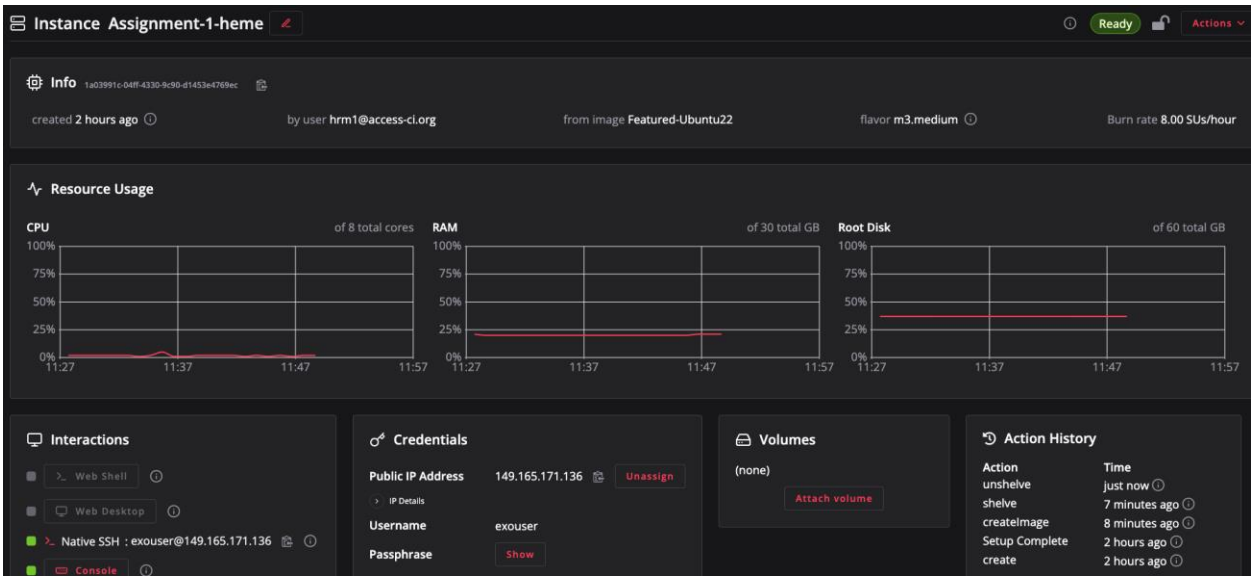
ENGR 516 Engineering cloud computing
Assignment 1

Hemesh Raaja Malathi heraaj@iu.edu)

First step is to create an Instance in Jetstream, after logging into ACCESS, with more than 20GB for storage, to able to perform the given task, so chose m3.medium.



I set up a Hadoop user account and updated the core-site.xml, mapred-site.xml, and yarn-site.xml configurations to guarantee the smooth startup of all Hadoop services without issues.

Log stat(Execution):
In the input directory, sample.log which was shared in the assignment was placed into the Hadoop Distributed File System. In mapper phase, the log file is used to get the IP information through the regex. In the reduced stage, a dictionary is used to come track of IP address, and the counts.

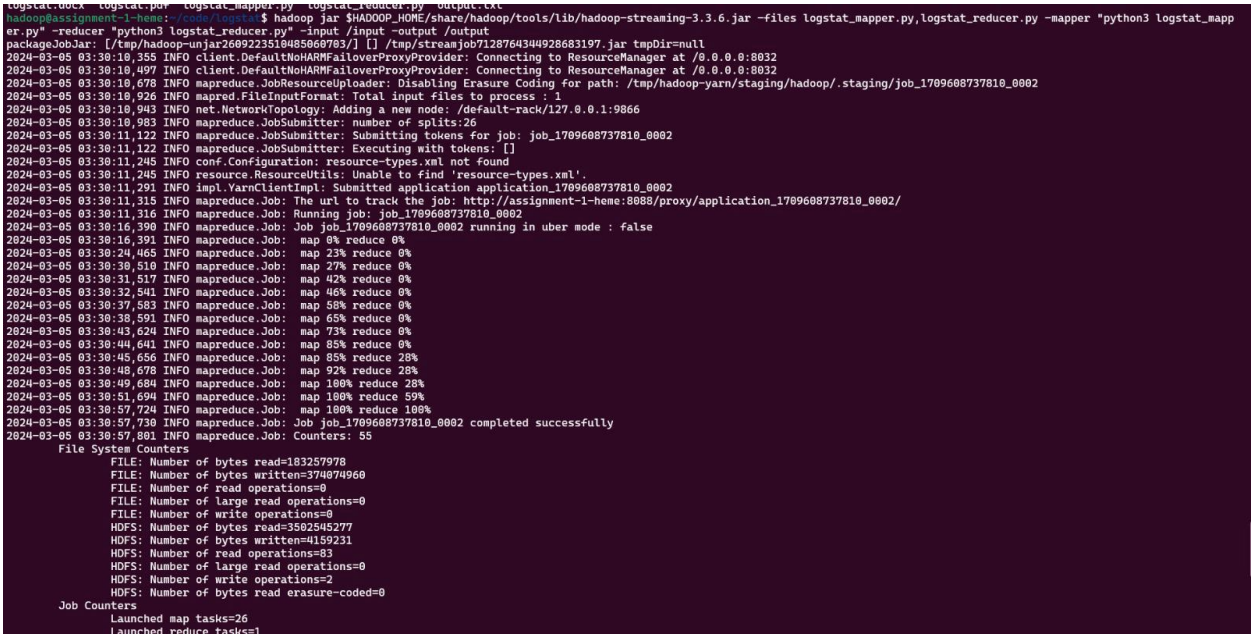
Cmd:
hdfs dfs -put sample.log /input

```
hadoop@assignment-1-heme:~/data$ hdfs dfs -put access.log  /input
hadoop@assignment-1-heme:~/data$
```

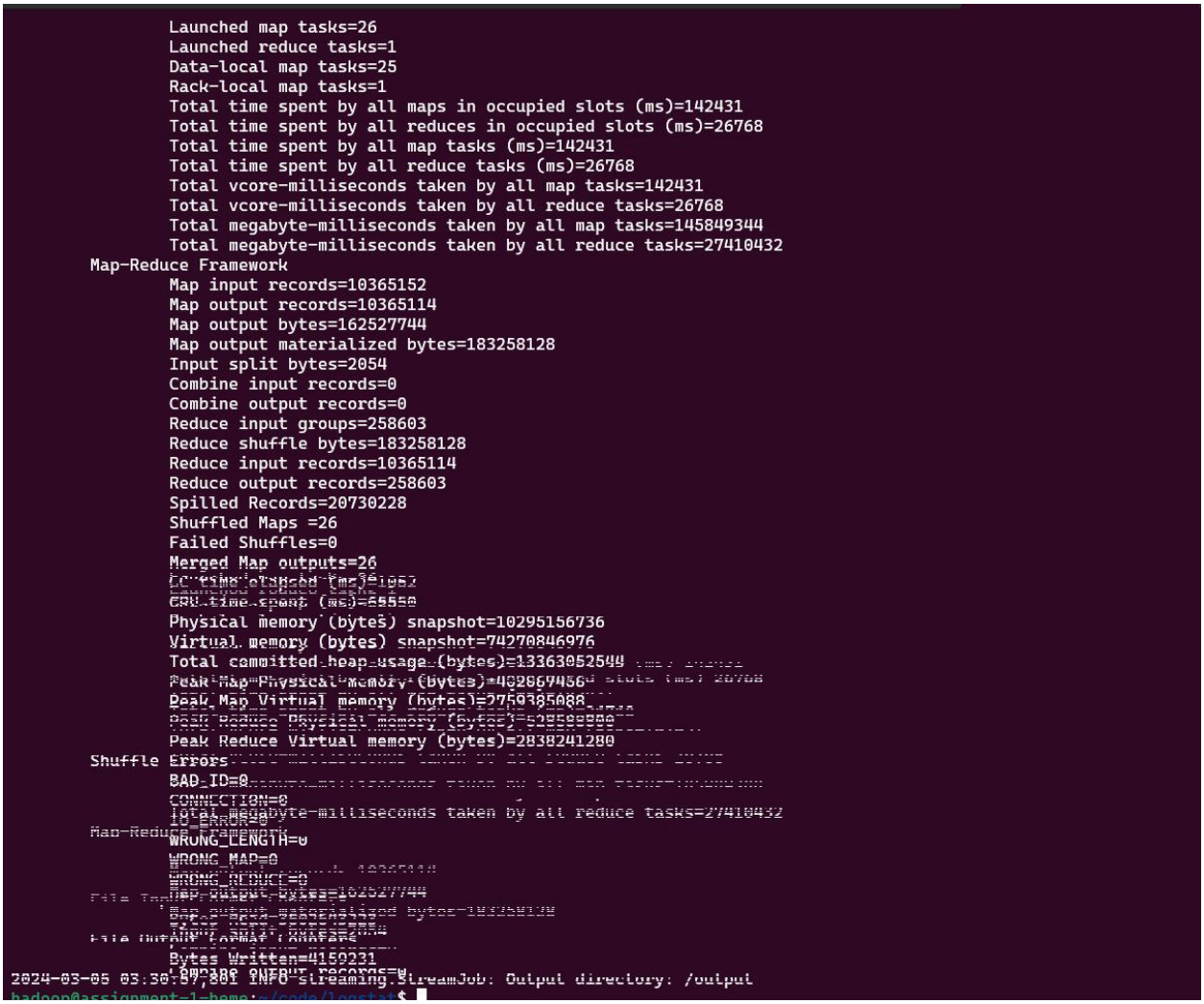
Then log stat mapper was executed followed reduced python scripts which gives the number of ip address as output.

