CISC 5950 Project 1

Manali Chordia - A20535490

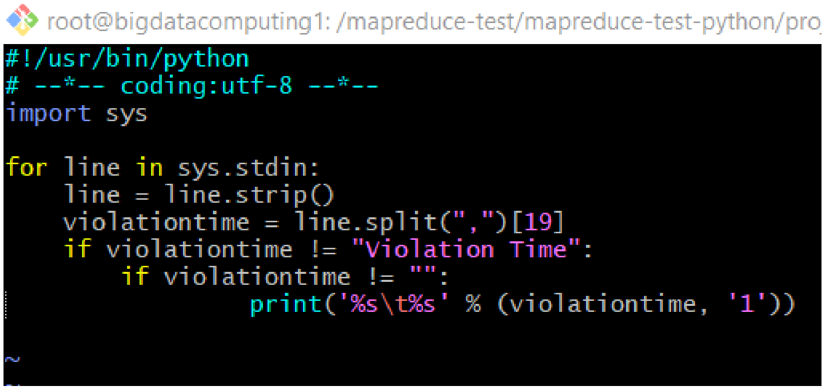
Yuying Zhou - A20163001

Part 1

Q1 When are the tickets most likely to be issued

Use the strip technique to remove the leading and trailing characters from each line after reading lines using sys.stdin. Commas are used to separate the word violation_time. In addition, index 19 is selected. When treating the csv file's header as a list, the matching index for "violation time" is 19. Get the value for each violation time in the value after filtering the column name and unexpected values, adjusting the key to equal the line number and the violation time in the line, and (violation_time,1).

Here is the mapper_p1q1.py:

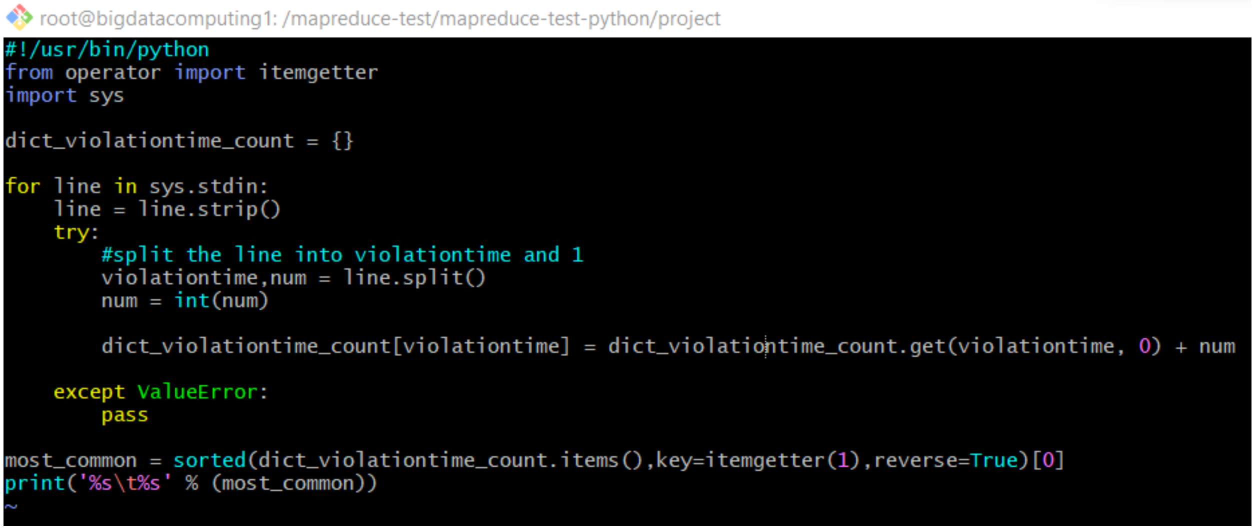


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/pro

```
#!/usr/bin/python
# --*-- coding:utf-8 --*--
import sys

for line in sys.stdin:
    line = line.strip()
    violationtime = line.split(",")[19]
    if violationtime != "Violation Time":
        if violationtime != "":
                print('%s\t%s' % (violationtime, '1'))

~
```

By setting the key to a violation time and the list of values to a list of counts, a sort function can be used to find the period when tickets are most likely to be issued (parameter num). To store future values, create an empty directory first, read lines from it using sys.stdin, then use the strip method to remove any strange characters. violation_time stands for the violation time since each time the violation time increases by one, num will be tallied once (key is equal to violation_time and the value is num). Indicate that the parameter num is an integer. The sorted function's itemgetter(n) method accepts an iterable object as input and returns the object's nth element. In this instance, itemgetter(1) uses the item (key, value) in dict_violation_time_count as input to extract the total counts for each violation time. Because reverse is set to True, the dict_violation_time_count is sorted in descending order by the number of counts. Since we only want the most likely to be issued time, index [0] will cause the sorting algorithm to only print the most common violation time.

Here is the reducer_p1q1.py file:

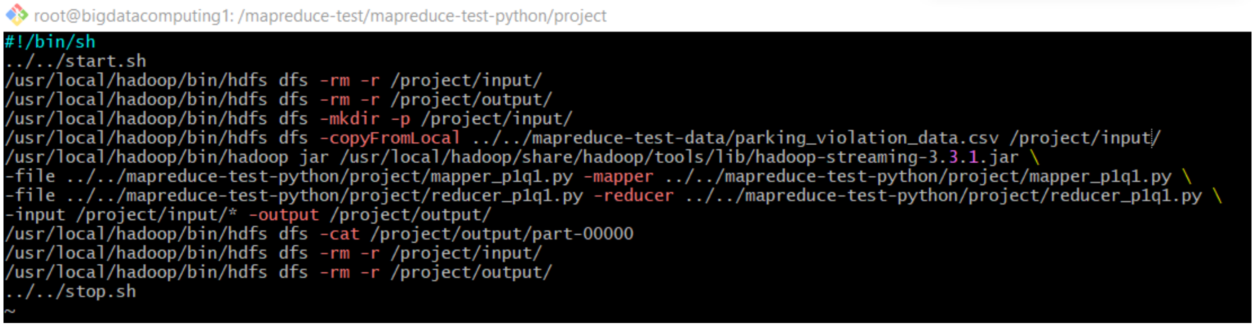


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
#!/usr/bin/python
from operator import itemgetter
import sys

dict_violationtime_count = {}

for line in sys.stdin:
    line = line.strip()
    try:
        #split the line into violationtime and 1
        violationtime,num = line.split()
        num = int(num)

        dict_violationtime_count[violationtime] = dict_violationtime_count.get(violationtime, 0) + num

    except ValueError:
        pass

most_common = sorted(dict_violationtime_count.items(),key=itemgetter(1),reverse=True)[0]
print('%s\t%s' % (most_common))
~
```

Start hdfs, yarn, and historyserver with start.sh, then exit safe mode. Delete the input and output directories to guarantee that no input or output remains to muck up the input data. Make a directory for the project so that it may copy the dataset from the local directory. Begin a Hadoop streaming job to perform a map and reduce operation. Save output and read it with the cat command. Stop hdfs, yarn, and historyserver by deleting the input and output directories and calling stop.sh.

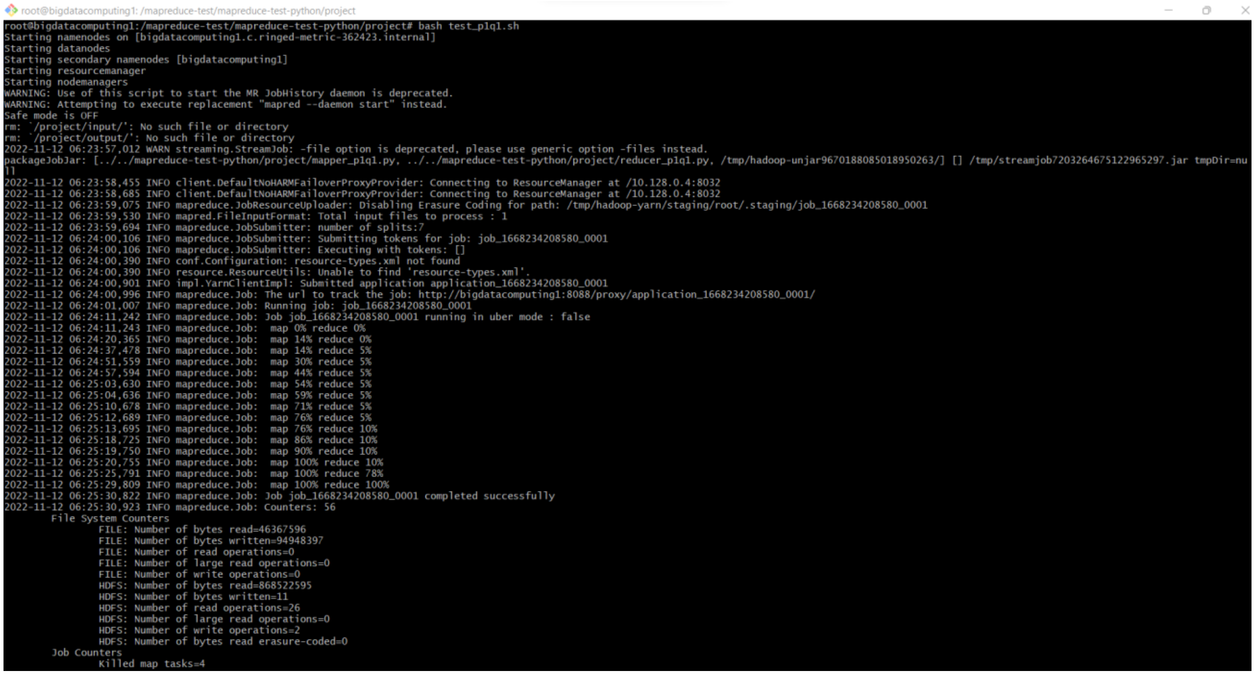
Here is the test_p1q1.sh file:



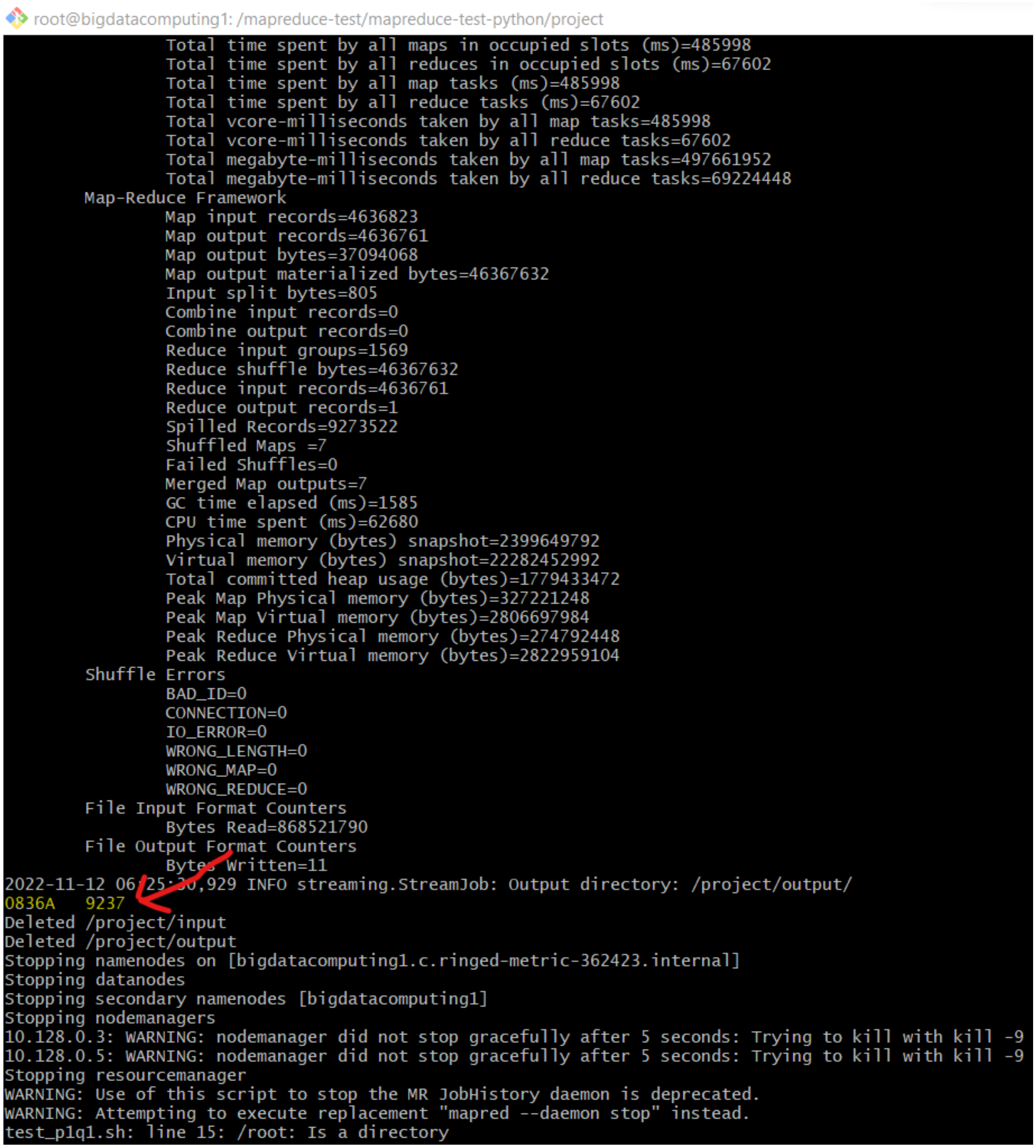
root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
#!/bin/sh
../../start.sh
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/input/
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/output/
/usr/local/hadoop/bin/hdfs dfs -mkdir -p /project/input/
/usr/local/hadoop/bin/hdfs dfs -copyFromLocal ../../mapreduce-test-data/parking_violation_data.csv /project/input/
/usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.1.jar \
-file ../../mapreduce-test-python/project/mapper_p1q1.py -mapper ../../mapreduce-test-python/project/mapper_p1q1.py \
-file ../../mapreduce-test-python/project/reducer_p1q1.py -reducer ../../mapreduce-test-python/project/reducer_p1q1.py \
-input /project/input/* -output /project/output/
/usr/local/hadoop/bin/hdfs dfs -cat /project/output/part-00000
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/input/
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/output/
../../stop.sh
~
```

Output:


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project# bash test_p1q1.sh
Starting namenodes on [bigdatacomputing1.c.ringed-metric-362423.internal]
Starting datanodes
Starting secondary namenodes [bigdatacomputing1]
Starting resourcemanager
Starting nodemanagers
WARNING: Use of this script to start the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon start" instead.
Safe mode is OFF
rm: `/project/input/': No such file or directory
rm: `/project/output/': No such file or directory
2022-11-12 06:23:57,012 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [../../mapreduce-test-python/project/mapper_p1q1.py, ../../mapreduce-test-python/project/reducer_p1q1.py, /tmp/hadoop-unjar9670188085018950263/] [] /tmp/streamjob7203264675122965297.jar tmpDir=nu
ll
2022-11-12 06:23:58,455 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /10.128.0.4:8032
2022-11-12 06:23:58,685 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /10.128.0.4:8032
2022-11-12 06:23:59,075 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1668234208580_0001
2022-11-12 06:23:59,530 INFO mapred.FileInputFormat: Total input files to process : 1
2022-11-12 06:23:59,694 INFO mapreduce.JobSubmitter: number of splits:7
2022-11-12 06:24:00,106 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1668234208580_0001
2022-11-12 06:24:00,106 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-11-12 06:24:00,390 INFO conf.Configuration: resource-types.xml not found
2022-11-12 06:24:00,390 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-11-12 06:24:00,901 INFO impl.YarnClientImpl: Submitted application application_1668234208580_0001
2022-11-12 06:24:00,996 INFO mapreduce.Job: The url to track the job: http://bigdatacomputing1:8088/proxy/application_1668234208580_0001/
2022-11-12 06:24:01,007 INFO mapreduce.Job: Running job: job_1668234208580_0001
2022-11-12 06:24:11,242 INFO mapreduce.Job: Job job_1668234208580_0001 running in uber mode : false
2022-11-12 06:24:11,243 INFO mapreduce.Job:  map 0% reduce 0%
2022-11-12 06:24:20,365 INFO mapreduce.Job:  map 14% reduce 0%
2022-11-12 06:24:37,478 INFO mapreduce.Job:  map 14% reduce 5%
2022-11-12 06:24:51,559 INFO mapreduce.Job:  map 30% reduce 5%
2022-11-12 06:24:57,594 INFO mapreduce.Job:  map 44% reduce 5%
2022-11-12 06:25:03,630 INFO mapreduce.Job:  map 54% reduce 5%
2022-11-12 06:25:04,636 INFO mapreduce.Job:  map 59% reduce 5%
2022-11-12 06:25:10,678 INFO mapreduce.Job:  map 71% reduce 5%
2022-11-12 06:25:12,689 INFO mapreduce.Job:  map 76% reduce 5%
2022-11-12 06:25:13,695 INFO mapreduce.Job:  map 76% reduce 10%
2022-11-12 06:25:18,725 INFO mapreduce.Job:  map 86% reduce 10%
2022-11-12 06:25:19,750 INFO mapreduce.Job:  map 90% reduce 10%
2022-11-12 06:25:20,755 INFO mapreduce.Job:  map 100% reduce 10%
2022-11-12 06:25:25,791 INFO mapreduce.Job:  map 100% reduce 78%
2022-11-12 06:25:29,809 INFO mapreduce.Job:  map 100% reduce 100%
2022-11-12 06:25:30,822 INFO mapreduce.Job: Job job_1668234208580_0001 completed successfully
2022-11-12 06:25:30,923 INFO mapreduce.Job: Counters: 56
        File System Counters
                FILE: Number of bytes read=46367596
                FILE: Number of bytes written=94948397
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=868522595
                HDFS: Number of bytes written=11
                HDFS: Number of read operations=26
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Killed map tasks=4
```



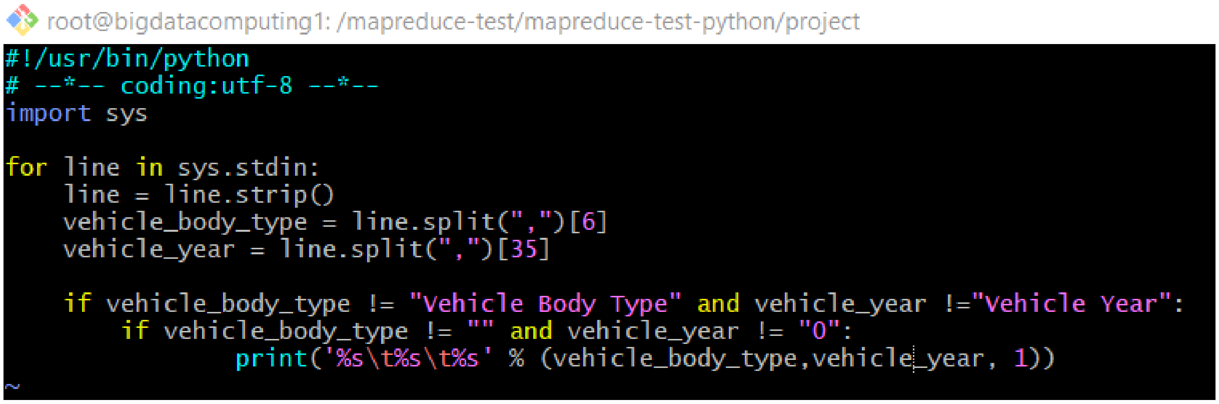
root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
                Total time spent by all maps in occupied slots (ms)=485998
                Total time spent by all reduces in occupied slots (ms)=67602
                Total time spent by all map tasks (ms)=485998
                Total time spent by all reduce tasks (ms)=67602
                Total vcore-milliseconds taken by all map tasks=485998
                Total vcore-milliseconds taken by all reduce tasks=67602
                Total megabyte-milliseconds taken by all map tasks=497661952
                Total megabyte-milliseconds taken by all reduce tasks=69224448
        Map-Reduce Framework
                Map input records=4636823
                Map output records=4636761
                Map output bytes=37094068
                Map output materialized bytes=46367632
                Input split bytes=805
                Combine input records=0
                Combine output records=0
                Reduce input groups=1569
                Reduce shuffle bytes=46367632
                Reduce input records=4636761
                Reduce output records=1
                Spilled Records=9273522
                Shuffled Maps =7
                Failed Shuffles=0
                Merged Map outputs=7
                GC time elapsed (ms)=1585
                CPU time spent (ms)=62680
                Physical memory (bytes) snapshot=2399649792
                Virtual memory (bytes) snapshot=22282452992
                Total committed heap usage (bytes)=1779433472
                Peak Map Physical memory (bytes)=327221248
                Peak Map Virtual memory (bytes)=2806697984
                Peak Reduce Physical memory (bytes)=274792448
                Peak Reduce Virtual memory (bytes)=2822959104
        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=868521790
        File Output Format Counters
                Bytes Written=11
2022-11-12 06:25:30,929 INFO streaming.StreamJob: Output directory: /project/output/
0836A   9237
Deleted /project/input
Deleted /project/output
Stopping namenodes on [bigdatacomputing1.c.ringed-metric-362423.internal]
Stopping datanodes
Stopping secondary namenodes [bigdatacomputing1]
Stopping nodemanagers
10.128.0.3: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill with kill -9
10.128.0.5: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill with kill -9
Stopping resourcemanager
WARNING: Use of this script to stop the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon stop" instead.
test_p1q1.sh: line 15: /root: Is a directory
```

Q2 What are the most common years and types of cars to be ticketed

The sys.stdin command can read lines. Find out the year and the type of vehicle. Using their respective indexes of 6 and 35, the line has been separated into the 7th column for vehicle type and the 36th column for vehicle year. Remove the year's missing value from the filter as well as the column name.

