Assignment 2

1. Create Instances

Launch two m3.tiny instances using the “E516-Hadoop-Image-V4” image, ensuring that each instance is allocated atleast 30GB of root disk storage.

2. Customize instances

Modify the /etc/hosts file on all instances.

```
127.0.0.1 localhost
10.3.5.210 node-master
10.3.5.165 node-worker1

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
~
```

Modify the /etc/hostname file on all instances. Set "node-master" in the /etc/hostname file of instance 1 and "node-worker1" in the /etc/hostname file of instance 2.

Grant access to your laptop's IP address to monitor HDFS and YARN Resource Manager at http://<node-master>:9870 and http://<node-master>:8088 by running:

$sudo ufw allow from [your-laptop-ip-address]

Allow communication across both the instances

$sudo ufw allow from 10.3.34.0/24

Ensure that workers file in $HADOOP_HOME/etc/hadoop is configured correctly.

2. Start Hadoop Services

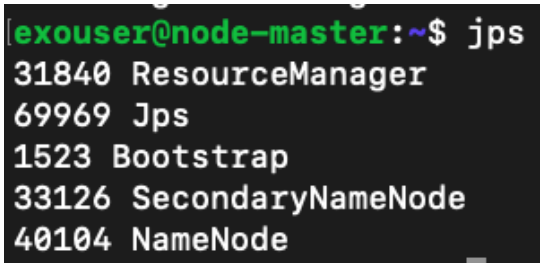
Format the namenode

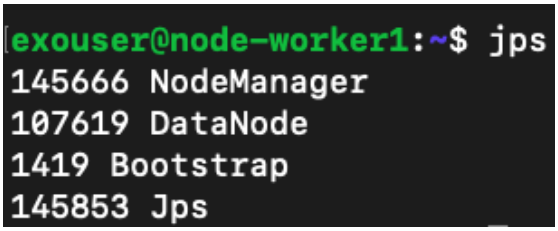
```
2024-11-09 14:35:52,226 INFO common.Storage: Storage directory /home/exouser/hadoop-3.4.0/data/nameNode has been successfully formatted.
```

Start all services on the master node

After starting the services, verify their status by running the jps command on both the master and worker nodes. Ensure that all relevant services (such as NameNode, DataNode, ResourceManager, NodeManager, etc.) are up and running without issues. This step will confirm that the Hadoop ecosystem is functioning properly on both nodes.

```
exouser@node-master:~$ ~/hadoop-3.4.0/sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as exouser in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [node-master]
Starting datanodes
Starting secondary namenodes [node-master]
Starting resourcemanager
Starting nodemanagers
```





3. Check Name Node UI and YARN UI

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).

Heap Memory used 143.85 MB of 251 MB Heap Memory. Max Heap Memory is 1.45 GB.

Non Heap Memory used 57.35 MB of 60.25 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	28.89 GB
Configured Remote Capacity:	0 B
DFS Used:	28 KB (0%)
Non DFS Used:	19.57 GB
DFS Remaining:	9.3 GB (32.2%)
Block Pool Used:	28 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0, In Maintenance: 0)
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)
Decommissioning Nodes	0
Entering Maintenance Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion (including replicas)	0
Block Deletion Start Time	Sat Nov 09 16:30:48 -0500 2024
Last Checkpoint Time	Sat Nov 09 16:30:00 -0500 2024
Last HA Transition Time	Never
Enabled Erasure Coding Policies	RS-6-3-1024k

NameNode Journal Status

Current transaction ID: 1

Journal Manager	State
FileJournalManager(root=/home/exouser/hadoop-3.4.0/data/nameNode)	EditLogFileOutputStream(/home/exouser/hadoop-3.4.0/data/nameNode/current/edits_inprogress_0000000000000000001)