Cmd:

hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.6.jar -files logstat_mapper.py,logstat_reducer.py -mapper "python3 logstat_mapper.py" -reducer "python3 logstat_reducer.py" -input /input -output /output



Screenshot of executing log stat map reduce



Running the map reduced on log stat

Output of the logstat:

```
96.126.104.16   12
96.126.104.226  3
96.126.105.139  86
96.126.113.125  1
96.126.115.151  58
96.126.116.214  10
96.31.67.12     1
96.41.104.3     1
96.44.144.98    5
96.66.15.147    27
96.70.31.155    9
96.9.142.138    39
97.107.132.87   157
97.107.137.22   87
97.107.138.62   1
97.107.141.106  1
97.107.209.4    24
97.113.24.90    108
98.1.80.42      1
98.176.113.4    4
98.200.11.185   1
98.206.114.40   3
98.207.129.108  30
98.207.84.103   2
98.23.40.35     2
98.248.3.114    15
99.100.6.45     2
99.100.76.33    13
99.171.130.25   175
99.188.25.107   1
99.203.23.117   1
99.227.140.55   2
99.227.204.206  2
99.228.154.237  1
99.228.156.167  14
99.228.174.11   1
99.229.160.69   39
99.229.17.167   1
99.229.20.212   1
99.229.40.159   1
99.229.54.10    67
99.237.214.84   2
99.240.108.108  1
99.243.47.93    124
99.246.134.169  273
99.246.164.168  56
99.246.247.185  1
99.253.184.236  16
99.99.188.195   14
hadoop@assignment-1-heme:~/code/logstat$
```

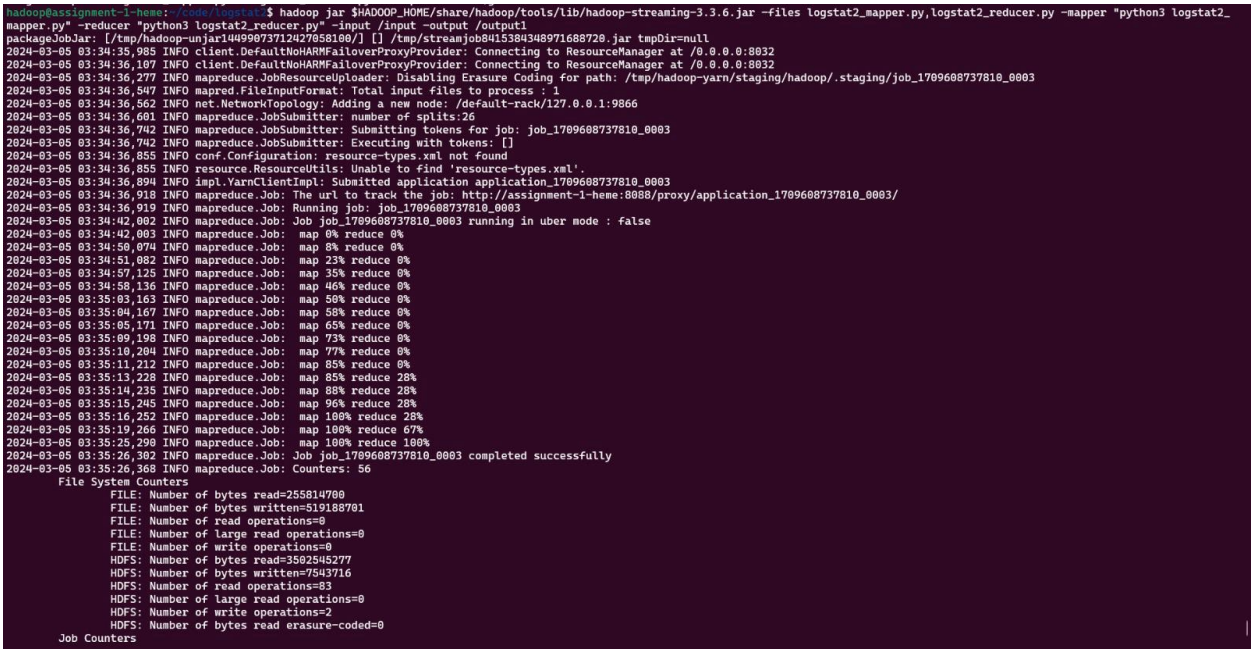
log stat map reduce op

Execution of logstat2 code:

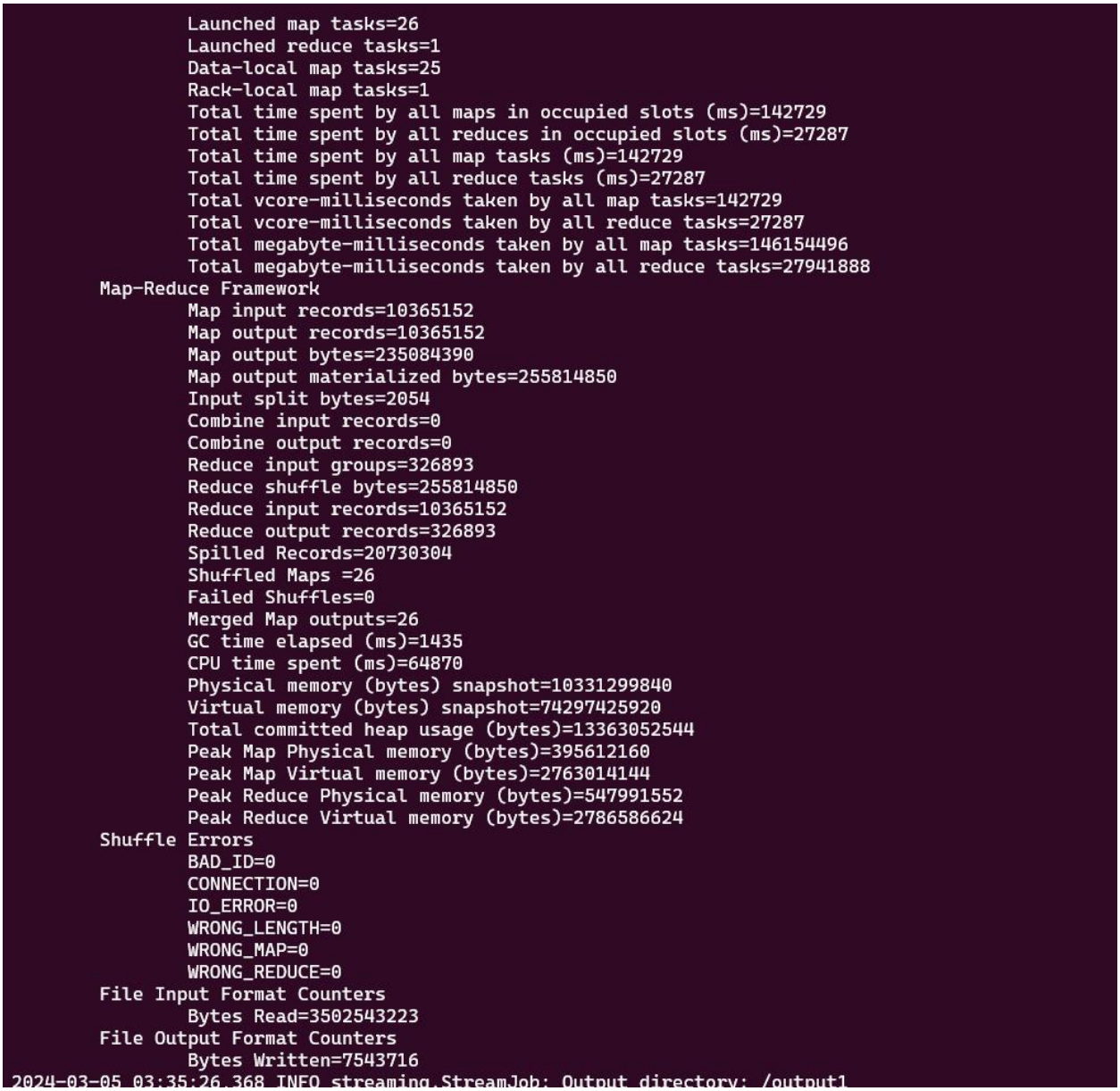
Now the log stat2 mapper and reducer python code are executed. This time the output has count of Ip address in the same hour. Now in the former phase(mapper), the log file is scanned and read, and related Ip's are extracted along with the hour information using regex. Then followed by the reducer, the extracted IP address are counted for the same hour is calculated and stored in the dictionary.

Cmd:

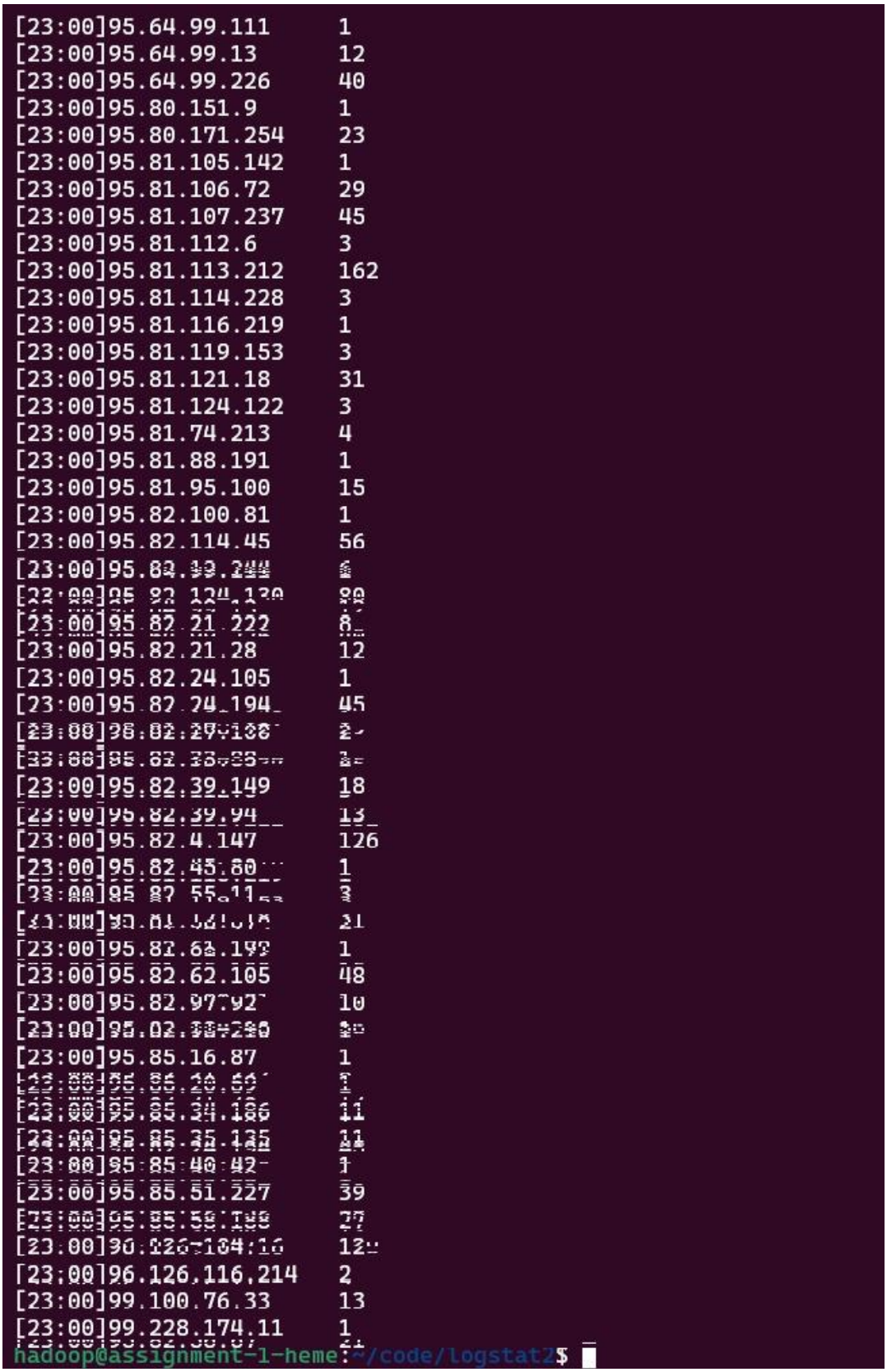
hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.6.jar -files logstat2_mapper.py,logstat2_reducer.py -mapper "python3 logstat2_mapper.py" -reducer "python3 logstat2_reducer.py" -input /input -output /output1