Here is the mapper_p1q2.py:

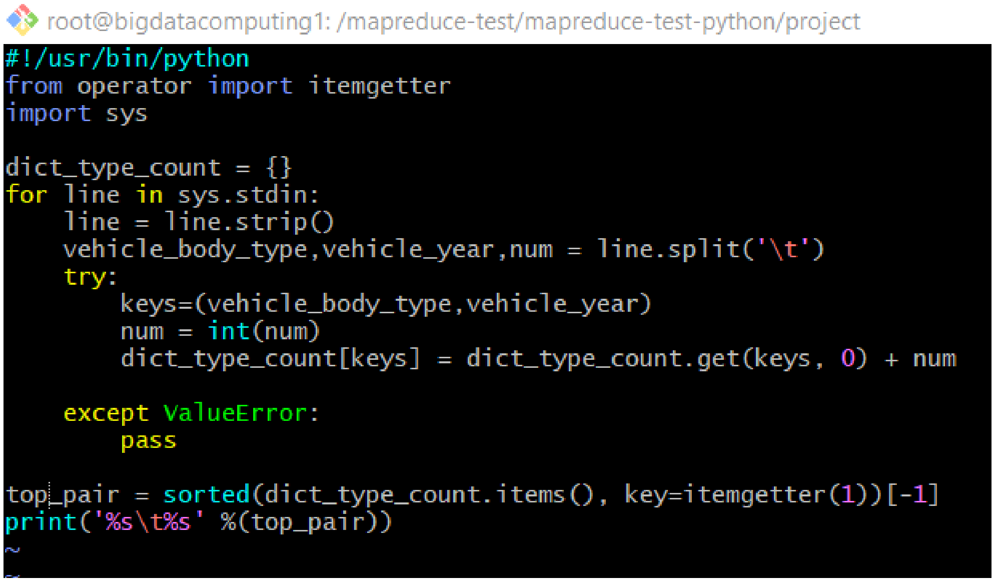


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
#!/usr/bin/python
# --*-- coding:utf-8 --*--
import sys

for line in sys.stdin:
    line = line.strip()
    vehicle_body_type = line.split(",")[6]
    vehicle_year = line.split(",")[35]

    if vehicle_body_type != "Vehicle Body Type" and vehicle_year !="Vehicle Year":
        if vehicle_body_type != "" and vehicle_year != "0":
                print('%s\t%s\t%s' % (vehicle_body_type,vehicle_year, 1))
~
```

First, make an empty dictionary to store the information on the count, year, and number of cars. Lines read from sys.stdin are then separated for the output by vehicle_body_type, vehicle_year, and num. Make num an integer for future calculations, then make a variable for each combination of car type and year. Use the empty dictionary and add num + 1 each time the same pair is discovered to store each pair of the automobile type and year as keys and the num as the value for counting. The final element, which reflects the most typical car type and year for receiving a parking ticket, was output after the dictionary was sorted using value(num).

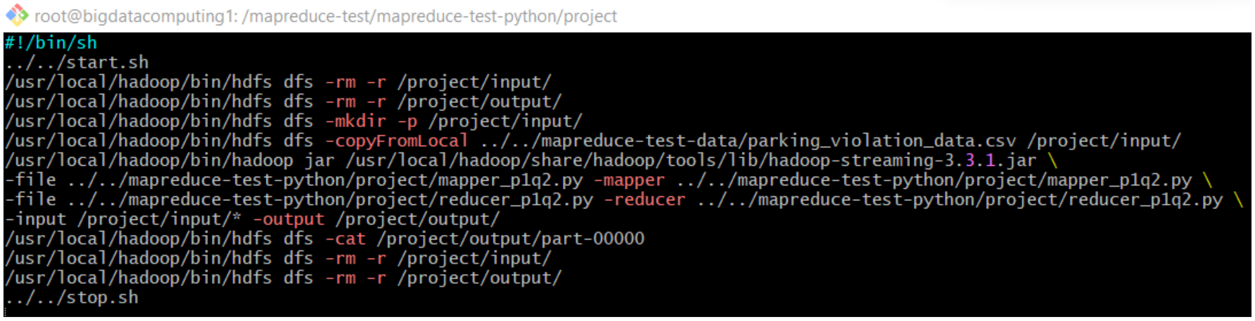
Here is the reducer_p1q2.py file:


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
#!/usr/bin/python
from operator import itemgetter
import sys

dict_type_count = {}
for line in sys.stdin:
    line = line.strip()
    vehicle_body_type,vehicle_year,num = line.split('\t')
    try:
        keys=(vehicle_body_type,vehicle_year)
        num = int(num)
        dict_type_count[keys] = dict_type_count.get(keys, 0) + num

    except ValueError:
        pass

top_pair = sorted(dict_type_count.items(), key=itemgetter(1))[-1]
print('%s\t%s' %(top_pair))
~
~
```

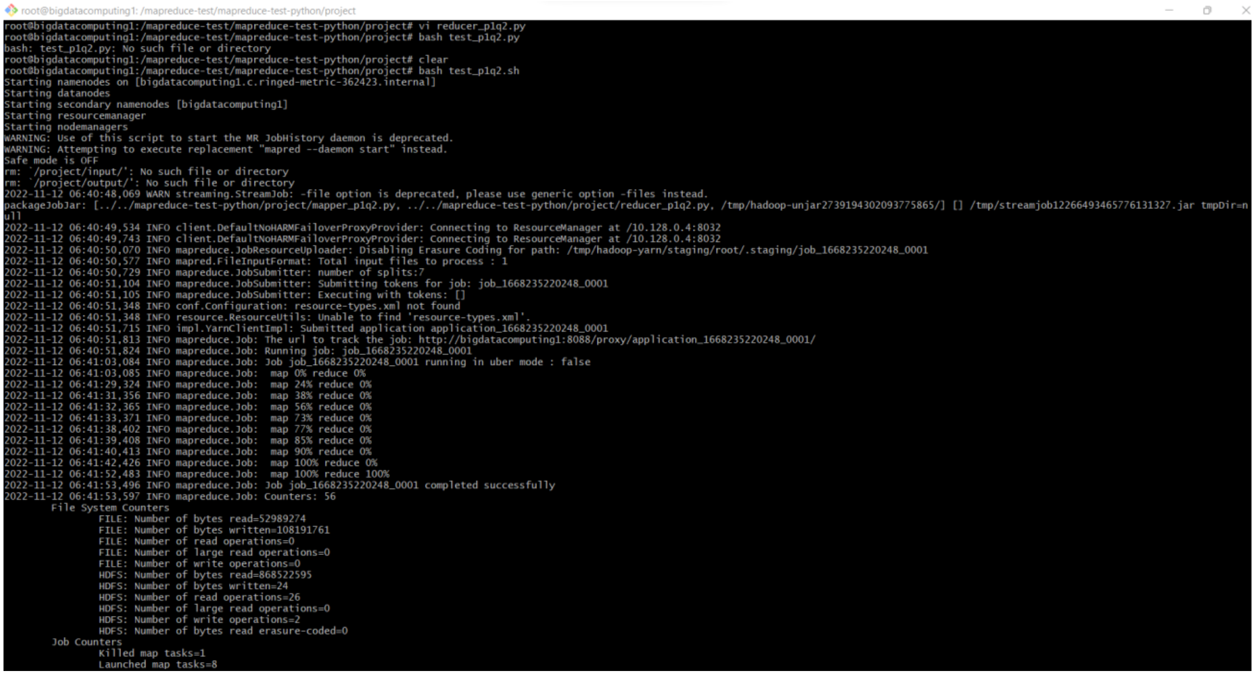
Start hdfs, yarn, and historyserver with start.sh, then exit safe mode. Delete the input and output directories to guarantee that no input or output remains to muck up the input data. Make a directory for the project so that it may copy the dataset from the local directory. Begin a Hadoop streaming job to perform a map and reduce operation. Save output and read it with the cat command. Stop hdfs, yarn, and historyserver by deleting the input and output directories and calling stop.sh.

Here is the test_p1q2.sh file:


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

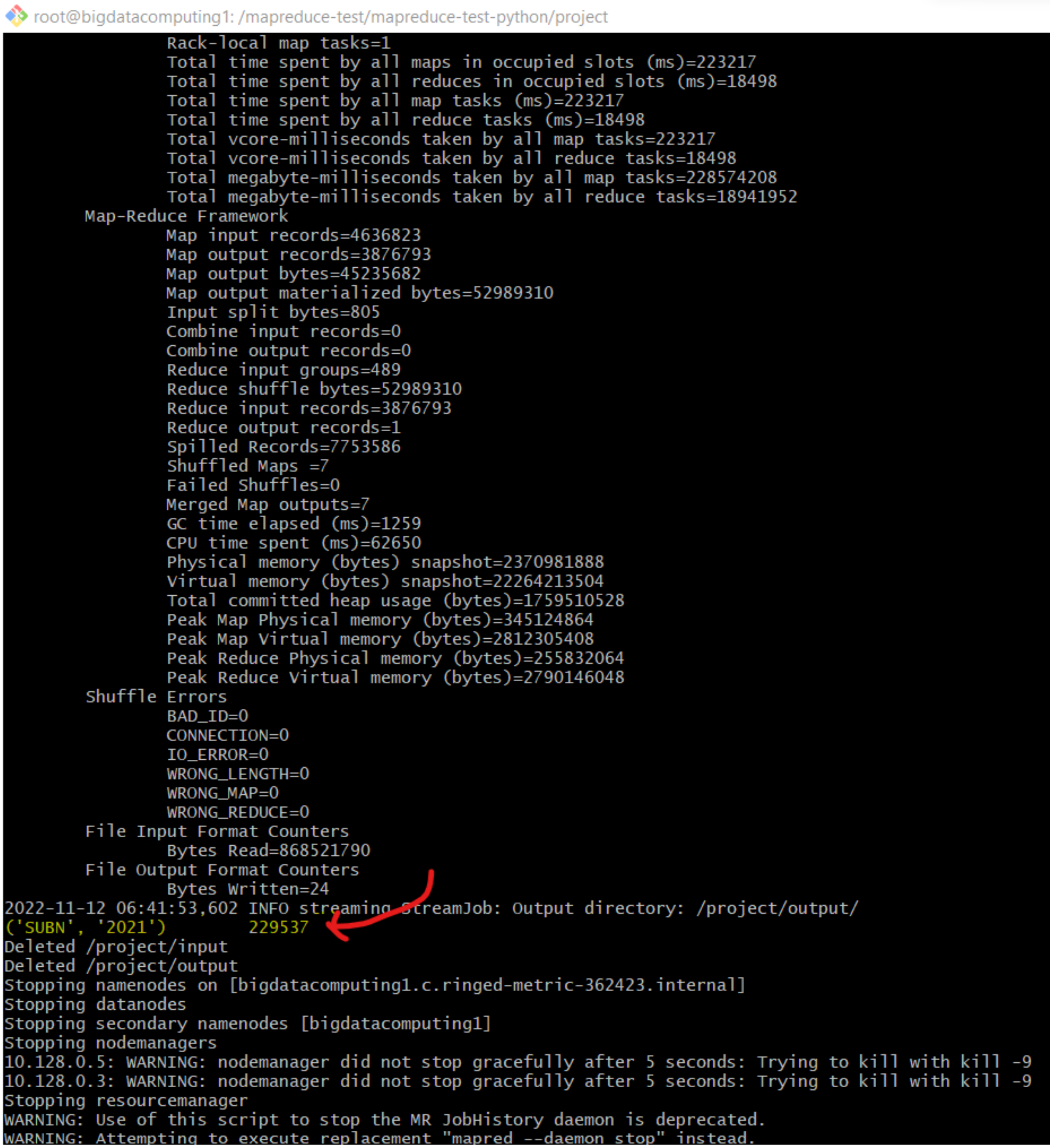
```
#!/bin/sh
../../start.sh
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/input/
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/output/
/usr/local/hadoop/bin/hdfs dfs -mkdir -p /project/input/
/usr/local/hadoop/bin/hdfs dfs -copyFromLocal ../../mapreduce-test-data/parking_violation_data.csv /project/input/
/usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.1.jar \
-file ../../mapreduce-test-python/project/mapper_p1q2.py -mapper ../../mapreduce-test-python/project/mapper_p1q2.py \
-file ../../mapreduce-test-python/project/reducer_p1q2.py -reducer ../../mapreduce-test-python/project/reducer_p1q2.py \
-input /project/input/* -output /project/output/
/usr/local/hadoop/bin/hdfs dfs -cat /project/output/part-00000
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/input/
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/output/
../../stop.sh
```

Output:



root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project# vi reducer_p1q2.py
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project# bash test_p1q2.py
bash: test_p1q2.py: No such file or directory
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project# clear
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project# bash test_p1q2.sh
Starting namenodes on [bigdatacomputing1.c.ringed-metric-362423.internal]
Starting datanodes
Starting secondary namenodes [bigdatacomputing1]
Starting resourcemanager
Starting nodemanagers
WARNING: Use of this script to start the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon start" instead.
Safe mode is OFF
rm: `/project/input/': No such file or directory
rm: `/project/output/': No such file or directory
2022-11-12 06:40:48,069 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [../../mapreduce-test-python/project/mapper_p1q2.py, ../../mapreduce-test-python/project/reducer_p1q2.py, /tmp/hadoop-unjar2739194302093775865/] [] /tmp/streamjob1226649346577613l327.jar tmpDir=n
ull
2022-11-12 06:40:49,534 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /10.128.0.4:8032
2022-11-12 06:40:49,743 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /10.128.0.4:8032
2022-11-12 06:40:50,070 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1668235220248_0001
2022-11-12 06:40:50,577 INFO mapred.FileInputFormat: Total input files to process : 1
2022-11-12 06:40:50,729 INFO mapreduce.JobSubmitter: number of splits:7
2022-11-12 06:40:51,104 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1668235220248_0001
2022-11-12 06:40:51,105 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-11-12 06:40:51,348 INFO conf.Configuration: resource-types.xml not found
2022-11-12 06:40:51,348 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-11-12 06:40:51,715 INFO impl.YarnClientImpl: Submitted application application_1668235220248_0001
2022-11-12 06:40:51,813 INFO mapreduce.Job: The url to track the job: http://bigdatacomputing1:8088/proxy/application_1668235220248_0001/
2022-11-12 06:40:51,824 INFO mapreduce.Job: Running job: job_1668235220248_0001
2022-11-12 06:41:03,084 INFO mapreduce.Job: Job job_1668235220248_0001 running in uber mode : false
2022-11-12 06:41:03,085 INFO mapreduce.Job:  map 0% reduce 0%
2022-11-12 06:41:29,324 INFO mapreduce.Job:  map 24% reduce 0%
2022-11-12 06:41:31,356 INFO mapreduce.Job:  map 38% reduce 0%
2022-11-12 06:41:32,365 INFO mapreduce.Job:  map 56% reduce 0%
2022-11-12 06:41:33,371 INFO mapreduce.Job:  map 73% reduce 0%
2022-11-12 06:41:38,402 INFO mapreduce.Job:  map 77% reduce 0%
2022-11-12 06:41:39,408 INFO mapreduce.Job:  map 85% reduce 0%
2022-11-12 06:41:40,413 INFO mapreduce.Job:  map 90% reduce 0%
2022-11-12 06:41:42,426 INFO mapreduce.Job:  map 100% reduce 0%
2022-11-12 06:41:52,483 INFO mapreduce.Job:  map 100% reduce 100%
2022-11-12 06:41:53,496 INFO mapreduce.Job: Job job_1668235220248_0001 completed successfully
2022-11-12 06:41:53,597 INFO mapreduce.Job: Counters: 56
        File System Counters
                FILE: Number of bytes read=52989274
                FILE: Number of bytes written=108191761
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=868522595
                HDFS: Number of bytes written=24
                HDFS: Number of read operations=26
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Killed map tasks=1
                Launched map tasks=8
```



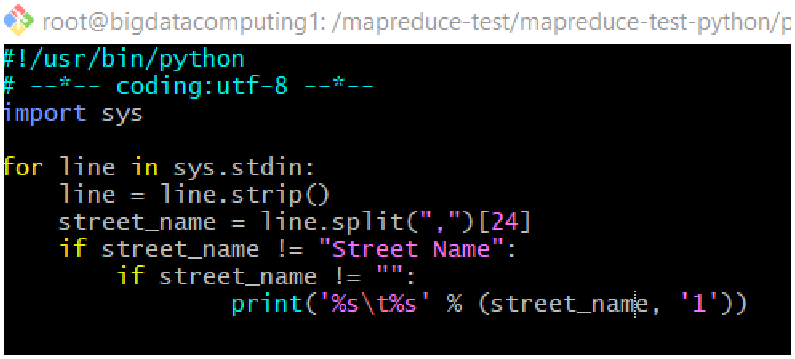
root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
                Rack-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=223217
                Total time spent by all reduces in occupied slots (ms)=18498
                Total time spent by all map tasks (ms)=223217
                Total time spent by all reduce tasks (ms)=18498
                Total vcore-milliseconds taken by all map tasks=223217
                Total vcore-milliseconds taken by all reduce tasks=18498
                Total megabyte-milliseconds taken by all map tasks=228574208
                Total megabyte-milliseconds taken by all reduce tasks=18941952
        Map-Reduce Framework
                Map input records=4636823
                Map output records=3876793
                Map output bytes=45235682
                Map output materialized bytes=52989310
                Input split bytes=805
                Combine input records=0
                Combine output records=0
                Reduce input groups=489
                Reduce shuffle bytes=52989310
                Reduce input records=3876793
                Reduce output records=1
                Spilled Records=7753586
                Shuffled Maps =7
                Failed Shuffles=0
                Merged Map outputs=7
                GC time elapsed (ms)=1259
                CPU time spent (ms)=62650
                Physical memory (bytes) snapshot=2370981888
                Virtual memory (bytes) snapshot=22264213504
                Total committed heap usage (bytes)=1759510528
                Peak Map Physical memory (bytes)=345124864
                Peak Map Virtual memory (bytes)=2812305408
                Peak Reduce Physical memory (bytes)=255832064
                Peak Reduce Virtual memory (bytes)=2790146048
        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=868521790
        File Output Format Counters
                Bytes Written=24
2022-11-12 06:41:53,602 INFO streaming.StreamJob: Output directory: /project/output/
('SUBN', '2021')        229537
Deleted /project/input
Deleted /project/output
Stopping namenodes on [bigdatacomputing1.c.ringed-metric-362423.internal]
Stopping datanodes
Stopping secondary namenodes [bigdatacomputing1]
Stopping nodemanagers
10.128.0.5: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill with kill -9
10.128.0.3: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill with kill -9
Stopping resourcemanager
WARNING: Use of this script to stop the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon stop" instead.
```

Q3 Where are tickets most commonly issued

The split function, whose index is 24, can be used to read lines from sys.stdin and extract data about street names. After that, use a filter to eliminate column names and empty values. Write the street name down.

