Course: CISC5950 Spring 2022

Professor: Ying Mao

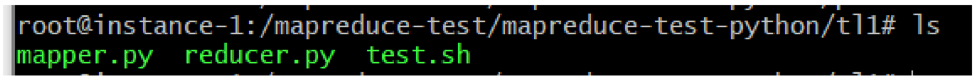
Student: Yuliya Akchurina

Lab1

Part 1.

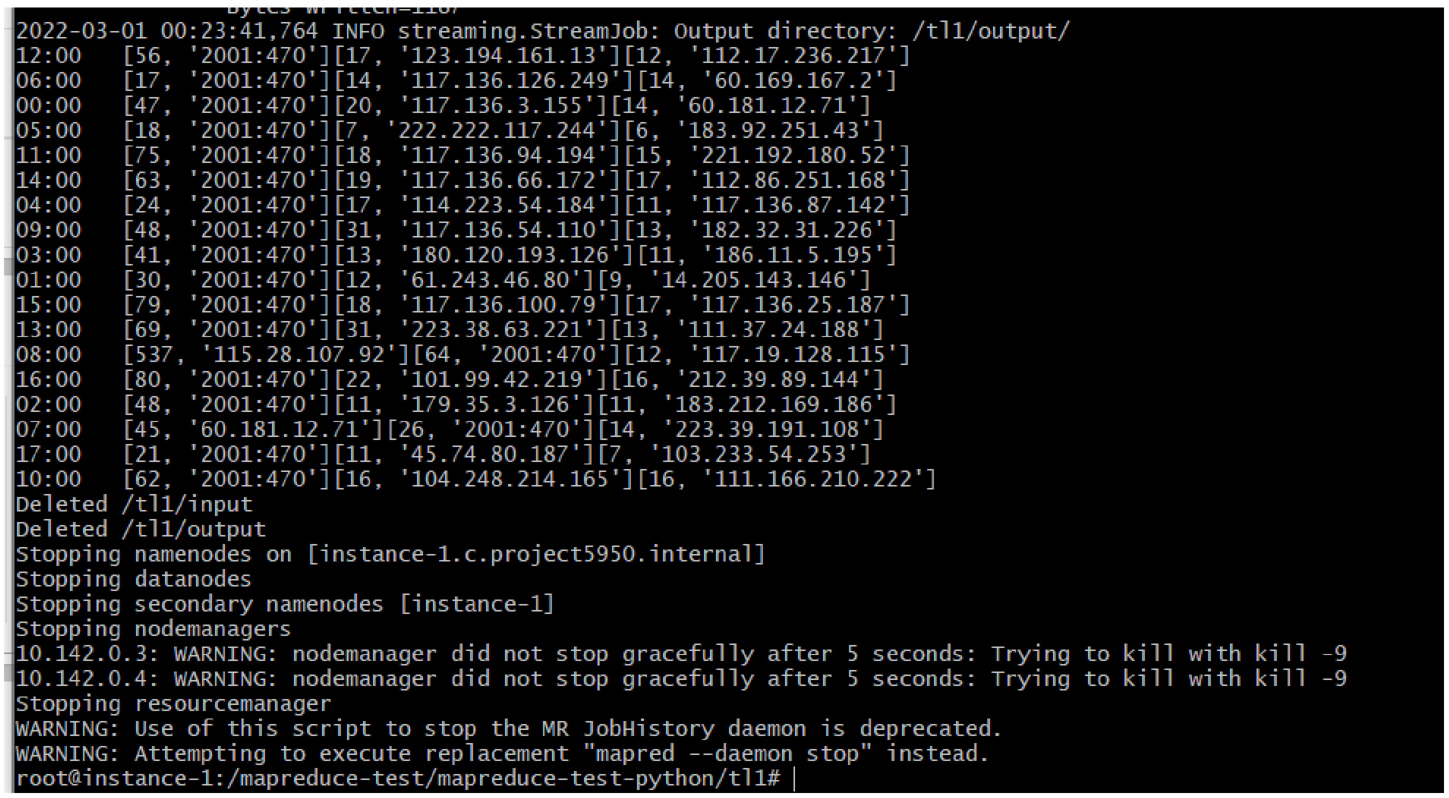
The reducer.py part of the code has been modified to separate the Hour from the IP address. Group by IP address and sort the values of the counts in a dictionary in descending order. After that top 3 hour are selected.