Logstat2 Mapreduce



Execution of map reduced in log stat 2

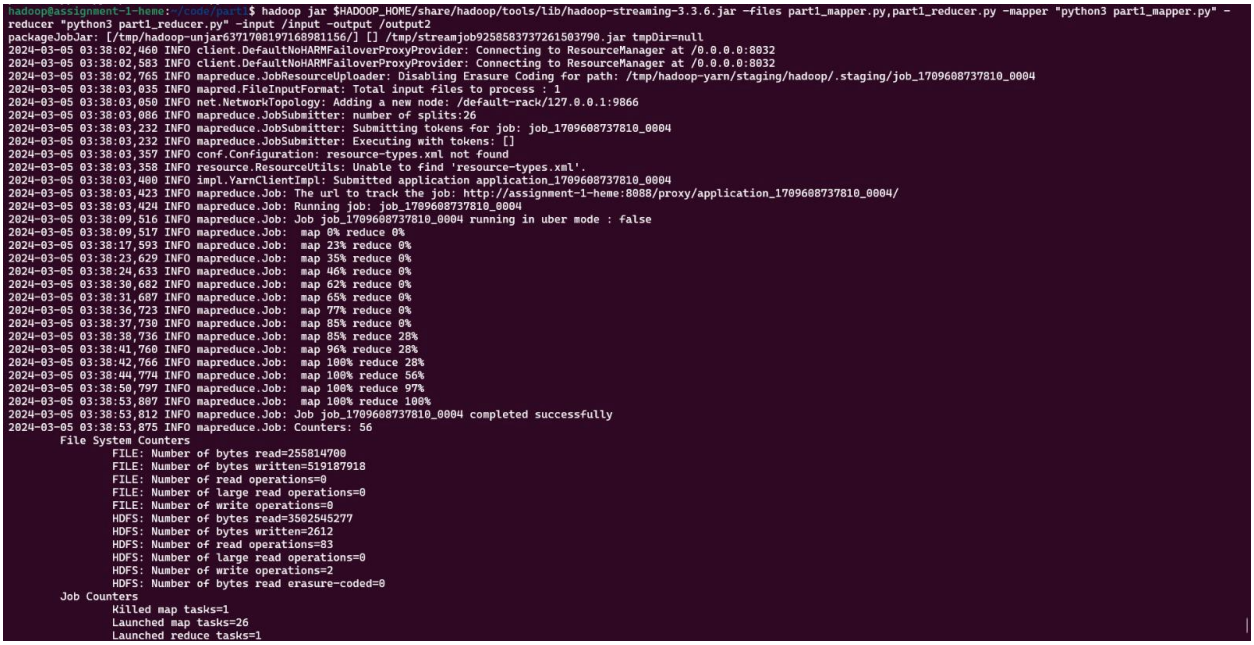


Output of the map reducer log stat 2

Part 1 - Top 3 IP address count for each hour

Now we need to calculate the top 3 ip address for each hour, which can be calculated by the following command.

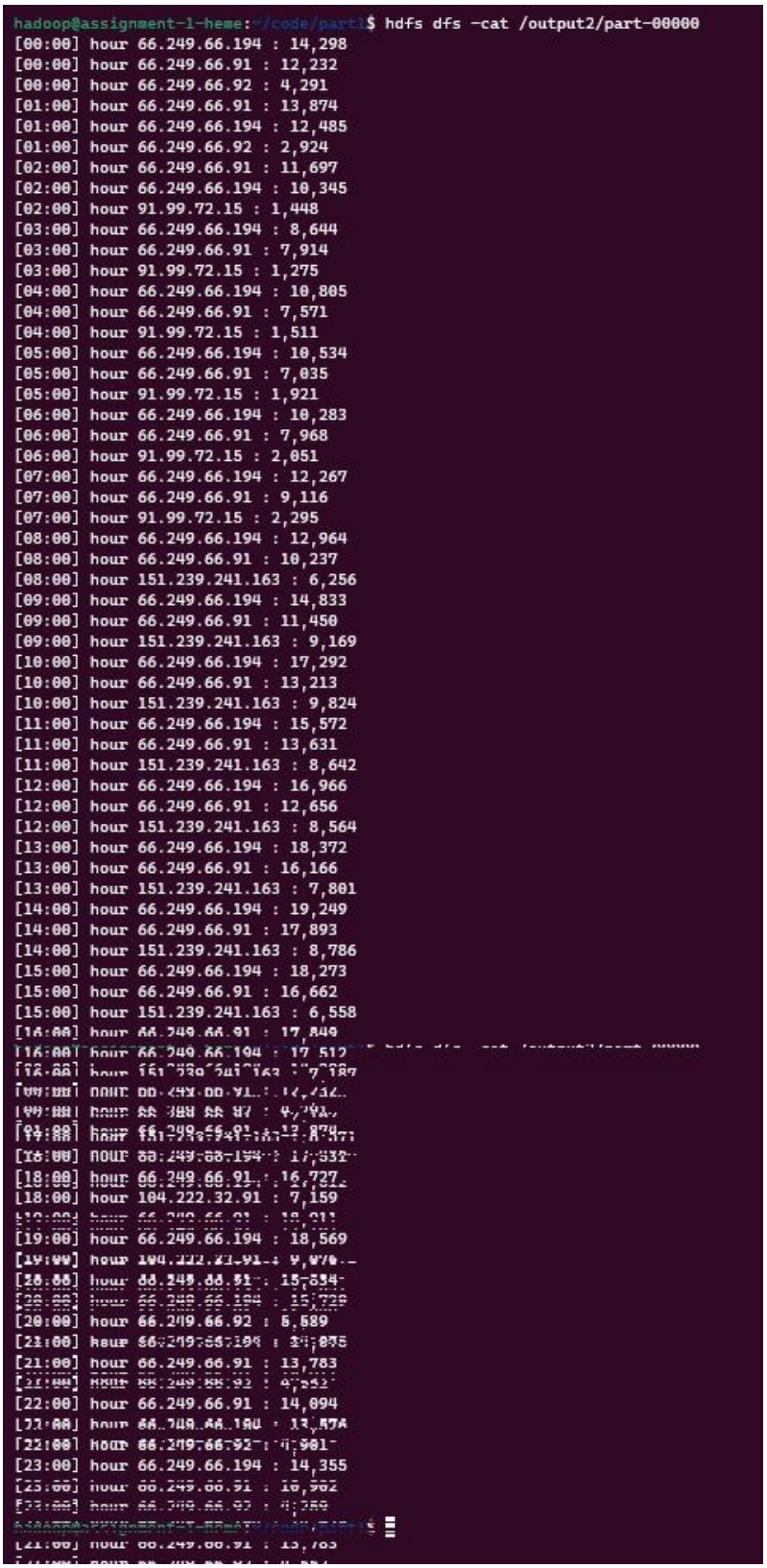
Cmd:
hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.6.jar -files part1_mapper.py,part1_reducer_mod.py -mapper "python3 part1_mapper.py" -reducer "python3 part1_reducer_mod.py" -input /input -output /output2



Part1 map reduce ip1

```
                Launched map tasks=26
                Launched reduce tasks=1
                Data-local map tasks=25
                Rack-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=142347
                Total time spent by all reduces in occupied slots (ms)=29469
                Total time spent by all map tasks (ms)=142347
                Total time spent by all reduce tasks (ms)=29469
                Total vcore-milliseconds taken by all map tasks=142347
                Total vcore-milliseconds taken by all reduce tasks=29469
                Total megabyte-milliseconds taken by all map tasks=145763328
                Total megabyte-milliseconds taken by all reduce tasks=30176256
        Map-Reduce Framework
                Map input records=10365152
                Map output records=10365152
                Map output bytes=235084390
                Map output materialized bytes=255814850
                Input split bytes=2054
                Combine input records=0
                Combine output records=0
                Reduce input groups=326893
                Reduce shuffle bytes=255814850
                Reduce input records=10365152
                Reduce output records=72
                Spilled Records=20730304
                Shuffled Maps =26
                Failed Shuffles=0
                Merged Map outputs=26
                GC time elapsed (ms)=1194
                CPU time spent (ms)=68480
                Physical memory (bytes) snapshot=10243825664
                Virtual memory (bytes) snapshot=74230550528
                Total committed heap usage (bytes)=13363052544
                Peak Map Physical memory (bytes)=389103616
                Peak Map Virtual memory (bytes)=2756751360
                Peak Reduce Physical memory (bytes)=535912448
                Peak Reduce Virtual memory (bytes)=2812657664
        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=3502543223
        File Output Format Counters
                Bytes Written=2612
2024-03-05 03:38:53,875 INFO streaming.StreamJob: Output directory: /output2
hadoop@assignment-1-heme:~/code/part1$
```

