs -mkdir /MinMaxIncomeMR
hadoop fs -mkdir /MinMaxIncomeMR/input

hadoop fs -copyFromLocal median_income_CA.csv /MinMaxIncomeMR/input

yarn jar MinMaxIncomeMR.jar ZipIncomeMinMax /MinMaxIncomeMR/input /MinMaxIncomeMR/output

echo "                         "
echo "printing 20 zipcode ranges min and max salary ........................."
echo "                         "

hadoop fs -cat /MinMaxIncomeMR/output/part-r-00000 | sort -nrk 1 | head -20


echo "                         "
echo "Total number of zipcode ranges with 10 zipcodes that are valid "
echo "                         "

hadoop fs -cat /MinMaxIncomeMR/output/part-r-00000 | wc -l
~
~
```

Explanation of Shell Script :

The shell script contains all the steps required to automate running a MapReduce job from within the project folder structure with just src and bin folder along with input data in it.

```
rm -rf minmaxincomemr_classes
```

removing the directory which holds all previously compiled java classes

```
rm MinMaxIncomeMR.jar
```

removing previously generated JAR files, just in case any change is made to the source files

```
mkdir minmaxincomemr_classes
```

recreating directory that will hold the compiled java classes

```
cd src
```

change directory to source directory

```
javac -classpath  `yarn classpath`  -d  ../minmaxincomemr_classes ZipIncomeMinMax.java MinMaxIncome.java
```

compiling java source files in src directory and saving it in the compiled java classes directory

```
cd ..
```

coming out of src directory to the project root

```
jar -cvf MinMaxIncome.jar  -C  minmaxincomemr_classes/ .
```

generating a JAR file in the project root using the compiled classes in the minmaxincomemr_classes directory

```
hadoop fs -rm -r /MinMaxIncomeMR
```

removing any previously created HDFS main directory for this job, which may contain output folder, as hadoop expects output directory to not exist beforehand

```
hadoop fs -mkdir /MinMaxIncomeMR
```

recreating the job specific main directory in HDFS

```
hadoop fs -mkdir /MinMaxIncomeMR/input
```

creating an subdirectory in the HDFS main directory created in the previous step for copying input data

```
hadoop fs -copyFromLocal median_income_CA.csv
/MinMaxIncomeMR/input
```

copying input data from local (available in the current project directory root) to the HDFS input data directory created in last step

```
yarn jar MinMaxIncomeMR.jar ZipIncomeMinMax
/MinMaxIncomeMR/input   /MinMaxIncomeMR/output
```

running MapReduce job using the "yarn jar" command which takes as other parameters the JAR file name, Java Class Name with main() method , HDFS input and output directories

```
hadoop fs -cat /MinMaxIncomeMR/output/part-r-00000  | sort -nrk 1 | head -20
```

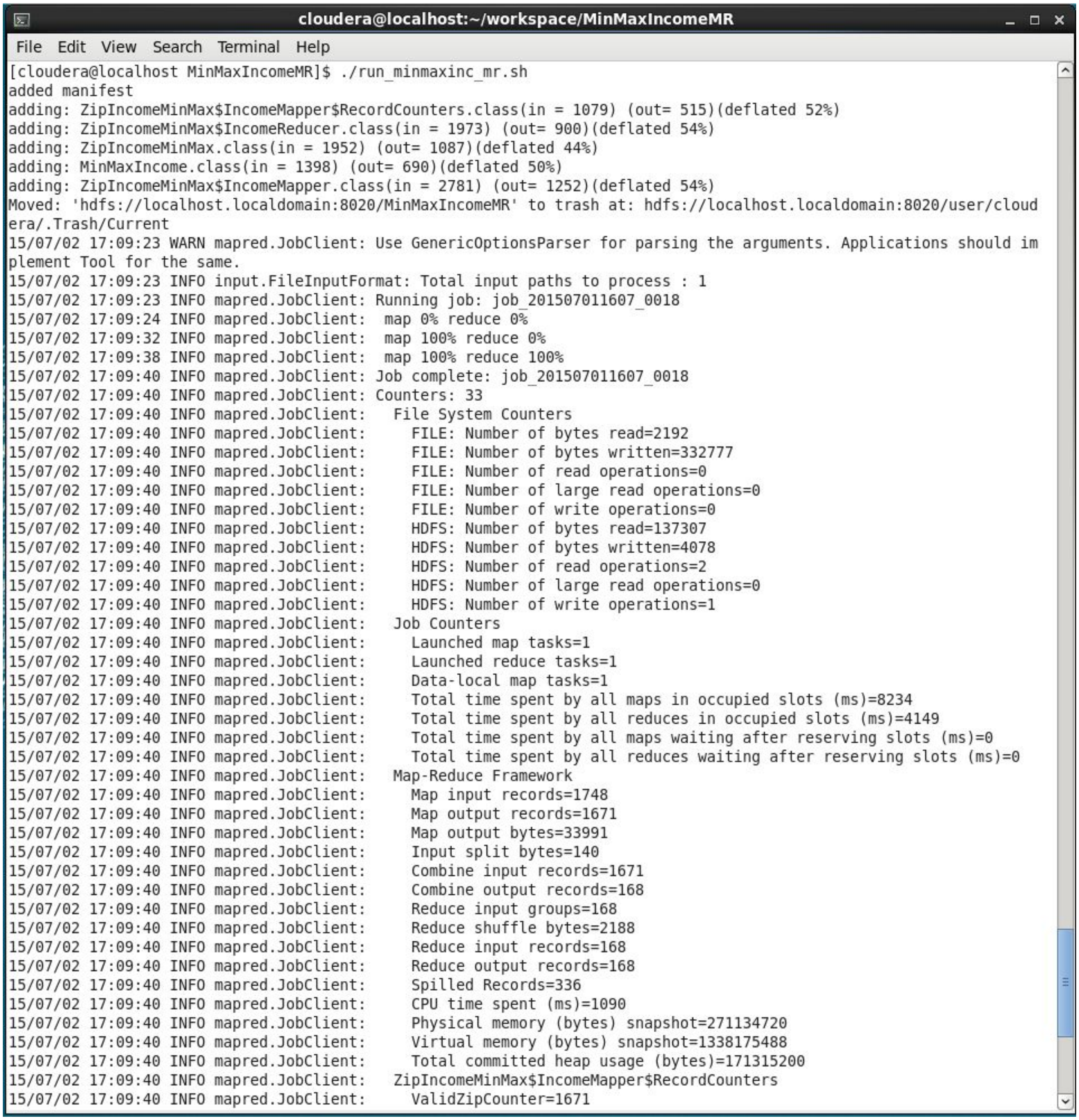
displaying the output generated in the output file part-r-00000 and piping it to sort in descending order by the first column and again piping it to restrict to 20 records from the top

```
hadoop fs -cat /MinMaxIncomeMR/output/part-r-00000 | wc -l
```

displaying the total number of lines in the output file part-r-00000

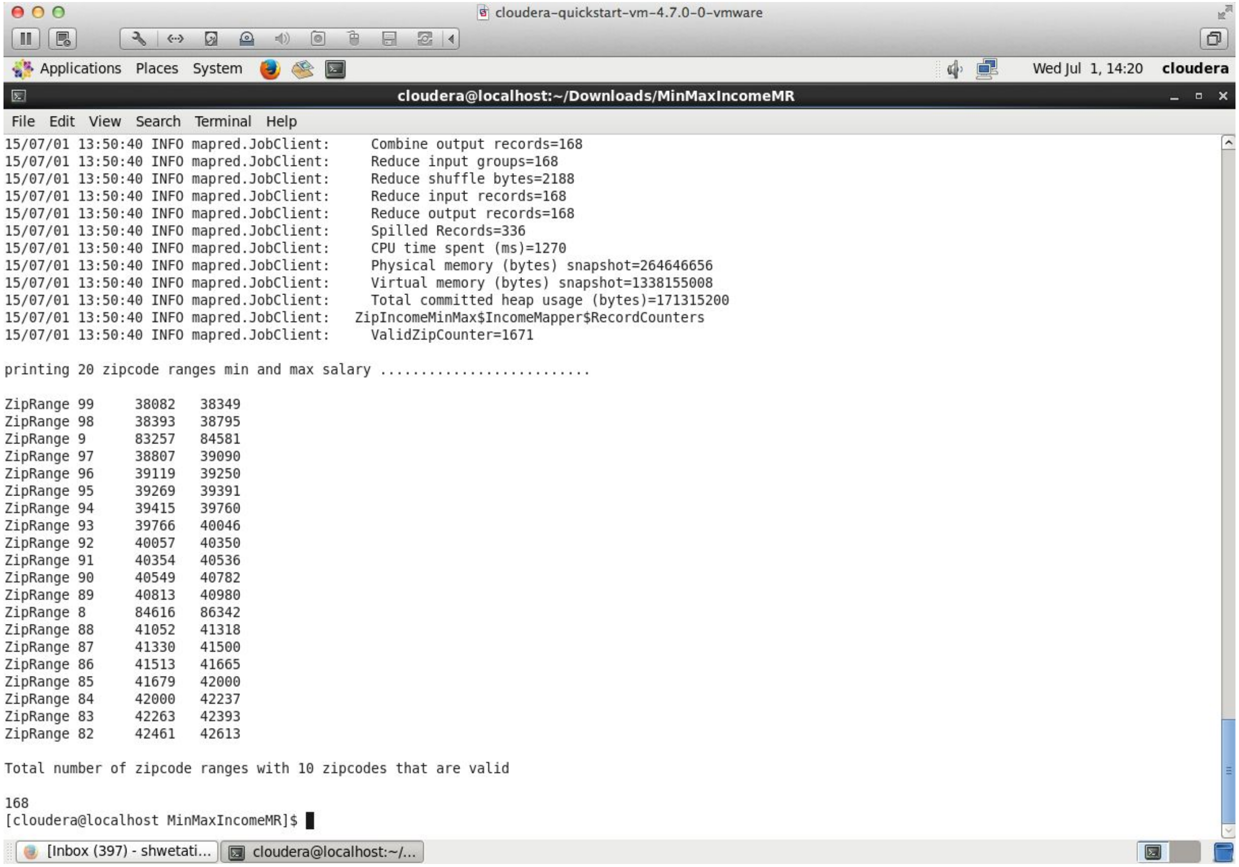
Note : This shell script should be run from the root of the project directory, where it is residing.

Execution of Script :



```
[cloudera@localhost MinMaxIncomeMR]$ ./run_minmaxinc_mr.sh
added manifest
adding: ZipIncomeMinMax$IncomeMapper$RecordCounters.class(in = 1079) (out= 515)(deflated 52%)
adding: ZipIncomeMinMax$IncomeReducer.class(in = 1973) (out= 900)(deflated 54%)
adding: ZipIncomeMinMax.class(in = 1952) (out= 1087)(deflated 44%)
adding: MinMaxIncome.class(in = 1398) (out= 690)(deflated 50%)
adding: ZipIncomeMinMax$IncomeMapper.class(in = 2781) (out= 1252)(deflated 54%)
Moved: 'hdfs://localhost.localdomain:8020/MinMaxIncomeMR' to trash at: hdfs://localhost.localdomain:8020/user/cloud
era/.Trash/Current
15/07/02 17:09:23 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should im
plement Tool for the same.
15/07/02 17:09:23 INFO input.FileInputFormat: Total input paths to process : 1
15/07/02 17:09:23 INFO mapred.JobClient: Running job: job_201507011607_0018
15/07/02 17:09:24 INFO mapred.JobClient:  map 0% reduce 0%
15/07/02 17:09:32 INFO mapred.JobClient:  map 100% reduce 0%
15/07/02 17:09:38 INFO mapred.JobClient:  map 100% reduce 100%
15/07/02 17:09:40 INFO mapred.JobClient: Job complete: job_201507011607_0018
15/07/02 17:09:40 INFO mapred.JobClient: Counters: 33
15/07/02 17:09:40 INFO mapred.JobClient:   File System Counters
15/07/02 17:09:40 INFO mapred.JobClient:     FILE: Number of bytes read=2192
15/07/02 17:09:40 INFO mapred.JobClient:     FILE: Number of bytes written=332777
15/07/02 17:09:40 INFO mapred.JobClient:     FILE: Number of read operations=0
15/07/02 17:09:40 INFO mapred.JobClient:     FILE: Number of large read operations=0
15/07/02 17:09:40 INFO mapred.JobClient:     FILE: Number of write operations=0
15/07/02 17:09:40 INFO mapred.JobClient:     HDFS: Number of bytes read=137307
15/07/02 17:09:40 INFO mapred.JobClient:     HDFS: Number of bytes written=4078
15/07/02 17:09:40 INFO mapred.JobClient:     HDFS: Number of read operations=2
15/07/02 17:09:40 INFO mapred.JobClient:     HDFS: Number of large read operations=0
15/07/02 17:09:40 INFO mapred.JobClient:     HDFS: Number of write operations=1
15/07/02 17:09:40 INFO mapred.JobClient:   Job Counters
15/07/02 17:09:40 INFO mapred.JobClient:     Launched map tasks=1
15/07/02 17:09:40 INFO mapred.JobClient:     Launched reduce tasks=1
15/07/02 17:09:40 INFO mapred.JobClient:     Data-local map tasks=1
15/07/02 17:09:40 INFO mapred.JobClient:     Total time spent by all maps in occupied slots (ms)=8234
15/07/02 17:09:40 INFO mapred.JobClient:     Total time spent by all reduces in occupied slots (ms)=4149
15/07/02 17:09:40 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
15/07/02 17:09:40 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
15/07/02 17:09:40 INFO mapred.JobClient:   Map-Reduce Framework
15/07/02 17:09:40 INFO mapred.JobClient:     Map input records=1748
15/07/02 17:09:40 INFO mapred.JobClient:     Map output records=1671
15/07/02 17:09:40 INFO mapred.JobClient:     Map output bytes=33991
15/07/02 17:09:40 INFO mapred.JobClient:     Input split bytes=140
15/07/02 17:09:40 INFO mapred.JobClient:     Combine input records=1671
15/07/02 17:09:40 INFO mapred.JobClient:     Combine output records=168
15/07/02 17:09:40 INFO mapred.JobClient:     Reduce input groups=168
15/07/02 17:09:40 INFO mapred.JobClient:     Reduce shuffle bytes=2188
15/07/02 17:09:40 INFO mapred.JobClient:     Reduce input records=168
15/07/02 17:09:40 INFO mapred.JobClient:     Reduce output records=168
15/07/02 17:09:40 INFO mapred.JobClient:     Spilled Records=336
15/07/02 17:09:40 INFO mapred.JobClient:     CPU time spent (ms)=1090
15/07/02 17:09:40 INFO mapred.JobClient:     Physical memory (bytes) snapshot=271134720
15/07/02 17:09:40 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=1338175488
15/07/02 17:09:40 INFO mapred.JobClient:     Total committed heap usage (bytes)=171315200
15/07/02 17:09:40 INFO mapred.JobClient:   ZipIncomeMinMax$IncomeMapper$RecordCounters
15/07/02 17:09:40 INFO mapred.JobClient:     ValidZipCounter=1671
```

Final Output :


cloudera-quickstart-vm-4.7.0-0-vmware
Applications Places System
Wed Jul 1, 14:20 cloudera
cloudera@localhost:~/Downloads/MinMaxIncomeMR
File Edit View Search Terminal Help
[Inbox (397) - shwetati...
cloudera@localhost:~/...

```
15/07/01 13:50:40 INFO mapred.JobClient:     Combine output records=168
15/07/01 13:50:40 INFO mapred.JobClient:     Reduce input groups=168
15/07/01 13:50:40 INFO mapred.JobClient:     Reduce shuffle bytes=2188
15/07/01 13:50:40 INFO mapred.JobClient:     Reduce input records=168
15/07/01 13:50:40 INFO mapred.JobClient:     Reduce output records=168
15/07/01 13:50:40 INFO mapred.JobClient:     Spilled Records=336
15/07/01 13:50:40 INFO mapred.JobClient:     CPU time spent (ms)=1270
15/07/01 13:50:40 INFO mapred.JobClient:     Physical memory (bytes) snapshot=264646656
15/07/01 13:50:40 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=1338155008
15/07/01 13:50:40 INFO mapred.JobClient:     Total committed heap usage (bytes)=171315200
15/07/01 13:50:40 INFO mapred.JobClient:   ZipIncomeMinMax$IncomeMapper$RecordCounters
15/07/01 13:50:40 INFO mapred.JobClient:     ValidZipCounter=1671

printing 20 zipcode ranges min and max salary .........................