The mapper.py, reducer.py and test.sh files are located in the /mapreduce-test/mapreduce-test-python/tl1/ folder.



```
root@instance-1:/mapreduce-test/mapreduce-test-python/tl1# ls
mapper.py  reducer.py  test.sh
```

Running the code through Bash command line on Google cloud instance1 result in the following output.



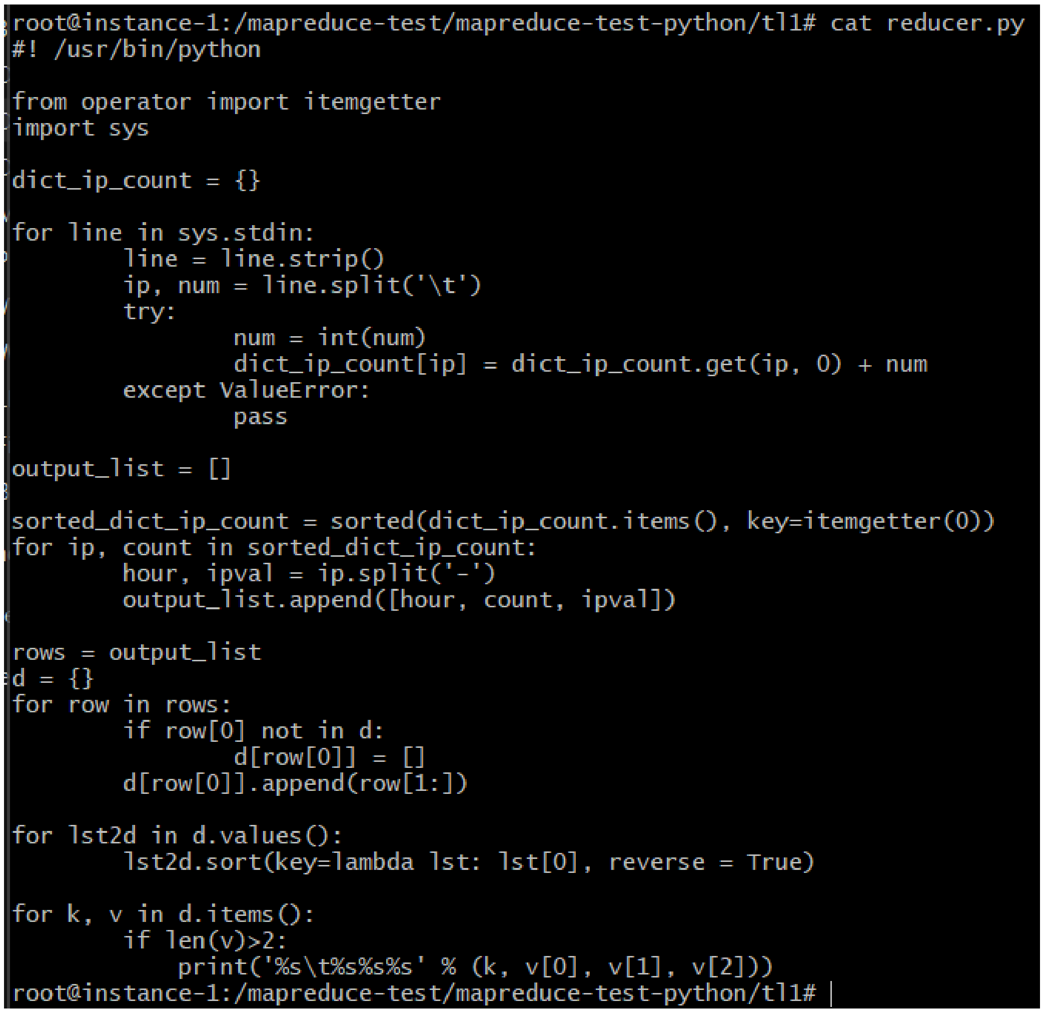
```
2022-03-01 00:23:41,764 INFO streaming.StreamJob: Output directory: /tl1/output/
12:00   [56, '2001:470'][17, '123.194.161.13'][12, '112.17.236.217']
06:00   [17, '2001:470'][14, '117.136.126.249'][14, '60.169.167.2']
00:00   [47, '2001:470'][20, '117.136.3.155'][14, '60.181.12.71']
05:00   [18, '2001:470'][7, '222.222.117.244'][6, '183.92.251.43']
11:00   [75, '2001:470'][18, '117.136.94.194'][15, '221.192.180.52']
14:00   [63, '2001:470'][19, '117.136.66.172'][17, '112.86.251.168']
04:00   [24, '2001:470'][17, '114.223.54.184'][11, '117.136.87.142']
09:00   [48, '2001:470'][31, '117.136.54.110'][13, '182.32.31.226']
03:00   [41, '2001:470'][13, '180.120.193.126'][11, '186.11.5.195']
01:00   [30, '2001:470'][12, '61.243.46.80'][9, '14.205.143.146']
15:00   [79, '2001:470'][18, '117.136.100.79'][17, '117.136.25.187']
13:00   [69, '2001:470'][31, '223.38.63.221'][13, '111.37.24.188']
08:00   [537, '115.28.107.92'][64, '2001:470'][12, '117.19.128.115']
16:00   [80, '2001:470'][22, '101.99.42.219'][16, '212.39.89.144']
02:00   [48, '2001:470'][11, '179.35.3.126'][11, '183.212.169.186']
07:00   [45, '60.181.12.71'][26, '2001:470'][14, '223.39.191.108']
17:00   [21, '2001:470'][11, '45.74.80.187'][7, '103.233.54.253']
10:00   [62, '2001:470'][16, '104.248.214.165'][16, '111.166.210.222']
Deleted /tl1/input
Deleted /tl1/output
Stopping namenodes on [instance-1.c.project5950.internal]
Stopping datanodes
Stopping secondary namenodes [instance-1]
Stopping nodemanagers
10.142.0.3: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill with kill -9
10.142.0.4: 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@instance-1:/mapreduce-test/mapreduce-test-python/tl1# |
```