Part1 map reduce for ip2



Output for the part -1

Part 2 - Like Database Search

For this part, it is more efficient to re use the above reducer code from the log stat program, since it is the same logic and same functionality, which is to read the log file and extract the IP information using regex for the hours within the given input range

```
hadoop@assignment-1-heme:~/code/part2$ cat part2_reducer_mod.py
import sys
import argparse
from operator import itemgetter
from collections import defaultdict

def parse_args():
    parser = argparse.ArgumentParser(description='Process IP addresses and counts.')
    parser.add_argument('--timerange', help='Specify a timerange in the format "hh-hh". For example, --timerange 03-04')
    return parser.parse_args()

def clean_time(time_str):
    return time_str.strip()

def main():
    args = parse_args()
    timerange_filter = args.timerange

    dict_ip_count = {}

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

    # Sort the IP addresses based on their count in descending order
    sorted_dict_ip_count = sorted(dict_ip_count.items(), key=lambda x: -x[1])

    result_dict = {}
    for key, value in sorted_dict_ip_count:
        result_dict.setdefault(key, 0)
        result_dict[key] += int(value)

    sorted_dict = list(sorted(result_dict.items(), key=lambda item: item[1], reverse=True))

    converted_data = []

    for entry in sorted_dict:
        # Split the first part of the tuple to separate time and IP
        time_ip_split = entry[0].strip().rsplit(']', 1)
        try:
            time, ip = time_ip_split[0] + "]", time_ip_split[1]
        except:
            continue
        # Create a dictionary for this entry
```

Part 2 mapper python script

Database search

Cmd:

hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.6.jar -files part2_mapper_mod.py,part2_reducer_mod.py -mapper "python3 part2_mapper_mod.py" -reducer "python3 part2_reducer_mod.py --timerange '00-01'" -input /input -output /output5

```
packageJobJar: [/tmp/hadoop-unjar1185351319187387042S/] [] /tmp/streamjob1486472912259757717S.jar tmpDir=null
2024-03-04 22:45:33,196 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-03-04 22:45:33,471 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-03-04 22:45:33,789 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1709510660825_0053
2024-03-04 22:45:34,227 INFO mapred.FileInputFormat: Total input files to process : 1
2024-03-04 22:45:34,250 INFO net.NetworkTopology: Adding a new node: /default-rack/127.0.0.1:9866
2024-03-04 22:45:34,334 INFO mapreduce.JobSubmitter: number of splits:26
2024-03-04 22:45:34,971 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1709510660825_0053
2024-03-04 22:45:34,971 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-03-04 22:45:34,764 INFO conf.Configuration: resource-types.xml not found
2024-03-04 22:45:34,764 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-03-04 22:45:34,834 INFO impl.YarnClientImpl: Submitted application application_1709510660825_0053
2024-03-04 22:45:34,884 INFO mapreduce.Job: The url to track the job: http://dev-instance-1:8088/proxy/application_1709510660825_0053/
2024-03-04 22:45:34,885 INFO mapreduce.Job: Running job: job_1709510660825_0053
2024-03-04 22:45:43,033 INFO mapreduce.Job: Job job_1709510660825_0053 running in uber mode : false
2024-03-04 22:45:43,034 INFO mapreduce.Job:  map 0% reduce 0%
2024-03-04 22:45:58,152 INFO mapreduce.Job:  map 4% reduce 0%
2024-03-04 22:45:59,161 INFO mapreduce.Job:  map 8% reduce 0%
2024-03-04 22:46:01,173 INFO mapreduce.Job:  map 15% reduce 0%
2024-03-04 22:46:02,178 INFO mapreduce.Job:  map 19% reduce 0%
2024-03-04 22:46:04,190 INFO mapreduce.Job:  map 23% reduce 0%
2024-03-04 22:46:12,241 INFO mapreduce.Job:  map 31% reduce 0%
2024-03-04 22:46:14,257 INFO mapreduce.Job:  map 35% reduce 0%
2024-03-04 22:46:15,261 INFO mapreduce.Job:  map 38% reduce 0%
2024-03-04 22:46:16,267 INFO mapreduce.Job:  map 42% reduce 0%
2024-03-04 22:46:18,279 INFO mapreduce.Job:  map 46% reduce 0%
2024-03-04 22:46:25,326 INFO mapreduce.Job:  map 50% reduce 0%
2024-03-04 22:46:26,332 INFO mapreduce.Job:  map 54% reduce 0%
2024-03-04 22:46:27,341 INFO mapreduce.Job:  map 58% reduce 0%
2024-03-04 22:46:29,351 INFO mapreduce.Job:  map 62% reduce 0%
2024-03-04 22:46:31,366 INFO mapreduce.Job:  map 65% reduce 0%
2024-03-04 22:46:32,371 INFO mapreduce.Job:  map 65% reduce 22%
2024-03-04 22:46:38,413 INFO mapreduce.Job:  map 69% reduce 22%
2024-03-04 22:46:39,418 INFO mapreduce.Job:  map 77% reduce 22%
2024-03-04 22:46:42,448 INFO mapreduce.Job:  map 81% reduce 22%
2024-03-04 22:46:43,451 INFO mapreduce.Job:  map 85% reduce 22%
2024-03-04 22:46:44,458 INFO mapreduce.Job:  map 85% reduce 28%
2024-03-04 22:46:49,484 INFO mapreduce.Job:  map 88% reduce 28%
2024-03-04 22:46:50,492 INFO mapreduce.Job:  map 96% reduce 28%
2024-03-04 22:46:52,500 INFO mapreduce.Job:  map 100% reduce 28%
2024-03-04 22:46:56,517 INFO mapreduce.Job:  map 100% reduce 61%
2024-03-04 22:47:02,544 INFO mapreduce.Job:  map 100% reduce 84%
2024-03-04 22:47:08,571 INFO mapreduce.Job:  map 100% reduce 100%
2024-03-04 22:47:10,584 INFO mapreduce.Job: Job job_1709510660825_0053 completed successfully
2024-03-04 22:47:10,699 INFO mapreduce.Job: Counters: 56
        File System Counters
                FILE: Number of bytes read=258814700
                FILE: Number of bytes written=519204793
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=3502548277
                HDFS: Number of bytes written=188
                HDFS: Number of read operations=83
                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=26
                Launched reduce tasks=1
                Data-local map tasks=25
                Rack-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=311080
                Total time spent by all reduces in occupied slots (ms)=56530
                Total time spent by all map tasks (ms)=311080
                Total time spent by all reduce tasks (ms)=56530
                Total vcore-milliseconds taken by all map tasks=311080
                Total vcore-milliseconds taken by all reduce tasks=56530
                Total megabyte-milliseconds taken by all map tasks=318545920
                Total megabyte-milliseconds taken by all reduce tasks=57886720
        Map-Reduce Framework
                Map input records=10365152
                Map output records=10365152
                Map output bytes=235864398
                Map output materialized bytes=258814850
```

```
-04 22:47:10,699 INFO mapreduce.Job: Counters: 56
File System Counters
        FILE: Number of bytes read=255814700
        FILE: Number of bytes written=519204793
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=3502545277
        HDFS: Number of bytes written=108
        HDFS: Number of read operations=83
        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=26
        Launched reduce tasks=1
        Data-local map tasks=25
        Rack-local map tasks=1
        Total time spent by all maps in occupied slots (ms)=311080
        Total time spent by all reduces in occupied slots (ms)=56530
        Total time spent by all map tasks (ms)=311080
        Total time spent by all reduce tasks (ms)=56530
        Total vcore-milliseconds taken by all map tasks=311080
        Total vcore-milliseconds taken by all reduce tasks=56530
        Total megabyte-milliseconds taken by all map tasks=318545920
        Total megabyte-milliseconds taken by all reduce tasks=57886720
Map-Reduce Framework
        Map input records=10365152
        Map output records=10365152
        Map output bytes=235084390
        Map output materialized bytes=255814850
        Input split bytes=2054
        Combine input records=0
        Combine output records=0
        Reduce input groups=326893
        Reduce shuffle bytes=255814850
        Reduce input records=10365152
        Reduce output records=3
        Spilled Records=20730304
        Shuffled Maps =26
        Failed Shuffles=0
        Merged Map outputs=26
        GC time elapsed (ms)=738
        CPU time spent (ms)=67130
        Physical memory (bytes) snapshot=8985698304
        Virtual memory (bytes) snapshot=73988972544
        Total committed heap usage (bytes)=6737100800
        Peak Map Physical memory (bytes)=349601792
        Peak Map Virtual memory (bytes)=2748928000
        Peak Reduce Physical memory (bytes)=764551168
        Peak Reduce Virtual memory (bytes)=3099250688
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=3502543223
File Output Format Counters
        Bytes Written=108
```

Running mapreducer on part 2 for the range 0 -1

Output:

```
[00:00] hour 66.249.66.194 : 14,298
[00:00] hour 66.249.66.91 : 12,232
[00:00] hour 66.249.66.92 : 4,291
```

Fair and Capacity Scheduler

I ran this mapreduce along with the word count, sort, grep mapreduce example. I modified the yarn-site.xml to set up fair and capacity schedulers. Also added fair-scheduler.xml file.

Then map reduce is executed along with the word count, sort, grep mapreduce. To do this Yarn-site.xml is modified(fair and capacity schedulers). Another file called fair-scheduler.xml is also added.