NameNode Storage

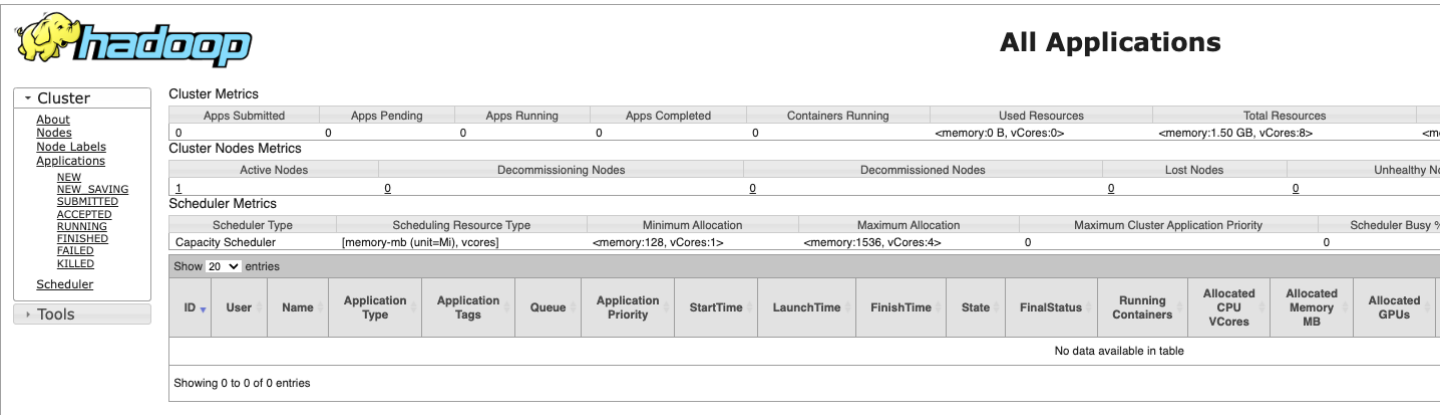
Storage Directory	Type	State
/home/exouser/hadoop-3.4.0/data/nameNode	IMAGE_AND_EDITS	Active

DFS Storage Types

Storage Type	Configured Capacity	Capacity Used	Capacity Remaining	Block Pool Used	Nodes In Service
DISK	28.89 GB	28 KB (0%)	9.3 GB (32.2%)	28 KB	1

Hadoop, 2024.

The NameNode on node-master:9000 is active and running Hadoop version 3.4.0. The cluster has a total configured capacity of 28.89 GB, with 14.25 MB of DFS storage used (0.05%) and 9.87 GB remaining (34.17%). There are 208 files and directories with 158 blocks (all replicated). The system is in normal mode, with no decommissioned or dead nodes.



It provides an overview of cluster metrics, including application states like "Apps Pending," "Apps Running," and "Apps Completed." Additionally, it displays cluster node metrics such as memory and virtual cores used, along with node statuses like "Active Nodes" and "Decommissioning Nodes".

4. Top K IP Addresses

Copy sample log file from local computer to the instance

```
(base) mukundkomati@Mukunds-MacBook-Pro ecc % scp sample.log exouser@149.165.159.91:~/
exouser@149.165.159.91's password:
sample.log                                    100%  100KB   2.9MB/s   00:00
```

Create an input directory on hdfs and upload sample.log to this directory

```
exouser@node-master:~$ ls
Desktop  Documents  Downloads  How2Customize.README  Music  Pictures  Public  Templates  Videos  hadoop-3.4.0  sample.log
exouser@node-master:~$ hadoop fs -mkdir -p ~/hadoop/ipadd_input
exouser@node-master:~$ hadoop fs -ls /home/exouser/hadoop/
Found 1 items
drwxr-xr-x   - exouser supergroup          0 2024-11-09 16:56 /home/exouser/hadoop/ipadd_input
exouser@node-master:~$ hdfs dfs -put sample.log /home/exouser/hadoop/ipadd_input
exouser@node-master:~$ hadoop fs -ls /home/exouser/hadoop/ipadd_input
Found 1 items
-rw-r--r--   1 exouser supergroup     102399 2024-11-09 16:56 /home/exouser/hadoop/ipadd_input/sample.log
exouser@node-master:~$
```