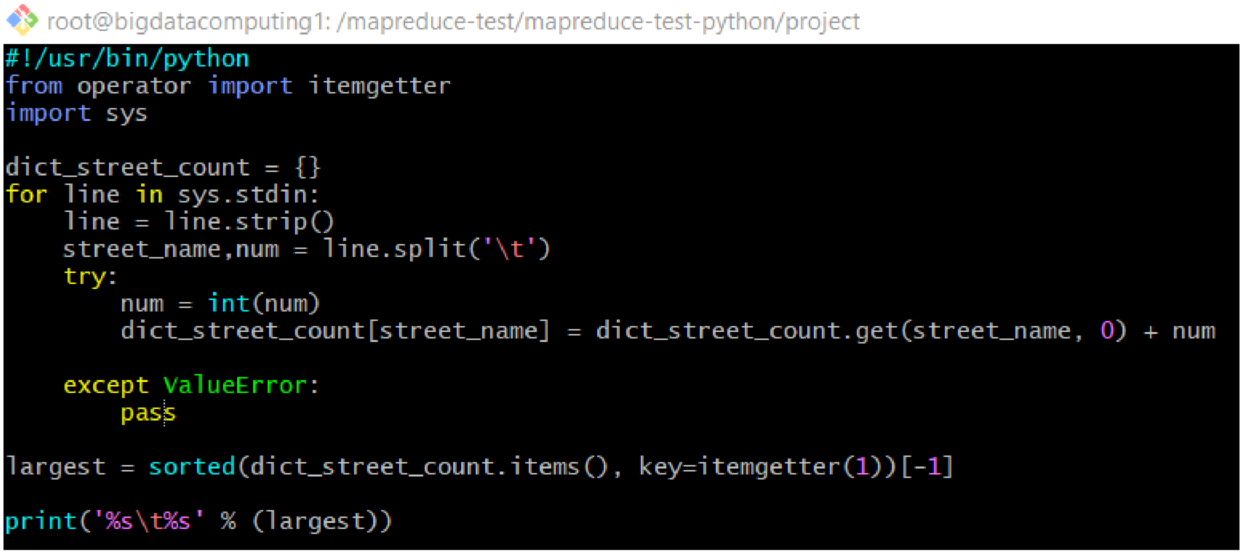
Here is the mapper_p1q3.py file:


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/p

```
#!/usr/bin/python
# --*-- coding:utf-8 --*--
import sys

for line in sys.stdin:
    line = line.strip()
    street_name = line.split(",")[24]
    if street_name != "Street Name":
        if street_name != "":
                print('%s\t%s' % (street_name, '1'))
```

Make a blank dictionary at first, and use sys.stdin to read lines. Use the strip function to tidy up lines and separate them into street and num parts. Make num an integer to make calculations easier in the future. street_name is the same as the key, and num is the value. Add one to the number each time the same key (street name) appears. After sorting the dictionary by value, the final element should be output (num).

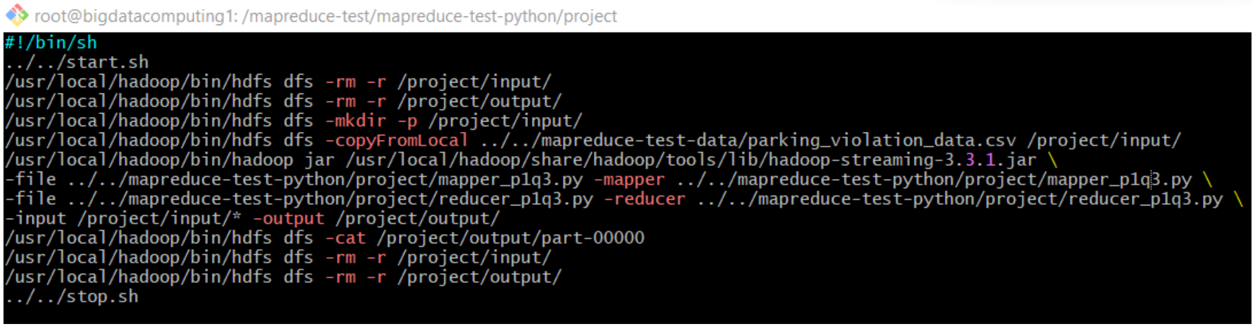
Here is the reducer_p1q3.py file:


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
#!/usr/bin/python
from operator import itemgetter
import sys

dict_street_count = {}
for line in sys.stdin:
    line = line.strip()
    street_name,num = line.split('\t')
    try:
        num = int(num)
        dict_street_count[street_name] = dict_street_count.get(street_name, 0) + num

    except ValueError:
        pass

largest = sorted(dict_street_count.items(), key=itemgetter(1))[-1]

print('%s\t%s' % (largest))
```

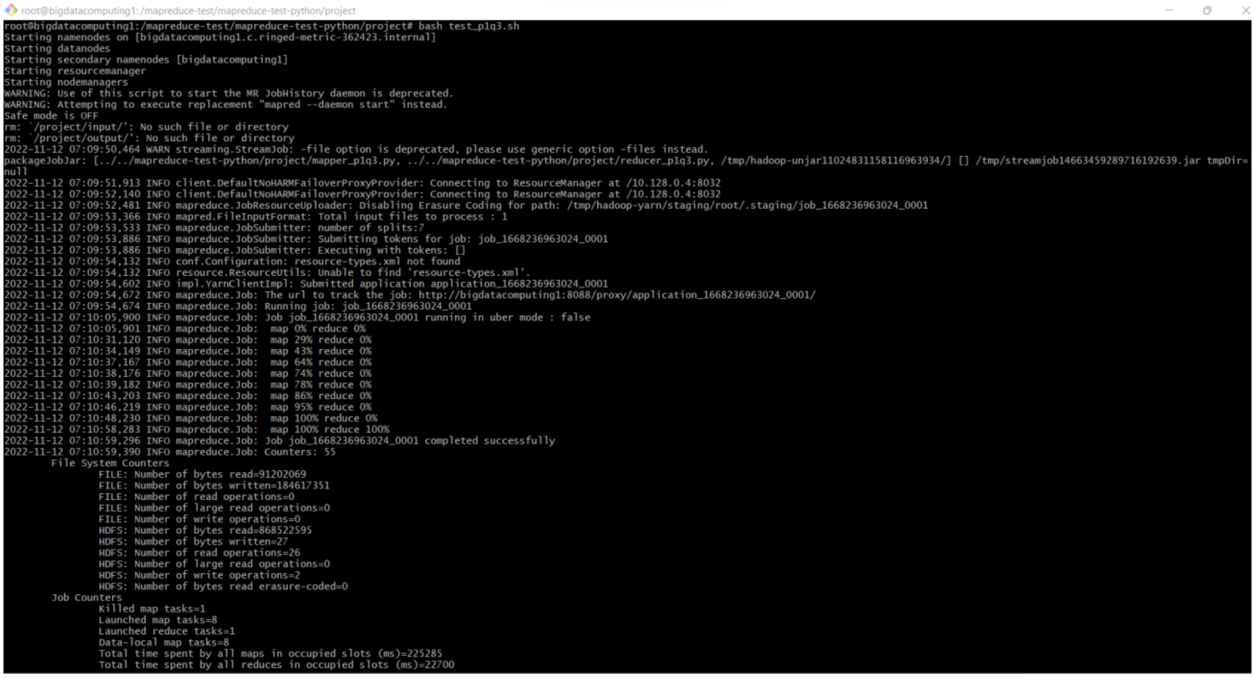
Start hdfs, yarn, and historyserver with start.sh, then exit safe mode. Delete the input and output directories to guarantee that no input or output remains to muck up the input data. Make a directory for the project so that it may copy the dataset from the local directory. Begin a Hadoop streaming job to perform a map and reduce operation. Save output and read it with the cat command. Stop hdfs, yarn, and historyserver by deleting the input and output directories and calling stop.sh.

Here is the test_p1q3.sh file:

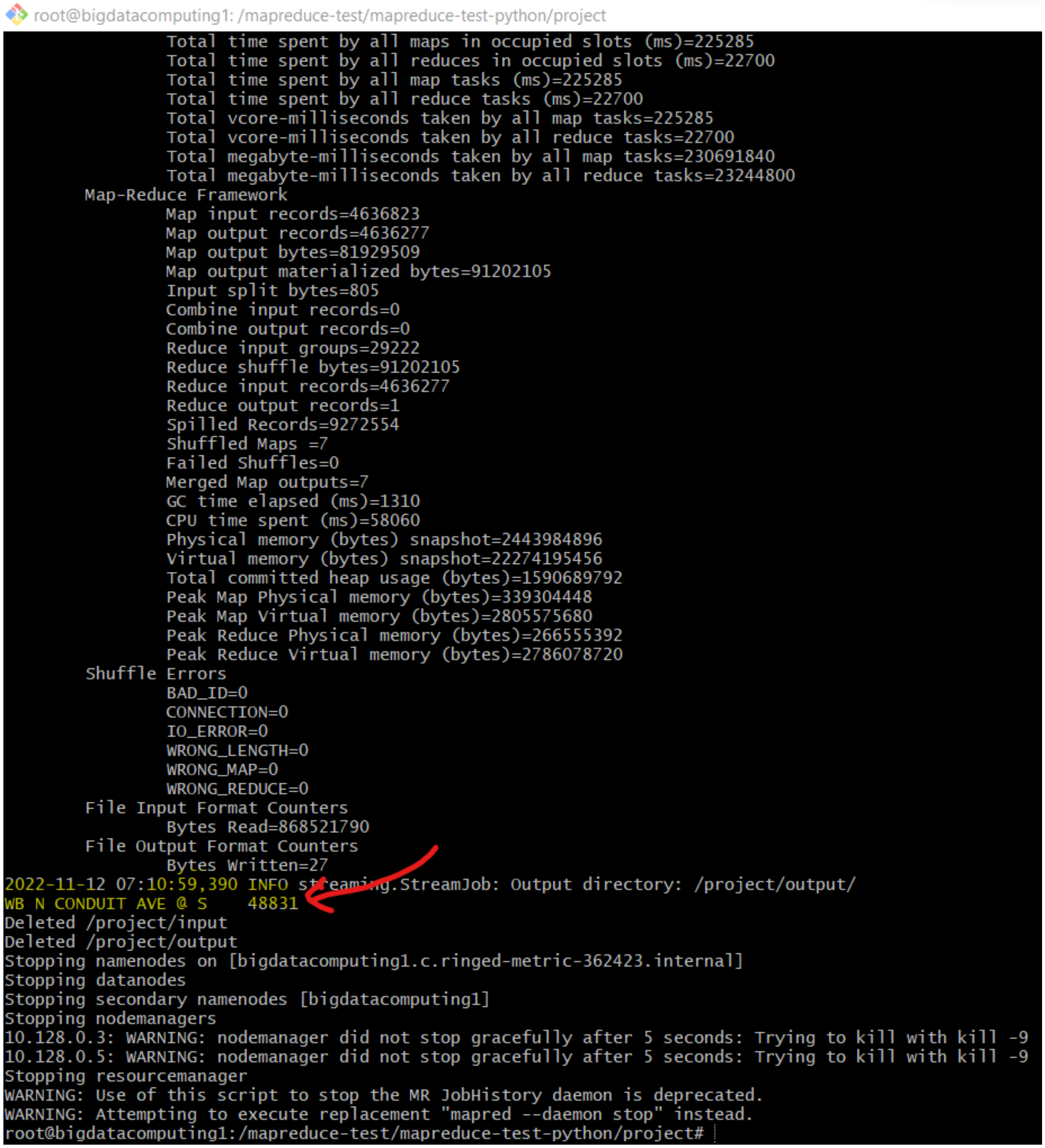

root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
#!/bin/sh
../../start.sh
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/input/
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/output/
/usr/local/hadoop/bin/hdfs dfs -mkdir -p /project/input/
/usr/local/hadoop/bin/hdfs dfs -copyFromLocal ../../mapreduce-test-data/parking_violation_data.csv /project/input/
/usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.1.jar \
-file ../../mapreduce-test-python/project/mapper_p1q3.py -mapper ../../mapreduce-test-python/project/mapper_p1q3.py \
-file ../../mapreduce-test-python/project/reducer_p1q3.py -reducer ../../mapreduce-test-python/project/reducer_p1q3.py \
-input /project/input/* -output /project/output/
/usr/local/hadoop/bin/hdfs dfs -cat /project/output/part-00000
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/input/
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/output/
../../stop.sh
```

Output:


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project# bash test_p1q3.sh
Starting namenodes on [bigdatacomputing1.c.ringed-metric-362423.internal]
Starting datanodes
Starting secondary namenodes [bigdatacomputing1]
Starting resourcemanager
Starting nodemanagers
WARNING: Use of this script to start the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon start" instead.
Safe mode is OFF
rm: `/project/input/': No such file or directory
rm: `/project/output/': No such file or directory
2022-11-12 07:09:50,464 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [../../mapreduce-test-python/project/mapper_p1q3.py, ../../mapreduce-test-python/project/reducer_p1q3.py, /tmp/hadoop-unjar11024831158116963934/] [] /tmp/streamjob14663459289716192639.jar tmpDir=
null
2022-11-12 07:09:51,913 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /10.128.0.4:8032
2022-11-12 07:09:52,140 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /10.128.0.4:8032
2022-11-12 07:09:52,481 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1668236963024_0001
2022-11-12 07:09:53,366 INFO mapred.FileInputFormat: Total input files to process : 1
2022-11-12 07:09:53,533 INFO mapreduce.JobSubmitter: number of splits:7
2022-11-12 07:09:53,886 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1668236963024_0001
2022-11-12 07:09:53,886 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-11-12 07:09:54,132 INFO conf.Configuration: resource-types.xml not found
2022-11-12 07:09:54,132 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-11-12 07:09:54,602 INFO impl.YarnClientImpl: Submitted application application_1668236963024_0001
2022-11-12 07:09:54,672 INFO mapreduce.Job: The url to track the job: http://bigdatacomputing1:8088/proxy/application_1668236963024_0001/
2022-11-12 07:09:54,674 INFO mapreduce.Job: Running job: job_1668236963024_0001
2022-11-12 07:10:05,900 INFO mapreduce.Job: Job job_1668236963024_0001 running in uber mode : false
2022-11-12 07:10:05,901 INFO mapreduce.Job:  map 0% reduce 0%
2022-11-12 07:10:31,120 INFO mapreduce.Job:  map 29% reduce 0%
2022-11-12 07:10:34,149 INFO mapreduce.Job:  map 43% reduce 0%
2022-11-12 07:10:37,167 INFO mapreduce.Job:  map 64% reduce 0%
2022-11-12 07:10:38,176 INFO mapreduce.Job:  map 74% reduce 0%
2022-11-12 07:10:39,182 INFO mapreduce.Job:  map 78% reduce 0%
2022-11-12 07:10:43,203 INFO mapreduce.Job:  map 86% reduce 0%
2022-11-12 07:10:46,219 INFO mapreduce.Job:  map 95% reduce 0%
2022-11-12 07:10:48,230 INFO mapreduce.Job:  map 100% reduce 0%
2022-11-12 07:10:58,283 INFO mapreduce.Job:  map 100% reduce 100%
2022-11-12 07:10:59,296 INFO mapreduce.Job: Job job_1668236963024_0001 completed successfully
2022-11-12 07:10:59,390 INFO mapreduce.Job: Counters: 55
        File System Counters
                FILE: Number of bytes read=91202069
                FILE: Number of bytes written=184617351
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=868522595
                HDFS: Number of bytes written=27
                HDFS: Number of read operations=26
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Killed map tasks=1
                Launched map tasks=8
                Launched reduce tasks=1
                Data-local map tasks=8
                Total time spent by all maps in occupied slots (ms)=225285
                Total time spent by all reduces in occupied slots (ms)=22700
```

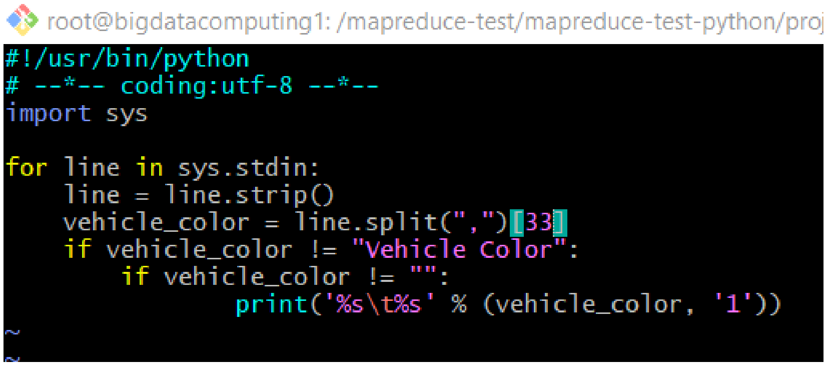

root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
                Total time spent by all maps in occupied slots (ms)=225285
                Total time spent by all reduces in occupied slots (ms)=22700
                Total time spent by all map tasks (ms)=225285
                Total time spent by all reduce tasks (ms)=22700
                Total vcore-milliseconds taken by all map tasks=225285
                Total vcore-milliseconds taken by all reduce tasks=22700
                Total megabyte-milliseconds taken by all map tasks=230691840
                Total megabyte-milliseconds taken by all reduce tasks=23244800
        Map-Reduce Framework
                Map input records=4636823
                Map output records=4636277
                Map output bytes=81929509
                Map output materialized bytes=91202105
                Input split bytes=805
                Combine input records=0
                Combine output records=0
                Reduce input groups=29222
                Reduce shuffle bytes=91202105
                Reduce input records=4636277
                Reduce output records=1
                Spilled Records=9272554
                Shuffled Maps =7
                Failed Shuffles=0
                Merged Map outputs=7
                GC time elapsed (ms)=1310
                CPU time spent (ms)=58060
                Physical memory (bytes) snapshot=2443984896
                Virtual memory (bytes) snapshot=22274195456
                Total committed heap usage (bytes)=1590689792
                Peak Map Physical memory (bytes)=339304448
                Peak Map Virtual memory (bytes)=2805575680
                Peak Reduce Physical memory (bytes)=266555392
                Peak Reduce Virtual memory (bytes)=2786078720
        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=868521790
        File Output Format Counters
                Bytes Written=27
2022-11-12 07:10:59,390 INFO streaming.StreamJob: Output directory: /project/output/
WB N CONDUIT AVE @ S    48831
Deleted /project/input
Deleted /project/output
Stopping namenodes on [bigdatacomputing1.c.ringed-metric-362423.internal]
Stopping datanodes
Stopping secondary namenodes [bigdatacomputing1]
Stopping nodemanagers
10.128.0.3: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill with kill -9
10.128.0.5: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill with kill -9
Stopping resourcemanager
WARNING: Use of this script to stop the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon stop" instead.
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project#
```

Q4 Which color of the vehicle is most likely to get a ticket

Use sys.stdin to read lines, just as with the last query. To get information about colors, use the split function, whose index is 33. After filtering out column names and missing values, output the color.