ZipRange 99     38082   38349
ZipRange 98     38393   38795
ZipRange 9      83257   84581
ZipRange 97     38807   39090
ZipRange 96     39119   39250
ZipRange 95     39269   39391
ZipRange 94     39415   39760
ZipRange 93     39766   40046
ZipRange 92     40057   40350
ZipRange 91     40354   40536
ZipRange 90     40549   40782
ZipRange 89     40813   40980
ZipRange 8      84616   86342
ZipRange 88     41052   41318
ZipRange 87     41330   41500
ZipRange 86     41513   41665
ZipRange 85     41679   42000
ZipRange 84     42000   42237
ZipRange 83     42263   42393
ZipRange 82     42461   42613

Total number of zipcode ranges with 10 zipcodes that are valid

168
[cloudera@localhost MinMaxIncomeMR]$ █
```

Output Explanation :

The output produced by this program is saved in the output directory /MinMaxIncomeMR/output of HDFS. The file part-r-00000 in this output contains all 168 valid 10 zip code ranges with minimum and maximum salary. The two columns in output files are the zip code range, and the minimum salary and maximum salary combo in that order. There were a total of 1671 zip codes that were valid from a total of 1748 zip codes provided in the data set. Our program omitted the zip codes that were not in proper format in the map stage of the job. Hence the remaining 1671 zip codes were split into Zip Code ranges of 10 zip codes each to find minimum and maximum salary in that range. We used 10 valid zip codes in consecutive order as it appears in the input file to group the zip code ranges.

Q2. This part involves some fun MapReduce processing using the White House Visitor Log. You can find the dataset at:

http://www.whitehouse.gov/files/disclosures/visitors/WhiteHouse-WAVESReleased-0827.csv

First download this dataset and copy it to HDFS (e.g., in the /usr/research/home/USERNAME directory, where USERNAME is replaced with your user name). Use the copyFromLocal command described at:

http://hadoop.apache.org/common/docs/r0.20.2/hdfs_shell.html

The attributes in this dataset are described at:

http://www.whitehouse.gov/files/disclosures/visitors/WhiteHouse-WAVES-Key- 1209.txt

Also, you can see a spreadsheet of the data at:

http://www.whitehouse.gov/briefing-room/disclosures/visitor-records

You are required to write efficient MapReduce programs to find the following information:
(i) The 10 most frequent visitors (NAMELAST, NAMEFIRST, NAMEMID) to the White House.
(ii) The 10 most frequently visited people (visitee_namelast, visitee_namefirst) in the White House.
(iii) The 10 most frequent visitor-visitee combinations.
(iv) Some other interesting statistic that you can think of.

Solution :

(i)

What this MapReduce program does :

The java file WhiteHouseMR1.java, contains all the hadoop job specific logic, which includes the mapper, reducer, Top Ten Mapper, Top Ten Reducer and driver etc.

This MapReduce program has 2 jobs which are chained to produce the final output.

For Job 1 :

In the Map Stage, records are emitted one by one to the map() tasks, which are then split into fields and the required fields (NAMELAST, NAMEFIRST, NAMEMID clubbed together) is taken as key and value as 1

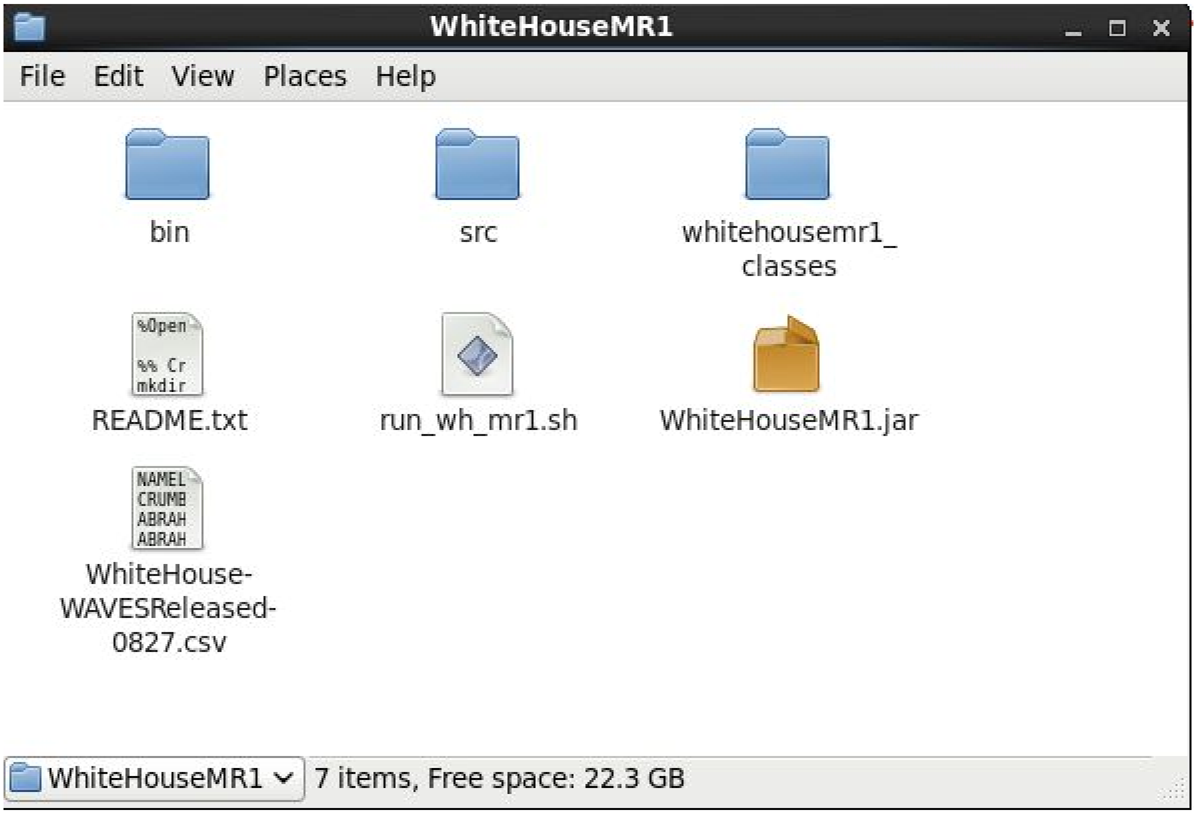
In the Reduce Stage, all the values of the same key (NAMELAST, NAMEFIRST, NAMEMID combo) are aggregated to get the overall count for each single visitor in the data set.

Output of job1 is saved into intermediate folder in HDFS to be read by the job 2.

For Job 2 :

The Top Ten Mapper and Top Ten Reducer are used to filter out the top 10 visitors and the output of this job is written to final output directory

Project Folder Structure :



7 items, Free space: 22.3 GB

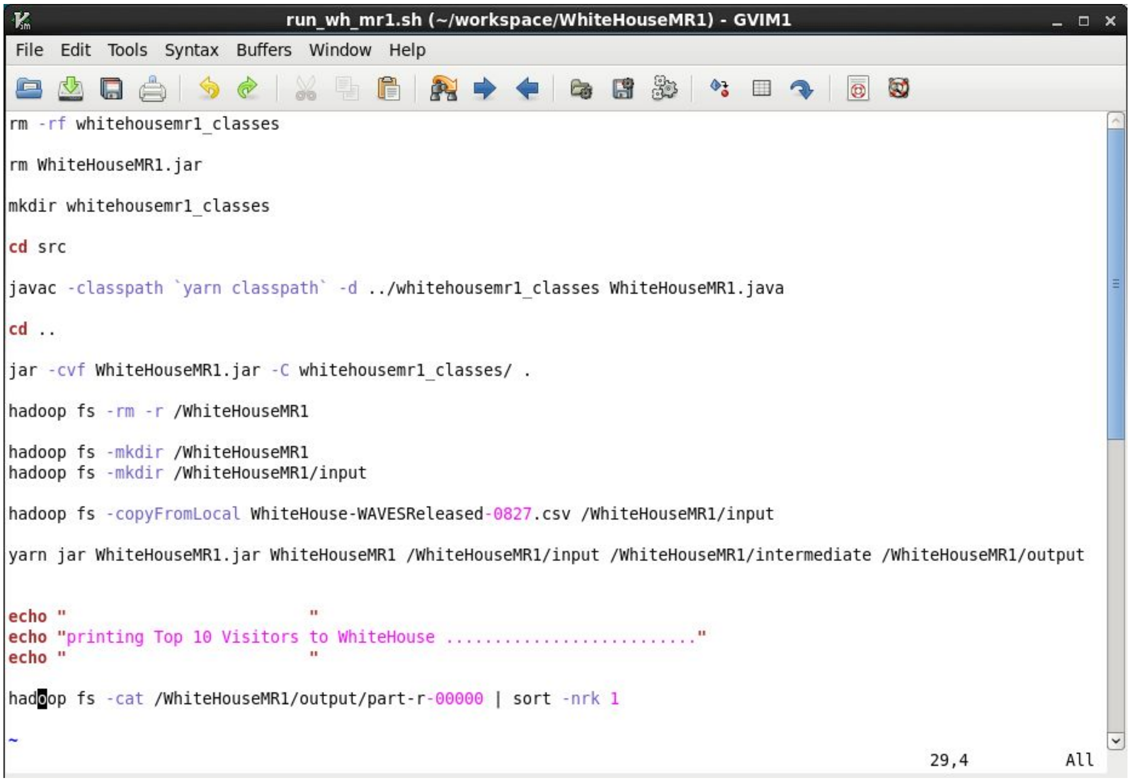
Project Source Folder Structure and Contents explanation :

The source folder contains following directories and files :

bin	This folder contains all the binaries
src	This folder contains the java file WhiteHouseMR1.java
whitehousemr1_classes	All the compiled Java Classes are in this folder
WhiteHouse-WAVESReleased-0827.csv	Input data for the MapReduce job

WhiteHouseMR1.jar	The Executable JAR generated for the MapReduce job
run_wh_mr1.sh	The shell script that automates compilation of classes, generation of JAR, creating input directory in HDFS , copying input data to the input folder in HDFS , running the map reduce job and saving the output in an output directory in HDFS , printing out necessary output from the output file.
README.txt	This file has step by step procedure of how a JAR is created given a Java Project with just src and bin folders.and how to execute it .

Shell Script :


run_wh_mr1.sh (~/workspace/WhiteHouseMR1) - GVIM1
File Edit Tools Syntax Buffers Window Help
29,4 All

```
rm -rf whitehousemr1_classes

rm WhiteHouseMR1.jar

mkdir whitehousemr1_classes

cd src

javac -classpath `yarn classpath` -d ../whitehousemr1_classes WhiteHouseMR1.java

cd ..

jar -cvf WhiteHouseMR1.jar -C whitehousemr1_classes/ .

hadoop fs -rm -r /WhiteHouseMR1

hadoop fs -mkdir /WhiteHouseMR1
hadoop fs -mkdir /WhiteHouseMR1/input

hadoop fs -copyFromLocal WhiteHouse-WAVESReleased-0827.csv /WhiteHouseMR1/input

yarn jar WhiteHouseMR1.jar WhiteHouseMR1 /WhiteHouseMR1/input /WhiteHouseMR1/intermediate /WhiteHouseMR1/output


echo "                         "
echo "printing Top 10 Visitors to WhiteHouse ..........................."
echo "                         "

hadoop fs -cat /WhiteHouseMR1/output/part-r-00000 | sort -nrk 1

~
```

Explanation of Shell Script :

The shell script contains all the steps required to automate running a MapReduce job from within the project folder structure with just src and bin folder along with input data in it.

```
rm -rf whitehousemr1_classes
```

removing the directory which holds all previously compiled java classes

```
rm WhiteHouseMR1.jar
```

removing previously generated JAR files, just in case any change is made to the source files

```
mkdir whitehousemr1_classes
```

recreating directory that will hold the compiled java classes

```
cd src
```

change directory to source directory

```
javac -classpath  `yarn classpath`  -d  ../whitehousemr1_classes WhiteHouseMR1.java
```

compiling java source file in src directory and saving it in the compiled java classes directory

```
cd ..
```

coming out of src directory to the project root

```
jar -cvf WhiteHouseMR1.jar  -C  whitehousemr1_classes/ .
```

generating a JAR file in the project root using the compiled classes in the whitehousemr1_classes directory

```
hadoop fs -rm -r /WhiteHouseMR1
```

removing any previously created HDFS main directory for this job, which may contain output folder, as hadoop expects output directory to not exist beforehand

```
hadoop fs -mkdir /WhiteHouseMR1
```

recreating the job specific main directory in HDFS

```
hadoop fs -mkdir /WhiteHouseMR1/input
```

creating an subdirectory in the HDFS main directory created in the previous step for copying input data

```
hadoop fs -copyFromLocal WhiteHouse-WAVESReleased-0827.csv
/WhiteHouseMR1/input
```

copying input data from local (available in the current project directory root) to the HDFS input data directory created in last step