Running the code on my laptop results in this output.

```
00:00   [47, '2001:470'][20, '117.136.3.155'][14, '60.181.12.71']
01:00   [30, '2001:470'][12, '61.243.46.80'][9, '14.205.143.146']
02:00   [48, '2001:470'][11, '179.35.3.126'][11, '183.212.169.186']
03:00   [41, '2001:470'][13, '180.120.193.126'][11, '186.11.5.195']
04:00   [24, '2001:470'][17, '114.223.54.184'][11, '117.136.87.142']
05:00   [18, '2001:470'][7, '222.222.117.244'][6, '183.92.251.43']
06:00   [17, '2001:470'][14, '117.136.126.249'][14, '60.169.167.2']
07:00   [45, '60.181.12.71'][26, '2001:470'][14, '223.39.191.108']
08:00   [537, '115.28.107.92'][64, '2001:470'][12, '117.19.128.115']
09:00   [48, '2001:470'][31, '117.136.54.110'][13, '182.32.31.226']
10:00   [62, '2001:470'][16, '104.248.214.165'][16, '111.166.210.222']
11:00   [75, '2001:470'][18, '117.136.94.194'][15, '221.192.180.52']
12:00   [56, '2001:470'][17, '123.194.161.13'][12, '112.17.236.217']
13:00   [69, '2001:470'][31, '223.38.63.221'][13, '111.37.24.188']
14:00   [63, '2001:470'][19, '117.136.66.172'][17, '112.86.251.168']
15:00   [79, '2001:470'][18, '117.136.100.79'][17, '117.136.25.187']
16:00   [80, '2001:470'][22, '101.99.42.219'][16, '212.39.89.144']
17:00   [21, '2001:470'][11, '45.74.80.187'][7, '103.233.54.253']
```

The code is the same but on the laptop the hours appear sorted, while on Google cloud they appear not sorted. Otherwise, the resulting values are the same.

The base code used is taken from logstat2. The primary changes to solve part 1 were made to the reducer.py. See the code below.