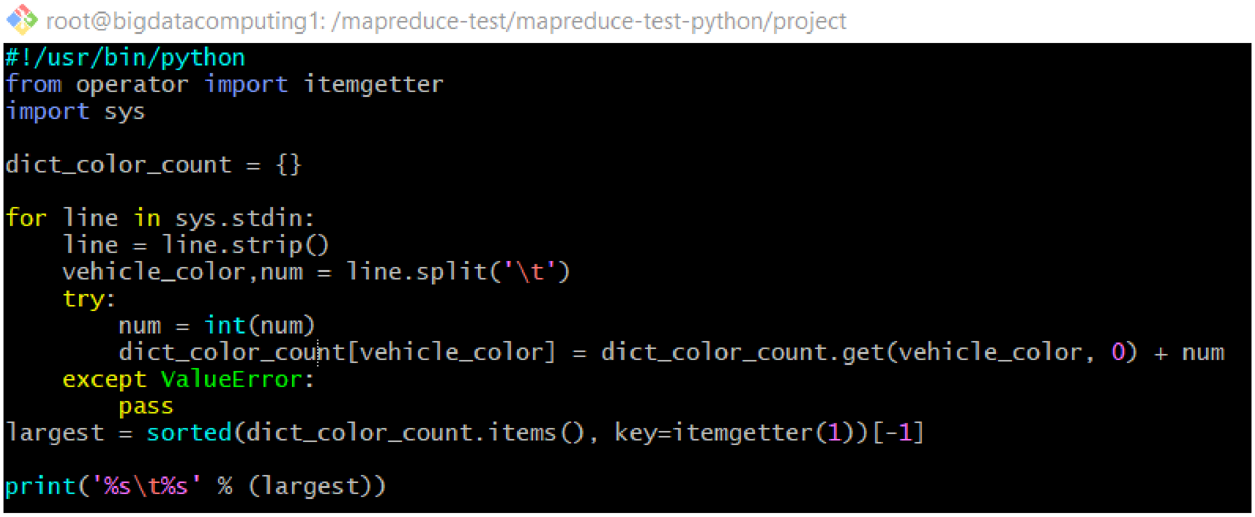
Here is the mapper_p1q4.py file:


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/proj

```
#!/usr/bin/python
# --*-- coding:utf-8 --*--
import sys

for line in sys.stdin:
    line = line.strip()
    vehicle_color = line.split(",")[33]
    if vehicle_color != "Vehicle Color":
        if vehicle_color != "":
                print('%s\t%s' % (vehicle_color, '1'))
~
~
```

Create a blank dictionary and use sys.stdin to read lines. Utilizing the strip function, split lines into street and num segments, then convert num to an integer for future calculations. Let num be the value and vehicle_color be the key. Add one to the number each time the same key (vehicle_color) appears. The final element should be output after sorting the dictionary by value.

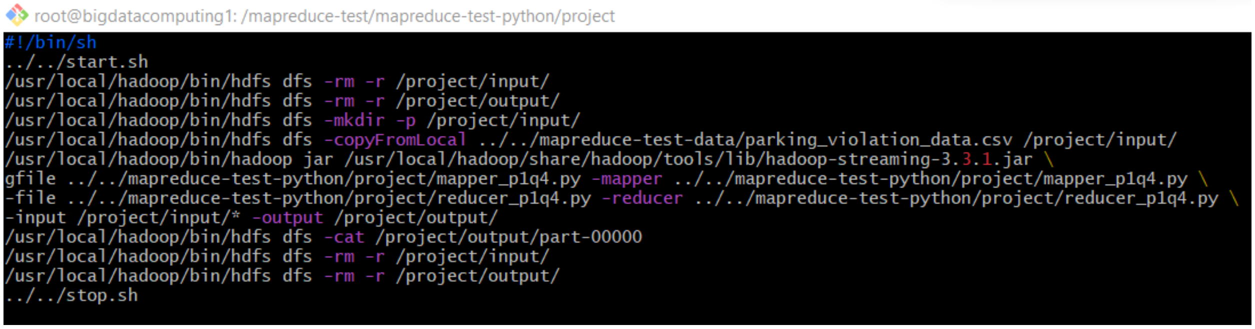
Here is the reducer_p1q4.py file:


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
#!/usr/bin/python
from operator import itemgetter
import sys

dict_color_count = {}

for line in sys.stdin:
    line = line.strip()
    vehicle_color,num = line.split('\t')
    try:
        num = int(num)
        dict_color_count[vehicle_color] = dict_color_count.get(vehicle_color, 0) + num
    except ValueError:
        pass
largest = sorted(dict_color_count.items(), key=itemgetter(1))[-1]

print('%s\t%s' % (largest))
```

Start hdfs, yarn, and historyserver with start.sh, then exit safe mode. Delete the input and output directories to guarantee that no input or output remains to muck up the input data. Make a directory for the project so that it may copy the dataset from the local directory. Begin a Hadoop streaming job to perform a map and reduce operation. Save output and read it with the cat command. Stop hdfs, yarn, and historyserver by deleting the input and output directories and calling stop.sh.

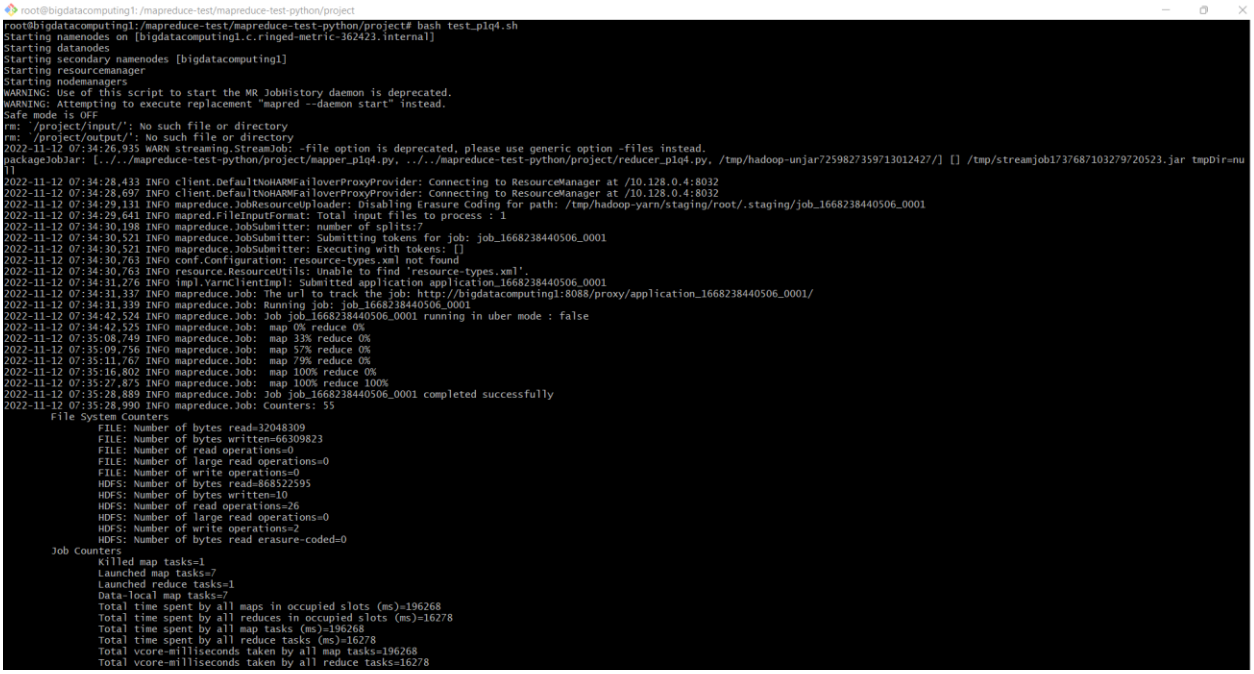
Here is the test_p1q4.sh file:



root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

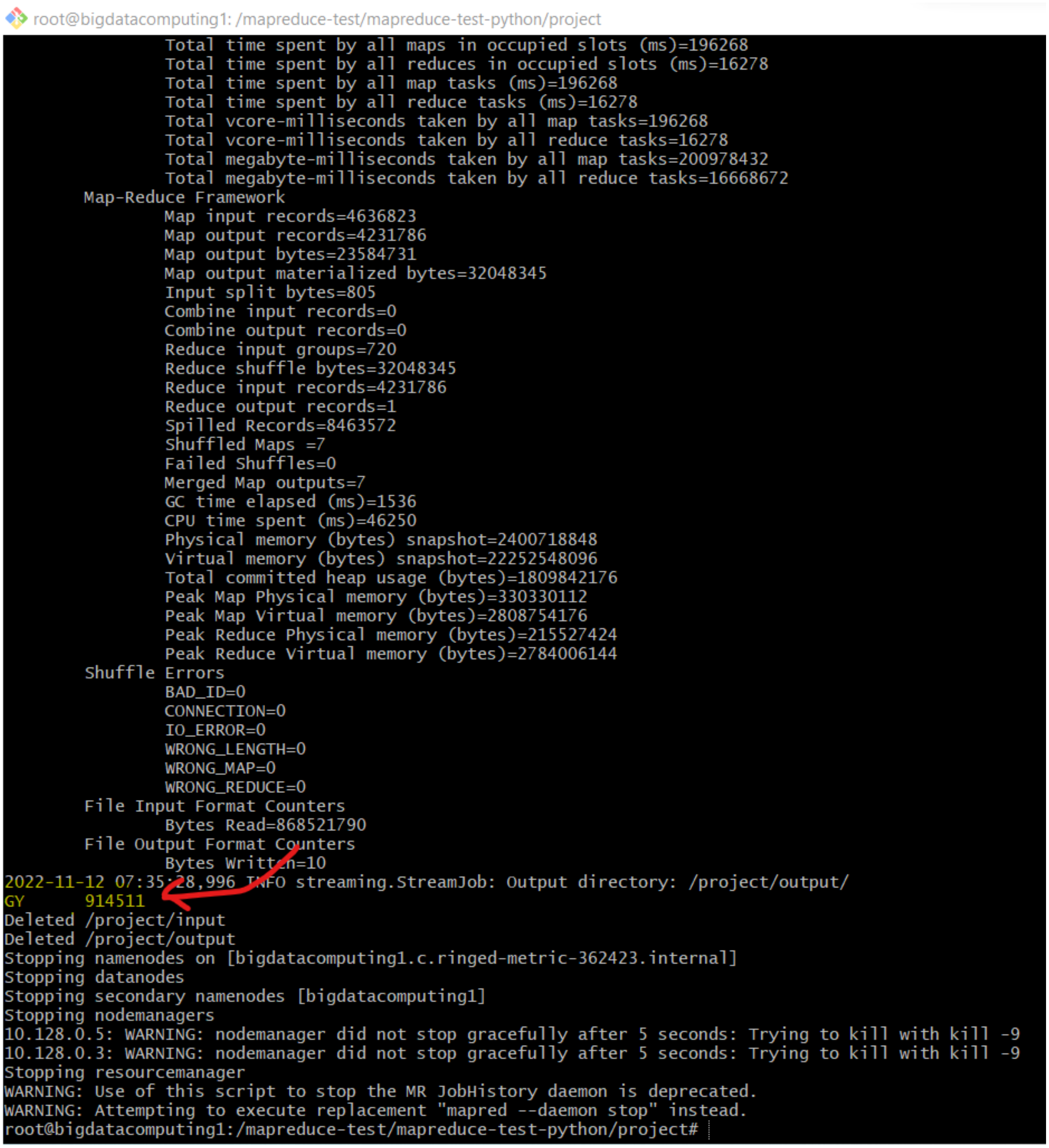
```
#!/bin/sh
../../start.sh
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/input/
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/output/
/usr/local/hadoop/bin/hdfs dfs -mkdir -p /project/input/
/usr/local/hadoop/bin/hdfs dfs -copyFromLocal ../../mapreduce-test-data/parking_violation_data.csv /project/input/
/usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.1.jar \
gfile ../../mapreduce-test-python/project/mapper_p1q4.py -mapper ../../mapreduce-test-python/project/mapper_p1q4.py \
-file ../../mapreduce-test-python/project/reducer_p1q4.py -reducer ../../mapreduce-test-python/project/reducer_p1q4.py \
-input /project/input/* -output /project/output/
/usr/local/hadoop/bin/hdfs dfs -cat /project/output/part-00000
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/input/
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/output/
../../stop.sh
```

Output:



root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project# bash test_p1q4.sh
Starting namenodes on [bigdatacomputing1.c.ringed-metric-362423.internal]
Starting datanodes
Starting secondary namenodes [bigdatacomputing1]
Starting resourcemanager
Starting nodemanagers
WARNING: Use of this script to start the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon start" instead.
Safe mode is OFF
rm: `/project/input/': No such file or directory
rm: `/project/output/': No such file or directory
2022-11-12 07:34:26,935 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [../../mapreduce-test-python/project/mapper_p1q4.py, ../../mapreduce-test-python/project/reducer_p1q4.py, /tmp/hadoop-unjar7259827359713012427/] [] /tmp/streamjob1737687103279720523.jar tmpDir=nu
ll
2022-11-12 07:34:28,433 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /10.128.0.4:8032
2022-11-12 07:34:28,697 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /10.128.0.4:8032
2022-11-12 07:34:29,131 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1668238440506_0001
2022-11-12 07:34:29,641 INFO mapred.FileInputFormat: Total input files to process : 1
2022-11-12 07:34:30,198 INFO mapreduce.JobSubmitter: number of splits:7
2022-11-12 07:34:30,521 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1668238440506_0001
2022-11-12 07:34:30,521 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-11-12 07:34:30,763 INFO conf.Configuration: resource-types.xml not found
2022-11-12 07:34:30,763 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-11-12 07:34:31,276 INFO impl.YarnClientImpl: Submitted application application_1668238440506_0001
2022-11-12 07:34:31,337 INFO mapreduce.Job: The url to track the job: http://bigdatacomputing1:8088/proxy/application_1668238440506_0001/
2022-11-12 07:34:31,339 INFO mapreduce.Job: Running job: job_1668238440506_0001
2022-11-12 07:34:42,524 INFO mapreduce.Job: Job job_1668238440506_0001 running in uber mode : false
2022-11-12 07:34:42,525 INFO mapreduce.Job:  map 0% reduce 0%
2022-11-12 07:35:08,749 INFO mapreduce.Job:  map 33% reduce 0%
2022-11-12 07:35:09,756 INFO mapreduce.Job:  map 57% reduce 0%
2022-11-12 07:35:11,767 INFO mapreduce.Job:  map 79% reduce 0%
2022-11-12 07:35:16,802 INFO mapreduce.Job:  map 100% reduce 0%
2022-11-12 07:35:27,875 INFO mapreduce.Job:  map 100% reduce 100%
2022-11-12 07:35:28,889 INFO mapreduce.Job: Job job_1668238440506_0001 completed successfully
2022-11-12 07:35:28,990 INFO mapreduce.Job: Counters: 55
        File System Counters
                FILE: Number of bytes read=32048309
                FILE: Number of bytes written=66309823
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=868522595
                HDFS: Number of bytes written=10
                HDFS: Number of read operations=26
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Killed map tasks=1
                Launched map tasks=7
                Launched reduce tasks=1
                Data-local map tasks=7
                Total time spent by all maps in occupied slots (ms)=196268
                Total time spent by all reduces in occupied slots (ms)=16278
                Total time spent by all map tasks (ms)=196268
                Total time spent by all reduce tasks (ms)=16278
                Total vcore-milliseconds taken by all map tasks=196268
                Total vcore-milliseconds taken by all reduce tasks=16278
```

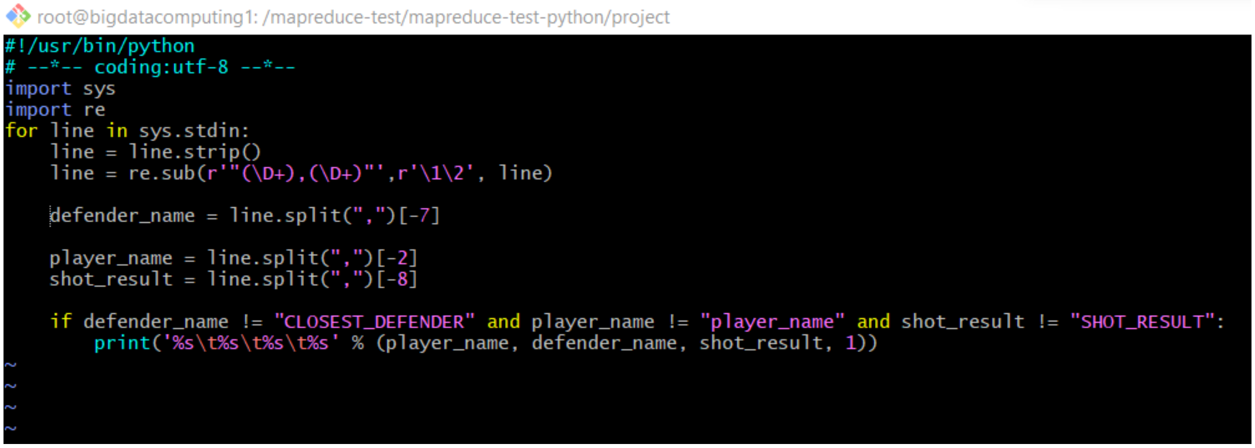

root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
                Total time spent by all maps in occupied slots (ms)=196268
                Total time spent by all reduces in occupied slots (ms)=16278
                Total time spent by all map tasks (ms)=196268
                Total time spent by all reduce tasks (ms)=16278
                Total vcore-milliseconds taken by all map tasks=196268
                Total vcore-milliseconds taken by all reduce tasks=16278
                Total megabyte-milliseconds taken by all map tasks=200978432
                Total megabyte-milliseconds taken by all reduce tasks=16668672
        Map-Reduce Framework
                Map input records=4636823
                Map output records=4231786
                Map output bytes=23584731
                Map output materialized bytes=32048345
                Input split bytes=805
                Combine input records=0
                Combine output records=0
                Reduce input groups=720
                Reduce shuffle bytes=32048345
                Reduce input records=4231786
                Reduce output records=1
                Spilled Records=8463572
                Shuffled Maps =7
                Failed Shuffles=0
                Merged Map outputs=7
                GC time elapsed (ms)=1536
                CPU time spent (ms)=46250
                Physical memory (bytes) snapshot=2400718848
                Virtual memory (bytes) snapshot=22252548096
                Total committed heap usage (bytes)=1809842176
                Peak Map Physical memory (bytes)=330330112
                Peak Map Virtual memory (bytes)=2808754176
                Peak Reduce Physical memory (bytes)=215527424
                Peak Reduce Virtual memory (bytes)=2784006144
        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=868521790
        File Output Format Counters
                Bytes Written=10
2022-11-12 07:35:28,996 INFO streaming.StreamJob: Output directory: /project/output/
GY      914511
Deleted /project/input
Deleted /project/output
Stopping namenodes on [bigdatacomputing1.c.ringed-metric-362423.internal]
Stopping datanodes
Stopping secondary namenodes [bigdatacomputing1]
Stopping nodemanagers
10.128.0.5: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill with kill -9
10.128.0.3: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill with kill -9
Stopping resourcemanager
WARNING: Use of this script to stop the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon stop" instead.
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project#
```

Part 2

Q1 Based on the fear sore, for each player, please find out who is his “most unwanted defender”

Divides on commas and removes all whitespace to produce a list of strings. Afterward, search for any strings that do not contain "closest defender" or "player name." If you discover one, delete any leading zeros and replace all spaces with underscores. Next, determine whether defender and player are not equal to closet_defender and player_name, respectively. If so, remove any leading zeros from the results and replace all white space with an underscore. output the outcome lastly. Here is the mapper_p2q1.py file:

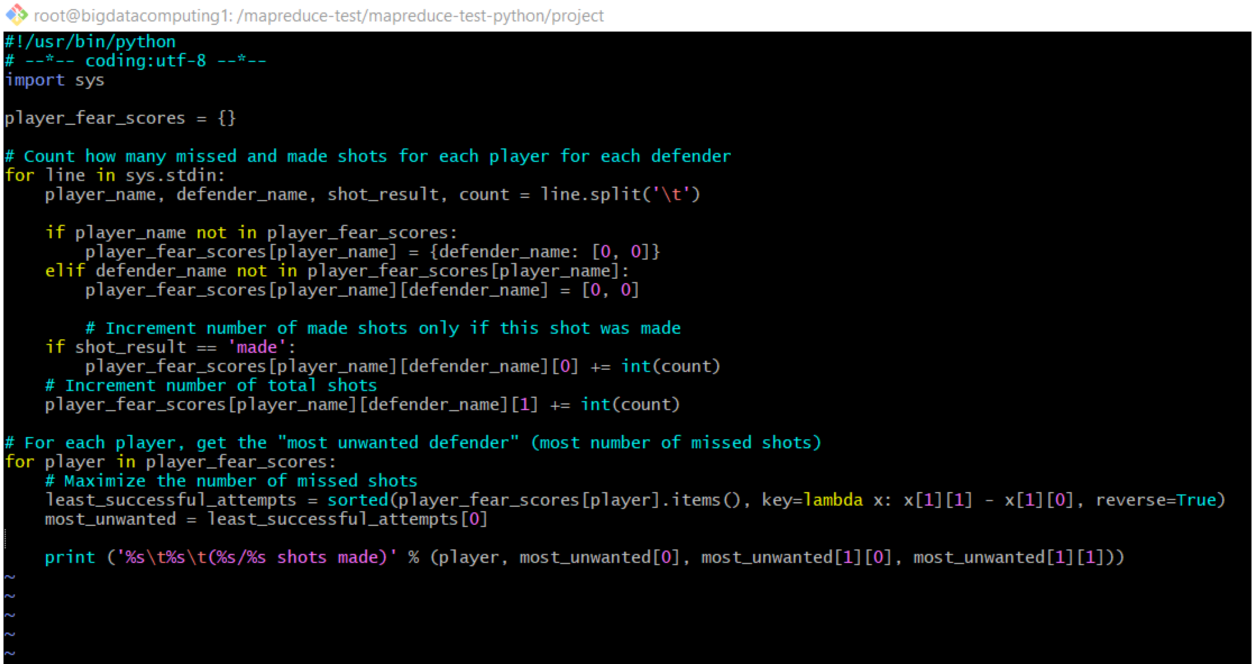


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
#!/usr/bin/python
# --*-- coding:utf-8 --*--
import sys
import re
for line in sys.stdin:
    line = line.strip()
    line = re.sub(r'"(\D+),(\D+)"',r'\1\2', line)

    defender_name = line.split(",")[-7]

    player_name = line.split(",")[-2]
    shot_result = line.split(",")[-8]

    if defender_name != "CLOSEST_DEFENDER" and player_name != "player_name" and shot_result != "SHOT_RESULT":
        print('%s\t%s\t%s\t%s' % (player_name, defender_name, shot_result, 1))
~
~
~
~
```

Use sys.stdin to read lines. After removing any newlines or tabs and separating the string into player and hit points, it iterates through each line of data one more time. The first portion is kept in a player variable, which has an empty list for hit points, and the second part is kept in a num variable, which has an integer value for the number of hit points to be added to that specific player's total hit points. Currently, each element in the players' list of hit points is being converted into a key-value pair in a dictionary called dic. All values connected to each key will be stored in this dictionary and kept there so they can be accessed at a later time. Iterate through the key-value pairs that make up the objects in the player_dict dictionary. Since "player" is the initial entry, a list will be created with "player" as the only item. As the second element, create a new list with values that are identical to those in the player dict. This procedure continues until there are no longer any keys to iterate over or lists to create. By going over the list of players and defenders and adding the most recent pair of names with each iteration, create a dictionary of player and defender pairs.