```
yarn jar WhiteHouseMR1.jar WhiteHouseMR1
/WhiteHouseMR1/input   /WhiteHouseMR1/intermediate
/WhiteHouseMR1/output
```

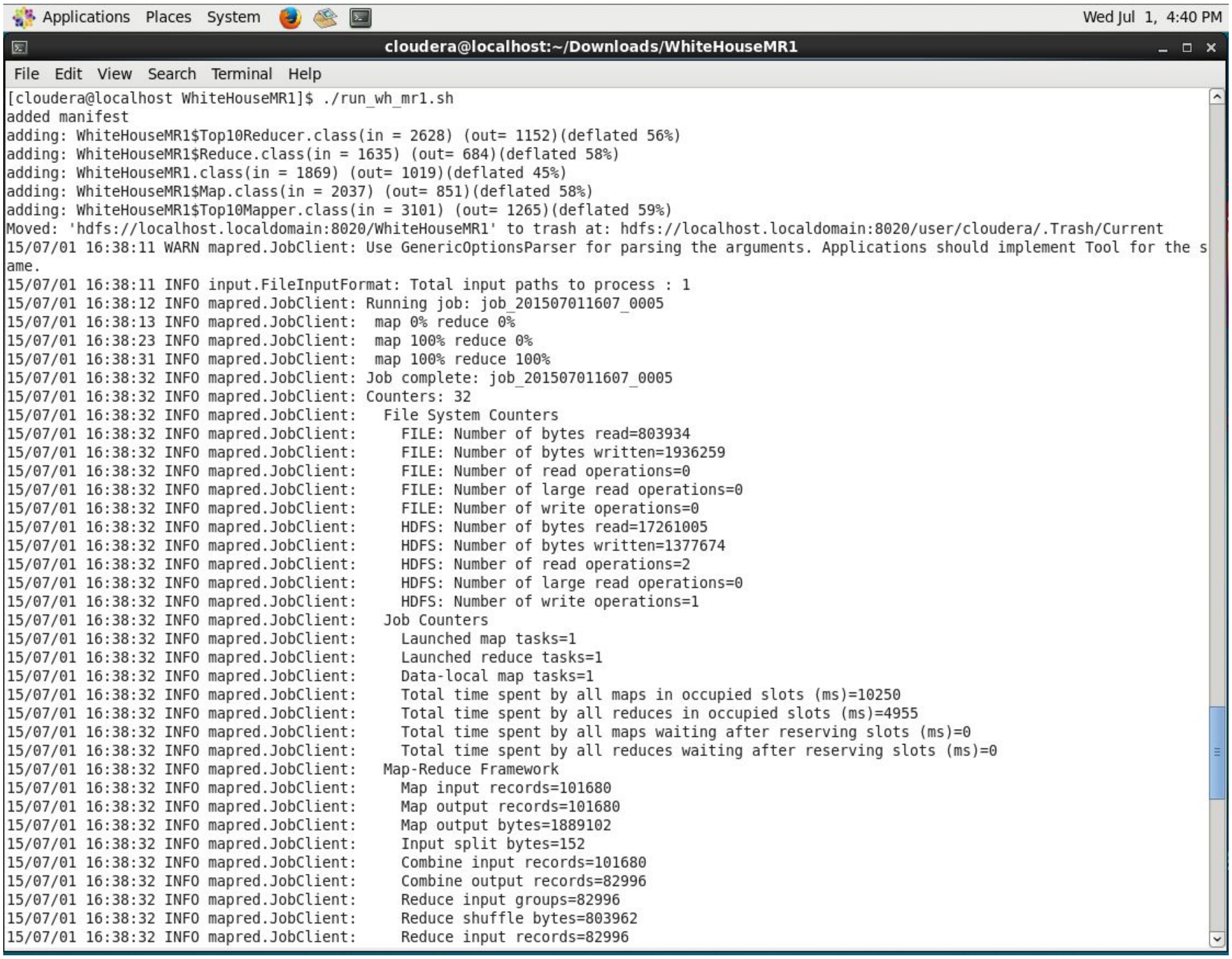
running MapReduce job using the "yarn jar" command which takes as other parameters the JAR file name, Java Class Name with main() method , HDFS input , intermediate and output directories

hadoop fs -cat /WhiteHouseMR1/output/part-r-00000 | sort -nrk 1

displaying the output generated in the output file part-r-00000 and piping it to sort in descending order by the first column

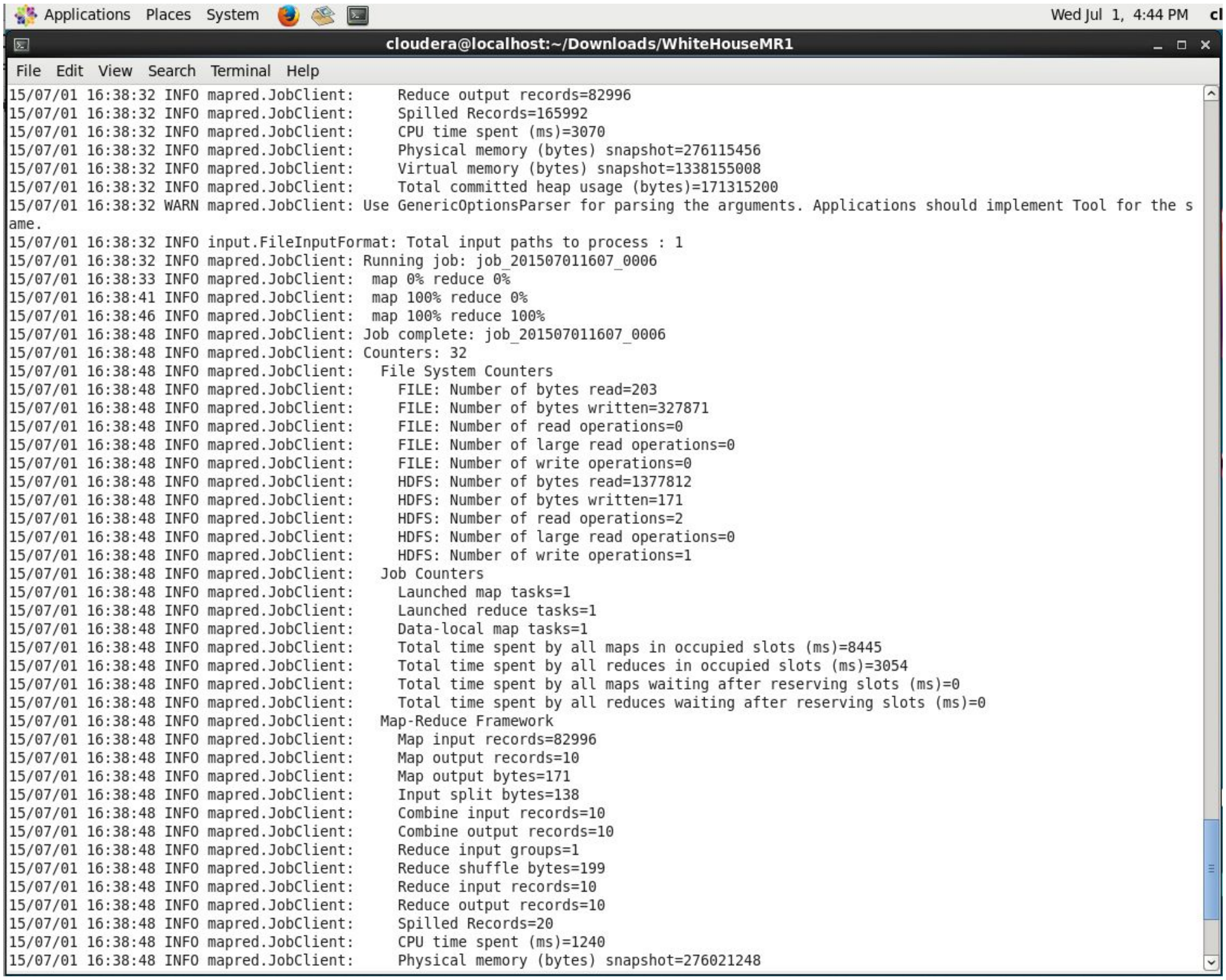
Note : This shell script should be run from the root of the project directory, where it is residing.

Execution Process :



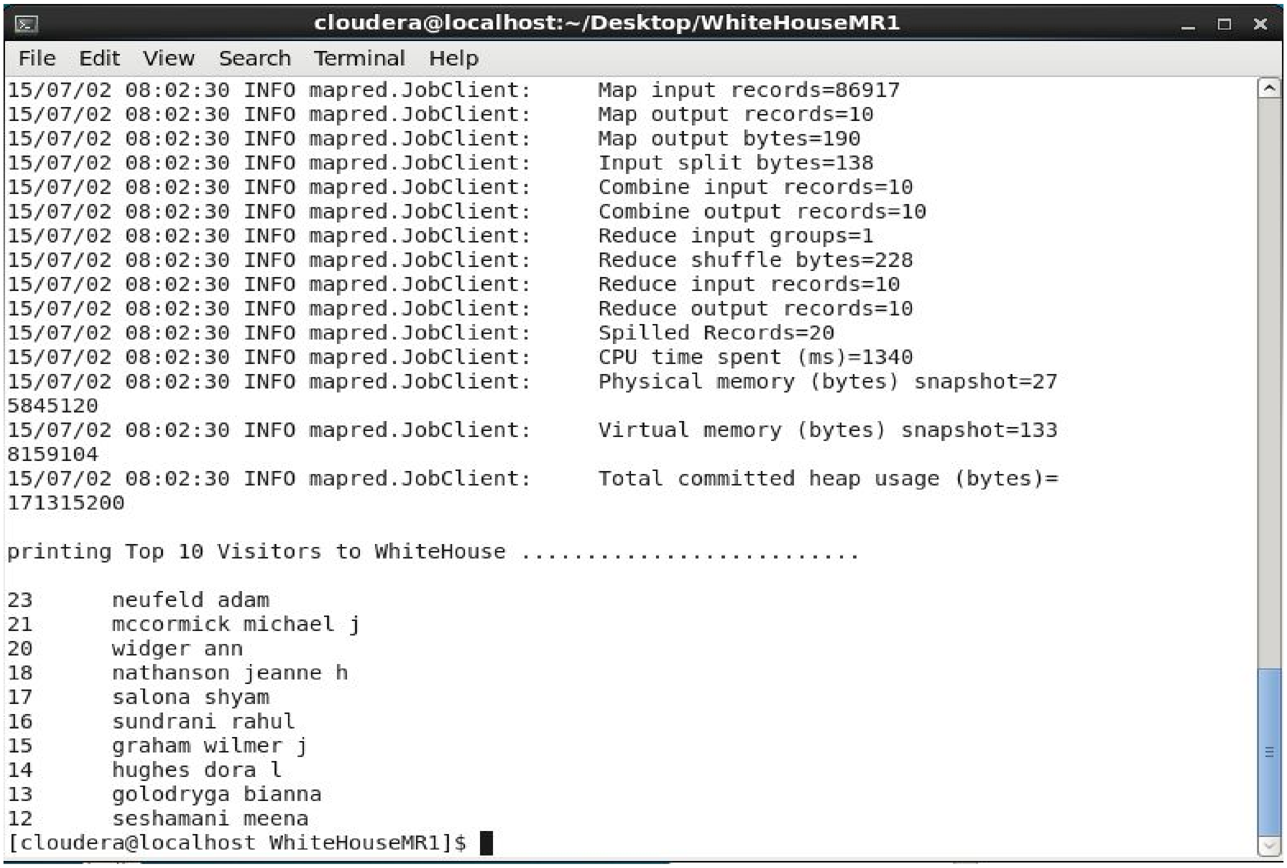
```
[cloudera@localhost WhiteHouseMR1]$ ./run_wh_mr1.sh
added manifest
adding: WhiteHouseMR1$Top10Reducer.class(in = 2628) (out= 1152)(deflated 56%)
adding: WhiteHouseMR1$Reduce.class(in = 1635) (out= 684)(deflated 58%)
adding: WhiteHouseMR1.class(in = 1869) (out= 1019)(deflated 45%)
adding: WhiteHouseMR1$Map.class(in = 2037) (out= 851)(deflated 58%)
adding: WhiteHouseMR1$Top10Mapper.class(in = 3101) (out= 1265)(deflated 59%)
Moved: 'hdfs://localhost.localdomain:8020/WhiteHouseMR1' to trash at: hdfs://localhost.localdomain:8020/user/cloudera/.Trash/Current
15/07/01 16:38:11 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the s
ame.
15/07/01 16:38:11 INFO input.FileInputFormat: Total input paths to process : 1
15/07/01 16:38:12 INFO mapred.JobClient: Running job: job_201507011607_0005
15/07/01 16:38:13 INFO mapred.JobClient:  map 0% reduce 0%
15/07/01 16:38:23 INFO mapred.JobClient:  map 100% reduce 0%
15/07/01 16:38:31 INFO mapred.JobClient:  map 100% reduce 100%
15/07/01 16:38:32 INFO mapred.JobClient: Job complete: job_201507011607_0005
15/07/01 16:38:32 INFO mapred.JobClient: Counters: 32
15/07/01 16:38:32 INFO mapred.JobClient:   File System Counters
15/07/01 16:38:32 INFO mapred.JobClient:     FILE: Number of bytes read=803934
15/07/01 16:38:32 INFO mapred.JobClient:     FILE: Number of bytes written=1936259
15/07/01 16:38:32 INFO mapred.JobClient:     FILE: Number of read operations=0
15/07/01 16:38:32 INFO mapred.JobClient:     FILE: Number of large read operations=0
15/07/01 16:38:32 INFO mapred.JobClient:     FILE: Number of write operations=0
15/07/01 16:38:32 INFO mapred.JobClient:     HDFS: Number of bytes read=17261005
15/07/01 16:38:32 INFO mapred.JobClient:     HDFS: Number of bytes written=1377674
15/07/01 16:38:32 INFO mapred.JobClient:     HDFS: Number of read operations=2
15/07/01 16:38:32 INFO mapred.JobClient:     HDFS: Number of large read operations=0
15/07/01 16:38:32 INFO mapred.JobClient:     HDFS: Number of write operations=1
15/07/01 16:38:32 INFO mapred.JobClient:   Job Counters
15/07/01 16:38:32 INFO mapred.JobClient:     Launched map tasks=1
15/07/01 16:38:32 INFO mapred.JobClient:     Launched reduce tasks=1
15/07/01 16:38:32 INFO mapred.JobClient:     Data-local map tasks=1
15/07/01 16:38:32 INFO mapred.JobClient:     Total time spent by all maps in occupied slots (ms)=10250
15/07/01 16:38:32 INFO mapred.JobClient:     Total time spent by all reduces in occupied slots (ms)=4955
15/07/01 16:38:32 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
15/07/01 16:38:32 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
15/07/01 16:38:32 INFO mapred.JobClient:   Map-Reduce Framework
15/07/01 16:38:32 INFO mapred.JobClient:     Map input records=101680
15/07/01 16:38:32 INFO mapred.JobClient:     Map output records=101680
15/07/01 16:38:32 INFO mapred.JobClient:     Map output bytes=1889102
15/07/01 16:38:32 INFO mapred.JobClient:     Input split bytes=152
15/07/01 16:38:32 INFO mapred.JobClient:     Combine input records=101680
15/07/01 16:38:32 INFO mapred.JobClient:     Combine output records=82996
15/07/01 16:38:32 INFO mapred.JobClient:     Reduce input groups=82996
15/07/01 16:38:32 INFO mapred.JobClient:     Reduce shuffle bytes=803962
15/07/01 16:38:32 INFO mapred.JobClient:     Reduce input records=82996
```

Execution Process Continues..


Applications Places System
Wed Jul 1, 4:44 PM
cl
cloudera@localhost:~/Downloads/WhiteHouseMR1
File Edit View Search Terminal Help

```
15/07/01 16:38:32 INFO mapred.JobClient:     Reduce output records=82996
15/07/01 16:38:32 INFO mapred.JobClient:     Spilled Records=165992
15/07/01 16:38:32 INFO mapred.JobClient:     CPU time spent (ms)=3070
15/07/01 16:38:32 INFO mapred.JobClient:     Physical memory (bytes) snapshot=276115456
15/07/01 16:38:32 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=1338155008
15/07/01 16:38:32 INFO mapred.JobClient:     Total committed heap usage (bytes)=171315200
15/07/01 16:38:32 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the s
ame.
15/07/01 16:38:32 INFO input.FileInputFormat: Total input paths to process : 1
15/07/01 16:38:32 INFO mapred.JobClient: Running job: job_201507011607_0006
15/07/01 16:38:33 INFO mapred.JobClient:  map 0% reduce 0%
15/07/01 16:38:41 INFO mapred.JobClient:  map 100% reduce 0%
15/07/01 16:38:46 INFO mapred.JobClient:  map 100% reduce 100%
15/07/01 16:38:48 INFO mapred.JobClient: Job complete: job_201507011607_0006
15/07/01 16:38:48 INFO mapred.JobClient: Counters: 32
15/07/01 16:38:48 INFO mapred.JobClient:   File System Counters
15/07/01 16:38:48 INFO mapred.JobClient:     FILE: Number of bytes read=203
15/07/01 16:38:48 INFO mapred.JobClient:     FILE: Number of bytes written=327871
15/07/01 16:38:48 INFO mapred.JobClient:     FILE: Number of read operations=0
15/07/01 16:38:48 INFO mapred.JobClient:     FILE: Number of large read operations=0
15/07/01 16:38:48 INFO mapred.JobClient:     FILE: Number of write operations=0
15/07/01 16:38:48 INFO mapred.JobClient:     HDFS: Number of bytes read=1377812
15/07/01 16:38:48 INFO mapred.JobClient:     HDFS: Number of bytes written=171
15/07/01 16:38:48 INFO mapred.JobClient:     HDFS: Number of read operations=2
15/07/01 16:38:48 INFO mapred.JobClient:     HDFS: Number of large read operations=0
15/07/01 16:38:48 INFO mapred.JobClient:     HDFS: Number of write operations=1
15/07/01 16:38:48 INFO mapred.JobClient:   Job Counters
15/07/01 16:38:48 INFO mapred.JobClient:     Launched map tasks=1
15/07/01 16:38:48 INFO mapred.JobClient:     Launched reduce tasks=1
15/07/01 16:38:48 INFO mapred.JobClient:     Data-local map tasks=1
15/07/01 16:38:48 INFO mapred.JobClient:     Total time spent by all maps in occupied slots (ms)=8445
15/07/01 16:38:48 INFO mapred.JobClient:     Total time spent by all reduces in occupied slots (ms)=3054
15/07/01 16:38:48 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
15/07/01 16:38:48 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
15/07/01 16:38:48 INFO mapred.JobClient:   Map-Reduce Framework
15/07/01 16:38:48 INFO mapred.JobClient:     Map input records=82996
15/07/01 16:38:48 INFO mapred.JobClient:     Map output records=10
15/07/01 16:38:48 INFO mapred.JobClient:     Map output bytes=171
15/07/01 16:38:48 INFO mapred.JobClient:     Input split bytes=138
15/07/01 16:38:48 INFO mapred.JobClient:     Combine input records=10
15/07/01 16:38:48 INFO mapred.JobClient:     Combine output records=10
15/07/01 16:38:48 INFO mapred.JobClient:     Reduce input groups=1
15/07/01 16:38:48 INFO mapred.JobClient:     Reduce shuffle bytes=199
15/07/01 16:38:48 INFO mapred.JobClient:     Reduce input records=10
15/07/01 16:38:48 INFO mapred.JobClient:     Reduce output records=10
15/07/01 16:38:48 INFO mapred.JobClient:     Spilled Records=20
15/07/01 16:38:48 INFO mapred.JobClient:     CPU time spent (ms)=1240
15/07/01 16:38:48 INFO mapred.JobClient:     Physical memory (bytes) snapshot=276021248
```

Final Output :


cloudera@localhost:~/Desktop/WhiteHouseMR1
File Edit View Search Terminal Help

```
15/07/02 08:02:30 INFO mapred.JobClient:     Map input records=86917
15/07/02 08:02:30 INFO mapred.JobClient:     Map output records=10
15/07/02 08:02:30 INFO mapred.JobClient:     Map output bytes=190
15/07/02 08:02:30 INFO mapred.JobClient:     Input split bytes=138
15/07/02 08:02:30 INFO mapred.JobClient:     Combine input records=10
15/07/02 08:02:30 INFO mapred.JobClient:     Combine output records=10
15/07/02 08:02:30 INFO mapred.JobClient:     Reduce input groups=1
15/07/02 08:02:30 INFO mapred.JobClient:     Reduce shuffle bytes=228
15/07/02 08:02:30 INFO mapred.JobClient:     Reduce input records=10
15/07/02 08:02:30 INFO mapred.JobClient:     Reduce output records=10
15/07/02 08:02:30 INFO mapred.JobClient:     Spilled Records=20
15/07/02 08:02:30 INFO mapred.JobClient:     CPU time spent (ms)=1340
15/07/02 08:02:30 INFO mapred.JobClient:     Physical memory (bytes) snapshot=27
5845120
15/07/02 08:02:30 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=133
8159104
15/07/02 08:02:30 INFO mapred.JobClient:     Total committed heap usage (bytes)=
171315200

printing Top 10 Visitors to WhiteHouse .........................