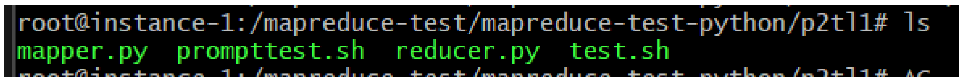
```
root@instance-1:/mapreduce-test/mapreduce-test-python/tl1# cat reducer.py
#! /usr/bin/python

from operator import itemgetter
import sys

dict_ip_count = {}

for line in sys.stdin:
        line = line.strip()
        ip, num = line.split('\t')
        try:
                num = int(num)
                dict_ip_count[ip] = dict_ip_count.get(ip, 0) + num
        except ValueError:
                pass

output_list = []

sorted_dict_ip_count = sorted(dict_ip_count.items(), key=itemgetter(0))
for ip, count in sorted_dict_ip_count:
        hour, ipval = ip.split('-')
        output_list.append([hour, count, ipval])

rows = output_list
d = {}
for row in rows:
        if row[0] not in d:
                d[row[0]] = []
        d[row[0]].append(row[1:])

for lst2d in d.values():
        lst2d.sort(key=lambda lst: lst[0], reverse = True)

for k, v in d.items():
        if len(v)>2:
            print('%s\t%s%s%s' % (k, v[0], v[1], v[2]))
root@instance-1:/mapreduce-test/mapreduce-test-python/tl1# |
```

1. Count the number of hits for each IP address per hour.
2. Split the hour and IP address into two values and save in the lust with changes order of hour, count, IP.
3. Cast the list into a dictionary where key is hour and values are count and IP. Group the dictionary values by the hour creating the nested list of values for each hour.
4. Reverse sort the dictionary by the count of hits for each IP address, for each hour.
5. For each hour output the IP address with the highest number of hits.

<u>Part 2.</u>

For the part 2 solution mapper.py is the same as in part1, the reducer.py and test.sh are updated.

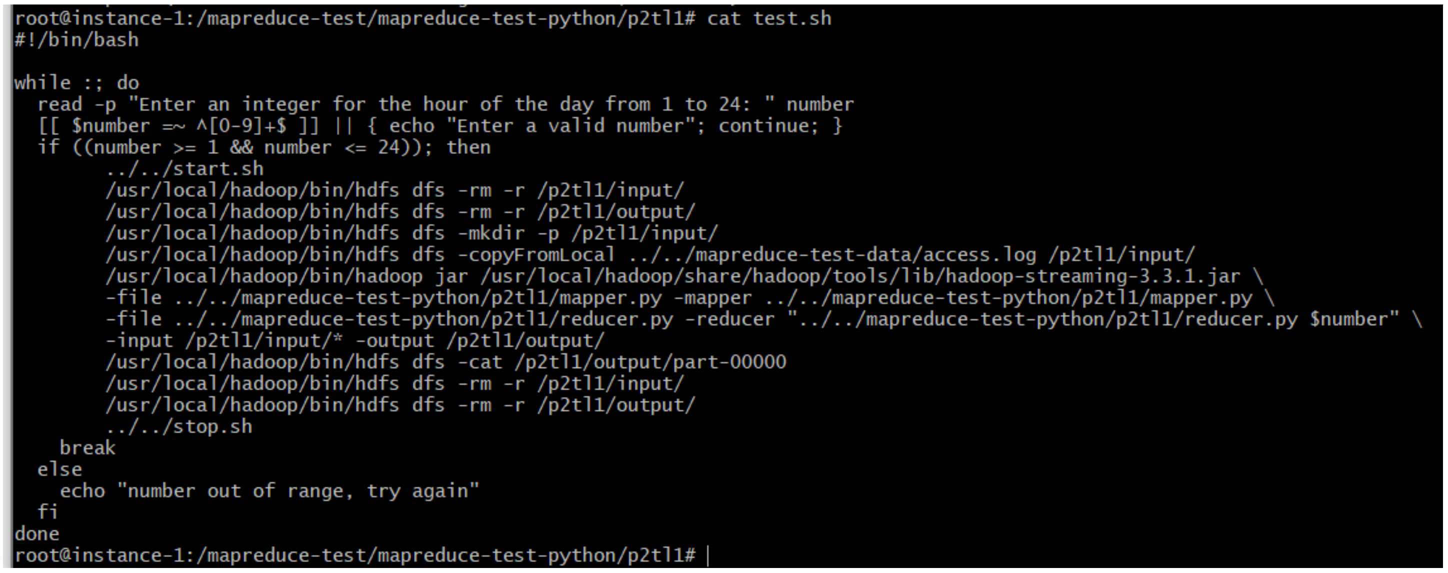
All 3 files are located on Google cloud instance in /mapreduce-test/mapreduce-test-python/p2tl1/ folder.



```
root@instance-1:/mapreduce-test/mapreduce-test-python/p2tl1# ls
mapper.py  prompttest.sh  reducer.py  test.sh
```

Prompttest.sh is a sample script used for testing, please disregard it.