Write the mapper and reducer python functions

```
exouser@node-master:~$ vi topk_mapper.py
exouser@node-master:~$ vi topk_reducer.py
```

```python
#!/usr/bin/env python3
import re
import sys

def map_function():
    """Map function to read log entries from standard input and output (hour, ip) pairs."""
    pattern = re.compile(r'(?P<ip>\d+\.\d+\.\d+\.\d+).*?\d{4}:(?P<hour>\d{2}):\d{2}.*?')

    for line in sys.stdin:
        match = pattern.search(line)
        if match:
            ip = match.group('ip')
            hour = match.group('hour')
            # Output the (hour, ip) pairs
            print(f"{hour}\t{ip}")

if __name__ == "__main__":
    map_function()
~
```

Reducer function takes K as an input

```python
#!/usr/bin/env python3
import sys
from collections import defaultdict, Counter

def reduce_function(K):
    """Reduce function to read (hour, ip) pairs from standard input and find the top K IPs for each hour."""
    hourly_ip_count = defaultdict(Counter)

    for line in sys.stdin:
        hour, ip = line.strip().split('\t')
        # Count occurrences of each IP for each hour
        hourly_ip_count[hour][ip] += 1

    # Output the top K IPs for each hour
    for hour, ip_counts in hourly_ip_count.items():
        top_ips = ip_counts.most_common(K)
        for ip, count in top_ips:
            print(f"{hour}\t{ip}\t{count}")

if __name__ == "__main__":
    # Retrieve the value of K from command-line arguments, defaulting to 5 if not provided
    K = int(sys.argv[1]) if len(sys.argv) > 1 else 5
    reduce_function(K)
~
```

Submit a job for the top 3 IP Addresses per hour

```
exouser@node-master:~$ hadoop jar /home/exouser/hadoop-3.4.0/share/hadoop/tools/lib/hadoop-streaming-3.4.0.jar \
    -input /home/exouser/hadoop/ipadd_input \
    -output /home/exouser/hadoop/ipadd_output \
    -mapper "/usr/bin/python3 topk_mapper.py" \
    -reducer "/usr/bin/python3 topk_reducer.py 2" \
    -file topk_mapper.py \
    -file topk_reducer.py
```

Job status is SUCCEEDED on YARN resource manager Web UI

Application application_1731187870591_0002

User:	exouser
Name:	TopK
Application Type:	MAPREDUCE
Application Tags:	
Application Priority:	0 (Higher Integer value indicates higher priority)
YarnApplicationState:	FINISHED
Queue:	root.default
FinalStatus Reported by AM:	SUCCEEDED
Started:	Sat Nov 09 17:10:22 -0500 2024
Launched:	Sat Nov 09 17:10:23 -0500 2024
Finished:	Sat Nov 09 17:10:47 -0500 2024
Elapsed:	24sec
Tracking URL:	History
Log Aggregation Status:	DISABLED
Application Timeout (Remaining Time):	Unlimited
Diagnostics:	
Unmanaged Application:	false
Application Node Label expression:	<Not set>
AM container Node Label expression:	<DEFAULT_PARTITION>

Print the output for top 3 IP Addresses per hour

```
exouser@node-master:~$ hadoop fs -cat /home/exouser/hadoop/ipadd_output/part-00000
03      66.111.54.249   38
03      5.211.97.39     36
```

Submit a job to output the top 5 IP Addresses per hour

```
exouser@node-master:~$ hadoop jar /home/exouser/hadoop-3.4.0/share/hadoop/tools/lib/hadoop-streaming-3.4.0.jar \
    -Dmapreduce.job.name="TopK" \
    -input /home/exouser/hadoop/ipadd_input \
    -output /home/exouser/hadoop/ipadd_output \
    -mapper "/usr/bin/python3 topk_mapper.py" \
    -reducer "/usr/bin/python3 topk_reducer.py 5" \
    -file topk_mapper.py \
    -file topk_reducer.py
```

Printing the output for top 5 IP Addresses per hour

```
exouser@node-master:~$ hadoop fs -cat /home/exouser/hadoop/ipadd_output/part-00000
03      66.111.54.249   38
03      5.211.97.39     36
03      66.249.66.194   31
03      31.56.96.51     22
03      5.209.200.218   21
```