23      neufeld adam
21      mccormick michael j
20      widger ann
18      nathanson jeanne h
17      salona shyam
16      sundrani rahul
15      graham wilmer j
14      hughes dora l
13      golodryga bianna
12      seshamani meena
[cloudera@localhost WhiteHouseMR1]$ 
```

Output Explanation :

The final output produced by this program is saved in the output directory /WhiteHouseMR1/output of HDFS. The file part-r-00000 in this output contains the top 10 visitors to the White House in descending order. The two columns in output files are the Visit Count, (NAMELAST, NAMEFIRST, NAMEMID clubbed together) in that order. This MapReduce main job included 2 sub jobs, first job dealt with identifying the visitors count for all the records in the data set. Later this job was chained with another job to only get the top 10 visitor count . Because of this, apart from the input and output folders in HDFS corresponding to this job, there was also an additional intermediate directory to write the output of the first job (/WhiteHouseMR1/intermediate), from where job 2 read the data to produce the final output .

(ii)

What this MapReduce program does :

The java file WhiteHouseMR2.java, contains all the hadoop job specific logic, which includes the mapper, reducer, Top Ten Mapper, Top Ten Reducer and driver etc.

This MapReduce program has 2 jobs which are chained to produce the final output.

For Job 1 :

In the Map Stage, records are emitted one by one to the map() tasks, which are then split into fields and the required fields (visitee_namelast, visitee_namefirst clubbed together) is taken as key and value as 1

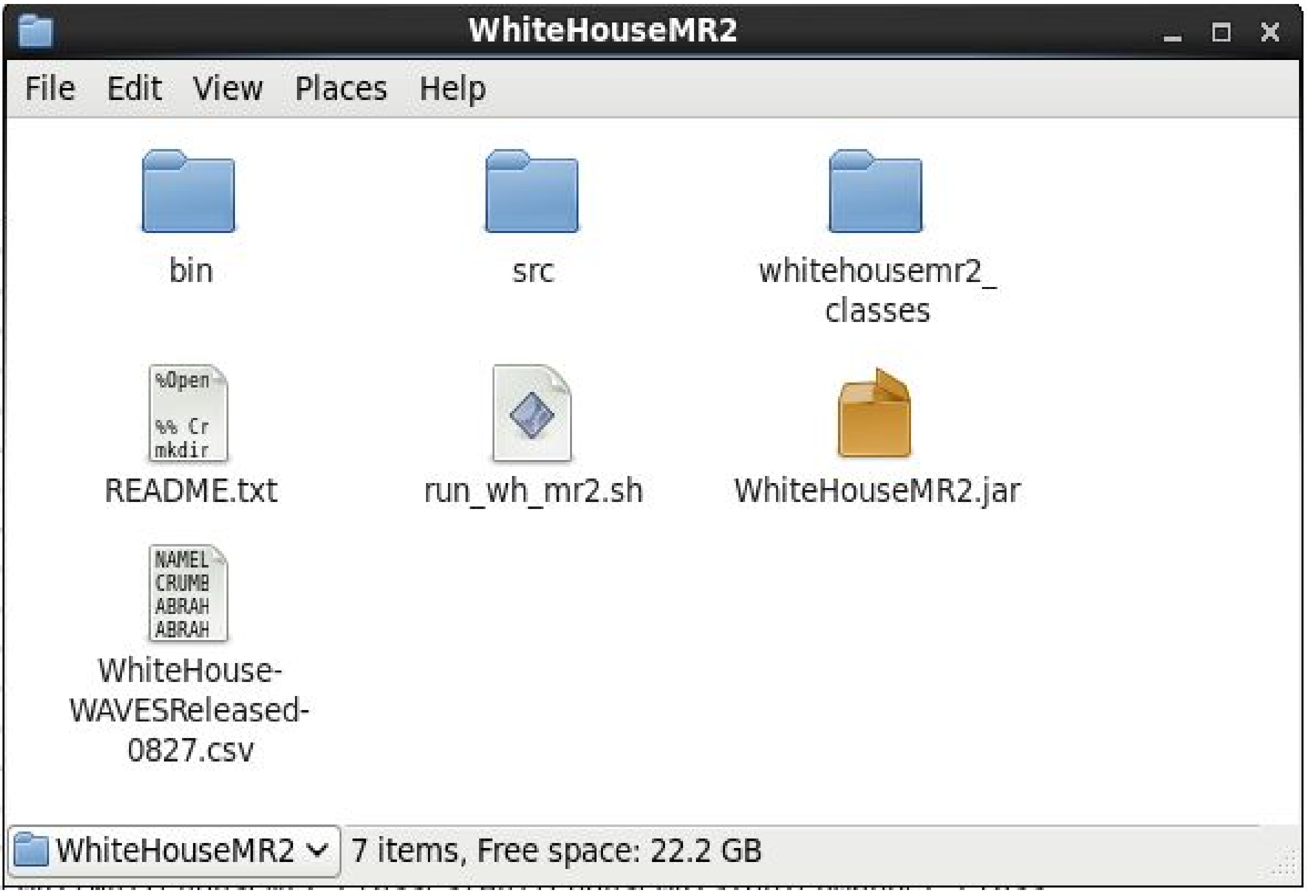
In the Reduce Stage, all the values of the same key (visitee_namelast, visitee_namefirst combo) are aggregated to get the overall count for each single visitee in the data set.

Output of job1 is saved into intermediate folder in HDFS to be read by the job 2.

For Job 2 :

The Top Ten Mapper and Top Ten Reducer are used to filter out the top 10 visitees and the output of this job is written to final output directory

Project Folder Structure :



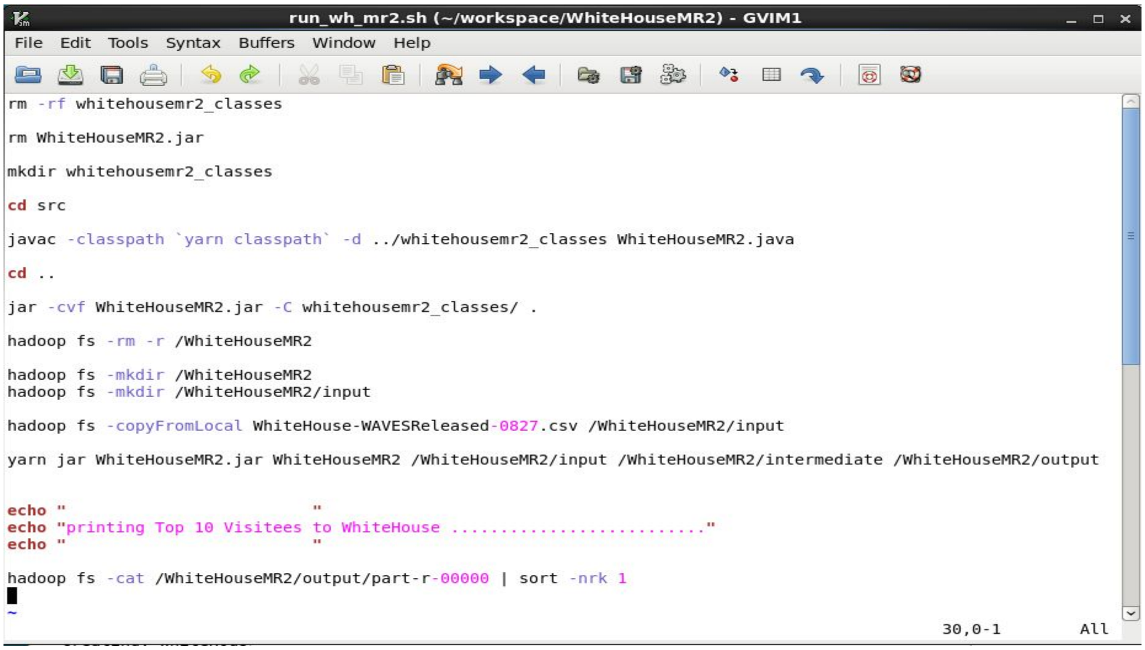
Project Source Folder Structure and Contents explanation :

The source folder contains following directories and files :

bin	This folder contains all the binaries
src	This folder contains the java file WhiteHouseMR2.java
whitehousemr2_classes	All the compiled Java Classes are in this folder

WhiteHouse-WAVESReleased-0827.csv	Input data for the MapReduce job
WhiteHouseMR2.jar	The Executable JAR generated for the MapReduce job
run_wh_mr2.sh	The shell script that automates compilation of classes, generation of JAR, creating input directory in HDFS , copying input data to the input folder in HDFS , running the map reduce job and saving the output in an output directory in HDFS , printing out necessary output from the output file.
README.txt	This file has step by step procedure of how a JAR is created given a Java Project with just src and bin folders.and how to execute it .

Shell Script :


run_wh_mr2.sh (~/workspace/WhiteHouseMR2) - GVIM1
File Edit Tools Syntax Buffers Window Help
30,0-1 All

```
rm -rf whitehousemr2_classes

rm WhiteHouseMR2.jar

mkdir whitehousemr2_classes

cd src

javac -classpath `yarn classpath` -d ../whitehousemr2_classes WhiteHouseMR2.java

cd ..

jar -cvf WhiteHouseMR2.jar -C whitehousemr2_classes/ .

hadoop fs -rm -r /WhiteHouseMR2

hadoop fs -mkdir /WhiteHouseMR2
hadoop fs -mkdir /WhiteHouseMR2/input

hadoop fs -copyFromLocal WhiteHouse-WAVESReleased-0827.csv /WhiteHouseMR2/input

yarn jar WhiteHouseMR2.jar WhiteHouseMR2 /WhiteHouseMR2/input /WhiteHouseMR2/intermediate /WhiteHouseMR2/output


echo "                          "
echo "printing Top 10 Visitees to WhiteHouse ..........................."
echo "                          "

hadoop fs -cat /WhiteHouseMR2/output/part-r-00000 | sort -nrk 1

~
```

Explanation of Shell Script :

The shell script contains all the steps required to automate running a MapReduce job from within the project folder structure with just src and bin folder along with input data in it.

```
rm -rf whitehousemr2_classes
```

removing the directory which holds all previously compiled java classes

```
rm WhiteHouseMR2.jar
```

removing previously generated JAR files, just in case any change is made to the source files

```
mkdir whitehousemr2_classes
```

recreating directory that will hold the compiled java classes

```
cd src
```

change directory to source directory

```
javac -classpath `yarn classpath` -d ../whitehousemr2_classes WhiteHouseMR2.java
```

compiling java source file in src directory and saving it in the compiled java classes directory

```
cd ..
```

coming out of src directory to the project root

```
jar -cvf WhiteHouseMR2.jar  -C  whitehousemr2_classes/ .
```

generating a JAR file in the project root using the compiled classes in the whitehousemr2_classes directory

```
hadoop fs -rm -r /WhiteHouseMR2
```

removing any previously created HDFS main directory for this job, which may contain output folder, as hadoop expects output directory to not exist beforehand

```
hadoop fs -mkdir /WhiteHouseMR2
```

recreating the job specific main directory in HDFS

```
hadoop fs -mkdir /WhiteHouseMR2/input
```

creating an sub directory in the HDFS main directory created in the previous step for copying input data

```
hadoop fs -copyFromLocal WhiteHouse-WAVESReleased-0827.csv /WhiteHouseMR2/input
```

copying input data from local (available in the current project directory root) to the HDFS input data directory created in last step

```
yarn jar WhiteHouseMR2.jar WhiteHouseMR2
/WhiteHouseMR2/input   /WhiteHouseMR2/intermediate
/WhiteHouseMR2/output
```

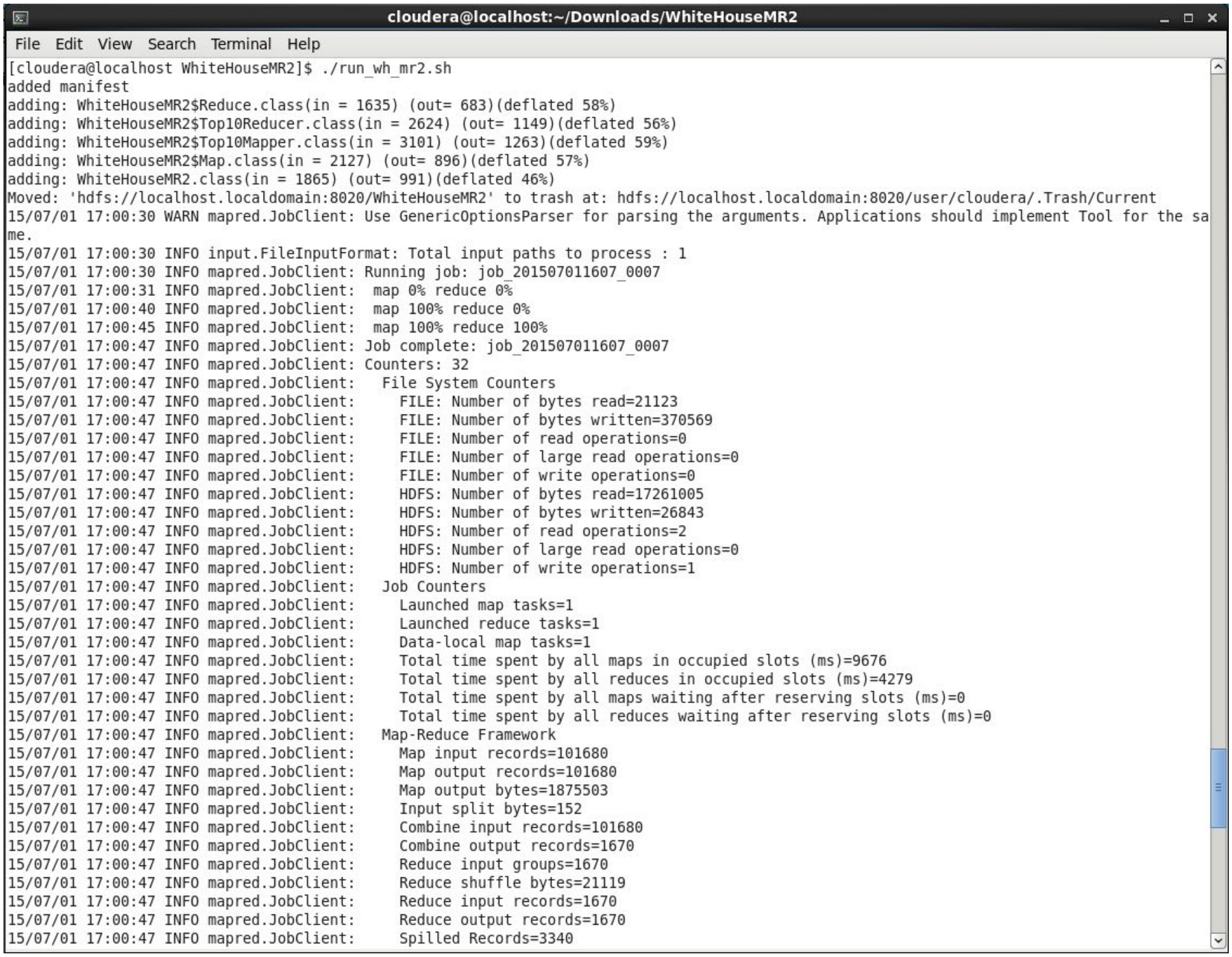
running MapReduce job using the "yarn jar" command which takes as other parameters the JAR file name, Java Class Name with main() method , HDFS input , intermediate and output directories

hadoop fs -cat /WhiteHouseMR2/output/part-r-00000 | sort -nrk 1

displaying the output generated in the output file part-r-00000 and piping it to sort in descending order by the first column

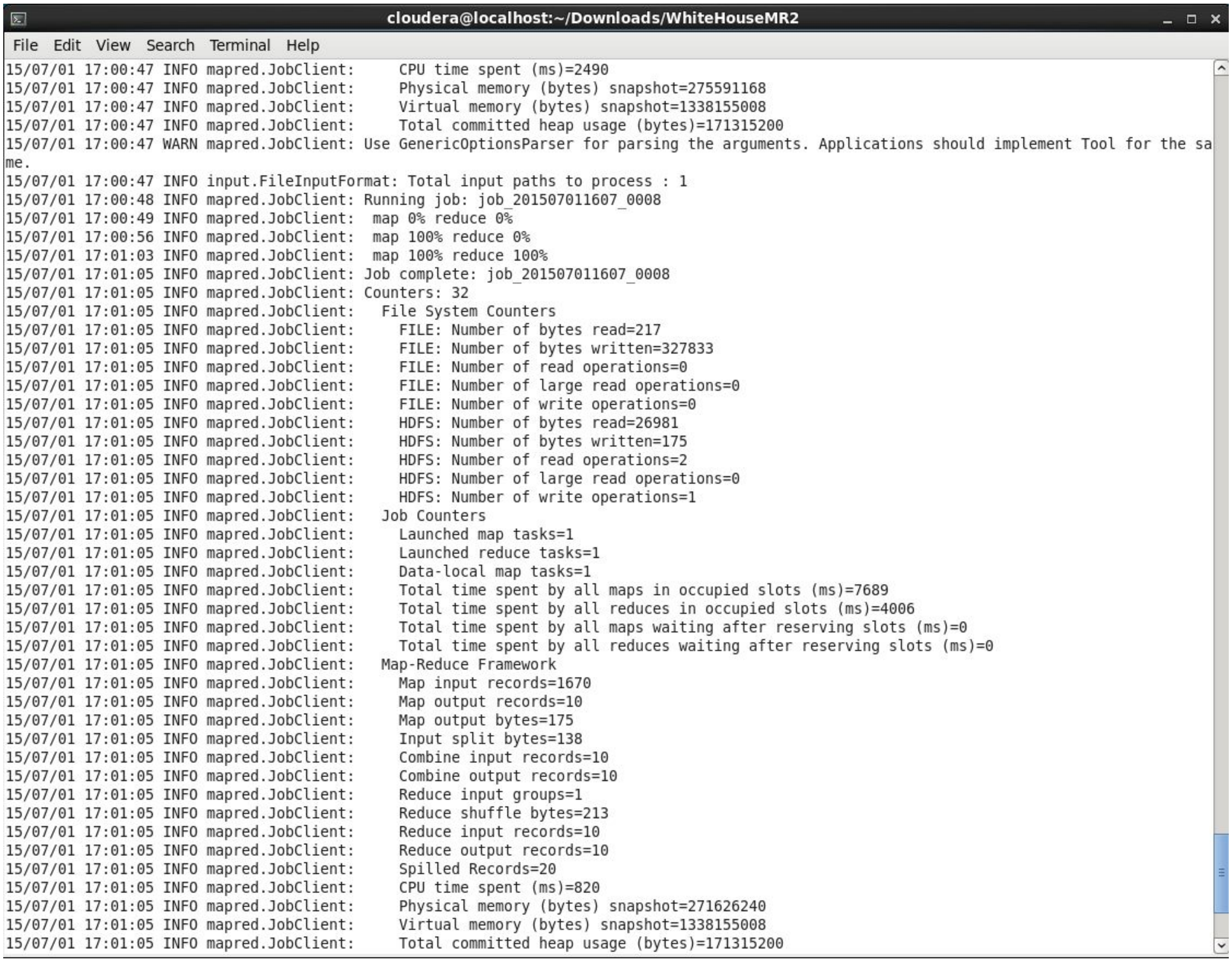
Note : This shell script should be run from the root of the project directory, where it is residing.

Execution Process :


cloudera@localhost:~/Downloads/WhiteHouseMR2
File Edit View Search Terminal Help

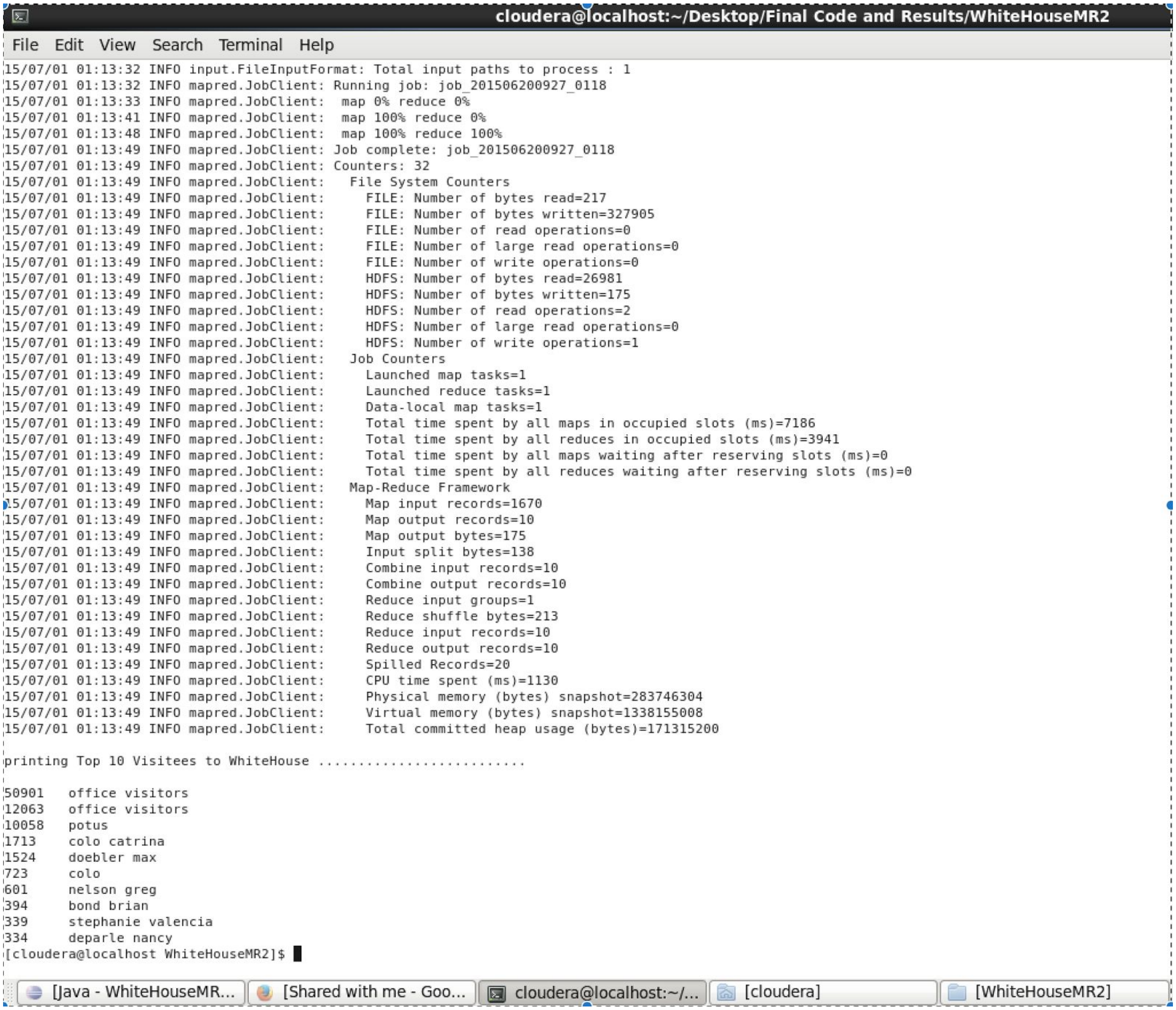
```
[cloudera@localhost WhiteHouseMR2]$ ./run_wh_mr2.sh
added manifest
adding: WhiteHouseMR2$Reduce.class(in = 1635) (out= 683)(deflated 58%)
adding: WhiteHouseMR2$Top10Reducer.class(in = 2624) (out= 1149)(deflated 56%)
adding: WhiteHouseMR2$Top10Mapper.class(in = 3101) (out= 1263)(deflated 59%)
adding: WhiteHouseMR2$Map.class(in = 2127) (out= 896)(deflated 57%)
adding: WhiteHouseMR2.class(in = 1865) (out= 991)(deflated 46%)
Moved: 'hdfs://localhost.localdomain:8020/WhiteHouseMR2' to trash at: hdfs://localhost.localdomain:8020/user/cloudera/.Trash/Current
15/07/01 17:00:30 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the sa
me.
15/07/01 17:00:30 INFO input.FileInputFormat: Total input paths to process : 1
15/07/01 17:00:30 INFO mapred.JobClient: Running job: job_201507011607_0007
15/07/01 17:00:31 INFO mapred.JobClient:  map 0% reduce 0%
15/07/01 17:00:40 INFO mapred.JobClient:  map 100% reduce 0%
15/07/01 17:00:45 INFO mapred.JobClient:  map 100% reduce 100%
15/07/01 17:00:47 INFO mapred.JobClient: Job complete: job_201507011607_0007
15/07/01 17:00:47 INFO mapred.JobClient: Counters: 32
15/07/01 17:00:47 INFO mapred.JobClient:   File System Counters
15/07/01 17:00:47 INFO mapred.JobClient:     FILE: Number of bytes read=21123
15/07/01 17:00:47 INFO mapred.JobClient:     FILE: Number of bytes written=370569
15/07/01 17:00:47 INFO mapred.JobClient:     FILE: Number of read operations=0
15/07/01 17:00:47 INFO mapred.JobClient:     FILE: Number of large read operations=0
15/07/01 17:00:47 INFO mapred.JobClient:     FILE: Number of write operations=0
15/07/01 17:00:47 INFO mapred.JobClient:     HDFS: Number of bytes read=17261005
15/07/01 17:00:47 INFO mapred.JobClient:     HDFS: Number of bytes written=26843
15/07/01 17:00:47 INFO mapred.JobClient:     HDFS: Number of read operations=2
15/07/01 17:00:47 INFO mapred.JobClient:     HDFS: Number of large read operations=0
15/07/01 17:00:47 INFO mapred.JobClient:     HDFS: Number of write operations=1
15/07/01 17:00:47 INFO mapred.JobClient:   Job Counters
15/07/01 17:00:47 INFO mapred.JobClient:     Launched map tasks=1
15/07/01 17:00:47 INFO mapred.JobClient:     Launched reduce tasks=1
15/07/01 17:00:47 INFO mapred.JobClient:     Data-local map tasks=1
15/07/01 17:00:47 INFO mapred.JobClient:     Total time spent by all maps in occupied slots (ms)=9676
15/07/01 17:00:47 INFO mapred.JobClient:     Total time spent by all reduces in occupied slots (ms)=4279
15/07/01 17:00:47 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
15/07/01 17:00:47 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
15/07/01 17:00:47 INFO mapred.JobClient:   Map-Reduce Framework
15/07/01 17:00:47 INFO mapred.JobClient:     Map input records=101680
15/07/01 17:00:47 INFO mapred.JobClient:     Map output records=101680
15/07/01 17:00:47 INFO mapred.JobClient:     Map output bytes=1875503
15/07/01 17:00:47 INFO mapred.JobClient:     Input split bytes=152
15/07/01 17:00:47 INFO mapred.JobClient:     Combine input records=101680
15/07/01 17:00:47 INFO mapred.JobClient:     Combine output records=1670
15/07/01 17:00:47 INFO mapred.JobClient:     Reduce input groups=1670
15/07/01 17:00:47 INFO mapred.JobClient:     Reduce shuffle bytes=21119
15/07/01 17:00:47 INFO mapred.JobClient:     Reduce input records=1670
15/07/01 17:00:47 INFO mapred.JobClient:     Reduce output records=1670
15/07/01 17:00:47 INFO mapred.JobClient:     Spilled Records=3340
```

Execution Process Continues..


cloudera@localhost:~/Downloads/WhiteHouseMR2
File Edit View Search Terminal Help