The test.sh was updated from part 1 to include the user prompt. It also verified that the input is a number between 1 and 24 and passes the value as the input to the reducer.py. See the code below.

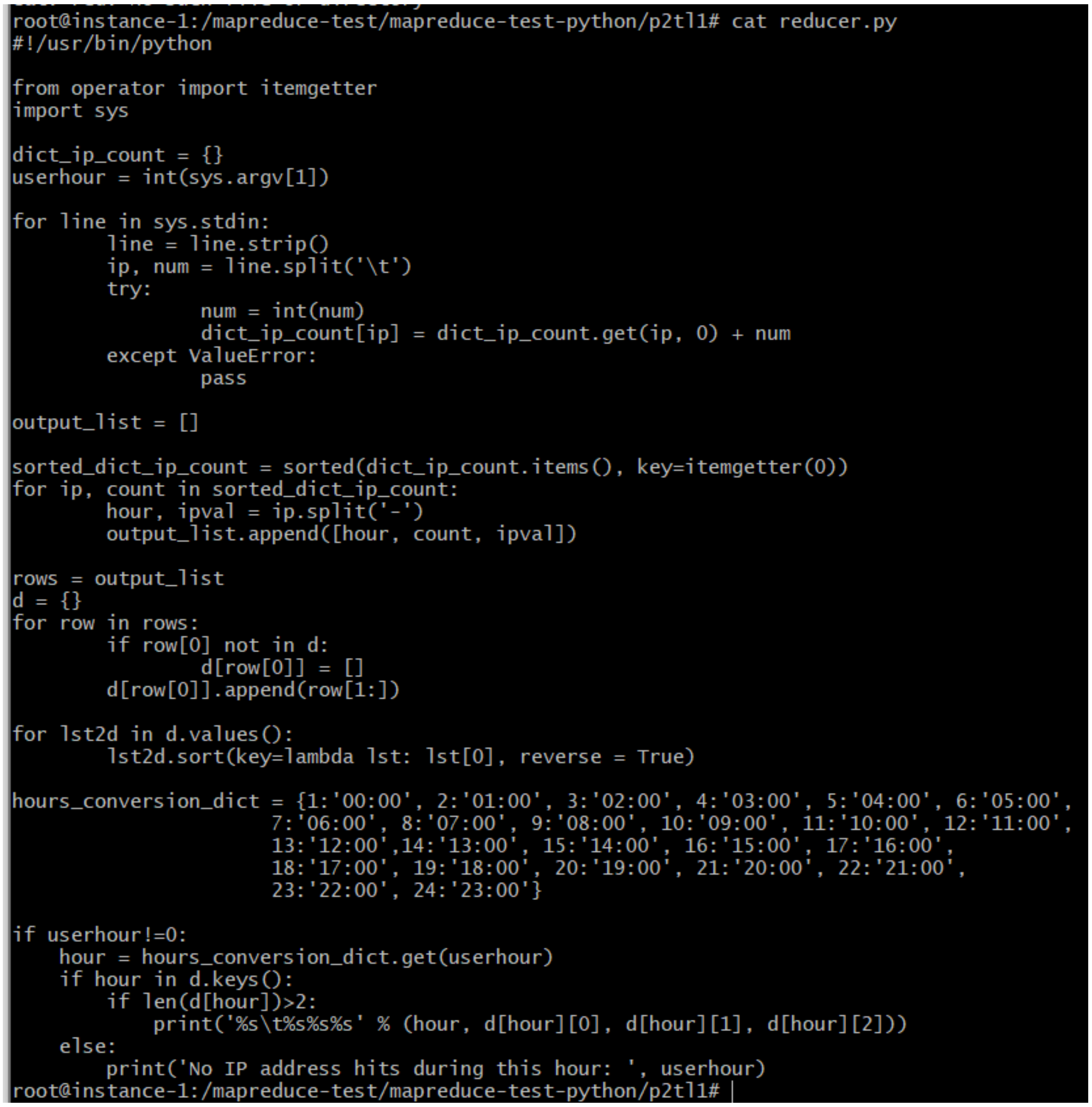


```
root@instance-1:/mapreduce-test/mapreduce-test-python/p2tl1# cat test.sh
#!/bin/bash

while :; do
  read -p "Enter an integer for the hour of the day from 1 to 24: " number
  [[ $number =~ ^[0-9]+$ ]] || { echo "Enter a valid number"; continue; }
  if ((number >= 1 && number <= 24)); then
        ../../start.sh
        /usr/local/hadoop/bin/hdfs dfs -rm -r /p2tl1/input/
        /usr/local/hadoop/bin/hdfs dfs -rm -r /p2tl1/output/
        /usr/local/hadoop/bin/hdfs dfs -mkdir -p /p2tl1/input/
        /usr/local/hadoop/bin/hdfs dfs -copyFromLocal ../../mapreduce-test-data/access.log /p2tl1/input/
        /usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.1.jar \
        -file ../../mapreduce-test-python/p2tl1/mapper.py -mapper ../../mapreduce-test-python/p2tl1/mapper.py \
        -file ../../mapreduce-test-python/p2tl1/reducer.py -reducer "../../mapreduce-test-python/p2tl1/reducer.py $number" \
        -input /p2tl1/input/* -output /p2tl1/output/
        /usr/local/hadoop/bin/hdfs dfs -cat /p2tl1/output/part-00000
        /usr/local/hadoop/bin/hdfs dfs -rm -r /p2tl1/input/
        /usr/local/hadoop/bin/hdfs dfs -rm -r /p2tl1/output/
        ../../stop.sh
    break
  else
    echo "number out of range, try again"
  fi
done
root@instance-1:/mapreduce-test/mapreduce-test-python/p2tl1# |
```