5. Top K IP Addresses in the given timeperiod

```
exouser@node-master:~$ vi topk_timeperiod_mapper.py
exouser@node-master:~$ vi topk_timeperiod_reducer.py
```

Write the mapper and reducer python functions

Mapper function takes time period as an input

```
#!/usr/bin/env python3
import sys
import re

def map_function(time_period):
    """Map function to read the log file and output (time_period, ip) pairs."""
    # Regular expression to match IP addresses and timestamps
    pattern = re.compile(r'(?P<ip>\d+\.\d+\.\d+\.\d+).*?\d{4}:(?P<hour>\d{2}):\d{2}.*?')

    start_hour, end_hour = map(int, time_period.split('-'))

    for line in sys.stdin:
        match = pattern.search(line)
        if match:
            ip = match.group('ip')
            hour = int(match.group('hour'))

            # Check if the hour is within the specified time period
            if start_hour <= hour < end_hour:
                # Output the (time_period, ip) pair
                print(f"{time_period}\t{ip}")

if __name__ == "__main__":
    # Read time period from command-line arguments
    if len(sys.argv) != 2:
        print("Usage: topk_timeperiod_mapper.py <time_period>")
        sys.exit(1)

    time_period = sys.argv[1]
    map_function(time_period)
~
```

Reducer function takes K as an input

```
#!/usr/bin/env python3
import sys
from collections import defaultdict, Counter

def reduce_function(K=5):
    """Reduce function to read (time_period, ip) pairs from standard input and find the top K IPs."""
    hourly_ip_count = defaultdict(Counter)

    for line in sys.stdin:
        time_period, ip = line.strip().split('\t')
        # Count occurrences of each IP for each time period
        hourly_ip_count[time_period][ip] += 1

    # Output the top K IPs for each time period
    for time_period, ip_counts in hourly_ip_count.items():
        top_ips = ip_counts.most_common(K)
        for ip, count in top_ips:
            print(f"{time_period}\t{ip}")

if __name__ == "__main__":
    K = int(sys.argv[1]) if len(sys.argv) > 1 else 5  # Get K from command-line argument
    reduce_function(K)
~
```

Submit a job for the Top 3 IP addresses in the time period from 00:00 to 04:00 hrs

```
exouser@node-master:~$ hadoop jar /home/exouser/hadoop-3.4.0/share/hadoop/tools/lib/hadoop-streaming-3.4.0.jar \
-Dmapreduce.job.name="TopK_TimePeriod" \
    -input /home/exouser/hadoop/ipadd_input \
    -output /home/exouser/hadoop/ipadd_output \
    -mapper "/usr/bin/python3 topk_timeperiod_mapper.py 0-4" \
    -reducer "/usr/bin/python3 topk_timeperiod_reducer.py 2" \
    -file topk_timeperiod_mapper.py \
    -file topk_timeperiod_reducer.py
```

Job status is succeeded on YARN resource manager Web UI

Application application_1731187870591_0007

User:	exouser
Name:	TopK_TimePeriod
Application Type:	MAPREDUCE
Application Tags:	
Application Priority:	0 (Higher Integer value indicates higher priority)
YarnApplicationState:	FINISHED
Queue:	root.default
FinalStatus Reported by AM:	SUCCEEDED
Started:	Sat Nov 09 17:28:26 -0500 2024
Launched:	Sat Nov 09 17:28:26 -0500 2024
Finished:	Sat Nov 09 17:28:48 -0500 2024
Elapsed:	22sec
Tracking URL:	History
Log Aggregation Status:	DISABLED
Application Timeout (Remaining Time):	Unlimited
Diagnostics:	
Unmanaged Application:	false
Application Node Label expression:	<Not set>
AM container Node Label expression:	<DEFAULT_PARTITION>

Print the output for top 2 IP Addresses in the time period from 00:00 to 04:00 hrs

```
exouser@node-master:~$ hadoop fs -cat /home/exouser/hadoop/ipadd_output/part-00000
0-4     66.111.54.249
0-4     5.211.97.39
```