```
15/07/01 17:00:47 INFO mapred.JobClient:     CPU time spent (ms)=2490
15/07/01 17:00:47 INFO mapred.JobClient:     Physical memory (bytes) snapshot=275591168
15/07/01 17:00:47 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=1338155008
15/07/01 17:00:47 INFO mapred.JobClient:     Total committed heap usage (bytes)=171315200
15/07/01 17:00:47 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the sa
me.
15/07/01 17:00:47 INFO input.FileInputFormat: Total input paths to process : 1
15/07/01 17:00:48 INFO mapred.JobClient: Running job: job_201507011607_0008
15/07/01 17:00:49 INFO mapred.JobClient:  map 0% reduce 0%
15/07/01 17:00:56 INFO mapred.JobClient:  map 100% reduce 0%
15/07/01 17:01:03 INFO mapred.JobClient:  map 100% reduce 100%
15/07/01 17:01:05 INFO mapred.JobClient: Job complete: job_201507011607_0008
15/07/01 17:01:05 INFO mapred.JobClient: Counters: 32
15/07/01 17:01:05 INFO mapred.JobClient:   File System Counters
15/07/01 17:01:05 INFO mapred.JobClient:     FILE: Number of bytes read=217
15/07/01 17:01:05 INFO mapred.JobClient:     FILE: Number of bytes written=327833
15/07/01 17:01:05 INFO mapred.JobClient:     FILE: Number of read operations=0
15/07/01 17:01:05 INFO mapred.JobClient:     FILE: Number of large read operations=0
15/07/01 17:01:05 INFO mapred.JobClient:     FILE: Number of write operations=0
15/07/01 17:01:05 INFO mapred.JobClient:     HDFS: Number of bytes read=26981
15/07/01 17:01:05 INFO mapred.JobClient:     HDFS: Number of bytes written=175
15/07/01 17:01:05 INFO mapred.JobClient:     HDFS: Number of read operations=2
15/07/01 17:01:05 INFO mapred.JobClient:     HDFS: Number of large read operations=0
15/07/01 17:01:05 INFO mapred.JobClient:     HDFS: Number of write operations=1
15/07/01 17:01:05 INFO mapred.JobClient:   Job Counters
15/07/01 17:01:05 INFO mapred.JobClient:     Launched map tasks=1
15/07/01 17:01:05 INFO mapred.JobClient:     Launched reduce tasks=1
15/07/01 17:01:05 INFO mapred.JobClient:     Data-local map tasks=1
15/07/01 17:01:05 INFO mapred.JobClient:     Total time spent by all maps in occupied slots (ms)=7689
15/07/01 17:01:05 INFO mapred.JobClient:     Total time spent by all reduces in occupied slots (ms)=4006
15/07/01 17:01:05 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
15/07/01 17:01:05 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
15/07/01 17:01:05 INFO mapred.JobClient:   Map-Reduce Framework
15/07/01 17:01:05 INFO mapred.JobClient:     Map input records=1670
15/07/01 17:01:05 INFO mapred.JobClient:     Map output records=10
15/07/01 17:01:05 INFO mapred.JobClient:     Map output bytes=175
15/07/01 17:01:05 INFO mapred.JobClient:     Input split bytes=138
15/07/01 17:01:05 INFO mapred.JobClient:     Combine input records=10
15/07/01 17:01:05 INFO mapred.JobClient:     Combine output records=10
15/07/01 17:01:05 INFO mapred.JobClient:     Reduce input groups=1
15/07/01 17:01:05 INFO mapred.JobClient:     Reduce shuffle bytes=213
15/07/01 17:01:05 INFO mapred.JobClient:     Reduce input records=10
15/07/01 17:01:05 INFO mapred.JobClient:     Reduce output records=10
15/07/01 17:01:05 INFO mapred.JobClient:     Spilled Records=20
15/07/01 17:01:05 INFO mapred.JobClient:     CPU time spent (ms)=820
15/07/01 17:01:05 INFO mapred.JobClient:     Physical memory (bytes) snapshot=271626240
15/07/01 17:01:05 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=1338155008
15/07/01 17:01:05 INFO mapred.JobClient:     Total committed heap usage (bytes)=171315200
```

Final Output :



cloudera@localhost:~/Desktop/Final Code and Results/WhiteHouseMR2
File Edit View Search Terminal Help
[Java - WhiteHouseMR...
[Shared with me - Goo...
cloudera@localhost:~/...
[cloudera]
[WhiteHouseMR2]

```
15/07/01 01:13:32 INFO input.FileInputFormat: Total input paths to process : 1
15/07/01 01:13:32 INFO mapred.JobClient: Running job: job_201506200927_0118
15/07/01 01:13:33 INFO mapred.JobClient:  map 0% reduce 0%
15/07/01 01:13:41 INFO mapred.JobClient:  map 100% reduce 0%
15/07/01 01:13:48 INFO mapred.JobClient:  map 100% reduce 100%
15/07/01 01:13:49 INFO mapred.JobClient: Job complete: job_201506200927_0118
15/07/01 01:13:49 INFO mapred.JobClient: Counters: 32
15/07/01 01:13:49 INFO mapred.JobClient:   File System Counters
15/07/01 01:13:49 INFO mapred.JobClient:     FILE: Number of bytes read=217
15/07/01 01:13:49 INFO mapred.JobClient:     FILE: Number of bytes written=327905
15/07/01 01:13:49 INFO mapred.JobClient:     FILE: Number of read operations=0
15/07/01 01:13:49 INFO mapred.JobClient:     FILE: Number of large read operations=0
15/07/01 01:13:49 INFO mapred.JobClient:     FILE: Number of write operations=0
15/07/01 01:13:49 INFO mapred.JobClient:     HDFS: Number of bytes read=26981
15/07/01 01:13:49 INFO mapred.JobClient:     HDFS: Number of bytes written=175
15/07/01 01:13:49 INFO mapred.JobClient:     HDFS: Number of read operations=2
15/07/01 01:13:49 INFO mapred.JobClient:     HDFS: Number of large read operations=0
15/07/01 01:13:49 INFO mapred.JobClient:     HDFS: Number of write operations=1
15/07/01 01:13:49 INFO mapred.JobClient:   Job Counters
15/07/01 01:13:49 INFO mapred.JobClient:     Launched map tasks=1
15/07/01 01:13:49 INFO mapred.JobClient:     Launched reduce tasks=1
15/07/01 01:13:49 INFO mapred.JobClient:     Data-local map tasks=1
15/07/01 01:13:49 INFO mapred.JobClient:     Total time spent by all maps in occupied slots (ms)=7186
15/07/01 01:13:49 INFO mapred.JobClient:     Total time spent by all reduces in occupied slots (ms)=3941
15/07/01 01:13:49 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
15/07/01 01:13:49 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
15/07/01 01:13:49 INFO mapred.JobClient:   Map-Reduce Framework
15/07/01 01:13:49 INFO mapred.JobClient:     Map input records=1670
15/07/01 01:13:49 INFO mapred.JobClient:     Map output records=10
15/07/01 01:13:49 INFO mapred.JobClient:     Map output bytes=175
15/07/01 01:13:49 INFO mapred.JobClient:     Input split bytes=138
15/07/01 01:13:49 INFO mapred.JobClient:     Combine input records=10
15/07/01 01:13:49 INFO mapred.JobClient:     Combine output records=10
15/07/01 01:13:49 INFO mapred.JobClient:     Reduce input groups=1
15/07/01 01:13:49 INFO mapred.JobClient:     Reduce shuffle bytes=213
15/07/01 01:13:49 INFO mapred.JobClient:     Reduce input records=10
15/07/01 01:13:49 INFO mapred.JobClient:     Reduce output records=10
15/07/01 01:13:49 INFO mapred.JobClient:     Spilled Records=20
15/07/01 01:13:49 INFO mapred.JobClient:     CPU time spent (ms)=1130
15/07/01 01:13:49 INFO mapred.JobClient:     Physical memory (bytes) snapshot=283746304
15/07/01 01:13:49 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=1338155008
15/07/01 01:13:49 INFO mapred.JobClient:     Total committed heap usage (bytes)=171315200

printing Top 10 Visitees to WhiteHouse .........................

50901   office visitors
12063   office visitors
10058   potus
1713    colo catrina
1524    doebler max
723     colo
601     nelson greg
394     bond brian
339     stephanie valencia
334     deparle nancy
[cloudera@localhost WhiteHouseMR2]$ 
```

Output Explanation :

The final output produced by this program is saved in the output directory /WhiteHouseMR2/output of HDFS. The file part-r-00000 in this output contains the top 10 visitees in White House in descending order. The two columns in output files are the Visit Count, (visitee_namelast, visitee_namefirst combo) in that order. This MapReduce main job included 2 sub jobs, first job dealt with identifying the visitees count for all the records in the data set. Later this job was chained with another job to only get the top 10 visitee count . Because of this, apart from the input and output folders in HDFS corresponding to this job, there was also an additional intermediate directory to write the output of the first job (/WhiteHouseMR2/intermediate), from where job 2 read the data to produce the final output .

(iii)

What this MapReduce program does :

The java file WhiteHouseMR3.java, contains all the hadoop job specific logic, which includes the mapper, reducer, Top Ten Mapper, Top Ten Reducer and driver etc.

This MapReduce program has 2 jobs which are chained to produce the final output.

For Job 1 :

In the Map Stage, records are emitted one by one to the map() tasks, which are then split into fields and the required fields (Visitor Name Fields and Visitee Name Fields clubbed together) is taken as key and value as 1

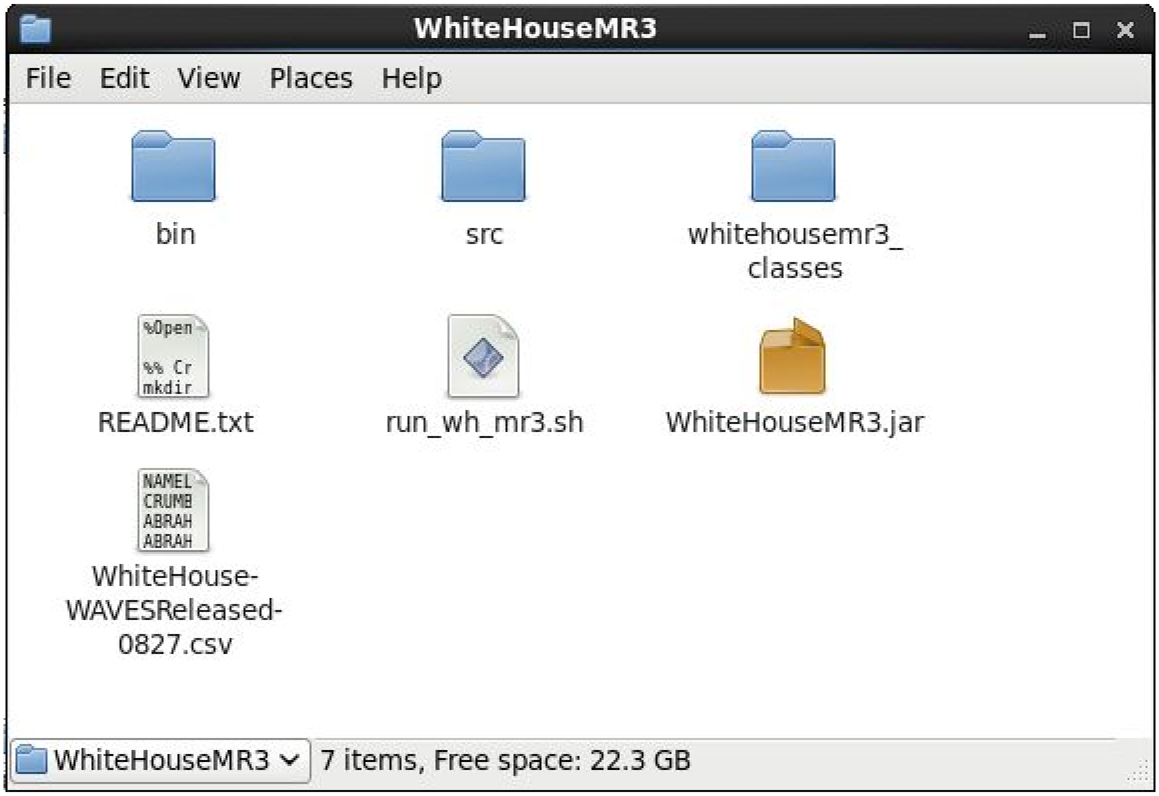
In the Reduce Stage, all the values of the same key (Visitor Name Fields and Visitee Name Fields combo) are aggregated to get the overall count for each single Visitor-Visitee combination in the data set.

Output of job1 is saved into intermediate folder in HDFS to be read by the job 2.

For Job 2 :

The Top Ten Mapper and Top Ten Reducer are used to filter out the top 10 Visitor-Visitee combination and the output of this job is written to final output directory

Project Folder Structure :



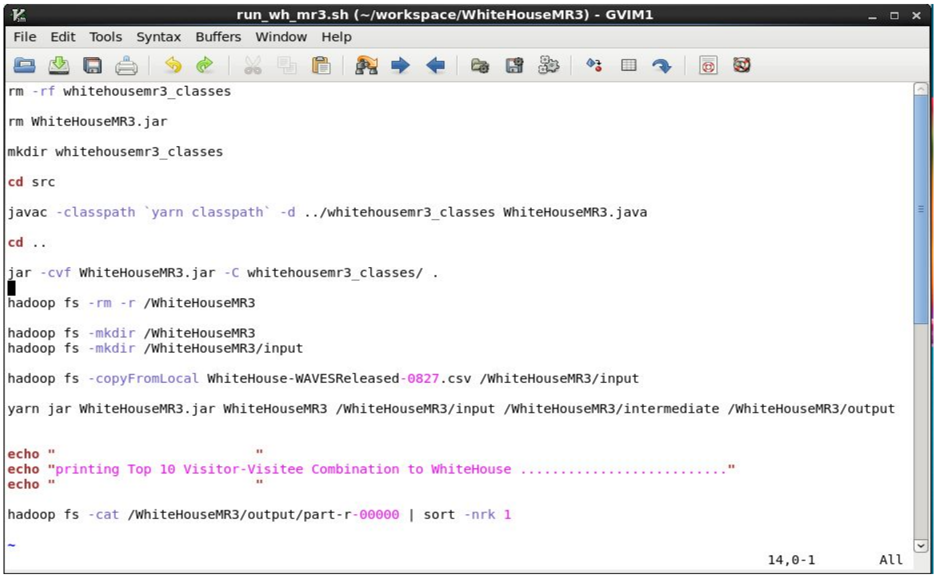
Project Source Folder Structure and Contents explanation:

The source folder contains following directories and files :

bin	This folder contains all the binaries
src	This folder contains the java file WhiteHouseMR3.java

whitehousemr3_classes	All the compiled Java Classes are in this folder
WhiteHouse-WAVESReleased-0827.csv	Input data for the MapReduce job
WhiteHouseMR3.jar	The Executable JAR generated for the MapReduce job
run_wh_mr3.sh	The shell script that automates compilation of classes, generation of JAR, creating input directory in HDFS , copying input data to the input folder in HDFS , running the map reduce job and saving the output in an output directory in HDFS , printing out necessary output from the output file.
README.txt	This file has step by step procedure of how a JAR is created given a Java Project with just src and bin folders.and how to execute it .

Shell Script :


run_wh_mr3.sh (~/workspace/WhiteHouseMR3) - GVIM1
File Edit Tools Syntax Buffers Window Help
14,0-1 All