The reducer was updated from part 1. See the code below.

The user input is passed as a number from 1 to 24. The input is converted to match “HH:00” format using dictionary.

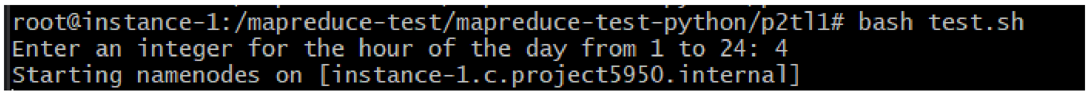
The result is printing of the 3 IP addresses with highest hits for the requested hour. If the hour is not available in the data, the message “No IP address hits during this hour” will be printed instead.



```
root@instance-1:/mapreduce-test/mapreduce-test-python/p2tl1# cat reducer.py
#!/usr/bin/python

from operator import itemgetter
import sys

dict_ip_count = {}
userhour = int(sys.argv[1])

for line in sys.stdin:
        line = line.strip()
        ip, num = line.split('\t')
        try:
                num = int(num)
                dict_ip_count[ip] = dict_ip_count.get(ip, 0) + num
        except ValueError:
                pass

output_list = []

sorted_dict_ip_count = sorted(dict_ip_count.items(), key=itemgetter(0))
for ip, count in sorted_dict_ip_count:
        hour, ipval = ip.split('-')
        output_list.append([hour, count, ipval])

rows = output_list
d = {}
for row in rows:
        if row[0] not in d:
                d[row[0]] = []
        d[row[0]].append(row[1:])

for lst2d in d.values():
        lst2d.sort(key=lambda lst: lst[0], reverse = True)

hours_conversion_dict = {1:'00:00', 2:'01:00', 3:'02:00', 4:'03:00', 5:'04:00', 6:'05:00',
                      7:'06:00', 8:'07:00', 9:'08:00', 10:'09:00', 11:'10:00', 12:'11:00',
                      13:'12:00',14:'13:00', 15:'14:00', 16:'15:00', 17:'16:00',
                      18:'17:00', 19:'18:00', 20:'19:00', 21:'20:00', 22:'21:00',
                      23:'22:00', 24:'23:00'}

if userhour!=0:
    hour = hours_conversion_dict.get(userhour)
    if hour in d.keys():
        if len(d[hour])>2:
            print('%s\t%s%s%s' % (hour, d[hour][0], d[hour][1], d[hour][2]))
    else:
        print('No IP address hits during this hour: ', userhour)
root@instance-1:/mapreduce-test/mapreduce-test-python/p2tl1# |
```