Submit a job to output the top 2 IP Addresses in the time period from 00:00 to 01:00 hrs

```
exouser@node-master:~$ hadoop jar /home/exouser/hadoop-3.4.0/share/hadoop/tools/lib/hadoop-streaming-3.4.0.jar \
-Dmapreduce.job.name="TopK_TimePeriod" \
    -input /home/exouser/hadoop/ipadd_input \
    -output /home/exouser/hadoop/ipadd_output \
    -mapper "/usr/bin/python3 topk_timeperiod_mapper.py 0-1" \
    -reducer "/usr/bin/python3 topk_timeperiod_reducer.py 2" \
    -file topk_timeperiod_mapper.py \
    -file topk_timeperiod_reducer.py
```

Job status is succeeded on YARN resource manager Web UI

Application application_1731187870591_0008

User:	exouser
Name:	TopK_TimePeriod
Application Type:	MAPREDUCE
Application Tags:	
Application Priority:	0 (Higher Integer value indicates higher priority)
YarnApplicationState:	FINISHED
Queue:	root.default
FinalStatus Reported by AM:	SUCCEEDED
Started:	Sat Nov 09 17:31:23 -0500 2024
Launched:	Sat Nov 09 17:31:24 -0500 2024
Finished:	Sat Nov 09 17:31:44 -0500 2024
Elapsed:	20sec
Tracking URL:	History
Log Aggregation Status:	DISABLED
Application Timeout (Remaining Time):	Unlimited
Diagnostics:	
Unmanaged Application:	false
Application Node Label expression:	<Not set>
AM container Node Label expression:	<DEFAULT_PARTITION>

Printing the output for top 2 IP Addresses in the time period from 00:00 to 01:00 hrs (No visits in the specified timeperiod in sample.log)

```
exouser@node-master:~$ hadoop fs -cat /home/exouser/hadoop/ipadd_output/part-00000
exouser@node-master:~$
```

6. Capacity Scheduler

6.1 yarn-site.xml

```
exouser@node-master:~$ vi ~/hadoop-3.4.0/etc/hadoop/yarn-site.xml
```

Insert the following XML snippet into the yarn-site.xml file to configure YARN to use the Capacity Scheduler

```
  <!-- Capacity Scheduler -->
    <property>
            <name>yarn.resourcemanager.scheduler.class</name>
            <value>org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityScheduler</value>
    </property>
</configuration>
```

```
exouser@node-master:~$ vi ~/hadoop-3.4.0/etc/hadoop/capacity-scheduler.xml
```

6.2 Capacity-scheduler.xml

Define 2 queues at the root level, each queue is allocated 50% of the clusters resources

```
    </description>
        </property>
        <property>
                <name>yarn.scheduler.capacity.root.queues</name>
                <value>queue1,queue2</value>
                <description>The queues at the root level.</description>
        </property>
        <property>
                <name>yarn.scheduler.capacity.root.queue1.capacity</name>
                <value>50</value>
                <description>Queue1 target capacity.</description>
        </property>
```

```
        <property>
                <name>yarn.scheduler.capacity.root.queue2.capacity</name>
                <value>50</value>
                <description>Queue2 target capacity.</description>
        </property>
        <property>
```

No queue-mapping required, as the user exouser needs to submit jobs to both the queues

```
        </property>
        <property>
                <name>yarn.scheduler.capacity.queue-mappings</name>
                <value></value>
        </property>
        <property>
                <name>yarn.scheduler.capacity.queue-mappings-override.enable</name>
                <value>false</value>
                <description>
      If a queue mapping is present, will it override the value specified
      by the user? This can be used by administrators to place jobs in queues
      that are different than the one specified by the user.
      The default is false.
```

6.3 Create input files for WordCount, Sort and Grep jobs

```
exouser@node-master:~$ vi sort_input.log
exouser@node-master:~$ vi word_count_input.log
exouser@node-master:~$ vi grep_input.log
```

grep_input.log

```
Hadoop is a framework for distributed storage and processing.
MapReduce is a programming model for processing large data sets.
Hadoop Streaming allows us to use any programming language for mapper and reducer.
Python is commonly used with Hadoop for data processing.
Hadoop can handle large amounts of data efficiently.
~
```

sort_input.log

```
banana
apple
cherry
banana
apple
date
cherry
fig
banana
grape
~
```

word_count.log

```
hello
world
hello
hadoop
world
hello
mapreduce
mapreduce
hadoop
~
```