```
rm -rf whitehousemr3_classes

rm WhiteHouseMR3.jar

mkdir whitehousemr3_classes

cd src

javac -classpath `yarn classpath` -d ../whitehousemr3_classes WhiteHouseMR3.java

cd ..

jar -cvf WhiteHouseMR3.jar -C whitehousemr3_classes/ .

hadoop fs -rm -r /WhiteHouseMR3

hadoop fs -mkdir /WhiteHouseMR3
hadoop fs -mkdir /WhiteHouseMR3/input

hadoop fs -copyFromLocal WhiteHouse-WAVESReleased-0827.csv /WhiteHouseMR3/input

yarn jar WhiteHouseMR3.jar WhiteHouseMR3 /WhiteHouseMR3/input /WhiteHouseMR3/intermediate /WhiteHouseMR3/output


echo "                         "
echo "printing Top 10 Visitor-Visitee Combination to WhiteHouse ..........................."
echo "                         "

hadoop fs -cat /WhiteHouseMR3/output/part-r-00000 | sort -nrk 1

~
```

Explanation of Shell Script :

The shell script contains all the steps required to automate running a MapReduce job from within the project folder structure with just src and bin folder along with input data in it.

```
rm -rf whitehousemr3_classes
```

removing the directory which holds all previously compiled java classes

```
rm WhiteHouseMR3.jar
```

removing previously generated JAR files, just in case any change is made to the source files

```
mkdir whitehousemr3_classes
```

recreating directory that will hold the compiled java classes

```
cd src
```

change directory to source directory

```
javac -classpath `yarn classpath` -d ../whitehousemr3_classes WhiteHouseMR3.java
```

compiling java source file in src directory and saving it in the compiled java classes directory

```
cd ..
```

coming out of src directory to the project root

```
jar -cvf WhiteHouseMR3.jar  -C  whitehousemr3_classes/ .
```

generating a JAR file in the project root using the compiled classes in the whitehousemr3_classes directory

```
hadoop fs -rm -r /WhiteHouseMR3
```

removing any previously created HDFS main directory for this job, which may contain output folder, as hadoop expects output directory to not exist beforehand

```
hadoop fs -mkdir /WhiteHouseMR3
```

recreating the job specific main directory in HDFS

```
hadoop fs -mkdir /WhiteHouseMR3/input
```

creating an sub directory in the HDFS main directory created in the previous step for copying input data

```
hadoop fs -copyFromLocal WhiteHouse-WAVESReleased-0827.csv
/WhiteHouseMR3/input
```

copying input data from local (available in the current project directory root) to the HDFS input data directory created in last step

```
yarn jar WhiteHouseMR3.jar WhiteHouseMR3
/WhiteHouseMR3/input   /WhiteHouseMR3/intermediate
/WhiteHouseMR3/output
```

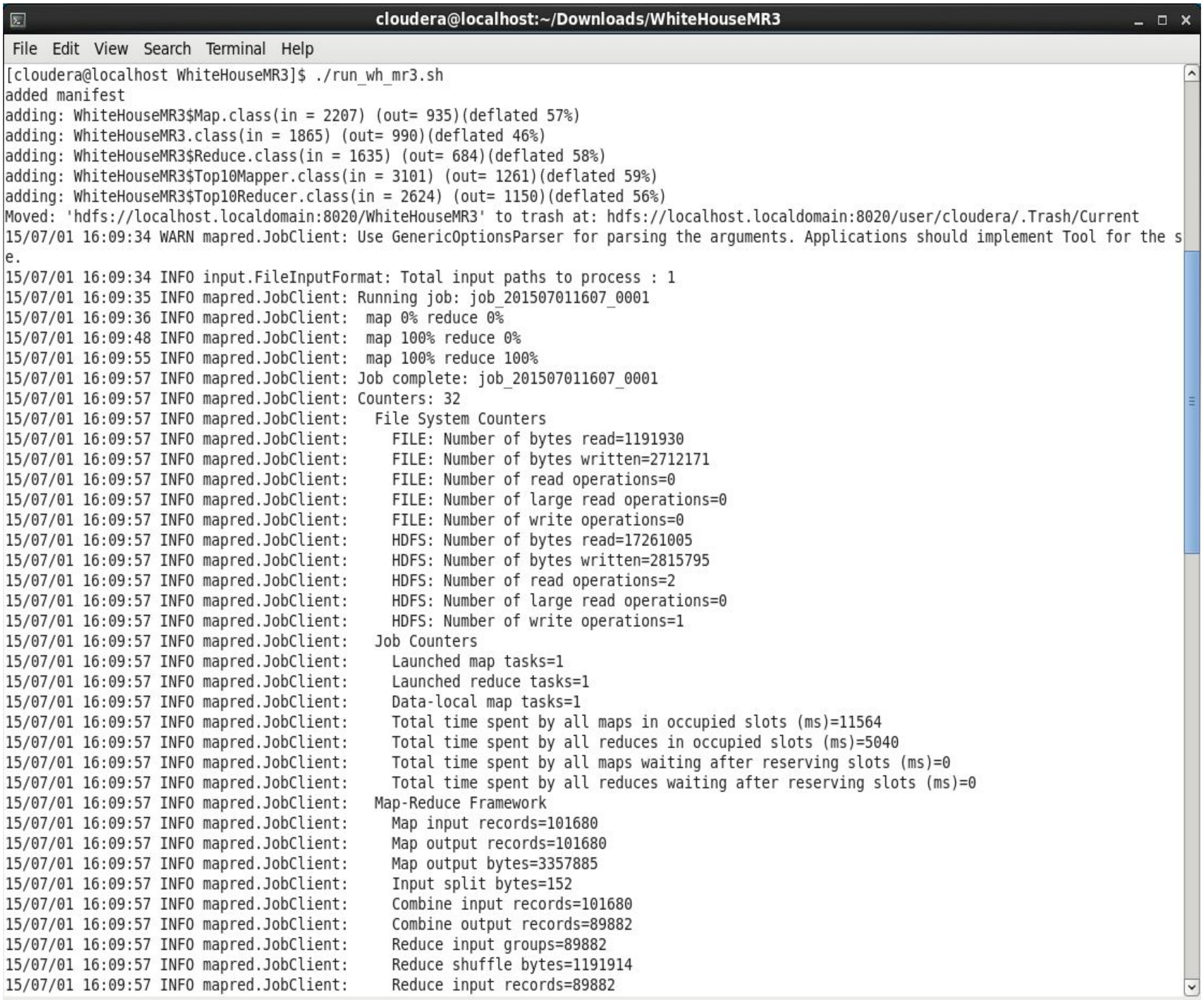
running MapReduce job using the "yarn jar" command which takes as other parameters the JAR file name, Java Class Name with main() method , HDFS input , intermediate and output directories

hadoop fs -cat /WhiteHouseMR3/output/part-r-00000 | sort -nrk 1

displaying the output generated in the output file part-r-00000 and piping it to sort in descending order by the first column

Note : This shell script should be run from the root of the project directory, where it is residing.

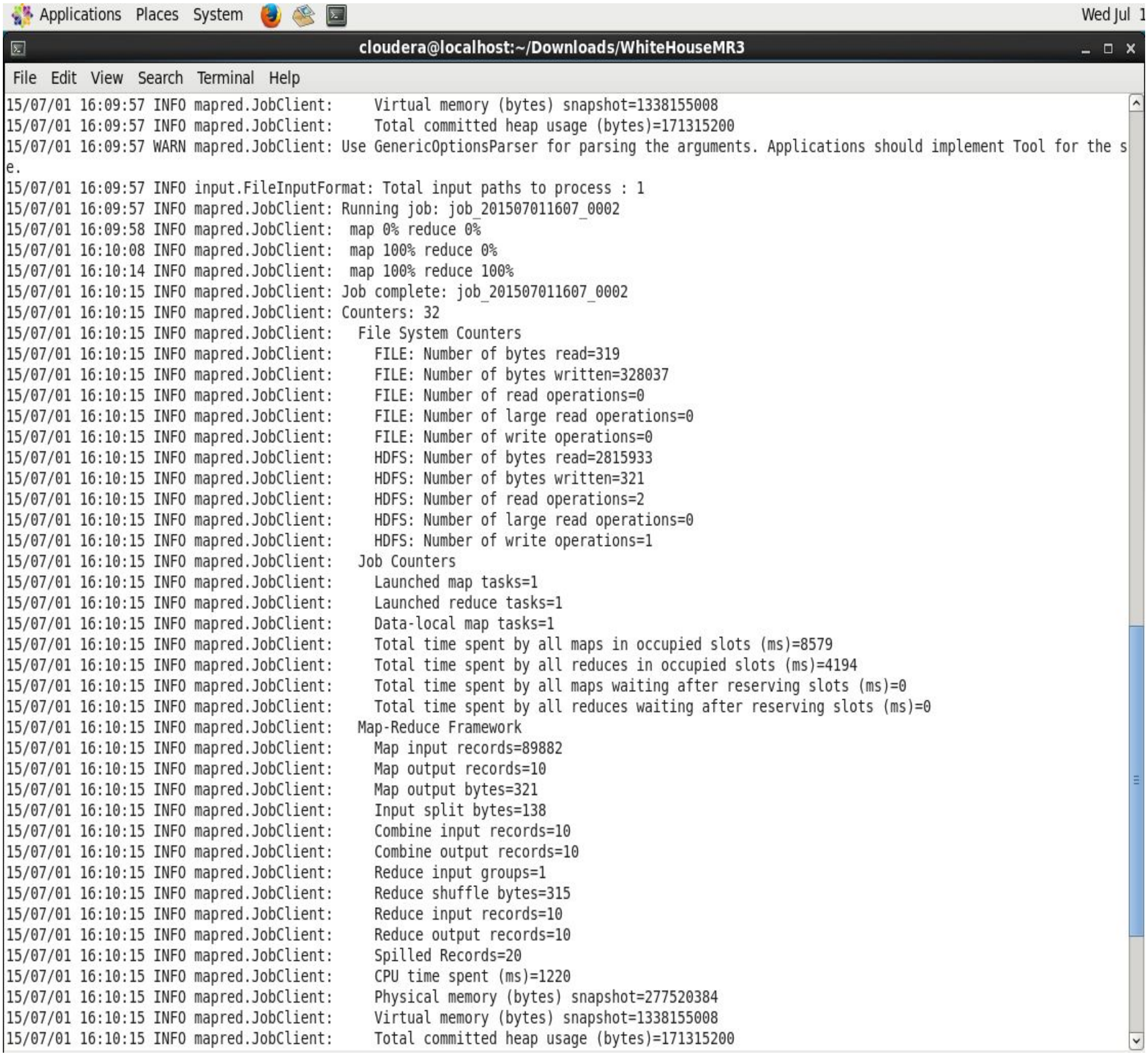
Execution Process :



cloudera@localhost:~/Downloads/WhiteHouseMR3
File Edit View Search Terminal Help

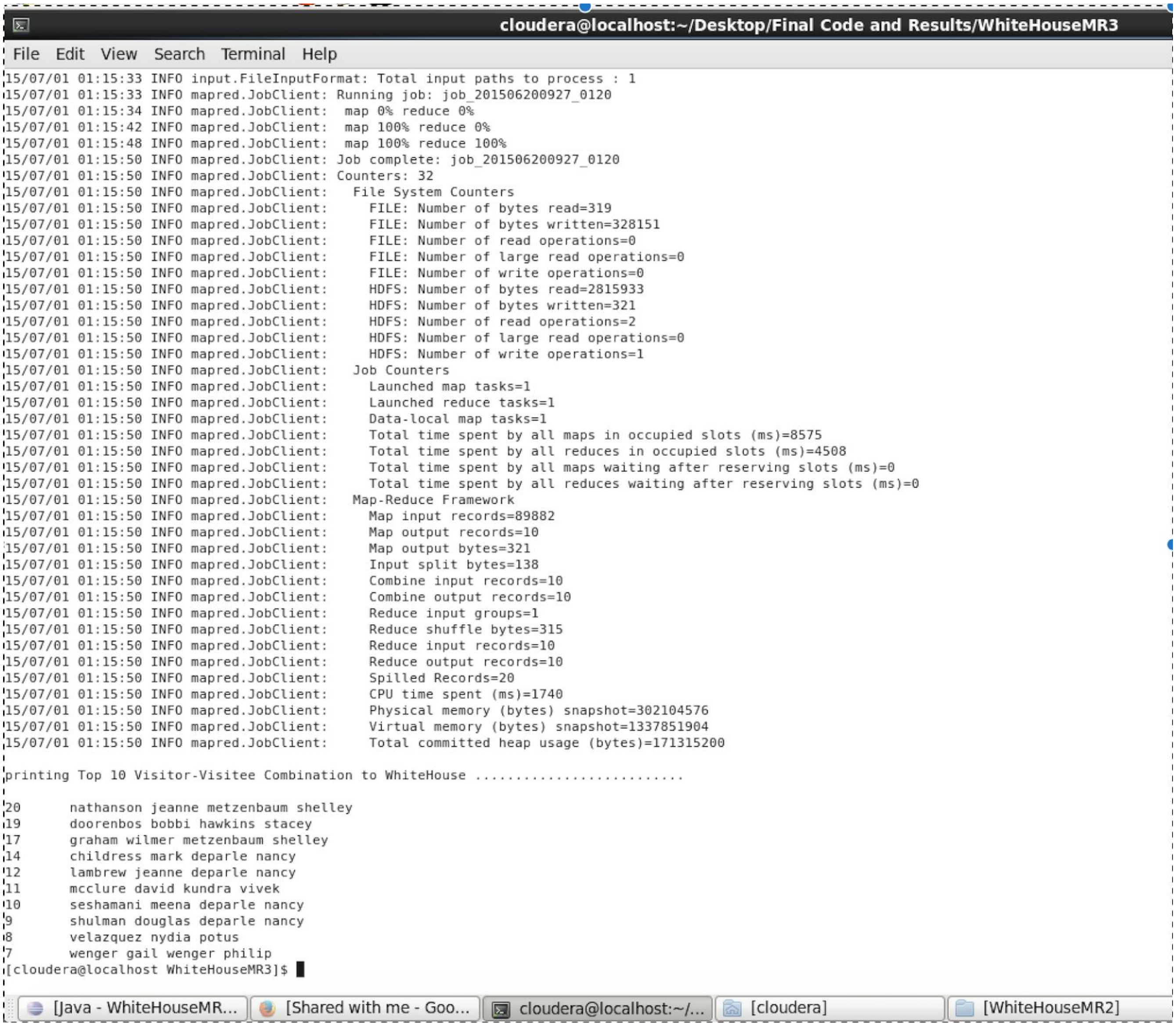
```
[cloudera@localhost WhiteHouseMR3]$ ./run_wh_mr3.sh
added manifest
adding: WhiteHouseMR3$Map.class(in = 2207) (out= 935)(deflated 57%)
adding: WhiteHouseMR3.class(in = 1865) (out= 990)(deflated 46%)
adding: WhiteHouseMR3$Reduce.class(in = 1635) (out= 684)(deflated 58%)
adding: WhiteHouseMR3$Top10Mapper.class(in = 3101) (out= 1261)(deflated 59%)
adding: WhiteHouseMR3$Top10Reducer.class(in = 2624) (out= 1150)(deflated 56%)
Moved: 'hdfs://localhost.localdomain:8020/WhiteHouseMR3' to trash at: hdfs://localhost.localdomain:8020/user/cloudera/.Trash/Current
15/07/01 16:09:34 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the s
e.
15/07/01 16:09:34 INFO input.FileInputFormat: Total input paths to process : 1
15/07/01 16:09:35 INFO mapred.JobClient: Running job: job_201507011607_0001
15/07/01 16:09:36 INFO mapred.JobClient:  map 0% reduce 0%
15/07/01 16:09:48 INFO mapred.JobClient:  map 100% reduce 0%
15/07/01 16:09:55 INFO mapred.JobClient:  map 100% reduce 100%
15/07/01 16:09:57 INFO mapred.JobClient: Job complete: job_201507011607_0001
15/07/01 16:09:57 INFO mapred.JobClient: Counters: 32
15/07/01 16:09:57 INFO mapred.JobClient:   File System Counters
15/07/01 16:09:57 INFO mapred.JobClient:     FILE: Number of bytes read=1191930
15/07/01 16:09:57 INFO mapred.JobClient:     FILE: Number of bytes written=2712171
15/07/01 16:09:57 INFO mapred.JobClient:     FILE: Number of read operations=0
15/07/01 16:09:57 INFO mapred.JobClient:     FILE: Number of large read operations=0
15/07/01 16:09:57 INFO mapred.JobClient:     FILE: Number of write operations=0
15/07/01 16:09:57 INFO mapred.JobClient:     HDFS: Number of bytes read=17261005
15/07/01 16:09:57 INFO mapred.JobClient:     HDFS: Number of bytes written=2815795
15/07/01 16:09:57 INFO mapred.JobClient:     HDFS: Number of read operations=2
15/07/01 16:09:57 INFO mapred.JobClient:     HDFS: Number of large read operations=0
15/07/01 16:09:57 INFO mapred.JobClient:     HDFS: Number of write operations=1
15/07/01 16:09:57 INFO mapred.JobClient:   Job Counters
15/07/01 16:09:57 INFO mapred.JobClient:     Launched map tasks=1
15/07/01 16:09:57 INFO mapred.JobClient:     Launched reduce tasks=1
15/07/01 16:09:57 INFO mapred.JobClient:     Data-local map tasks=1
15/07/01 16:09:57 INFO mapred.JobClient:     Total time spent by all maps in occupied slots (ms)=11564
15/07/01 16:09:57 INFO mapred.JobClient:     Total time spent by all reduces in occupied slots (ms)=5040
15/07/01 16:09:57 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
15/07/01 16:09:57 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
15/07/01 16:09:57 INFO mapred.JobClient:   Map-Reduce Framework
15/07/01 16:09:57 INFO mapred.JobClient:     Map input records=101680
15/07/01 16:09:57 INFO mapred.JobClient:     Map output records=101680
15/07/01 16:09:57 INFO mapred.JobClient:     Map output bytes=3357885
15/07/01 16:09:57 INFO mapred.JobClient:     Input split bytes=152
15/07/01 16:09:57 INFO mapred.JobClient:     Combine input records=101680
15/07/01 16:09:57 INFO mapred.JobClient:     Combine output records=89882
15/07/01 16:09:57 INFO mapred.JobClient:     Reduce input groups=89882
15/07/01 16:09:57 INFO mapred.JobClient:     Reduce shuffle bytes=1191914
15/07/01 16:09:57 INFO mapred.JobClient:     Reduce input records=89882
```

Execution Process Continues..


Applications Places System
Wed Jul 1
cloudera@localhost:~/Downloads/WhiteHouseMR3
File Edit View Search Terminal Help