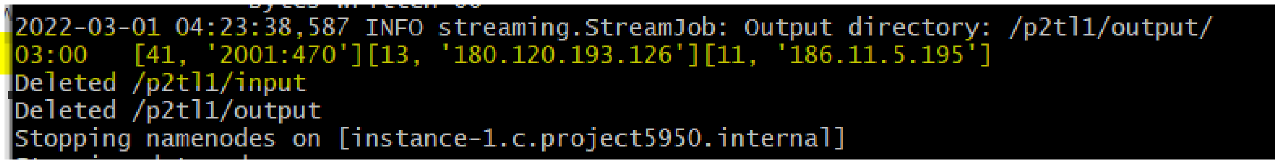
The outcome of Part 2 code run on google cloud instance:

User input:



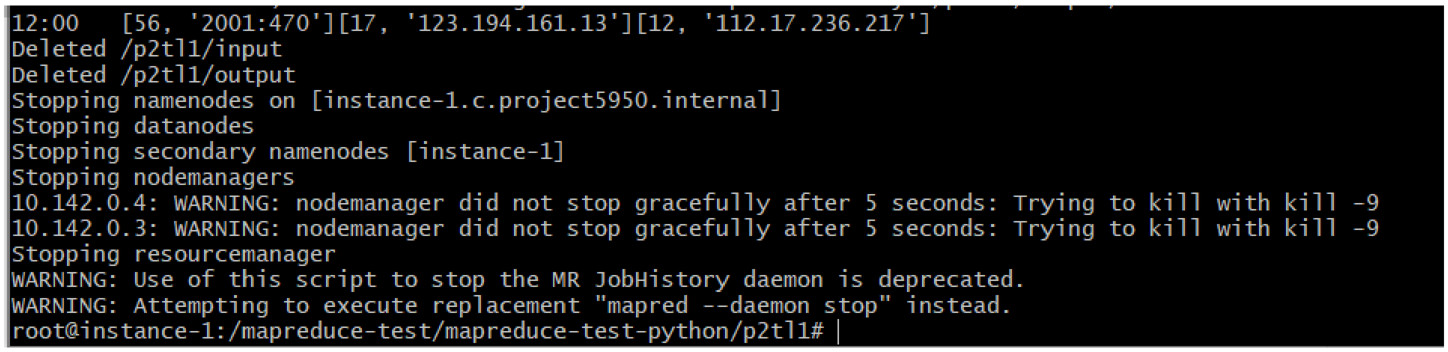
```
root@instance-1:/mapreduce-test/mapreduce-test-python/p2tl1# bash test.sh
Enter an integer for the hour of the day from 1 to 24: 4
Starting namenodes on [instance-1.c.project5950.internal]
```

The output line shows user input hour and top 3 IP addresses with the corresponding counts for that hour.



```
2022-03-01 04:23:38,587 INFO streaming.StreamJob: Output directory: /p2tl1/output/
03:00   [41, '2001:470'][13, '180.120.193.126'][11, '186.11.5.195']
Deleted /p2tl1/input
Deleted /p2tl1/output
Stopping namenodes on [instance-1.c.project5950.internal]
```

Another run:



```
12:00   [56, '2001:470'][17, '123.194.161.13'][12, '112.17.236.217']
Deleted /p2tl1/input
Deleted /p2tl1/output
Stopping namenodes on [instance-1.c.project5950.internal]
Stopping datanodes
Stopping secondary namenodes [instance-1]
Stopping nodemanagers
10.142.0.4: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill with kill -9
10.142.0.3: 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@instance-1:/mapreduce-test/mapreduce-test-python/p2tl1# 
```

Note: The regular expression equation provided in logstat2 produces inaccurate results when the values that are not IP addresses are counted as IP addresses. Some numbers that are presented as the IP addresses are incorrect data. This part needs further refinement.