Configuration setup for Capacity Scheduler:

```
hadoop@assignment-1-heme:~/hadoop/etc/hadoop$ cat mapred-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
<property>
  <name>yarn.app.mapreduce.am.env</name>
  <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
</property>
<property>
  <name>mapreduce.map.env</name>
  <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
</property>
<property>
  <name>mapreduce.reduce.env</name>
  <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
</property>
<property>
  <name>mapred.fairscheduler.allocation.file</name>
  <value>/hadoop/etc/hadoop/conf/fair-scheduler.xml</value>
</property>
</configuration>
```

Fair schedule is set in yarn-site.xml

```
<configuration>

  <property>
    <name>yarn.scheduler.capacity.maximum-applications</name>
    <value>10000</value>
    <description>
      Maximum number of applications that can be pending and running.
    </description>
  </property>

  <property>
    <name>yarn.scheduler.capacity.maximum-am-resource-percent</name>
    <value>0.1</value>
    <description>
      Maximum percent of resources in the cluster which can be used to run
      application masters i.e. controls number of concurrent running
      applications.
    </description>
  </property>

  <property>
    <name>yarn.scheduler.capacity.resource-calculator</name>
    <value>org.apache.hadoop.yarn.util.resource.DefaultResourceCalculator</value>
    <description>
      The ResourceCalculator implementation to be used to compare
      Resources in the scheduler.
      The default i.e. DefaultResourceCalculator only uses Memory while
      DominantResourceCalculator uses dominant-resource to compare
      multi-dimensional resources such as Memory, CPU etc.
    </description>
  </property>

  <property>
    <name>yarn.scheduler.capacity.root.queues</name>
    <value>default</value>
    <description>
      The queues at the this level (root is the root queue).
    </description>
  </property>
```

Fair schedule is set in fair-scheduler.xml

1. **Execution for wordcount example:**

hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar wordcount /input1 /output6

```
hadoop@assignment-1-heme:~/hadoop/etc/hadoop$ hadoop jar  $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar wordcount /input /output6
2024-03-05 03:51:46,873 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-03-05 03:51:47,158 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1709608737810_0006
2024-03-05 03:51:47,359 INFO input.FileInputFormat: Total input files to process : 1
2024-03-05 03:51:47,421 INFO mapreduce.JobSubmitter: number of splits:26
2024-03-05 03:51:47,571 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1709608737810_0006
2024-03-05 03:51:47,571 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-03-05 03:51:47,694 INFO conf.Configuration: resource-types.xml not found
2024-03-05 03:51:47,695 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-03-05 03:51:47,738 INFO impl.YarnClientImpl: Submitted application application_1709608737810_0006
2024-03-05 03:51:47,763 INFO mapreduce.Job: The url to track the job: http://assignment-1-heme:8088/proxy/application_1709608737810_0006/
2024-03-05 03:51:47,764 INFO mapreduce.Job: Running job: job_1709608737810_0006
2024-03-05 03:51:53,850 INFO mapreduce.Job: Job job_1709608737810_0006 running in uber mode : false
2024-03-05 03:51:53,852 INFO mapreduce.Job:  map 0% reduce 0%
2024-03-05 03:52:07,965 INFO mapreduce.Job:  map 4% reduce 0%
2024-03-05 03:52:08,972 INFO mapreduce.Job:  map 12% reduce 0%
2024-03-05 03:52:09,983 INFO mapreduce.Job:  map 19% reduce 0%
2024-03-05 03:52:10,991 INFO mapreduce.Job:  map 23% reduce 0%
2024-03-05 03:52:22,064 INFO mapreduce.Job:  map 27% reduce 0%
2024-03-05 03:52:23,070 INFO mapreduce.Job:  map 42% reduce 0%
2024-03-05 03:52:28,099 INFO mapreduce.Job:  map 42% reduce 14%
2024-03-05 03:52:35,137 INFO mapreduce.Job:  map 46% reduce 14%
2024-03-05 03:52:36,143 INFO mapreduce.Job:  map 54% reduce 14%
2024-03-05 03:52:37,150 INFO mapreduce.Job:  map 58% reduce 14%
2024-03-05 03:52:38,155 INFO mapreduce.Job:  map 62% reduce 14%
2024-03-05 03:52:40,176 INFO mapreduce.Job:  map 62% reduce 21%
2024-03-05 03:52:48,210 INFO mapreduce.Job:  map 65% reduce 21%
2024-03-05 03:52:49,217 INFO mapreduce.Job:  map 77% reduce 21%
2024-03-05 03:52:51,231 INFO mapreduce.Job:  map 81% reduce 21%
2024-03-05 03:52:52,238 INFO mapreduce.Job:  map 81% reduce 27%
2024-03-05 03:53:01,280 INFO mapreduce.Job:  map 88% reduce 27%
2024-03-05 03:53:02,286 INFO mapreduce.Job:  map 92% reduce 27%
2024-03-05 03:53:03,291 INFO mapreduce.Job:  map 100% reduce 27%
2024-03-05 03:53:04,295 INFO mapreduce.Job:  map 100% reduce 32%
2024-03-05 03:53:07,311 INFO mapreduce.Job:  map 100% reduce 100%
2024-03-05 03:53:07,315 INFO mapreduce.Job: Job job_1709608737810_0006 completed successfully
2024-03-05 03:53:07,398 INFO mapreduce.Job: Counters: 55
        File System Counters
                FILE: Number of bytes read=1037247315
                FILE: Number of bytes written=1528561269
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=3502545615
                HDFS: Number of bytes written=376527705
                HDFS: Number of read operations=83
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
```

Screenshot of map reduce on first input example

```
hadoop@assignment-1-heme:~/data$ hadoop jar  $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar sort /input1 /output_sort
2024-03-05 04:11:35,340 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
Running on 1 nodes to sort from hdfs://localhost:9000/input1 into hdfs://localhost:9000/output_sort with 1 reduces.
Job started: Tue Mar 05 04:11:35 UTC 2024
2024-03-05 04:11:35,960 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-03-05 04:11:36,094 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1709608737810_0008
2024-03-05 04:11:36,284 INFO input.FileInputFormat: Total input files to process : 1
2024-03-05 04:11:36,339 INFO mapreduce.JobSubmitter: number of splits:1
2024-03-05 04:11:36,477 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1709608737810_0008
2024-03-05 04:11:36,477 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-03-05 04:11:36,597 INFO conf.Configuration: resource-types.xml not found
2024-03-05 04:11:36,597 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-03-05 04:11:36,637 INFO impl.YarnClientImpl: Submitted application application_1709608737810_0008
2024-03-05 04:11:36,661 INFO mapreduce.Job: The url to track the job: http://assignment-1-heme:8088/proxy/application_1709608737810_0008/
2024-03-05 04:11:36,661 INFO mapreduce.Job: Running job: job_1709608737810_0008
2024-03-05 04:11:42,746 INFO mapreduce.Job: Job job_1709608737810_0008 running in uber mode : false
2024-03-05 04:11:42,747 INFO mapreduce.Job:  map 0% reduce 0%
2024-03-05 04:11:46,791 INFO mapreduce.Job:  map 100% reduce 0%
2024-03-05 04:11:51,815 INFO mapreduce.Job:  map 100% reduce 100%
2024-03-05 04:11:51,821 INFO mapreduce.Job: Job job_1709608737810_0008 completed successfully
2024-03-05 04:11:51,890 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=1441
                FILE: Number of bytes written=556327
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=3067
                HDFS: Number of bytes written=2131
                HDFS: Number of read operations=9
                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=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=2005
                Total time spent by all reduces in occupied slots (ms)=2079
                Total time spent by all map tasks (ms)=2005
                Total time spent by all reduce tasks (ms)=2079
                Total vcore-milliseconds taken by all map tasks=2005
                Total vcore-milliseconds taken by all reduce tasks=2079
                Total megabyte-milliseconds taken by all map tasks=2053120
                Total megabyte-milliseconds taken by all reduce tasks=2128896
        Map-Reduce Framework
                Map input records=100
                Map output records=100
```

Word count map reduce on second input example

```
spark   72
spider/4.0(+http://www.sogou.com/docs/help/webmasters.htm#07)"  11
spring  21
ss.iwihuj/at/gmail.com)"        21
subscribers;    37
sun4u;  2
thanks  78
thinkphp'       18
thl_4400        25
thonkphp        36
to      369
touch;  86
tr-tr;  109
universal5422;  200
universal5430;  2
unknown 33
v0.4)"  1
vivo    6
web     11
wv)     45161
www.MihanGsm.com;       68
www.amiami.com:443      1
www.fars-gsm.com        40
www.mellarmobile.com    1
www.msftncsi.com:443    6
www.pars-gsm.com        104
www.pars-gsm.com)       85
x33&page=1"     2
x4      48
x4)     126
x64)    1823217
x64;    1162749
x86     79
x86'    28
x86_64  347
x86_64) 42874
x86_64; 6349
x86_64;fa)      1
yahoo.adquality.lwd.desktop/1548117389-0"       1
yahoo.adquality.lwd.desktop/1548117392-0"       1
yahoo.adquality.lwd.desktop/1548203995-0"       1
yahoo.adquality.lwd.desktop/1548203998-0"       1
yahoo.adquality.lwd.desktop/1548292923-0"       1
yahoo.adquality.lwd.desktop/1548292926-0"       1
yahoo.adquality.lwd.desktop/1548377471-0"       1
yahoo.adquality.lwd.desktop/1548377472-0"       1
yie8)"  4
zanbil.ir       50
zgrab/0.x       21
zgrab/0.x"      33
zh-CN;  1024
zh-TW)  1
zh-cn)  214
zh-cn)AppleWebKit/534.46.0(KHTML,       35
zh-cn;  633
zlib/1.2.3"     137
zvav;   206
~'13\xCC        32
```

Screenshot of word count mapreduce example output

2. Execution for sort example:

Then sequence file(seq file) which contains the number in un sorted at row level and column level, was used. This seq file is then uploaded or placed into hdfs.

```
2024-03-03 17:01:48,363 INFO zlib.ZlibFactory: Successfully loaded & initialized native-zlib library
2024-03-03 17:01:48,363 INFO compress.CodecPool: Got brand-new decompressor [.deflate]
30      30
31      32
32      34
33      36
34      38
35      31 30
36      31 32
37      31 34
38      31 36
39      31 38
31 30   32 30
31 31   32 32
31 32   32 34
31 33   32 36
31 34   32 38
31 35   33 30
31 36   33 32
31 37   33 34
31 38   33 36
31 39   33 38
32 30   34 30
32 31   34 32
32 32   34 34
32 33   34 36
32 34   34 38
32 35   35 30
32 36   35 32
32 37   35 34
32 38   35 36
32 39   35 38
33 30   36 30
```