```
15/07/01 16:09:57 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=1338155008
15/07/01 16:09:57 INFO mapred.JobClient:     Total committed heap usage (bytes)=171315200
15/07/01 16:09:57 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the s
e.
15/07/01 16:09:57 INFO input.FileInputFormat: Total input paths to process : 1
15/07/01 16:09:57 INFO mapred.JobClient: Running job: job_201507011607_0002
15/07/01 16:09:58 INFO mapred.JobClient:  map 0% reduce 0%
15/07/01 16:10:08 INFO mapred.JobClient:  map 100% reduce 0%
15/07/01 16:10:14 INFO mapred.JobClient:  map 100% reduce 100%
15/07/01 16:10:15 INFO mapred.JobClient: Job complete: job_201507011607_0002
15/07/01 16:10:15 INFO mapred.JobClient: Counters: 32
15/07/01 16:10:15 INFO mapred.JobClient:   File System Counters
15/07/01 16:10:15 INFO mapred.JobClient:     FILE: Number of bytes read=319
15/07/01 16:10:15 INFO mapred.JobClient:     FILE: Number of bytes written=328037
15/07/01 16:10:15 INFO mapred.JobClient:     FILE: Number of read operations=0
15/07/01 16:10:15 INFO mapred.JobClient:     FILE: Number of large read operations=0
15/07/01 16:10:15 INFO mapred.JobClient:     FILE: Number of write operations=0
15/07/01 16:10:15 INFO mapred.JobClient:     HDFS: Number of bytes read=2815933
15/07/01 16:10:15 INFO mapred.JobClient:     HDFS: Number of bytes written=321
15/07/01 16:10:15 INFO mapred.JobClient:     HDFS: Number of read operations=2
15/07/01 16:10:15 INFO mapred.JobClient:     HDFS: Number of large read operations=0
15/07/01 16:10:15 INFO mapred.JobClient:     HDFS: Number of write operations=1
15/07/01 16:10:15 INFO mapred.JobClient:   Job Counters
15/07/01 16:10:15 INFO mapred.JobClient:     Launched map tasks=1
15/07/01 16:10:15 INFO mapred.JobClient:     Launched reduce tasks=1
15/07/01 16:10:15 INFO mapred.JobClient:     Data-local map tasks=1
15/07/01 16:10:15 INFO mapred.JobClient:     Total time spent by all maps in occupied slots (ms)=8579
15/07/01 16:10:15 INFO mapred.JobClient:     Total time spent by all reduces in occupied slots (ms)=4194
15/07/01 16:10:15 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
15/07/01 16:10:15 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
15/07/01 16:10:15 INFO mapred.JobClient:   Map-Reduce Framework
15/07/01 16:10:15 INFO mapred.JobClient:     Map input records=89882
15/07/01 16:10:15 INFO mapred.JobClient:     Map output records=10
15/07/01 16:10:15 INFO mapred.JobClient:     Map output bytes=321
15/07/01 16:10:15 INFO mapred.JobClient:     Input split bytes=138
15/07/01 16:10:15 INFO mapred.JobClient:     Combine input records=10
15/07/01 16:10:15 INFO mapred.JobClient:     Combine output records=10
15/07/01 16:10:15 INFO mapred.JobClient:     Reduce input groups=1
15/07/01 16:10:15 INFO mapred.JobClient:     Reduce shuffle bytes=315
15/07/01 16:10:15 INFO mapred.JobClient:     Reduce input records=10
15/07/01 16:10:15 INFO mapred.JobClient:     Reduce output records=10
15/07/01 16:10:15 INFO mapred.JobClient:     Spilled Records=20
15/07/01 16:10:15 INFO mapred.JobClient:     CPU time spent (ms)=1220
15/07/01 16:10:15 INFO mapred.JobClient:     Physical memory (bytes) snapshot=277520384
15/07/01 16:10:15 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=1338155008
15/07/01 16:10:15 INFO mapred.JobClient:     Total committed heap usage (bytes)=171315200
```

Final Output :


cloudera@localhost:~/Desktop/Final Code and Results/WhiteHouseMR3
File Edit View Search Terminal Help
[Java - WhiteHouseMR...
[Shared with me - Goo...
cloudera@localhost:~/...
[cloudera]
[WhiteHouseMR2]

```
15/07/01 01:15:33 INFO input.FileInputFormat: Total input paths to process : 1
15/07/01 01:15:33 INFO mapred.JobClient: Running job: job_201506200927_0120
15/07/01 01:15:34 INFO mapred.JobClient:  map 0% reduce 0%
15/07/01 01:15:42 INFO mapred.JobClient:  map 100% reduce 0%
15/07/01 01:15:48 INFO mapred.JobClient:  map 100% reduce 100%
15/07/01 01:15:50 INFO mapred.JobClient: Job complete: job_201506200927_0120
15/07/01 01:15:50 INFO mapred.JobClient: Counters: 32
15/07/01 01:15:50 INFO mapred.JobClient:   File System Counters
15/07/01 01:15:50 INFO mapred.JobClient:     FILE: Number of bytes read=319
15/07/01 01:15:50 INFO mapred.JobClient:     FILE: Number of bytes written=328151
15/07/01 01:15:50 INFO mapred.JobClient:     FILE: Number of read operations=0
15/07/01 01:15:50 INFO mapred.JobClient:     FILE: Number of large read operations=0
15/07/01 01:15:50 INFO mapred.JobClient:     FILE: Number of write operations=0
15/07/01 01:15:50 INFO mapred.JobClient:     HDFS: Number of bytes read=2815933
15/07/01 01:15:50 INFO mapred.JobClient:     HDFS: Number of bytes written=321
15/07/01 01:15:50 INFO mapred.JobClient:     HDFS: Number of read operations=2
15/07/01 01:15:50 INFO mapred.JobClient:     HDFS: Number of large read operations=0
15/07/01 01:15:50 INFO mapred.JobClient:     HDFS: Number of write operations=1
15/07/01 01:15:50 INFO mapred.JobClient:   Job Counters
15/07/01 01:15:50 INFO mapred.JobClient:     Launched map tasks=1
15/07/01 01:15:50 INFO mapred.JobClient:     Launched reduce tasks=1
15/07/01 01:15:50 INFO mapred.JobClient:     Data-local map tasks=1
15/07/01 01:15:50 INFO mapred.JobClient:     Total time spent by all maps in occupied slots (ms)=8575
15/07/01 01:15:50 INFO mapred.JobClient:     Total time spent by all reduces in occupied slots (ms)=4508
15/07/01 01:15:50 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
15/07/01 01:15:50 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
15/07/01 01:15:50 INFO mapred.JobClient:   Map-Reduce Framework
15/07/01 01:15:50 INFO mapred.JobClient:     Map input records=89882
15/07/01 01:15:50 INFO mapred.JobClient:     Map output records=10
15/07/01 01:15:50 INFO mapred.JobClient:     Map output bytes=321
15/07/01 01:15:50 INFO mapred.JobClient:     Input split bytes=138
15/07/01 01:15:50 INFO mapred.JobClient:     Combine input records=10
15/07/01 01:15:50 INFO mapred.JobClient:     Combine output records=10
15/07/01 01:15:50 INFO mapred.JobClient:     Reduce input groups=1
15/07/01 01:15:50 INFO mapred.JobClient:     Reduce shuffle bytes=315
15/07/01 01:15:50 INFO mapred.JobClient:     Reduce input records=10
15/07/01 01:15:50 INFO mapred.JobClient:     Reduce output records=10
15/07/01 01:15:50 INFO mapred.JobClient:     Spilled Records=20
15/07/01 01:15:50 INFO mapred.JobClient:     CPU time spent (ms)=1740
15/07/01 01:15:50 INFO mapred.JobClient:     Physical memory (bytes) snapshot=302104576
15/07/01 01:15:50 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=1337851904
15/07/01 01:15:50 INFO mapred.JobClient:     Total committed heap usage (bytes)=171315200

printing Top 10 Visitor-Visitee Combination to WhiteHouse .........................

20      nathanson jeanne metzenbaum shelley
19      doorenbos bobbi hawkins stacey
17      graham wilmer metzenbaum shelley
14      childress mark deparle nancy
12      lambrew jeanne deparle nancy
11      mcclure david kundra vivek
10      seshamani meena deparle nancy
9       shulman douglas deparle nancy
8       velazquez nydia potus
7       wenger gail wenger philip
[cloudera@localhost WhiteHouseMR3]$ 
```

Output Explanation :

The final output produced by this program is saved in the output directory /WhiteHouseMR3/output of HDFS. The file part-r-00000 in this output contains the top 10 Visitor-Visitee combo of White House in descending order. The two columns in output files are the Visit Count, (Visitor Name Fields and Visitee Name Fields combo) in that order. This MapReduce main job included 2 sub jobs, first job dealt with identifying the Visitor-Visitee count for all the records in the data set. Later this job was chained with another job to only get the top 10 Visitor-Visitee count . Because of this, apart from the input and output folders in HDFS corresponding to this job, there was also an additional intermediate directory to write the output of the first job (/WhiteHouseMR3/intermediate), from where job 2 read the data to produce the final output .

(iv)

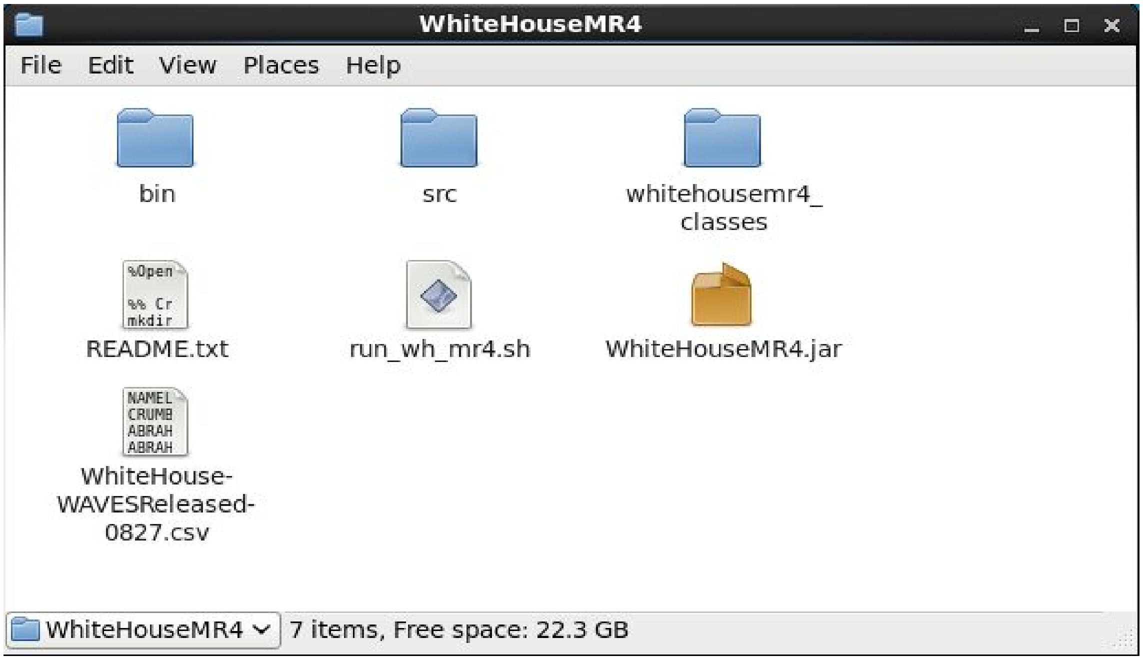
What this MapReduce program does :

The java file WhiteHouseMR4.java, contains all the hadoop job specific logic, which includes the mapper, reducer and driver etc.

In the Map Stage, records are emitted one by one to the map() tasks, which are then split into fields and the required field (APPT_MADE_DATE) is used to identify the key and value. Here the month part of this entire date was taken as key and corresponding year in this date was taken as value.

In the Reduce Stage, all the values of the same key (month) are aggregated to get the overall count for each single month in the data set and finally the average appointments made per month across all years are written to the output file, which yields our final output part-r-0000 . This output can be later used for human inspection or execution of next job or altogether a new program.

Project Folder Structure :



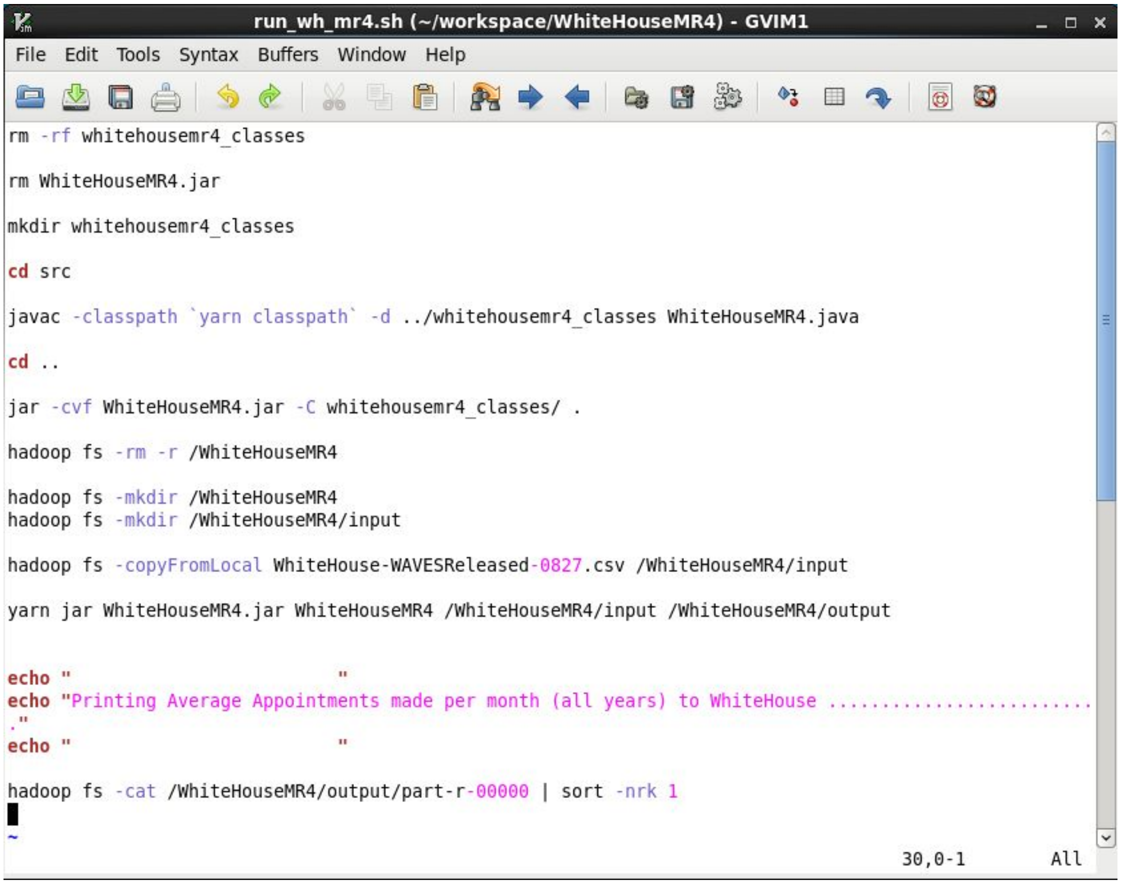
Project Source Folder Structure and Contents :

The source folder contains following directories and files :

bin	This folder contains all the binaries
src	This folder contains the java file WhiteHouseMR4.java
whitehousemr4_classes	All the compiled Java Classes are in this folder
WhiteHouse-WAVESReleased-0827.csv	Input data for the MapReduce job
WhiteHouseMR4.jar	The Executable JAR generated for the MapReduce job

run_wh_mr4.sh	The shell script that automates compilation of classes, generation of JAR, creating input directory in HDFS , copying input data to the input folder in HDFS , running the map reduce job and saving the output in an output directory in HDFS , printing out necessary output from the output file.
README.txt	This file has step by step procedure of how a JAR is created given a Java Project with just src and bin folders.and how to execute it .

Shell Script :



```
rm -rf whitehousemr4_classes

rm WhiteHouseMR4.jar

mkdir whitehousemr4_classes

cd src

javac -classpath `yarn classpath` -d ../whitehousemr4_classes WhiteHouseMR4.java

cd ..

jar -cvf WhiteHouseMR4.jar -C whitehousemr4_classes/ .

hadoop fs -rm -r /WhiteHouseMR4

hadoop fs -mkdir /WhiteHouseMR4
hadoop fs -mkdir /WhiteHouseMR4/input

hadoop fs -copyFromLocal WhiteHouse-WAVESReleased-0827.csv /WhiteHouseMR4/input

yarn jar WhiteHouseMR4.jar WhiteHouseMR4 /WhiteHouseMR4/input /WhiteHouseMR4/output


echo "                         "
echo "Printing Average Appointments made per month (all years) to WhiteHouse ........................
."
echo "                         "

hadoop fs -cat /WhiteHouseMR4/output/part-r-00000 | sort -nrk 1

~
                                                                                30,0-1        All
```

Explanation of Shell Script :

The shell script contains all the steps required to automate running a MapReduce job from within the project folder structure with just src and bin folder along with input data in it.