Here is the reducer_p2q1.py file:

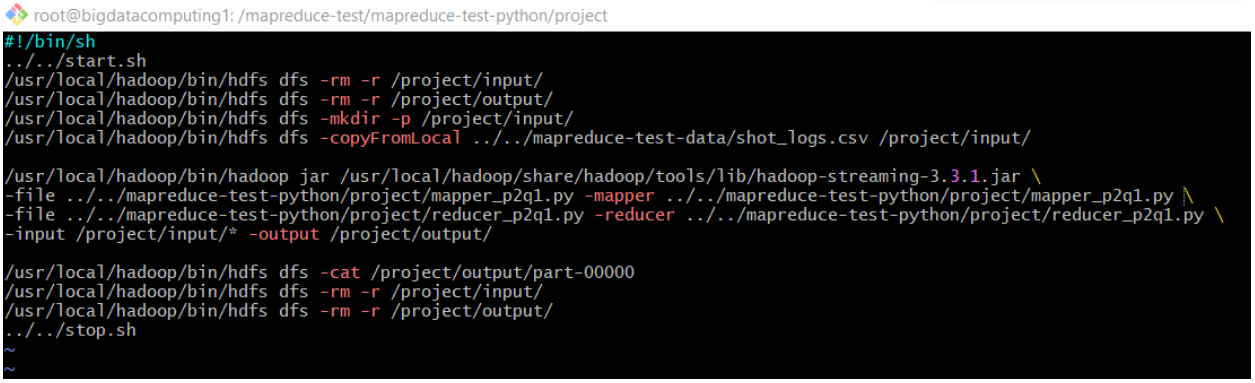


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
#!/usr/bin/python
# --*-- coding:utf-8 --*--
import sys

player_fear_scores = {}

# Count how many missed and made shots for each player for each defender
for line in sys.stdin:
    player_name, defender_name, shot_result, count = line.split('\t')

    if player_name not in player_fear_scores:
        player_fear_scores[player_name] = {defender_name: [0, 0]}
    elif defender_name not in player_fear_scores[player_name]:
        player_fear_scores[player_name][defender_name] = [0, 0]

        # Increment number of made shots only if this shot was made
    if shot_result == 'made':
        player_fear_scores[player_name][defender_name][0] += int(count)
    # Increment number of total shots
    player_fear_scores[player_name][defender_name][1] += int(count)

# For each player, get the "most unwanted defender" (most number of missed shots)
for player in player_fear_scores:
    # Maximize the number of missed shots
    least_successful_attempts = sorted(player_fear_scores[player].items(), key=lambda x: x[1][1] - x[1][0], reverse=True)
    most_unwanted = least_successful_attempts[0]

    print ('%s\t%s\t(%s/%s shots made)' % (player, most_unwanted[0], most_unwanted[1][0], most_unwanted[1][1]))
~
~
~
~
~
```

Start hdfs, yarn, and historyserver with start.sh, then exit safe mode. Delete the input and output directories to guarantee that no input or output remains to muck up the input data. Make a directory for the project so that it may copy the dataset from the local directory. Begin a Hadoop streaming job to perform a map and reduce operation. Save output and read it with the cat command. Stop hdfs, yarn, and historyserver by deleting the input and output directories and calling stop.sh.

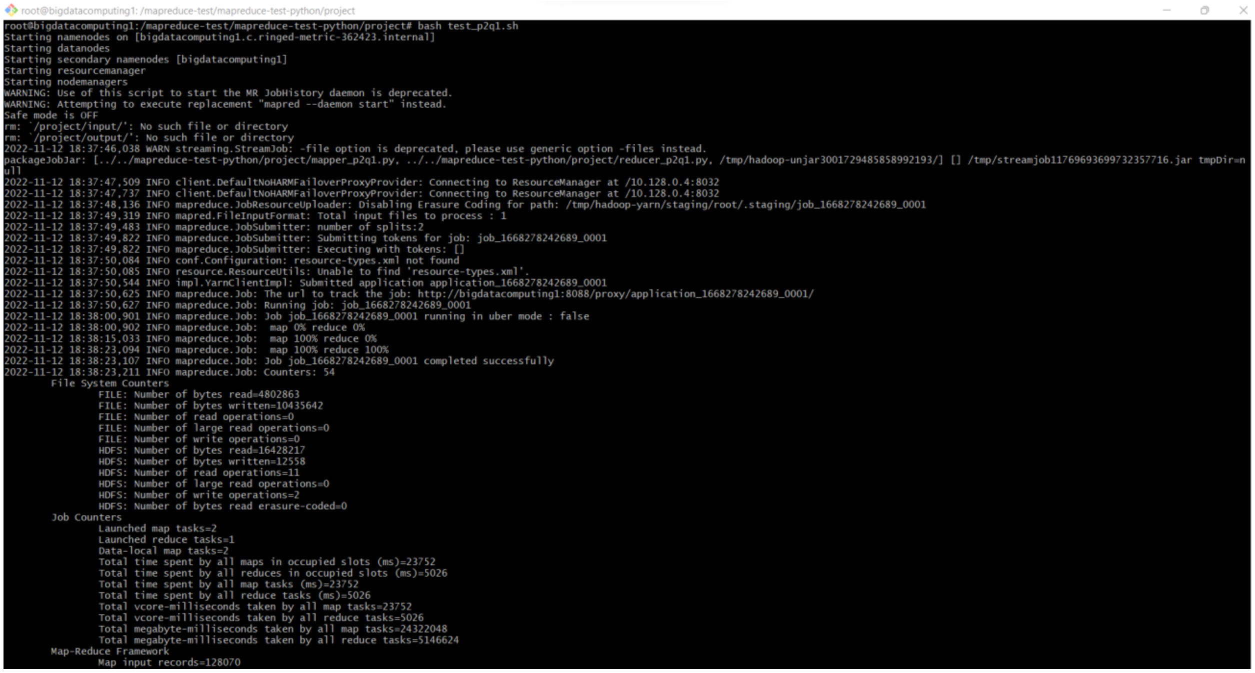
Here is the test_p2q1.sh file:



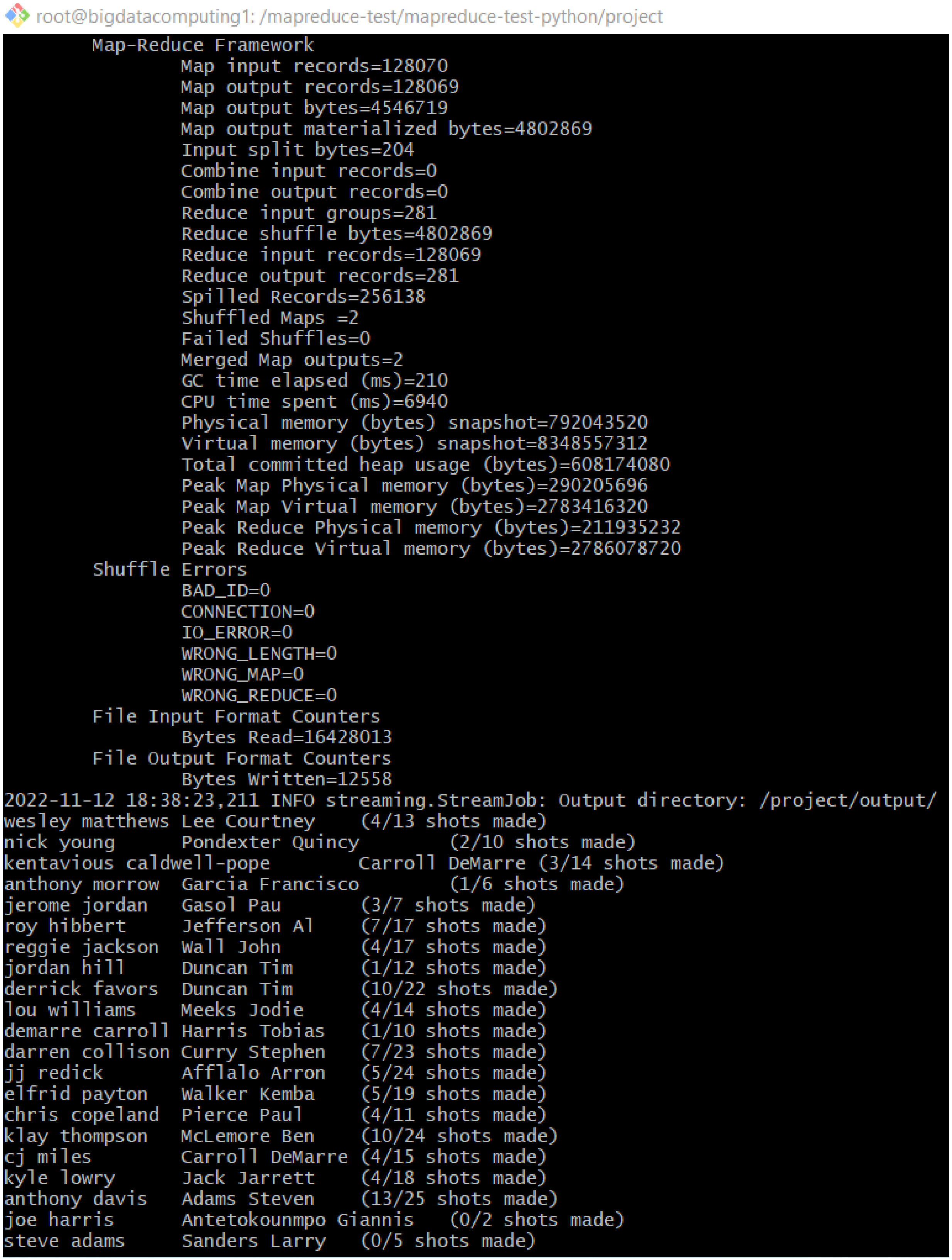
root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
#!/bin/sh
../../start.sh
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/input/
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/output/
/usr/local/hadoop/bin/hdfs dfs -mkdir -p /project/input/
/usr/local/hadoop/bin/hdfs dfs -copyFromLocal ../../mapreduce-test-data/shot_logs.csv /project/input/

/usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.1.jar \
-file ../../mapreduce-test-python/project/mapper_p2q1.py -mapper ../../mapreduce-test-python/project/mapper_p2q1.py \
-file ../../mapreduce-test-python/project/reducer_p2q1.py -reducer ../../mapreduce-test-python/project/reducer_p2q1.py \
-input /project/input/* -output /project/output/

/usr/local/hadoop/bin/hdfs dfs -cat /project/output/part-00000
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/input/
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/output/
../../stop.sh
~
~
```

Output:

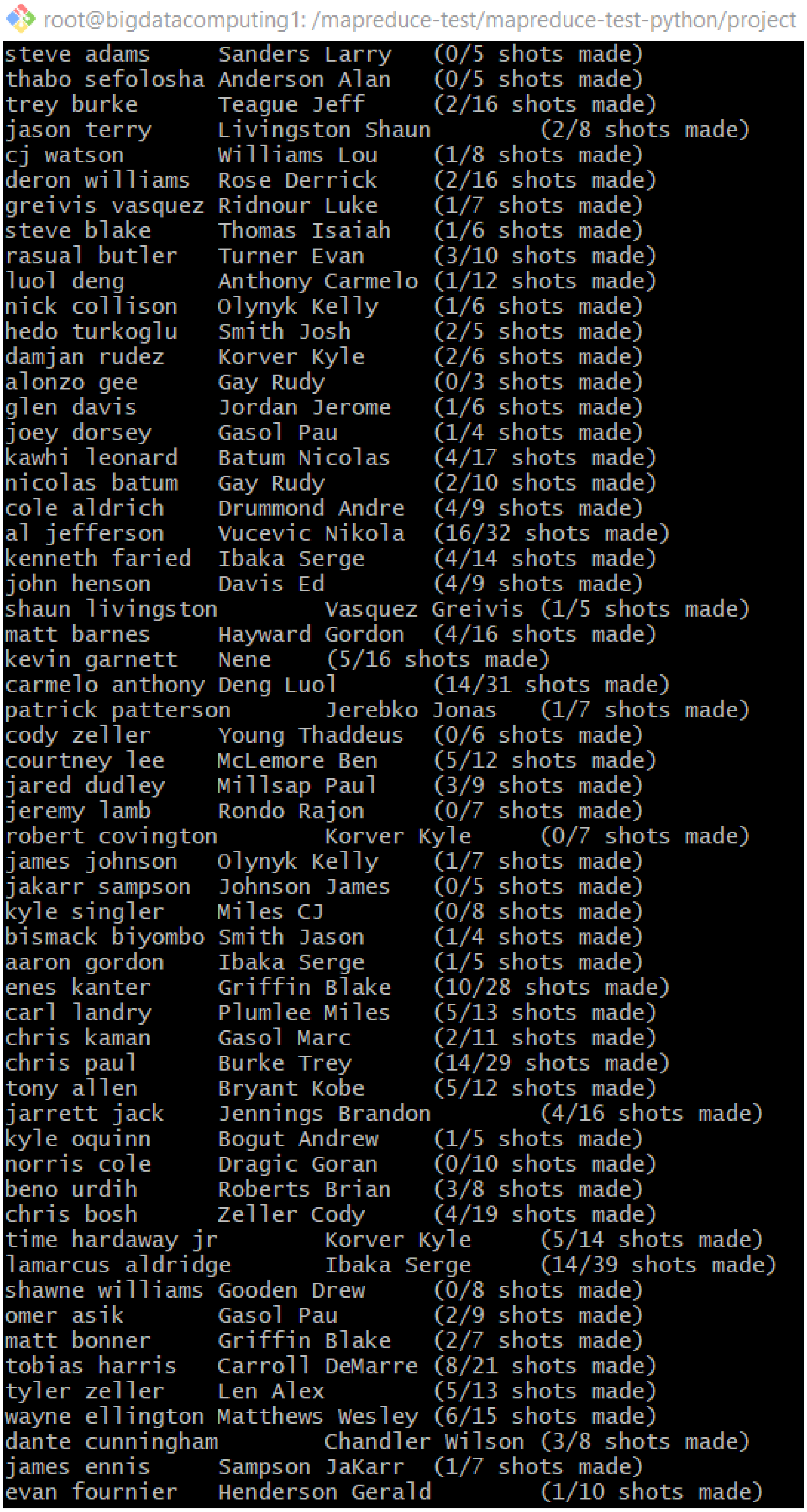

root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project# bash test_p2q1.sh
Starting namenodes on [bigdatacomputing1.c.ringed-metric-362423.internal]
Starting datanodes
Starting secondary namenodes [bigdatacomputing1]
Starting resourcemanager
Starting nodemanagers
WARNING: Use of this script to start the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon start" instead.
Safe mode is OFF
rm: `/project/input/': No such file or directory
rm: `/project/output/': No such file or directory
2022-11-12 18:37:46,038 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [../../mapreduce-test-python/project/mapper_p2q1.py, ../../mapreduce-test-python/project/reducer_p2q1.py, /tmp/hadoop-unjar3001729485858992193/] [] /tmp/streamjob1176969369973235716.jar tmpDir=n
ull
2022-11-12 18:37:47,509 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /10.128.0.4:8032
2022-11-12 18:37:47,737 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /10.128.0.4:8032
2022-11-12 18:37:48,136 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1668278242689_0001
2022-11-12 18:37:49,319 INFO mapred.FileInputFormat: Total input files to process : 1
2022-11-12 18:37:49,483 INFO mapreduce.JobSubmitter: number of splits:2
2022-11-12 18:37:49,822 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1668278242689_0001
2022-11-12 18:37:49,822 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-11-12 18:37:50,084 INFO conf.Configuration: resource-types.xml not found
2022-11-12 18:37:50,085 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-11-12 18:37:50,544 INFO impl.YarnClientImpl: Submitted application application_1668278242689_0001
2022-11-12 18:37:50,625 INFO mapreduce.Job: The url to track the job: http://bigdatacomputing1:8088/proxy/application_1668278242689_0001/
2022-11-12 18:37:50,627 INFO mapreduce.Job: Running job: job_1668278242689_0001
2022-11-12 18:38:00,901 INFO mapreduce.Job: Job job_1668278242689_0001 running in uber mode : false
2022-11-12 18:38:00,902 INFO mapreduce.Job:  map 0% reduce 0%
2022-11-12 18:38:15,033 INFO mapreduce.Job:  map 100% reduce 0%
2022-11-12 18:38:23,094 INFO mapreduce.Job:  map 100% reduce 100%
2022-11-12 18:38:23,107 INFO mapreduce.Job: Job job_1668278242689_0001 completed successfully
2022-11-12 18:38:23,211 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=4802863
                FILE: Number of bytes written=10435642
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=16428217
                HDFS: Number of bytes written=12558
                HDFS: Number of read operations=11
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=23752
                Total time spent by all reduces in occupied slots (ms)=5026
                Total time spent by all map tasks (ms)=23752
                Total time spent by all reduce tasks (ms)=5026
                Total vcore-milliseconds taken by all map tasks=23752
                Total vcore-milliseconds taken by all reduce tasks=5026
                Total megabyte-milliseconds taken by all map tasks=24322048
                Total megabyte-milliseconds taken by all reduce tasks=5146624
        Map-Reduce Framework
                Map input records=128070
```



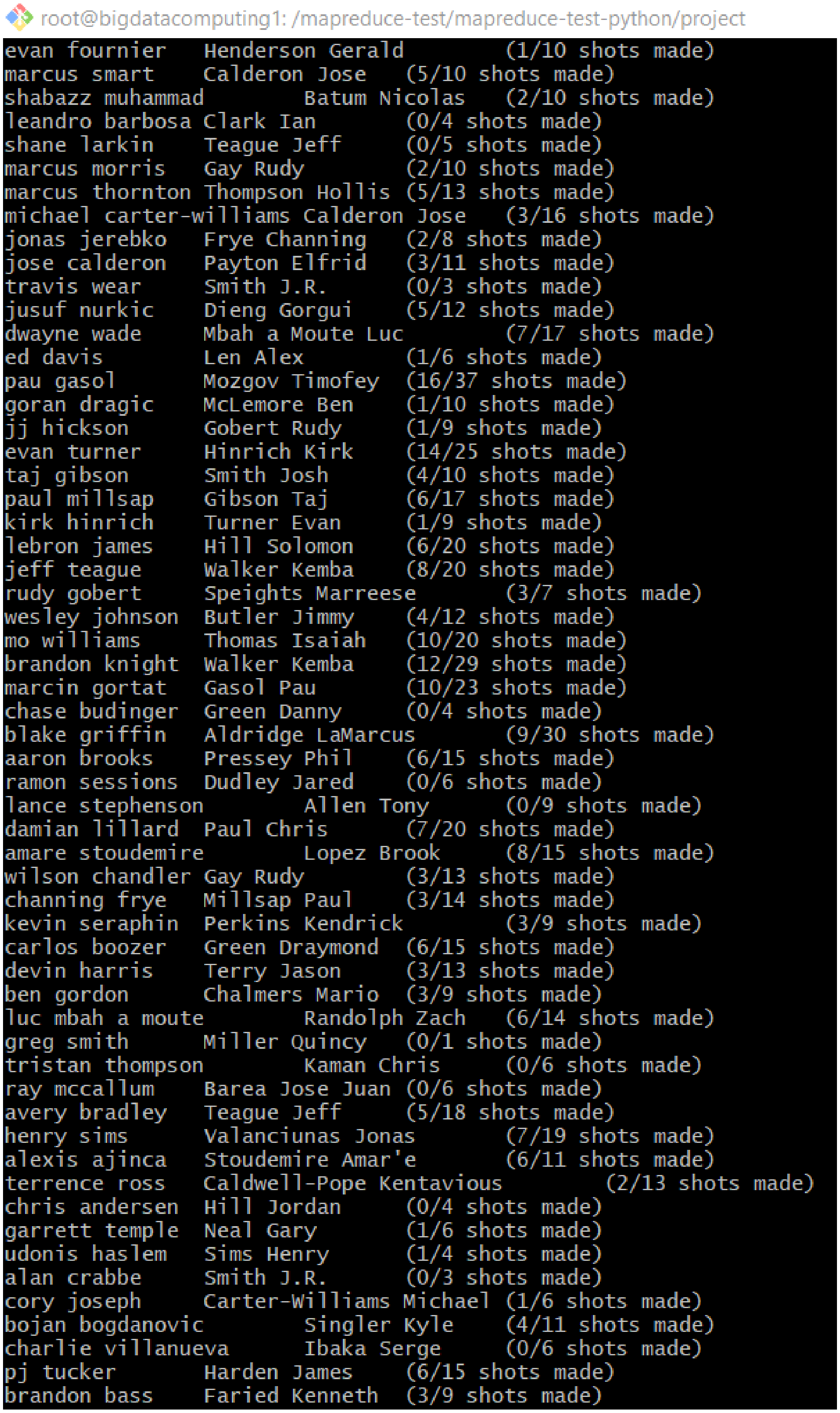
root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
        Map-Reduce Framework
                Map input records=128070
                Map output records=128069
                Map output bytes=4546719
                Map output materialized bytes=4802869
                Input split bytes=204
                Combine input records=0
                Combine output records=0
                Reduce input groups=281
                Reduce shuffle bytes=4802869
                Reduce input records=128069
                Reduce output records=281
                Spilled Records=256138
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=210
                CPU time spent (ms)=6940
                Physical memory (bytes) snapshot=792043520
                Virtual memory (bytes) snapshot=8348557312
                Total committed heap usage (bytes)=608174080
                Peak Map Physical memory (bytes)=290205696
                Peak Map Virtual memory (bytes)=2783416320
                Peak Reduce Physical memory (bytes)=211935232
                Peak Reduce Virtual memory (bytes)=2786078720
        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=16428013
        File Output Format Counters
                Bytes Written=12558
2022-11-12 18:38:23,211 INFO streaming.StreamJob: Output directory: /project/output/
wesley matthews Lee Courtney    (4/13 shots made)
nick young      Pondexter Quincy        (2/10 shots made)
kentavious caldwell-pope        Carroll DeMarre (3/14 shots made)
anthony morrow  Garcia Francisco        (1/6 shots made)
jerome jordan   Gasol Pau       (3/7 shots made)
roy hibbert     Jefferson Al    (7/17 shots made)
reggie jackson  Wall John       (4/17 shots made)
jordan hill     Duncan Tim      (1/12 shots made)
derrick favors  Duncan Tim      (10/22 shots made)
lou williams    Meeks Jodie     (4/14 shots made)
demarre carroll Harris Tobias   (1/10 shots made)
darren collison Curry Stephen   (7/23 shots made)
jj redick       Afflalo Arron   (5/24 shots made)
elfrid payton   Walker Kemba    (5/19 shots made)
chris copeland  Pierce Paul     (4/11 shots made)
klay thompson   McLemore Ben    (10/24 shots made)
cj miles        Carroll DeMarre (4/15 shots made)
kyle lowry      Jack Jarrett    (4/18 shots made)
anthony davis   Adams Steven    (13/25 shots made)
joe harris      Antetokounmpo Giannis   (0/2 shots made)
steve adams     Sanders Larry   (0/5 shots made)
```



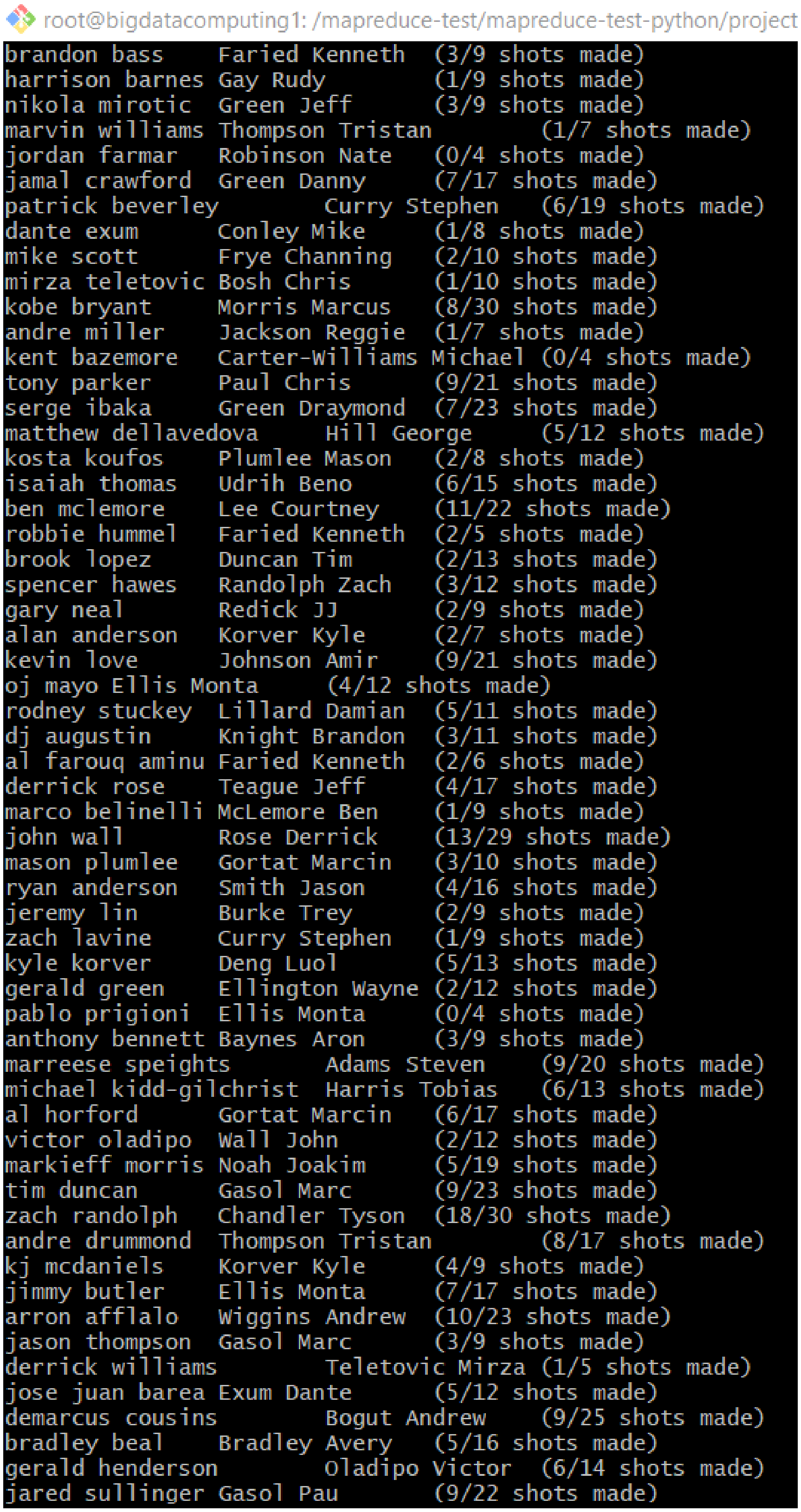
root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
steve adams     Sanders Larry   (0/5 shots made)
thabo sefolosha Anderson Alan   (0/5 shots made)
trey burke      Teague Jeff     (2/16 shots made)
jason terry     Livingston Shaun        (2/8 shots made)
cj watson       Williams Lou    (1/8 shots made)
deron williams  Rose Derrick    (2/16 shots made)
greivis vasquez Ridnour Luke    (1/7 shots made)
steve blake     Thomas Isaiah   (1/6 shots made)
rasual butler   Turner Evan     (3/10 shots made)
luol deng       Anthony Carmelo (1/12 shots made)
nick collison   Olynyk Kelly    (1/6 shots made)
hedo turkoglu   Smith Josh      (2/5 shots made)
damjan rudez    Korver Kyle     (2/6 shots made)
alonzo gee      Gay Rudy        (0/3 shots made)
glen davis      Jordan Jerome   (1/6 shots made)
joey dorsey     Gasol Pau       (1/4 shots made)
kawhi leonard   Batum Nicolas   (4/17 shots made)
nicolas batum   Gay Rudy        (2/10 shots made)
cole aldrich    Drummond Andre  (4/9 shots made)
al jefferson    Vucevic Nikola  (16/32 shots made)
kenneth faried  Ibaka Serge     (4/14 shots made)
john henson     Davis Ed        (4/9 shots made)
shaun livingston        Vasquez Greivis (1/5 shots made)
matt barnes     Hayward Gordon  (4/16 shots made)
kevin garnett   Nene    (5/16 shots made)
carmelo anthony Deng Luol       (14/31 shots made)
patrick patterson       Jerebko Jonas   (1/7 shots made)
cody zeller     Young Thaddeus  (0/6 shots made)
courtney lee    McLemore Ben    (5/12 shots made)
jared dudley    Millsap Paul    (3/9 shots made)
jeremy lamb     Rondo Rajon     (0/7 shots made)
robert covington        Korver Kyle     (0/7 shots made)
james johnson   Olynyk Kelly    (1/7 shots made)
jakarr sampson  Johnson James   (0/5 shots made)
kyle singler    Miles CJ        (0/8 shots made)
bismack biyombo Smith Jason     (1/4 shots made)
aaron gordon    Ibaka Serge     (1/5 shots made)
enes kanter     Griffin Blake   (10/28 shots made)
carl landry     Plumlee Miles   (5/13 shots made)
chris kaman     Gasol Marc      (2/11 shots made)
chris paul      Burke Trey      (14/29 shots made)
tony allen      Bryant Kobe     (5/12 shots made)
jarrett jack    Jennings Brandon        (4/16 shots made)
kyle oquinn     Bogut Andrew    (1/5 shots made)
norris cole     Dragic Goran    (0/10 shots made)
beno urdih      Roberts Brian   (3/8 shots made)
chris bosh      Zeller Cody     (4/19 shots made)
time hardaway jr        Korver Kyle     (5/14 shots made)
lamarcus aldridge       Ibaka Serge     (14/39 shots made)
shawne williams Gooden Drew     (0/8 shots made)
omer asik       Gasol Pau       (2/9 shots made)
matt bonner     Griffin Blake   (2/7 shots made)
tobias harris   Carroll DeMarre (8/21 shots made)
tyler zeller    Len Alex        (5/13 shots made)
wayne ellington Matthews Wesley (6/15 shots made)
dante cunningham        Chandler Wilson (3/8 shots made)
james ennis     Sampson JaKarr  (1/7 shots made)
evan fournier   Henderson Gerald        (1/10 shots made)
```



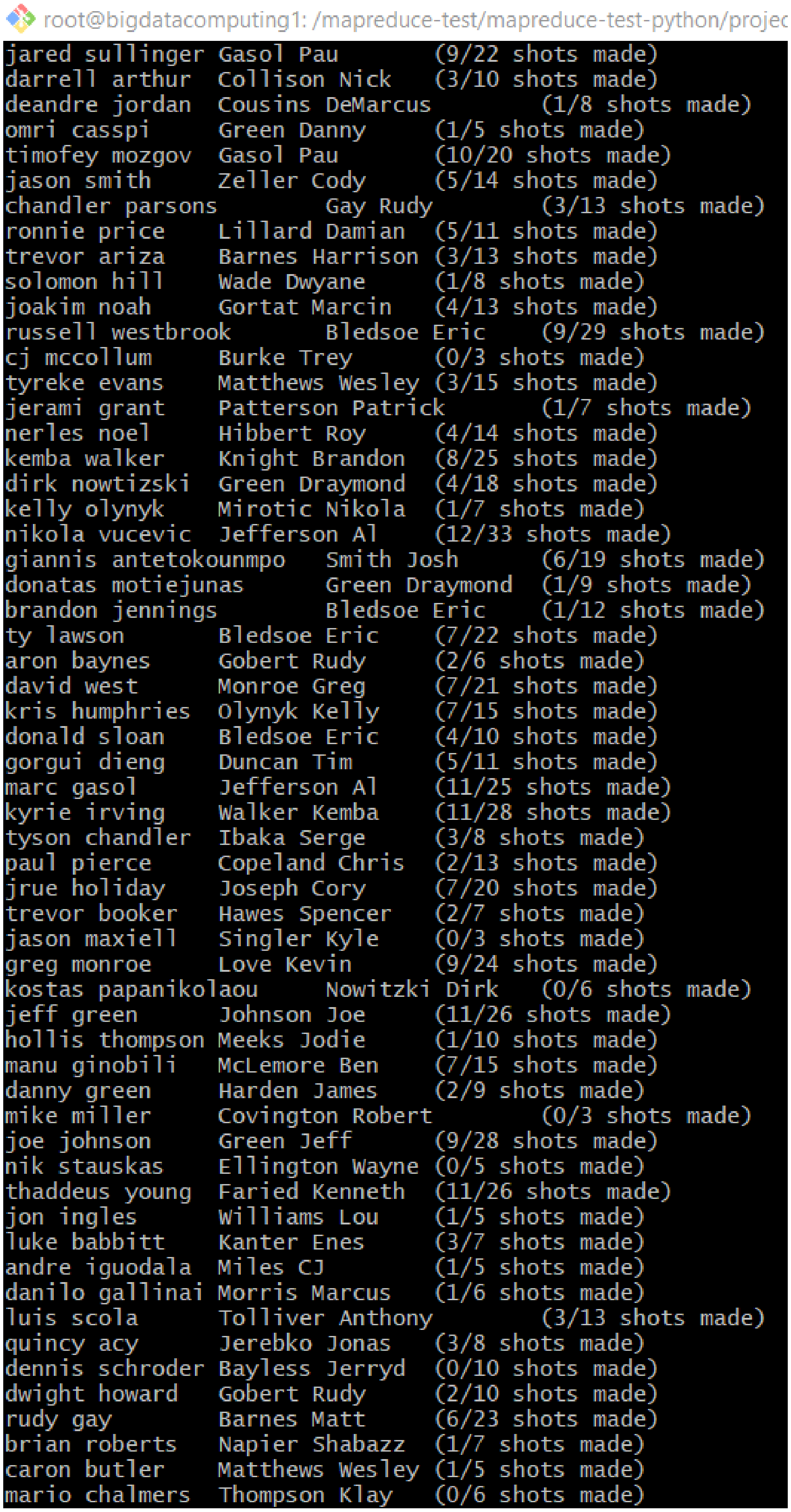
root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
evan fournier   Henderson Gerald        (1/10 shots made)
marcus smart    Calderon Jose   (5/10 shots made)
shabazz muhammad        Batum Nicolas   (2/10 shots made)
leandro barbosa Clark Ian       (0/4 shots made)
shane larkin    Teague Jeff     (0/5 shots made)
marcus morris   Gay Rudy        (2/10 shots made)
marcus thornton Thompson Hollis (5/13 shots made)
michael carter-williams Calderon Jose   (3/16 shots made)
jonas jerebko   Frye Channing   (2/8 shots made)
jose calderon   Payton Elfrid   (3/11 shots made)
travis wear     Smith J.R.      (0/3 shots made)
jusuf nurkic    Dieng Gorgui    (5/12 shots made)
dwayne wade     Mbah a Moute Luc        (7/17 shots made)
ed davis        Len Alex        (1/6 shots made)
pau gasol       Mozgov Timofey  (16/37 shots made)
goran dragic    McLemore Ben    (1/10 shots made)
jj hickson      Gobert Rudy     (1/9 shots made)
evan turner     Hinrich Kirk    (14/25 shots made)
taj gibson      Smith Josh      (4/10 shots made)
paul millsap    Gibson Taj      (6/17 shots made)
kirk hinrich    Turner Evan     (1/9 shots made)
lebron james    Hill Solomon    (6/20 shots made)
jeff teague     Walker Kemba    (8/20 shots made)
rudy gobert     Speights Marreese       (3/7 shots made)
wesley johnson  Butler Jimmy    (4/12 shots made)
mo williams     Thomas Isaiah   (10/20 shots made)
brandon knight  Walker Kemba    (12/29 shots made)
marcin gortat   Gasol Pau       (10/23 shots made)
chase budinger  Green Danny     (0/4 shots made)
blake griffin   Aldridge LaMarcus       (9/30 shots made)
aaron brooks    Pressey Phil    (6/15 shots made)
ramon sessions  Dudley Jared    (0/6 shots made)
lance stephenson        Allen Tony      (0/9 shots made)
damian lillard  Paul Chris      (7/20 shots made)
amare stoudemire        Lopez Brook     (8/15 shots made)
wilson chandler Gay Rudy        (3/13 shots made)
channing frye   Millsap Paul    (3/14 shots made)
kevin seraphin  Perkins Kendrick        (3/9 shots made)
carlos boozer   Green Draymond  (6/15 shots made)
devin harris    Terry Jason     (3/13 shots made)
ben gordon      Chalmers Mario  (3/9 shots made)
luc mbah a moute        Randolph Zach   (6/14 shots made)
greg smith      Miller Quincy   (0/1 shots made)
tristan thompson        Kaman Chris     (0/6 shots made)
ray mccallum    Barea Jose Juan (0/6 shots made)
avery bradley   Teague Jeff     (5/18 shots made)
henry sims      Valanciunas Jonas       (7/19 shots made)
alexis ajinca   Stoudemire Amar'e       (6/11 shots made)
terrence ross   Caldwell-Pope Kentavious        (2/13 shots made)
chris andersen  Hill Jordan     (0/4 shots made)
garrett temple  Neal Gary       (1/6 shots made)
udonis haslem   Sims Henry      (1/4 shots made)
alan crabbe     Smith J.R.      (0/3 shots made)
cory joseph     Carter-Williams Michael (1/6 shots made)
bojan bogdanovic        Singler Kyle    (4/11 shots made)
charlie villanueva      Ibaka Serge     (0/6 shots made)
pj tucker       Harden James    (6/15 shots made)
brandon bass    Faried Kenneth  (3/9 shots made)
```