Screenshot of placing sequence file into hdfs

Cmd to execute sort example:

hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar sort /input1 /output_sort

```
hadoop@assignment-1-heme:~/data$ hadoop jar  $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar sort /input1 /output_sort
2024-03-05 04:11:35,340 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
Running on 1 nodes to sort from hdfs://localhost:9000/input1 into hdfs://localhost:9000/output_sort with 1 reduces.
Job started: Tue Mar 05 04:11:35 UTC 2024
2024-03-05 04:11:35,960 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-03-05 04:11:36,094 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1709608737810_0008
2024-03-05 04:11:36,284 INFO input.FileInputFormat: Total input files to process : 1
2024-03-05 04:11:36,339 INFO mapreduce.JobSubmitter: number of splits:1
2024-03-05 04:11:36,477 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1709608737810_0008
2024-03-05 04:11:36,477 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-03-05 04:11:36,597 INFO conf.Configuration: resource-types.xml not found
2024-03-05 04:11:36,597 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-03-05 04:11:36,637 INFO impl.YarnClientImpl: Submitted application application_1709608737810_0008
2024-03-05 04:11:36,661 INFO mapreduce.Job: The url to track the job: http://assignment-1-heme:8088/proxy/application_1709608737810_0008/
2024-03-05 04:11:36,661 INFO mapreduce.Job: Running job: job_1709608737810_0008
2024-03-05 04:11:42,746 INFO mapreduce.Job: Job job_1709608737810_0008 running in uber mode : false
2024-03-05 04:11:42,747 INFO mapreduce.Job:  map 0% reduce 0%
2024-03-05 04:11:46,791 INFO mapreduce.Job:  map 100% reduce 0%
2024-03-05 04:11:51,815 INFO mapreduce.Job:  map 100% reduce 100%
2024-03-05 04:11:51,821 INFO mapreduce.Job: Job job_1709608737810_0008 completed successfully
2024-03-05 04:11:51,890 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=1441
                FILE: Number of bytes written=556327
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=3067
                HDFS: Number of bytes written=2131
                HDFS: Number of read operations=9
                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=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=2005
                Total time spent by all reduces in occupied slots (ms)=2079
                Total time spent by all map tasks (ms)=2005
                Total time spent by all reduce tasks (ms)=2079
                Total vcore-milliseconds taken by all map tasks=2005
                Total vcore-milliseconds taken by all reduce tasks=2079
                Total megabyte-milliseconds taken by all map tasks=2053120
                Total megabyte-milliseconds taken by all reduce tasks=2128896
        Map-Reduce Framework
                Map input records=100
                Map output records=100
```

sort mapreduce example input1

```
        Map-Reduce Framework
                Map input records=100
                Map output records=100
                Map output bytes=1235
                Map output materialized bytes=1441
                Input split bytes=93
                Combine input records=0
                Combine output records=0
                Reduce input groups=100
                Reduce shuffle bytes=1441
                Reduce input records=100
                Reduce output records=100
                Spilled Records=200
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=52
                CPU time spent (ms)=1010
                Physical memory (bytes) snapshot=531935232
                Virtual memory (bytes) snapshot=5476925440
                Total committed heap usage (bytes)=494927872
                Peak Map Physical memory (bytes)=311296000
                Peak Map Virtual memory (bytes)=2736181248
                Peak Reduce Physical memory (bytes)=220639232
                Peak Reduce Virtual memory (bytes)=2740744192
        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=2974
        File Output Format Counters
                Bytes Written=2131
```

Sort mapreduce example input2

Output:

After the sort code being executed on the unsorted sequence file, we get an output which is now organized such that each row is arranged in ascending order according to the data.

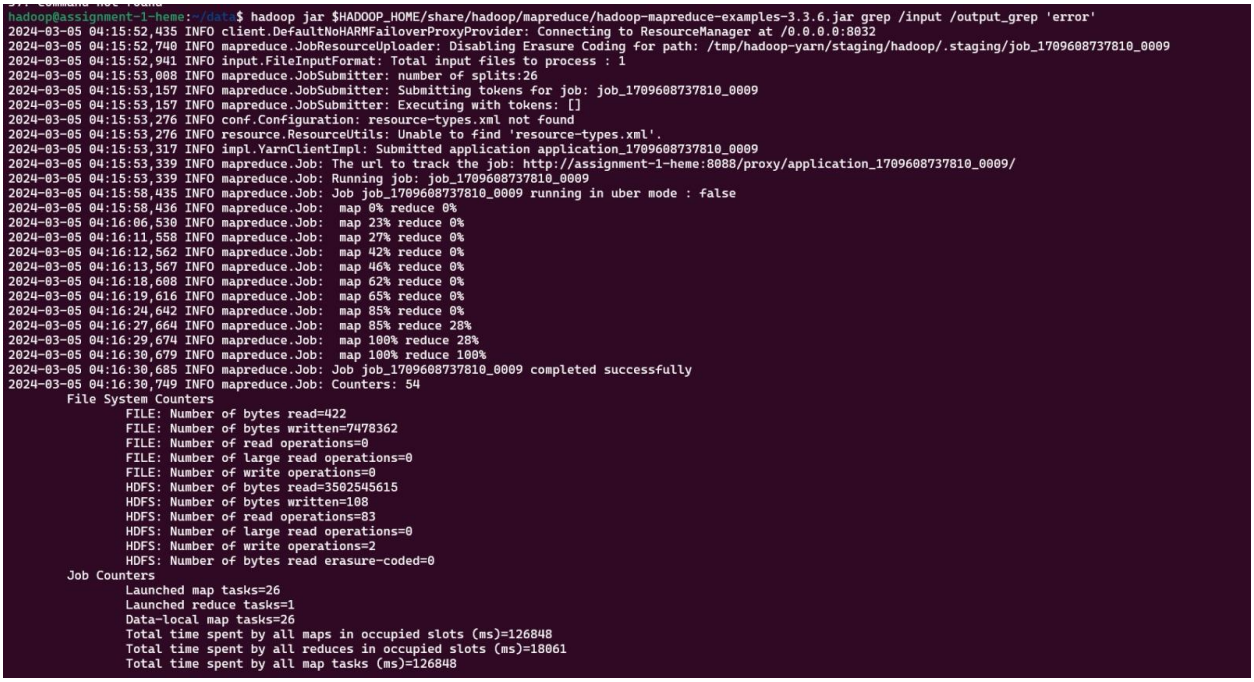
```
hadoop@assignment-1-heme:~/data$ hdfs dfs -text  /output_sort/part-r-00000
30      30
31      32
31 30   32 30
31 31   32 32
31 32   32 34
31 33   32 36
31 34   32 38
31 35   33 30
31 36   33 32
31 37   33 34
31 38   33 36
31 39   33 38
32      34
32 30   34 30
32 31   34 32
32 32   34 34
32 33   34 36
32 34   34 38
32 35   35 30
32 36   35 32
32 37   35 34
32 38   35 36
32 39   35 38
33      36
33 30   36 30
33 31   36 32
33 32   36 34
33 33   36 36
33 34   36 38
33 35   37 30
33 36   37 32
33 37   37 34
33 38   37 36
33 39   37 38
34      38
34 30   38 30
34 31   38 32
34 32   38 34
```

Sort map reduce for example op

3. Execution for grep example:

I am executing a grep example to count the number of errors present in a sample.log file located in HDFS at /input3. This is achieved by executing the command below, which performs the grep operation to obtain the required count.

Cmd:
hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar grep /input /output_grep 'error'



Screenshot of grep mapreduce example input1

```
                FILE: Number of bytes written=553449
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=239
                HDFS: Number of bytes written=12
                HDFS: Number of read operations=9
                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=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=1594
                Total time spent by all reduces in occupied slots (ms)=1653
                Total time spent by all map tasks (ms)=1594
                Total time spent by all reduce tasks (ms)=1653
                Total vcore-milliseconds taken by all map tasks=1594
                Total vcore-milliseconds taken by all reduce tasks=1653
                Total megabyte-milliseconds taken by all map tasks=1632256
                Total megabyte-milliseconds taken by all reduce tasks=1692672
        Map-Reduce Framework
                Map input records=1
                Map output records=1
                Map output bytes=14
                Map output materialized bytes=22
                Input split bytes=131
                Combine input records=0
                Combine output records=0
                Reduce input groups=1
                Reduce shuffle bytes=22
                Reduce input records=1
                Reduce output records=1
                Spilled Records=2
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=27
                CPU time spent (ms)=780
                Physical memory (bytes) snapshot=529129472
                Virtual memory (bytes) snapshot=5477249024
                Total committed heap usage (bytes)=494927872
                Peak Map Physical memory (bytes)=308137984
                Peak Map Virtual memory (bytes)=2729844736
                Peak Reduce Physical memory (bytes)=220991488
                Peak Reduce Virtual memory (bytes)=2747404288
        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=108
        File Output Format Counters
                Bytes Written=12
```

Screenshot of grep mapreduce example input2

```
hadoop@assignment-1-heme:~/data$ hdfs dfs -cat /output_grep/part-r-00000
27678   error
```

Screenshot of grep mapreduce example output

Enabling Fair Scheduler:

To Enable Fair Scheduler following changes are amended in the respective files.

1. **Configuration setup in yarn-site.xml**

```
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>

  <!-- Set the scheduler to Fair Scheduler -->
  <property>
    <name>yarn.resourcemanager.scheduler.class</name>
    <value>org.apache.hadoop.yarn.server.resourcemanager.scheduler.fair.FairScheduler</value>
  </property>

  <!-- Specify the path to the Fair Scheduler configuration file -->
  <property>
    <name>yarn.scheduler.fair.allocation.file</name>
    <value>/home/hadoop/hadoop/etc/hadoop/fair-scheduler.xml</value>
  </property>

  <!-- Enable Fair Scheduler preemption if needed -->
  <property>
    <name>yarn.scheduler.fair.preemption</name>
    <value>true</value>
  </property>

  <!-- Site specific YARN configuration properties -->
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>

</configuration>
```

Screenshot of yarn-site.xml

2. **Configuration setup in Fair Scheduler.xml:**

```
<?xml version="1.0"?>
<allocations>

  <!-- Example Pool Configurations -->
  <pool name="pool1">
    <!-- Minimum and Maximum Resources for Maps and Reduces -->
    <minMaps>10</minMaps>
    <minReduces>5</minReduces>
    <maxMaps>20</maxMaps>
    <maxReduces>10</maxReduces>

    <!-- Maximum Running Jobs in the Pool -->
    <maxRunningJobs>5</maxRunningJobs>

    <!-- Preemption Timeout in Seconds -->
    <minSharePreemptionTimeout>300</minSharePreemptionTimeout>

    <!-- Pool's Weight for Fair Sharing Calculations -->
    <weight>1.0</weight>
  </pool>

  <pool name="pool2">
    <minMaps>5</minMaps>
    <minReduces>2</minReduces>
    <maxMaps>15</maxMaps>
    <maxReduces>8</maxReduces>
    <maxRunningJobs>3</maxRunningJobs>
    <minSharePreemptionTimeout>240</minSharePreemptionTimeout>
    <weight>0.5</weight>
  </pool>

  <!-- Example User Configuration -->
  <user name="user1">
    <!-- Maximum Running Jobs for the User -->
    <maxRunningJobs>8</maxRunningJobs>
  </user>

  <!-- Default Running Job Limits for Pools and Users -->
  <poolMaxJobsDefault>10</poolMaxJobsDefault>
  <userMaxJobsDefault>5</userMaxJobsDefault>

  <!-- Default Minimum Share Preemption Timeout for Pools -->
  <defaultMinSharePreemptionTimeout>600</defaultMinSharePreemptionTimeout>
```