```
rm -rf whitehousemr4_classes
```

removing the directory which holds all previously compiled java classes

```
rm WhiteHouseMR4.jar
```

removing previously generated JAR files, just in case any change is made to the source files

```
mkdir whitehousemr4_classes
```

recreating directory that will hold the compiled java classes

```
cd src
```

change directory to source directory

```
javac -classpath  `yarn classpath`  -d  ../whitehousemr4_classes WhiteHouseMR4.java
```

compiling java source file in src directory and saving it in the compiled java classes directory

```
cd ..
```

coming out of src directory to the project root

```
jar -cvf WhiteHouseMR4.jar  -C  whitehousemr4_classes/ .
```

generating a JAR file in the project root using the compiled classes in the whitehousemr4_classes directory

```
hadoop fs -rm -r /WhiteHouseMR4
```

removing any previously created HDFS main directory for this job, which may contain output folder, as hadoop expects output directory to not exist beforehand

```
hadoop fs -mkdir /WhiteHouseMR4
```

recreating the job specific main directory in HDFS

```
hadoop fs -mkdir /WhiteHouseMR4/input
```

creating an sub directory in the HDFS main directory created in the previous step for copying input data

```
hadoop fs -copyFromLocal WhiteHouse-WAVESReleased-0827.csv
/WhiteHouseMR4/input
```

copying input data from local (available in the current project directory root) to the HDFS input data directory created in last step

```
yarn jar WhiteHouseMR4.jar WhiteHouseMR4
/WhiteHouseMR4/input  /WhiteHouseMR4/output
```

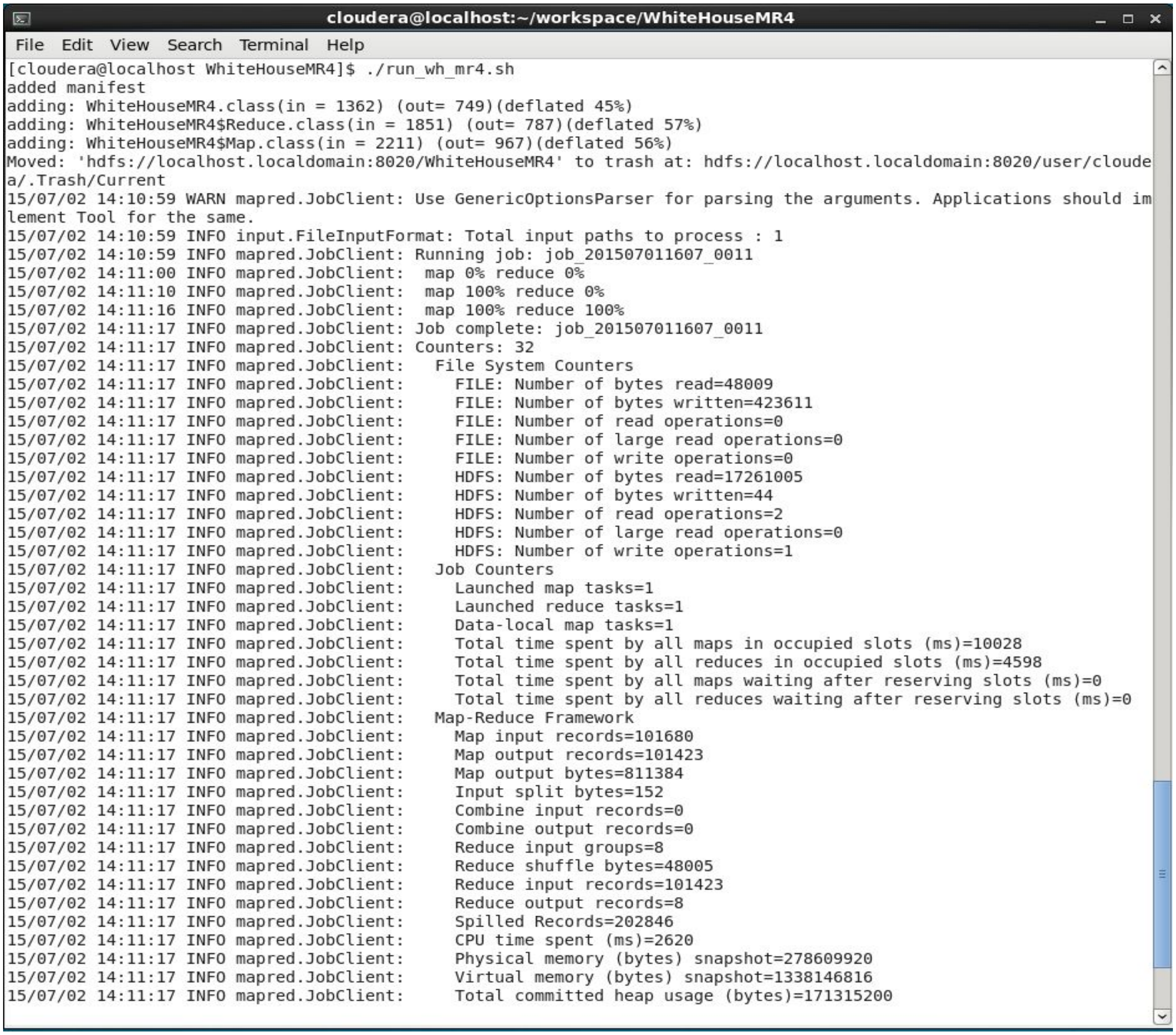
running MapReduce job using the "yarn jar" command which takes as other parameters the JAR file name, Java Class Name with main() method , HDFS input and output directories

hadoop fs -cat /WhiteHouseMR4/output/part-r-00000 | sort -nrk 1

displaying the output generated in the output file part-r-00000 and piping it to sort in descending order by the first column

Note : This shell script should be run from the root of the project directory, where it is residing.

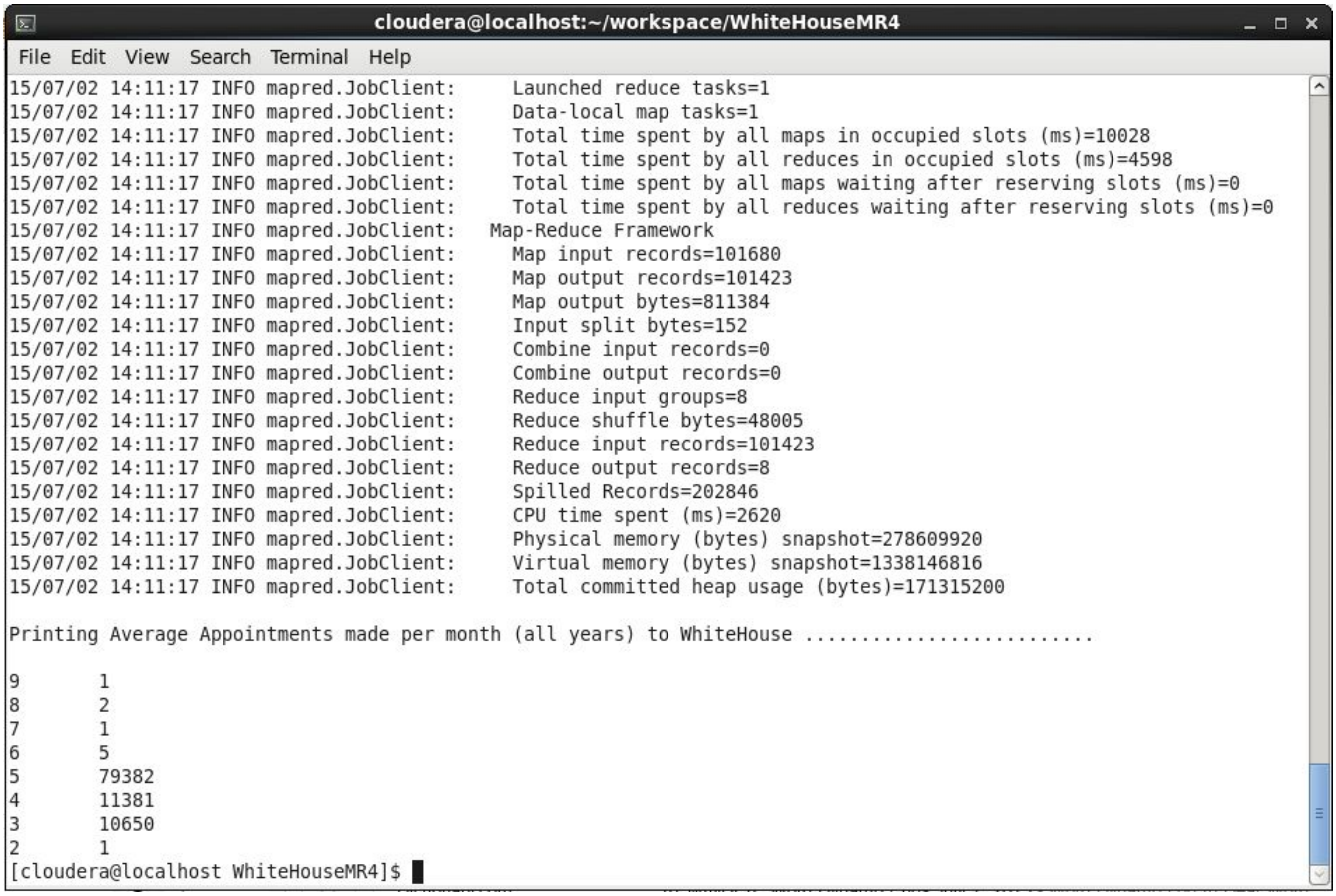
Execution Process :



cloudera@localhost:~/workspace/WhiteHouseMR4
File Edit View Search Terminal Help

```
[cloudera@localhost WhiteHouseMR4]$ ./run_wh_mr4.sh
added manifest
adding: WhiteHouseMR4.class(in = 1362) (out= 749)(deflated 45%)
adding: WhiteHouseMR4$Reduce.class(in = 1851) (out= 787)(deflated 57%)
adding: WhiteHouseMR4$Map.class(in = 2211) (out= 967)(deflated 56%)
Moved: 'hdfs://localhost.localdomain:8020/WhiteHouseMR4' to trash at: hdfs://localhost.localdomain:8020/user/cloude
a/.Trash/Current
15/07/02 14:10:59 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should im
lement Tool for the same.
15/07/02 14:10:59 INFO input.FileInputFormat: Total input paths to process : 1
15/07/02 14:10:59 INFO mapred.JobClient: Running job: job_201507011607_0011
15/07/02 14:11:00 INFO mapred.JobClient:  map 0% reduce 0%
15/07/02 14:11:10 INFO mapred.JobClient:  map 100% reduce 0%
15/07/02 14:11:16 INFO mapred.JobClient:  map 100% reduce 100%
15/07/02 14:11:17 INFO mapred.JobClient: Job complete: job_201507011607_0011
15/07/02 14:11:17 INFO mapred.JobClient: Counters: 32
15/07/02 14:11:17 INFO mapred.JobClient:   File System Counters
15/07/02 14:11:17 INFO mapred.JobClient:     FILE: Number of bytes read=48009
15/07/02 14:11:17 INFO mapred.JobClient:     FILE: Number of bytes written=423611
15/07/02 14:11:17 INFO mapred.JobClient:     FILE: Number of read operations=0
15/07/02 14:11:17 INFO mapred.JobClient:     FILE: Number of large read operations=0
15/07/02 14:11:17 INFO mapred.JobClient:     FILE: Number of write operations=0
15/07/02 14:11:17 INFO mapred.JobClient:     HDFS: Number of bytes read=17261005
15/07/02 14:11:17 INFO mapred.JobClient:     HDFS: Number of bytes written=44
15/07/02 14:11:17 INFO mapred.JobClient:     HDFS: Number of read operations=2
15/07/02 14:11:17 INFO mapred.JobClient:     HDFS: Number of large read operations=0
15/07/02 14:11:17 INFO mapred.JobClient:     HDFS: Number of write operations=1
15/07/02 14:11:17 INFO mapred.JobClient:   Job Counters
15/07/02 14:11:17 INFO mapred.JobClient:     Launched map tasks=1
15/07/02 14:11:17 INFO mapred.JobClient:     Launched reduce tasks=1
15/07/02 14:11:17 INFO mapred.JobClient:     Data-local map tasks=1
15/07/02 14:11:17 INFO mapred.JobClient:     Total time spent by all maps in occupied slots (ms)=10028
15/07/02 14:11:17 INFO mapred.JobClient:     Total time spent by all reduces in occupied slots (ms)=4598
15/07/02 14:11:17 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
15/07/02 14:11:17 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
15/07/02 14:11:17 INFO mapred.JobClient:   Map-Reduce Framework
15/07/02 14:11:17 INFO mapred.JobClient:     Map input records=101680
15/07/02 14:11:17 INFO mapred.JobClient:     Map output records=101423
15/07/02 14:11:17 INFO mapred.JobClient:     Map output bytes=811384
15/07/02 14:11:17 INFO mapred.JobClient:     Input split bytes=152
15/07/02 14:11:17 INFO mapred.JobClient:     Combine input records=0
15/07/02 14:11:17 INFO mapred.JobClient:     Combine output records=0
15/07/02 14:11:17 INFO mapred.JobClient:     Reduce input groups=8
15/07/02 14:11:17 INFO mapred.JobClient:     Reduce shuffle bytes=48005
15/07/02 14:11:17 INFO mapred.JobClient:     Reduce input records=101423
15/07/02 14:11:17 INFO mapred.JobClient:     Reduce output records=8
15/07/02 14:11:17 INFO mapred.JobClient:     Spilled Records=202846
15/07/02 14:11:17 INFO mapred.JobClient:     CPU time spent (ms)=2620
15/07/02 14:11:17 INFO mapred.JobClient:     Physical memory (bytes) snapshot=278609920
15/07/02 14:11:17 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=1338146816
15/07/02 14:11:17 INFO mapred.JobClient:     Total committed heap usage (bytes)=171315200
```

Final Output :


cloudera@localhost:~/workspace/WhiteHouseMR4
File Edit View Search Terminal Help

```
15/07/02 14:11:17 INFO mapred.JobClient:     Launched reduce tasks=1
15/07/02 14:11:17 INFO mapred.JobClient:     Data-local map tasks=1
15/07/02 14:11:17 INFO mapred.JobClient:     Total time spent by all maps in occupied slots (ms)=10028
15/07/02 14:11:17 INFO mapred.JobClient:     Total time spent by all reduces in occupied slots (ms)=4598
15/07/02 14:11:17 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
15/07/02 14:11:17 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
15/07/02 14:11:17 INFO mapred.JobClient:   Map-Reduce Framework
15/07/02 14:11:17 INFO mapred.JobClient:     Map input records=101680
15/07/02 14:11:17 INFO mapred.JobClient:     Map output records=101423
15/07/02 14:11:17 INFO mapred.JobClient:     Map output bytes=811384
15/07/02 14:11:17 INFO mapred.JobClient:     Input split bytes=152
15/07/02 14:11:17 INFO mapred.JobClient:     Combine input records=0
15/07/02 14:11:17 INFO mapred.JobClient:     Combine output records=0
15/07/02 14:11:17 INFO mapred.JobClient:     Reduce input groups=8
15/07/02 14:11:17 INFO mapred.JobClient:     Reduce shuffle bytes=48005
15/07/02 14:11:17 INFO mapred.JobClient:     Reduce input records=101423
15/07/02 14:11:17 INFO mapred.JobClient:     Reduce output records=8
15/07/02 14:11:17 INFO mapred.JobClient:     Spilled Records=202846
15/07/02 14:11:17 INFO mapred.JobClient:     CPU time spent (ms)=2620
15/07/02 14:11:17 INFO mapred.JobClient:     Physical memory (bytes) snapshot=278609920
15/07/02 14:11:17 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=1338146816
15/07/02 14:11:17 INFO mapred.JobClient:     Total committed heap usage (bytes)=171315200

Printing Average Appointments made per month (all years) to WhiteHouse ..........................

9       1
8       2
7       1
6       5
5       79382
4       11381
3       10650
2       1
[cloudera@localhost WhiteHouseMR4]$ 
```

Output Explanation :

The final output produced by this program is saved in the output directory /WhiteHouseMR4/output of HDFS. The file part-r-00000 in this output contains the Average Appointments made per month to the White House in descending order. The two columns in output files are the Month Number, Average appointments made in that month in that order. There were a total of 8 months in which appointments were made , and 4 months had no appointments at all. The month of May had the maximum average appointments made across all the years available in the data set.