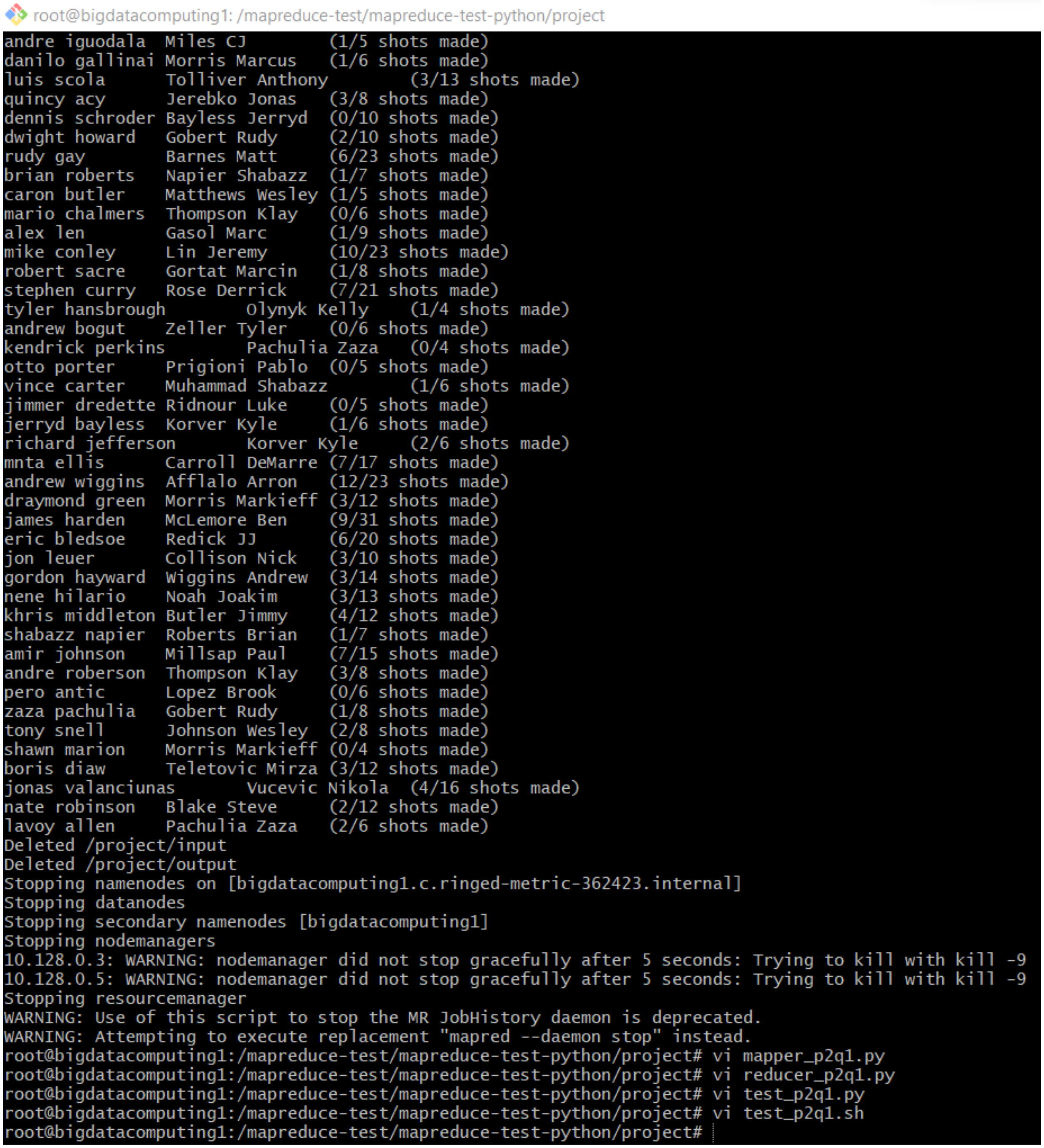
root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
brandon bass    Faried Kenneth  (3/9 shots made)
harrison barnes Gay Rudy        (1/9 shots made)
nikola mirotic  Green Jeff      (3/9 shots made)
marvin williams Thompson Tristan        (1/7 shots made)
jordan farmar   Robinson Nate   (0/4 shots made)
jamal crawford  Green Danny     (7/17 shots made)
patrick beverley        Curry Stephen   (6/19 shots made)
dante exum      Conley Mike     (1/8 shots made)
mike scott      Frye Channing   (2/10 shots made)
mirza teletovic Bosh Chris      (1/10 shots made)
kobe bryant     Morris Marcus   (8/30 shots made)
andre miller    Jackson Reggie  (1/7 shots made)
kent bazemore   Carter-Williams Michael (0/4 shots made)
tony parker     Paul Chris      (9/21 shots made)
serge ibaka     Green Draymond  (7/23 shots made)
matthew dellavedova     Hill George     (5/12 shots made)
kosta koufos    Plumlee Mason   (2/8 shots made)
isaiah thomas   Udrih Beno      (6/15 shots made)
ben mclemore    Lee Courtney    (11/22 shots made)
robbie hummel   Faried Kenneth  (2/5 shots made)
brook lopez     Duncan Tim      (2/13 shots made)
spencer hawes   Randolph Zach   (3/12 shots made)
gary neal       Redick JJ       (2/9 shots made)
alan anderson   Korver Kyle     (2/7 shots made)
kevin love      Johnson Amir    (9/21 shots made)
oj mayo Ellis Monta     (4/12 shots made)
rodney stuckey  Lillard Damian  (5/11 shots made)
dj augustin     Knight Brandon  (3/11 shots made)
al farouq aminu Faried Kenneth  (2/6 shots made)
derrick rose    Teague Jeff     (4/17 shots made)
marco belinelli McLemore Ben    (1/9 shots made)
john wall       Rose Derrick    (13/29 shots made)
mason plumlee   Gortat Marcin   (3/10 shots made)
ryan anderson   Smith Jason     (4/16 shots made)
jeremy lin      Burke Trey      (2/9 shots made)
zach lavine     Curry Stephen   (1/9 shots made)
kyle korver     Deng Luol       (5/13 shots made)
gerald green    Ellington Wayne (2/12 shots made)
pablo prigioni  Ellis Monta     (0/4 shots made)
anthony bennett Baynes Aron     (3/9 shots made)
marreese speights       Adams Steven    (9/20 shots made)
michael kidd-gilchrist  Harris Tobias   (6/13 shots made)
al horford      Gortat Marcin   (6/17 shots made)
victor oladipo  Wall John       (2/12 shots made)
markieff morris Noah Joakim     (5/19 shots made)
tim duncan      Gasol Marc      (9/23 shots made)
zach randolph   Chandler Tyson  (18/30 shots made)
andre drummond  Thompson Tristan        (8/17 shots made)
kj mcdaniels    Korver Kyle     (4/9 shots made)
jimmy butler    Ellis Monta     (7/17 shots made)
arron afflalo   Wiggins Andrew  (10/23 shots made)
jason thompson  Gasol Marc      (3/9 shots made)
derrick williams        Teletovic Mirza (1/5 shots made)
jose juan barea Exum Dante      (5/12 shots made)
demarcus cousins        Bogut Andrew    (9/25 shots made)
bradley beal    Bradley Avery   (5/16 shots made)
gerald henderson        Oladipo Victor  (6/14 shots made)
jared sullinger Gasol Pau       (9/22 shots made)
```



root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/projec

```
jared sullinger Gasol Pau       (9/22 shots made)
darrell arthur  Collison Nick   (3/10 shots made)
deandre jordan  Cousins DeMarcus        (1/8 shots made)
omri casspi     Green Danny     (1/5 shots made)
timofey mozgov  Gasol Pau       (10/20 shots made)
jason smith     Zeller Cody     (5/14 shots made)
chandler parsons        Gay Rudy        (3/13 shots made)
ronnie price    Lillard Damian  (5/11 shots made)
trevor ariza    Barnes Harrison (3/13 shots made)
solomon hill    Wade Dwyane     (1/8 shots made)
joakim noah     Gortat Marcin   (4/13 shots made)
russell westbrook       Bledsoe Eric    (9/29 shots made)
cj mccollum     Burke Trey      (0/3 shots made)
tyreke evans    Matthews Wesley (3/15 shots made)
jerami grant    Patterson Patrick       (1/7 shots made)
nerles noel     Hibbert Roy     (4/14 shots made)
kemba walker    Knight Brandon  (8/25 shots made)
dirk nowtizski  Green Draymond  (4/18 shots made)
kelly olynyk    Mirotic Nikola  (1/7 shots made)
nikola vucevic  Jefferson Al    (12/33 shots made)
giannis antetokounmpo   Smith Josh      (6/19 shots made)
donatas motiejunas      Green Draymond  (1/9 shots made)
brandon jennings        Bledsoe Eric    (1/12 shots made)
ty lawson       Bledsoe Eric    (7/22 shots made)
aron baynes     Gobert Rudy     (2/6 shots made)
david west      Monroe Greg     (7/21 shots made)
kris humphries  Olynyk Kelly    (7/15 shots made)
donald sloan    Bledsoe Eric    (4/10 shots made)
gorgui dieng    Duncan Tim      (5/11 shots made)
marc gasol      Jefferson Al    (11/25 shots made)
kyrie irving    Walker Kemba    (11/28 shots made)
tyson chandler  Ibaka Serge     (3/8 shots made)
paul pierce     Copeland Chris  (2/13 shots made)
jrue holiday    Joseph Cory     (7/20 shots made)
trevor booker   Hawes Spencer   (2/7 shots made)
jason maxiell   Singler Kyle    (0/3 shots made)
greg monroe     Love Kevin      (9/24 shots made)
kostas papanikolaou     Nowitzki Dirk   (0/6 shots made)
jeff green      Johnson Joe     (11/26 shots made)
hollis thompson Meeks Jodie     (1/10 shots made)
manu ginobili   McLemore Ben    (7/15 shots made)
danny green     Harden James    (2/9 shots made)
mike miller     Covington Robert        (0/3 shots made)
joe johnson     Green Jeff      (9/28 shots made)
nik stauskas    Ellington Wayne (0/5 shots made)
thaddeus young  Faried Kenneth  (11/26 shots made)
jon ingles      Williams Lou    (1/5 shots made)
luke babbitt    Kanter Enes     (3/7 shots made)
andre iguodala  Miles CJ        (1/5 shots made)
danilo gallinai Morris Marcus   (1/6 shots made)
luis scola      Tolliver Anthony        (3/13 shots made)
quincy acy      Jerebko Jonas   (3/8 shots made)
dennis schroder Bayless Jerryd  (0/10 shots made)
dwight howard   Gobert Rudy     (2/10 shots made)
rudy gay        Barnes Matt     (6/23 shots made)
brian roberts   Napier Shabazz  (1/7 shots made)
caron butler    Matthews Wesley (1/5 shots made)
mario chalmers  Thompson Klay   (0/6 shots made)
```



root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
andre iguodala  Miles CJ        (1/5 shots made)
danilo gallinai Morris Marcus   (1/6 shots made)
luis scola      Tolliver Anthony        (3/13 shots made)
quincy acy      Jerebko Jonas   (3/8 shots made)
dennis schroder Bayless Jerryd  (0/10 shots made)
dwight howard   Gobert Rudy     (2/10 shots made)
rudy gay        Barnes Matt     (6/23 shots made)
brian roberts   Napier Shabazz  (1/7 shots made)
caron butler    Matthews Wesley (1/5 shots made)
mario chalmers  Thompson Klay   (0/6 shots made)
alex len        Gasol Marc      (1/9 shots made)
mike conley     Lin Jeremy      (10/23 shots made)
robert sacre    Gortat Marcin   (1/8 shots made)
stephen curry   Rose Derrick    (7/21 shots made)
tyler hansbrough        Olynyk Kelly    (1/4 shots made)
andrew bogut    Zeller Tyler    (0/6 shots made)
kendrick perkins        Pachulia Zaza   (0/4 shots made)
otto porter     Prigioni Pablo  (0/5 shots made)
vince carter    Muhammad Shabazz        (1/6 shots made)
jimmer dredette Ridnour Luke    (0/5 shots made)
jerryd bayless  Korver Kyle     (1/6 shots made)
richard jefferson       Korver Kyle     (2/6 shots made)
mnta ellis      Carroll DeMarre (7/17 shots made)
andrew wiggins  Afflalo Arron   (12/23 shots made)
draymond green  Morris Markieff (3/12 shots made)
james harden    McLemore Ben    (9/31 shots made)
eric bledsoe    Redick JJ       (6/20 shots made)
jon leuer       Collison Nick   (3/10 shots made)
gordon hayward  Wiggins Andrew  (3/14 shots made)
nene hilario    Noah Joakim     (3/13 shots made)
khris middleton Butler Jimmy    (4/12 shots made)
shabazz napier  Roberts Brian   (1/7 shots made)
amir johnson    Millsap Paul    (7/15 shots made)
andre roberson  Thompson Klay   (3/8 shots made)
pero antic      Lopez Brook     (0/6 shots made)
zaza pachulia   Gobert Rudy     (1/8 shots made)
tony snell      Johnson Wesley  (2/8 shots made)
shawn marion    Morris Markieff (0/4 shots made)
boris diaw      Teletovic Mirza (3/12 shots made)
jonas valanciunas       Vucevic Nikola  (4/16 shots made)
nate robinson   Blake Steve     (2/12 shots made)
lavoy allen     Pachulia Zaza   (2/6 shots made)
Deleted /project/input
Deleted /project/output
Stopping namenodes on [bigdatacomputing1.c.ringed-metric-362423.internal]
Stopping datanodes
Stopping secondary namenodes [bigdatacomputing1]
Stopping nodemanagers
10.128.0.3: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill with kill -9
10.128.0.5: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill with kill -9
Stopping resourcemanager
WARNING: Use of this script to stop the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon stop" instead.
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project# vi mapper_p2q1.py
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project# vi reducer_p2q1.py
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project# vi test_p2q1.py
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project# vi test_p2q1.sh
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project#
```

Q2 Considering the hit rate, which zone is the best for James Harden, Chris Paul, Stephen Curry, and Lebron James

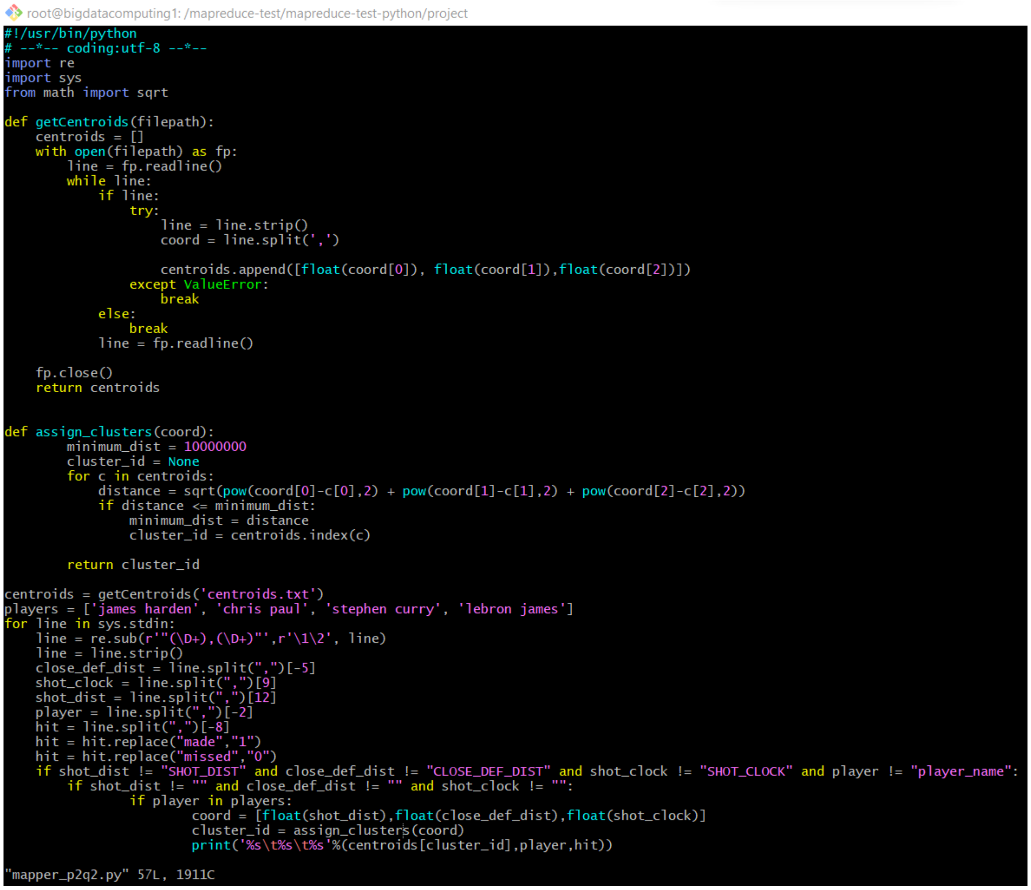
First, run centroids.txt to randomly create the first 4 centroids and save them. In order to help categorize the records, create the assign clusters function, which is based on how far apart from the four centroids each player is. By stripping out the unique characters and comma-separating the lines, we can divide all of the player's records into four zones. Filter the column names and missing values in the file.

In order to determine the distances between each data point and the centroids, create a function called assign centroids that squares the difference between each data point and the latter. Create an empty list called centroids and open the file as fp to store future values. Using the strip method, the unexpected characters are eliminated from each line. After specifying the type of cord as floating points, add them to the previously constructed empty list (centroids). Add the list of centroids once more.

Define the function assign_cluster. Set a starting value of 10000000 for min dist. We calculate the distance between each item in the centroid list and the min dist, then compare the results. If the dist is less than or equal to the min_dist, we also update the min_dist to the current dist and write the index of centroids to cluster_id. Return to the cluster_id at the end. The assign cluster function helps categorize each player's stats based on how far they are from the four centroids.

After reading the txt file to get centroids, make a list of the target players. Using sys.stdin, read lines from the file. After that, use the strip function to get rid of any further strange characters. Apply index -5,9,12,-2,-8 to obtain close_def_dist, short_clock, shot_dist, player, and hit. In the hit, made and missed are replaced by the numbers 1 and 0. Centroids[clus_id], player, and hit of those four players ('james harden,' 'chris paul,''stephen curry,' and 'lebron james') will be given to the second round mapper by the first round mapper.