Screenshot of fair-scheduler.xml

```
  <!-- Default Running Job Limits for Pools and Users -->
  <poolMaxJobsDefault>10</poolMaxJobsDefault>
  <userMaxJobsDefault>5</userMaxJobsDefault>

  <!-- Default Minimum Share Preemption Timeout for Pools -->
  <defaultMinSharePreemptionTimeout>600</defaultMinSharePreemptionTimeout>

  <!-- Fair Share Preemption Timeout for Jobs Below Their Fair Share -->
  <fairSharePreemptionTimeout>600</fairSharePreemptionTimeout>

</allocations>
```

Screenshot of fair-scheduler.xml

3. Enabling Fair Scheduler in Mapred-site.xml :

```
<configuration>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
<property>
  <name>yarn.app.mapreduce.am.env</name>
  <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
</property>
<property>
  <name>mapreduce.map.env</name>
  <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
</property>
<property>
  <name>mapreduce.reduce.env</name>
  <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
</property>
    <!-- Specify the path to the Fair Scheduler configuration file -->
  <property>
    <name>mapred.fairscheduler.allocation.file</name>
    <value>/home/hadoop/hadoop/etc/hadoop/fair-scheduler.xml</value>
  </property>
</configuration>
```

Screenshot of mapred-site.xml

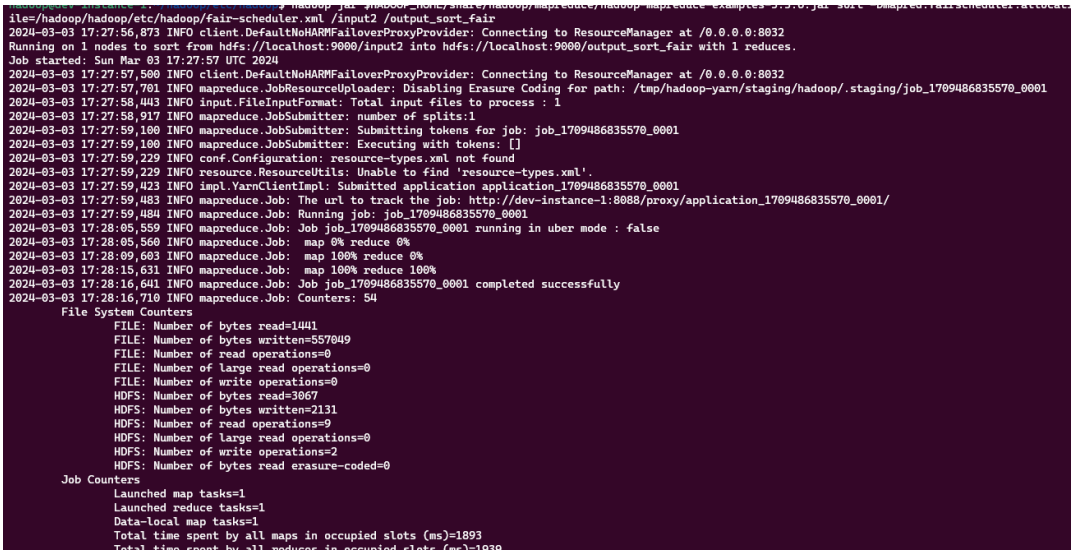
Executing all the above 3 examples with Fair Scheduler enabled:

1. **Execution of sort example.**

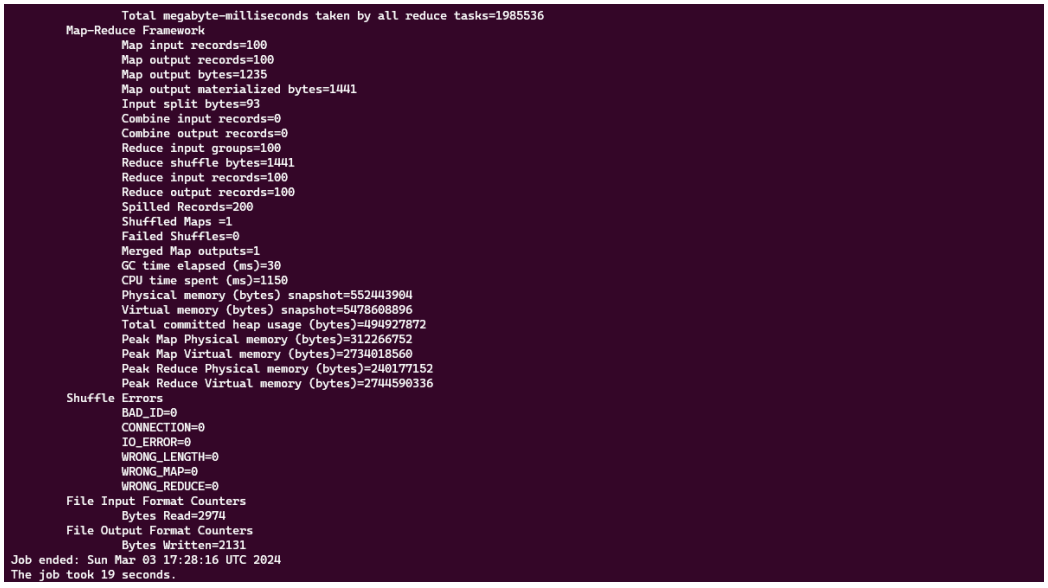
By executing below command by adding location of fair scheduler as a parameter, it uses fair scheduler to run mapper and reducer code for sort.

Command:

hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar sort -Dmapred.fairscheduler.allocation.file=/hadoop/hadoop/etc/hadoop/fair-scheduler.xml /input /output_sort



Screenshot of map reduce on first input example



Word count map reduce on second input example

Output:

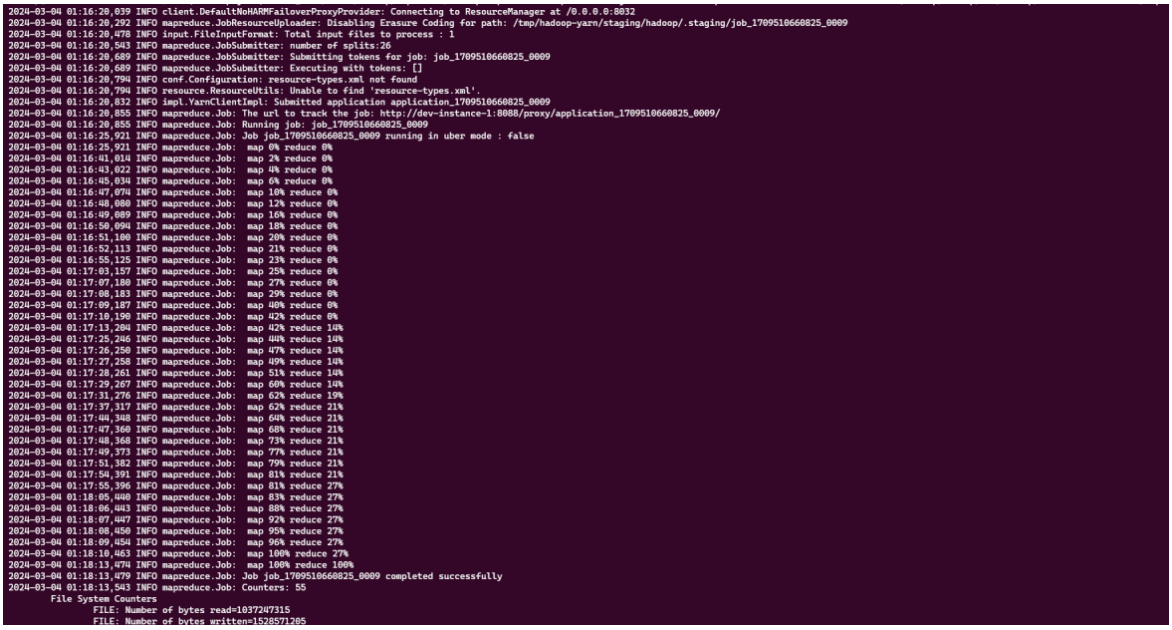
```
hadoop@assignment-1-heme:~/data$ hdfs dfs -text  /output_sort/part-r-00000
30      30
31      32
31 30   32 30
31 31   32 32
31 32   32 34
31 33   32 36
31 34   32 38
31 35   33 30
31 36   33 32
31 37   33 34
31 38   33 36
31 39   33 38
32      34
32 30   34 30
32 31   34 32
32 32   34 34
32 33   34 36
32 34   34 38
32 35   35 30
32 36   35 32
32 37   35 34
32 38   35 36
32 39   35 38
33      36
33 30   36 30
33 31   36 32
33 32   36 34
33 33   36 36
33 34   36 38
33 35   37 30
33 36   37 32
33 37   37 34
33 38   37 36
33 39   37 38
34      38
34 30   38 30
34 31   38 32
34 32   38 34
```

Screenshot of sort mapreduce example output

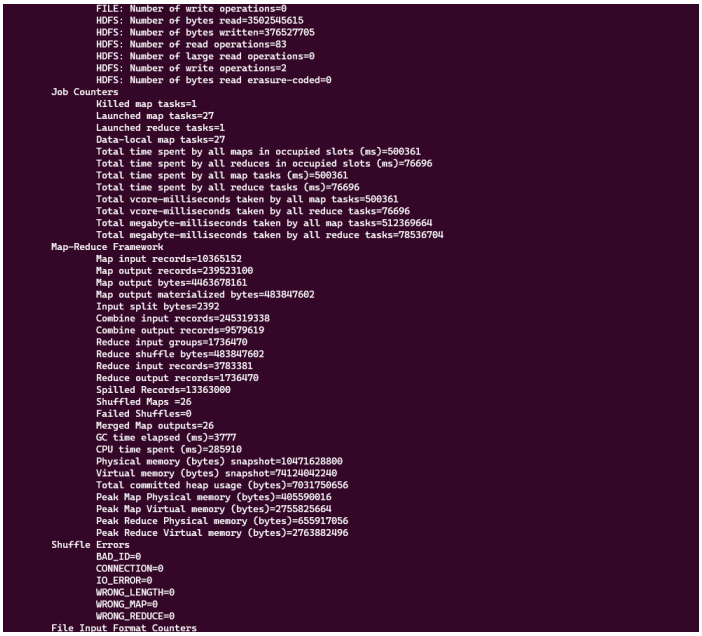
2. **Execution of wordcount example:**

To ensure fair scheduler is used by the program, we need to pass it as input parameter. Below commands takes fair scheduler as param, additional to previous command which we have executed.