Here is the mapper_p2q2.py file:

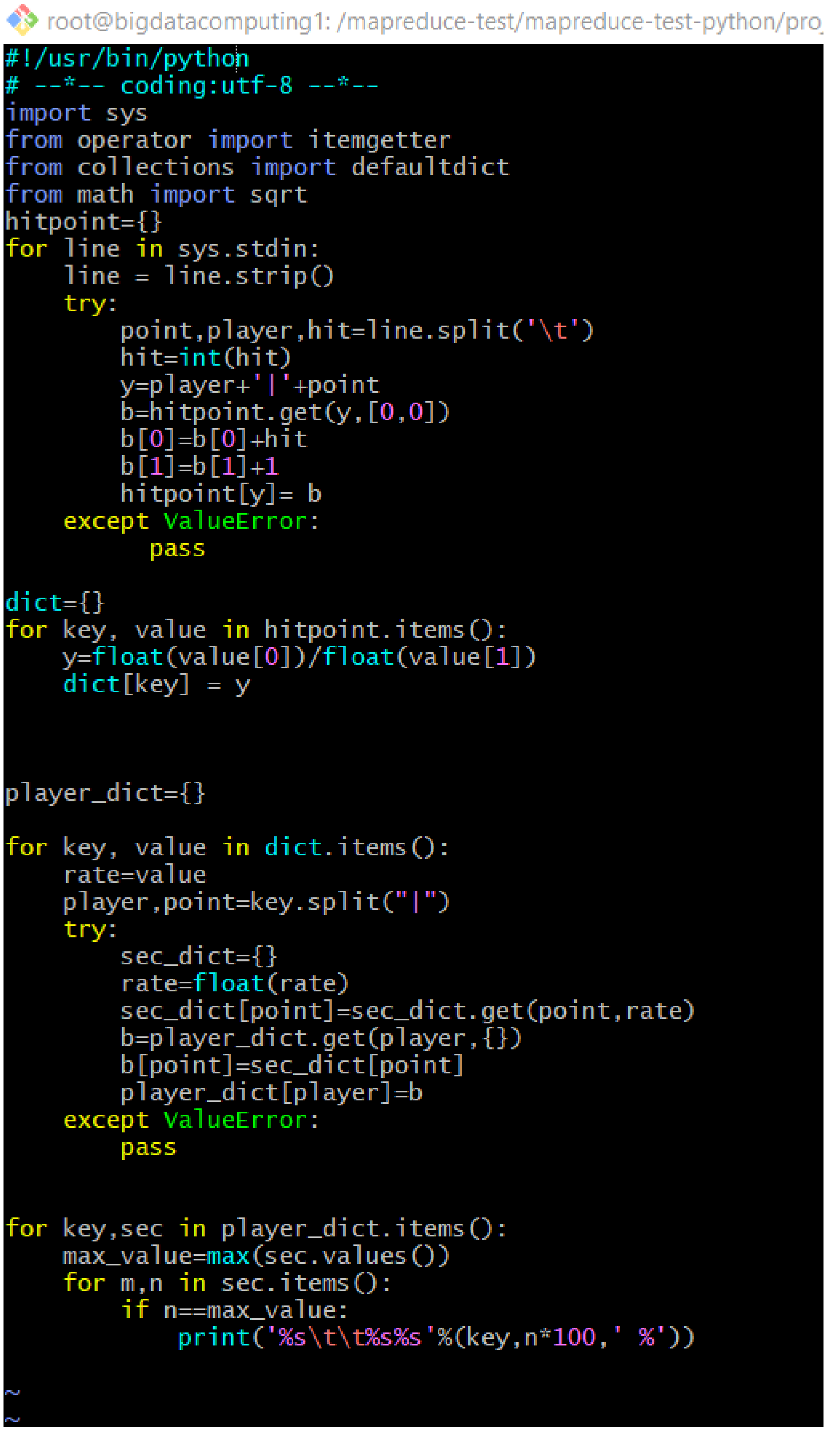


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
#!/usr/bin/python
# --*-- coding:utf-8 --*--
import re
import sys
from math import sqrt

def getCentroids(filepath):
    centroids = []
    with open(filepath) as fp:
        line = fp.readline()
        while line:
            if line:
                try:
                    line = line.strip()
                    coord = line.split(',')

                    centroids.append([float(coord[0]), float(coord[1]),float(coord[2])])
                except ValueError:
                    break
            else:
                break
            line = fp.readline()

    fp.close()
    return centroids


def assign_clusters(coord):
        minimum_dist = 10000000
        cluster_id = None
        for c in centroids:
            distance = sqrt(pow(coord[0]-c[0],2) + pow(coord[1]-c[1],2) + pow(coord[2]-c[2],2))
            if distance <= minimum_dist:
                minimum_dist = distance
                cluster_id = centroids.index(c)

        return cluster_id

centroids = getCentroids('centroids.txt')
players = ['james harden', 'chris paul', 'stephen curry', 'lebron james']
for line in sys.stdin:
    line = re.sub(r'"(\D+),(\D+)"',r'\1\2', line)
    line = line.strip()
    close_def_dist = line.split(",")[-5]
    shot_clock = line.split(",")[9]
    shot_dist = line.split(",")[12]
    player = line.split(",")[-2]
    hit = line.split(",")[-8]
    hit = hit.replace("made","1")
    hit = hit.replace("missed","0")
    if shot_dist != "SHOT_DIST" and close_def_dist != "CLOSE_DEF_DIST" and shot_clock != "SHOT_CLOCK" and player != "player_name":
        if shot_dist != "" and close_def_dist != "" and shot_clock != "":
                if player in players:
                        coord = [float(shot_dist),float(close_def_dist),float(shot_clock)]
                        cluster_id = assign_clusters(coord)
                        print('%s\t%s\t%s'%(centroids[cluster_id],player,hit))

"mapper_p2q2.py" 57L, 1911C
```

We update the centroid for this cluster for each location contained inside a certain centroid and build four new centroids by averaging the shot_dist, close_def _dist, and shot_clock for each cluster. After the tenth iteration, the first MapReduce round will receive these changed centroids. In order to calculate the hit and total shooting times, create an empty dictionary. The comfy zones and players are established as the keys, and lists of values are used as the form of values. No matter who the defender is, a[0] represents the number of hits and a[1] represents the total number of shooting attempts for this player. To get the hit rate, use a for loop to divide the number of hits per total number of shots (a[0]/a[1]). Define a nested dictionary, in which the key is the player and the value is a dictionary with the keys "comfortable zones" and "hit rate." As a result, there are 16 key-value pairs in the dictionary (4 players * 4 centroids), where the values represent the corresponding hit rates. We identify the optimal zones for Lebron James, James Harden, Chris Paul, and Stephen Curry based on their highest hit rates by comparing the hit rate which is values in the dictionary.

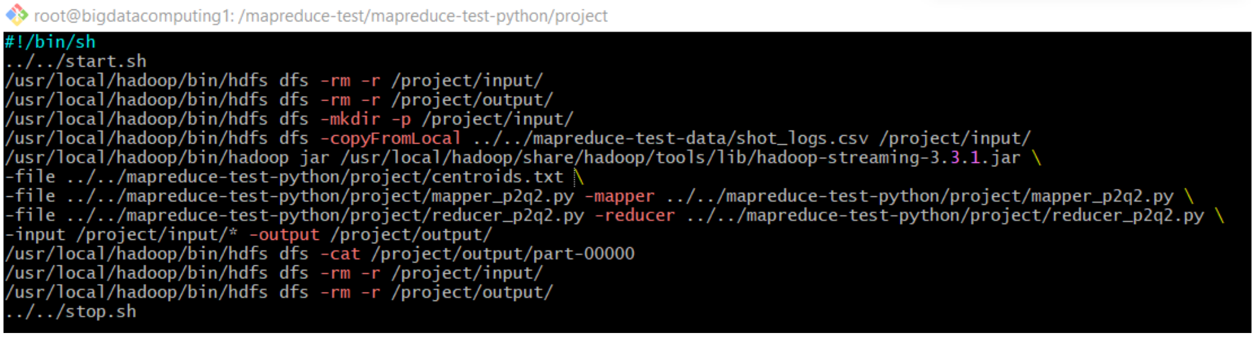
Here is the reducer_p2q2.py file:


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/pro

```
#!/usr/bin/python
# --*-- coding:utf-8 --*--
import sys
from operator import itemgetter
from collections import defaultdict
from math import sqrt
hitpoint={}
for line in sys.stdin:
    line = line.strip()
    try:
        point,player,hit=line.split('\t')
        hit=int(hit)
        y=player+'|'+point
        b=hitpoint.get(y,[0,0])
        b[0]=b[0]+hit
        b[1]=b[1]+1
        hitpoint[y]= b
    except ValueError:
          pass

dict={}
for key, value in hitpoint.items():
    y=float(value[0])/float(value[1])
    dict[key] = y



player_dict={}

for key, value in dict.items():
    rate=value
    player,point=key.split("|")
    try:
        sec_dict={}
        rate=float(rate)
        sec_dict[point]=sec_dict.get(point,rate)
        b=player_dict.get(player,{})
        b[point]=sec_dict[point]
        player_dict[player]=b
    except ValueError:
        pass


for key,sec in player_dict.items():
    max_value=max(sec.values())
    for m,n in sec.items():
        if n==max_value:
            print('%s\t\t%s%s'%(key,n*100,' %'))

~
~
```

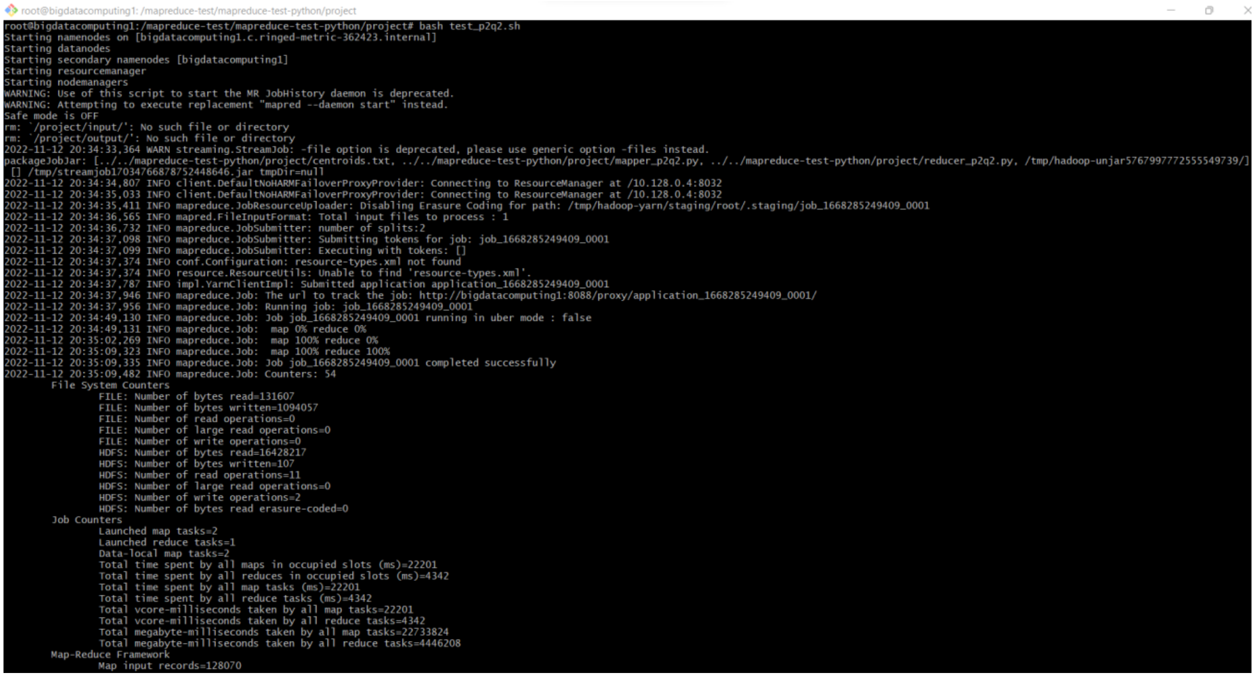
Start hdfs, yarn, and historyserver with start.sh, then exit safe mode. Delete the input and output directories to guarantee that no input or output remains to muck up the input data. Make a directory for the project so that it may copy the dataset from the local directory. Add the centroids.txt file to read. Begin a Hadoop streaming job to perform a map and reduce operation. Save output and read it with the cat command. Stop hdfs, yarn, and historyserver by deleting the input and output directories and calling stop.sh.

Here is the test_p2q2.sh file:

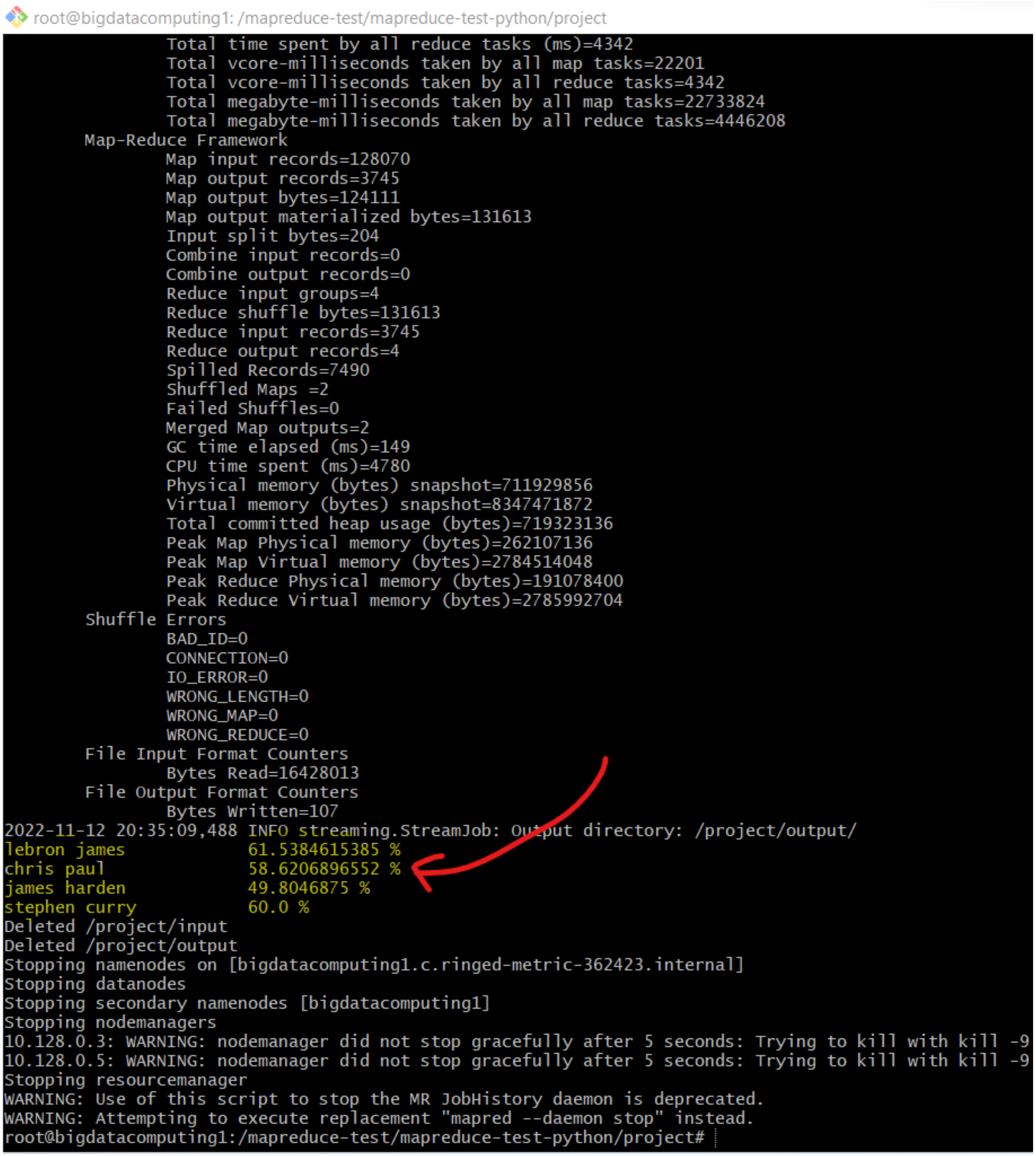

root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
#!/bin/sh
../../start.sh
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/input/
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/output/
/usr/local/hadoop/bin/hdfs dfs -mkdir -p /project/input/
/usr/local/hadoop/bin/hdfs dfs -copyFromLocal ../../mapreduce-test-data/shot_logs.csv /project/input/
/usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.1.jar \
-file ../../mapreduce-test-python/project/centroids.txt \
-file ../../mapreduce-test-python/project/mapper_p2q2.py -mapper ../../mapreduce-test-python/project/mapper_p2q2.py \
-file ../../mapreduce-test-python/project/reducer_p2q2.py -reducer ../../mapreduce-test-python/project/reducer_p2q2.py \
-input /project/input/* -output /project/output/
/usr/local/hadoop/bin/hdfs dfs -cat /project/output/part-00000
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/input/
/usr/local/hadoop/bin/hdfs dfs -rm -r /project/output/
../../stop.sh
```

Output:


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project# bash test_p2q2.sh
Starting namenodes on [bigdatacomputing1.c.ringed-metric-362423.internal]
Starting datanodes
Starting secondary namenodes [bigdatacomputing1]
Starting resourcemanager
Starting nodemanagers
WARNING: Use of this script to start the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon start" instead.
Safe mode is OFF
rm: `/project/input/': No such file or directory
rm: `/project/output/': No such file or directory
2022-11-12 20:34:33,364 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [../../mapreduce-test-python/project/centroids.txt, ../../mapreduce-test-python/project/mapper_p2q2.py, ../../mapreduce-test-python/project/reducer_p2q2.py, /tmp/hadoop-unjar5767997772555549739/]
 [] /tmp/streamjob17034766878752448646.jar tmpDir=null
2022-11-12 20:34:34,807 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /10.128.0.4:8032
2022-11-12 20:34:35,033 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /10.128.0.4:8032
2022-11-12 20:34:35,411 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1668285249409_0001
2022-11-12 20:34:36,565 INFO mapred.FileInputFormat: Total input files to process : 1
2022-11-12 20:34:36,732 INFO mapreduce.JobSubmitter: number of splits:2
2022-11-12 20:34:37,098 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1668285249409_0001
2022-11-12 20:34:37,099 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-11-12 20:34:37,374 INFO conf.Configuration: resource-types.xml not found
2022-11-12 20:34:37,374 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-11-12 20:34:37,787 INFO impl.YarnClientImpl: Submitted application application_1668285249409_0001
2022-11-12 20:34:37,946 INFO mapreduce.Job: The url to track the job: http://bigdatacomputing1:8088/proxy/application_1668285249409_0001/
2022-11-12 20:34:37,956 INFO mapreduce.Job: Running job: job_1668285249409_0001
2022-11-12 20:34:49,130 INFO mapreduce.Job: Job job_1668285249409_0001 running in uber mode : false
2022-11-12 20:34:49,131 INFO mapreduce.Job:  map 0% reduce 0%
2022-11-12 20:35:02,269 INFO mapreduce.Job:  map 100% reduce 0%
2022-11-12 20:35:09,323 INFO mapreduce.Job:  map 100% reduce 100%
2022-11-12 20:35:09,335 INFO mapreduce.Job: Job job_1668285249409_0001 completed successfully
2022-11-12 20:35:09,482 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=131607
                FILE: Number of bytes written=1094057
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=16428217
                HDFS: Number of bytes written=107
                HDFS: Number of read operations=11
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=22201
                Total time spent by all reduces in occupied slots (ms)=4342
                Total time spent by all map tasks (ms)=22201
                Total time spent by all reduce tasks (ms)=4342
                Total vcore-milliseconds taken by all map tasks=22201
                Total vcore-milliseconds taken by all reduce tasks=4342
                Total megabyte-milliseconds taken by all map tasks=22733824
                Total megabyte-milliseconds taken by all reduce tasks=4446208
        Map-Reduce Framework
                Map input records=128070
```


root@bigdatacomputing1: /mapreduce-test/mapreduce-test-python/project

```
                Total time spent by all reduce tasks (ms)=4342
                Total vcore-milliseconds taken by all map tasks=22201
                Total vcore-milliseconds taken by all reduce tasks=4342
                Total megabyte-milliseconds taken by all map tasks=22733824
                Total megabyte-milliseconds taken by all reduce tasks=4446208
        Map-Reduce Framework
                Map input records=128070
                Map output records=3745
                Map output bytes=124111
                Map output materialized bytes=131613
                Input split bytes=204
                Combine input records=0
                Combine output records=0
                Reduce input groups=4
                Reduce shuffle bytes=131613
                Reduce input records=3745
                Reduce output records=4
                Spilled Records=7490
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=149
                CPU time spent (ms)=4780
                Physical memory (bytes) snapshot=711929856
                Virtual memory (bytes) snapshot=8347471872
                Total committed heap usage (bytes)=719323136
                Peak Map Physical memory (bytes)=262107136
                Peak Map Virtual memory (bytes)=2784514048
                Peak Reduce Physical memory (bytes)=191078400
                Peak Reduce Virtual memory (bytes)=2785992704
        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=16428013
        File Output Format Counters
                Bytes Written=107
2022-11-12 20:35:09,488 INFO streaming.StreamJob: Output directory: /project/output/
lebron james            61.5384615385 %
chris paul              58.6206896552 %
james harden            49.8046875 %
stephen curry           60.0 %
Deleted /project/input
Deleted /project/output
Stopping namenodes on [bigdatacomputing1.c.ringed-metric-362423.internal]
Stopping datanodes
Stopping secondary namenodes [bigdatacomputing1]
Stopping nodemanagers
10.128.0.3: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill with kill -9
10.128.0.5: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill with kill -9
Stopping resourcemanager
WARNING: Use of this script to stop the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon stop" instead.
root@bigdatacomputing1:/mapreduce-test/mapreduce-test-python/project#
```