Command:
hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar wordcount -Dmapred.fairscheduler.allocation.file=/hadoop/hadoop/etc/hadoop/fair-scheduler.xml /input /output_file



Screenshot of sort mapreduce example input1



Screenshot of sort mapreduce example input2

Output:

```
wv)     45161
www.MihanGsm.com;       68
www.amiami.com:443      1
www.fars-gsm.com        40
www.mellarmobile.com    1
www.msftncsi.com:443    6
www.pars-gsm.com        104
www.pars-gsm.com)       85
x33&page=1"     2
x4      48
x4)     126
x64)    1823217
x64;    1162749
x86     79
x86'    28
x86_64  347
x86_64) 42874
x86_64; 6349
x86_64;fa)      1
yahoo.adquality.lwd.desktop/1548117389-0"       1
yahoo.adquality.lwd.desktop/1548117392-0"       1
yahoo.adquality.lwd.desktop/1548203995-0"       1
yahoo.adquality.lwd.desktop/1548203998-0"       1
yahoo.adquality.lwd.desktop/1548292923-0"       1
yahoo.adquality.lwd.desktop/1548292926-0"       1
yahoo.adquality.lwd.desktop/1548377471-0"       1
yahoo.adquality.lwd.desktop/1548377472-0"       1
yie8)"  4
zanbil.ir       50
zgrab/0.x       21
zgrab/0.x"      33
zh-CN;  1024
zh-TW)  1
zh-cn)  214
zh-cn)AppleWebKit/534.46.0(KHTML,       35
zh-cn;  633
zlib/1.2.3"     137
```

Screenshot of wordcount mapreduce example output

3. Execution of grep example:

Similarly we take fair scheduler as an input param, to execute the sort code, similar to previous example.

Command:

hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar grep -Dmapred.fairscheduler.allocation.file=/hadoop/hadoop/etc/hadoop/fair-scheduler.xml /input /output 'error'

```
doop/hadoop/etc/hadoop/fair-scheduler.xml /input /output_grep_fair 'error'
2024-03-04 01:35:23,709 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-03-04 01:35:23,968 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1709510660825_0010
2024-03-04 01:35:24,156 INFO input.FileInputFormat: Total input files to process : 1
2024-03-04 01:35:24,209 INFO mapreduce.JobSubmitter: number of splits:26
2024-03-04 01:35:24,357 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1709510660825_0010
2024-03-04 01:35:24,357 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-03-04 01:35:24,463 INFO conf.Configuration: resource-types.xml not found
2024-03-04 01:35:24,463 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-03-04 01:35:24,501 INFO impl.YarnClientImpl: Submitted application application_1709510660825_0010
2024-03-04 01:35:24,523 INFO mapreduce.Job: The url to track the job: http://dev-instance-1:8088/proxy/application_1709510660825_0010/
2024-03-04 01:35:24,523 INFO mapreduce.Job: Running job: job_1709510660825_0010
2024-03-04 01:35:29,579 INFO mapreduce.Job: Job job_1709510660825_0010 running in uber mode : false
2024-03-04 01:35:29,580 INFO mapreduce.Job:  map 0% reduce 0%
2024-03-04 01:35:34,622 INFO mapreduce.Job:  map 4% reduce 0%
2024-03-04 01:35:37,633 INFO mapreduce.Job:  map 8% reduce 0%
2024-03-04 01:35:39,646 INFO mapreduce.Job:  map 12% reduce 0%
2024-03-04 01:35:41,663 INFO mapreduce.Job:  map 15% reduce 0%
2024-03-04 01:35:43,672 INFO mapreduce.Job:  map 19% reduce 0%
2024-03-04 01:35:44,678 INFO mapreduce.Job:  map 23% reduce 0%
2024-03-04 01:35:45,688 INFO mapreduce.Job:  map 31% reduce 0%
2024-03-04 01:35:48,708 INFO mapreduce.Job:  map 35% reduce 0%
2024-03-04 01:35:50,730 INFO mapreduce.Job:  map 38% reduce 0%
2024-03-04 01:35:51,739 INFO mapreduce.Job:  map 42% reduce 0%
2024-03-04 01:35:53,749 INFO mapreduce.Job:  map 50% reduce 0%
2024-03-04 01:35:55,763 INFO mapreduce.Job:  map 54% reduce 0%
2024-03-04 01:35:57,772 INFO mapreduce.Job:  map 58% reduce 0%
2024-03-04 01:35:59,780 INFO mapreduce.Job:  map 62% reduce 0%
2024-03-04 01:36:00,789 INFO mapreduce.Job:  map 65% reduce 0%
2024-03-04 01:36:01,792 INFO mapreduce.Job:  map 69% reduce 0%
2024-03-04 01:36:02,799 INFO mapreduce.Job:  map 73% reduce 0%
2024-03-04 01:36:03,805 INFO mapreduce.Job:  map 73% reduce 24%
2024-03-04 01:36:04,809 INFO mapreduce.Job:  map 77% reduce 24%
2024-03-04 01:36:06,815 INFO mapreduce.Job:  map 81% reduce 24%
2024-03-04 01:36:08,823 INFO mapreduce.Job:  map 85% reduce 24%
2024-03-04 01:36:09,828 INFO mapreduce.Job:  map 92% reduce 27%
2024-03-04 01:36:10,832 INFO mapreduce.Job:  map 96% reduce 27%
2024-03-04 01:36:11,835 INFO mapreduce.Job:  map 100% reduce 100%
```

Screenshot of grep mapreduce example input1

```
        Total vcore-milliseconds taken by all reduce tasks=1677
        Total megabyte-milliseconds taken by all map tasks=1678336
        Total megabyte-milliseconds taken by all reduce tasks=171724
lap-Reduce Framework
        Map input records=1
        Map output records=1
        Map output bytes=14
        Map output materialized bytes=22
        Input split bytes=131
        Combine input records=0
        Combine output records=0
        Reduce input groups=1
        Reduce shuffle bytes=22
        Reduce input records=1
        Reduce output records=1
        Spilled Records=2
        Shuffled Maps =1
        Failed Shuffles=0
        Merged Map outputs=1
        GC time elapsed (ms)=29
        CPU time spent (ms)=810
        Physical memory (bytes) snapshot=5146132480
        Virtual memory (bytes) snapshot=5465944064
        Total committed heap usage (bytes)=494927872
        Peak Map Physical memory (bytes)=305217536
        Peak Map Virtual memory (bytes)=2729836544
        Peak Reduce Physical memory (bytes)=209395712
        Peak Reduce Virtual memory (bytes)=2736107520
huffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
ile Input Format Counters
        Bytes Read=108
ile Output Format Counters
```

Screenshot of grep mapreduce example input2

Output:

1. Describe your Jetstream2 instance. What was your cloud.init script? Which size instance did you use?

For this assignment, initially I have chosen, m3.quad, but after working on assignment, I have released that m3.quad didn't have enough storage to download and unzip the access file, so my initial though was to attach extra volume, but then there were only limited volume instances available at the time, so could not get an volume, so scraped the full instance and did the whole installation process and worked on part 1 on m3.medium which has 100gb of volume attached to it. Which has sufficient storage space and computing resource to run the code of the assignment, once the assignment was done the instance was freed for others to use.

The use of cloud.init script is for configuring the Jetstream instance. This script had steps for initial setup procedure, dependencies, how to install dependencies, configuring the user account, and steps for initiating a various service.

2. Did you use the Console, Web Shell, or Web Desktop? If you used more than one interface, which did you prefer?

For this assignment, I have used both the web shell and web desktop, for different tasks. I have used web shell(command line access), for installing dependencies, and debug the error while installing dependencies, so therefore, I preferred console for configuration part and executing the code, as it easier to interact and debug the error. But when there is some tasks which requires more graphical assistance, I have used web desktop. For majority of the work web shell was used, but for some tasks which would be easier to use graphical interface web desktop is used, so it all depends on the tasks, since the assignment had more component related to configuration and executing scripts, I have used web shell more.

3. Do you have any feedback on your experience with this instance and interface(s)?
ßßßßßßßßß
It is an positive experience with Jetstream except few cons as discussed below.

Pros:

1. It has intuitive interface and easy to navigate
2. The computational needs has been met, without much latency.
3. It has good web shell which makes interacting with the system much more easier, and there were not much restrictions in that part.
4. Web desktop was clean and minimum design which was good, because it had only the necessities installed.

Cons:

1. Initially only limited number of instance was allowed, which was much less than class size, so it was difficult to get an instance.

2. Similar problem with attaching volume, in my initial setup assuming 20GB would be enough I have used m2.quad, but turns out the system files are themselves 18.5GB so could not download necessary files, when though of attaching volume to the instance again very few volumes were available. This forced me to create another instance next available size(m2.medium) of much bigger size(100 GB) which is not necessary for the assignment, here it takes the flexibility one of the key advantage of cloud computing, by not having enough volume instance available so only necessary volume could be added.
3. The web desktop was laggy for some part though it is not that bad.
4. And on 4th Tuesday, I have received an email stating that my instance was shut down because my Ip was exposed, and DDoS attack has happened, it would be great if we were taught about the preventive measure against the attack, so we can prevent this from happening. As I am new to cloud I am not sure what settings, or what configuration led to such an attack, I did not anything consciously which would jeopardize the instance.

**Since there was not many instance available in the initial stage of assignment(recently 4th Tuesday some extra instances were added) I have executed some of code(part 2 in my friends instance before Tuesday, so there would be some name changes in the instance names).*