Upload the input files to HDFS

```
exouser@node-master:~$ hadoop fs -mkdir -p ~/hadoop/sort_input
exouser@node-master:~$ hadoop fs -mkdir -p ~/hadoop/word_count_input
exouser@node-master:~$ hadoop fs -mkdir -p ~/hadoop/grep_input
exouser@node-master:~$ hadoop fs -ls /home/exouser/hadoop/
Found 5 items
drwxr-xr-x   - exouser supergroup          0 2024-11-09 17:50 /home/exouser/hadoop/grep_input
-rw-r--r--   1 exouser supergroup     102399 2024-11-09 17:15 /home/exouser/hadoop/ipadd_input
drwxr-xr-x   - exouser supergroup          0 2024-11-09 17:31 /home/exouser/hadoop/ipadd_output
drwxr-xr-x   - exouser supergroup          0 2024-11-09 17:50 /home/exouser/hadoop/sort_input
drwxr-xr-x   - exouser supergroup          0 2024-11-09 17:50 /home/exouser/hadoop/word_count_input
exouser@node-master:~$ hdfs dfs -put sort_input.log /home/exouser/hadoop/sort_input
exouser@node-master:~$ hdfs dfs -put word_count_input.log /home/exouser/hadoop/word_count_input
exouser@node-master:~$ hdfs dfs -put grep_input.log /home/exouser/hadoop/grep_input
```

6.4 Write the map-reduce functions

Sort Functions

```
#!/usr/bin/env python3
import sys

def sort_mapper():
    """Mapper function for Sort."""
    for line in sys.stdin:
        # Strip any trailing whitespace and emit each word as the key with a placeholder value
        word = line.strip()
        print(f"{word}\t1")

if __name__ == "__main__":
    sort_mapper()
~
```

```
#!/usr/bin/env python3
import sys

def sort_reducer():
    """Reducer function for Sort."""
    for line in sys.stdin:
        # Output each sorted word as is
        word, _ = line.strip().split('\t')
        print(word)

if __name__ == "__main__":
    sort_reducer()
~
```

Word Count Functions

```
#!/usr/bin/env python3
import sys

def wordcount_mapper():
    """Mapper function for WordCount."""
    for line in sys.stdin:
        # Strip whitespace and split the line into words
        words = line.strip().split()
        # Emit each word with a count of 1
        for word in words:
            print(f"{word}\t1")

if __name__ == "__main__":
    wordcount_mapper()
~
```

```
#!/usr/bin/env python3
import sys

def wordcount_reducer():
    """Reducer function for WordCount."""
    current_word = None
    current_count = 0

    for line in sys.stdin:
        # Split the input line into word and count
        word, count = line.strip().split('\t')
        count = int(count)

        # If the word changes (new word), print the count for the previous word
        if current_word == word:
            current_count += count
        else:
            if current_word is not None:
                print(f"{current_word}\t{current_count}")
            current_word = word
            current_count = count

    # Print the last word count
    if current_word is not None:
        print(f"{current_word}\t{current_count}")

if __name__ == "__main__":
    wordcount_reducer()
```

Grep Functions

```python
#!/usr/bin/env python3
import sys
import re

def grep_mapper(pattern):
    """Mapper function for Grep. Outputs lines containing the specified pattern."""
    regex = re.compile(pattern)

    for line in sys.stdin:
        line = line.strip()
        if regex.search(line):  # Check if the line contains the pattern
            print(line)  # Output the matching line

if __name__ == "__main__":
    # Example pattern; change this to any desired pattern
    pattern = sys.argv[1] if len(sys.argv) > 1 else "Hadoop"
    grep_mapper(pattern)
~
```

```python
#!/usr/bin/env python3
import sys

def grep_reducer():
    """Reducer function for Grep. Forwards each line it receives."""
    for line in sys.stdin:
        print(line.strip())  # Output each line as is

if __name__ == "__main__":
    grep_reducer()
~
```

6.5 Run Jobs parallelly using capacity scheduler

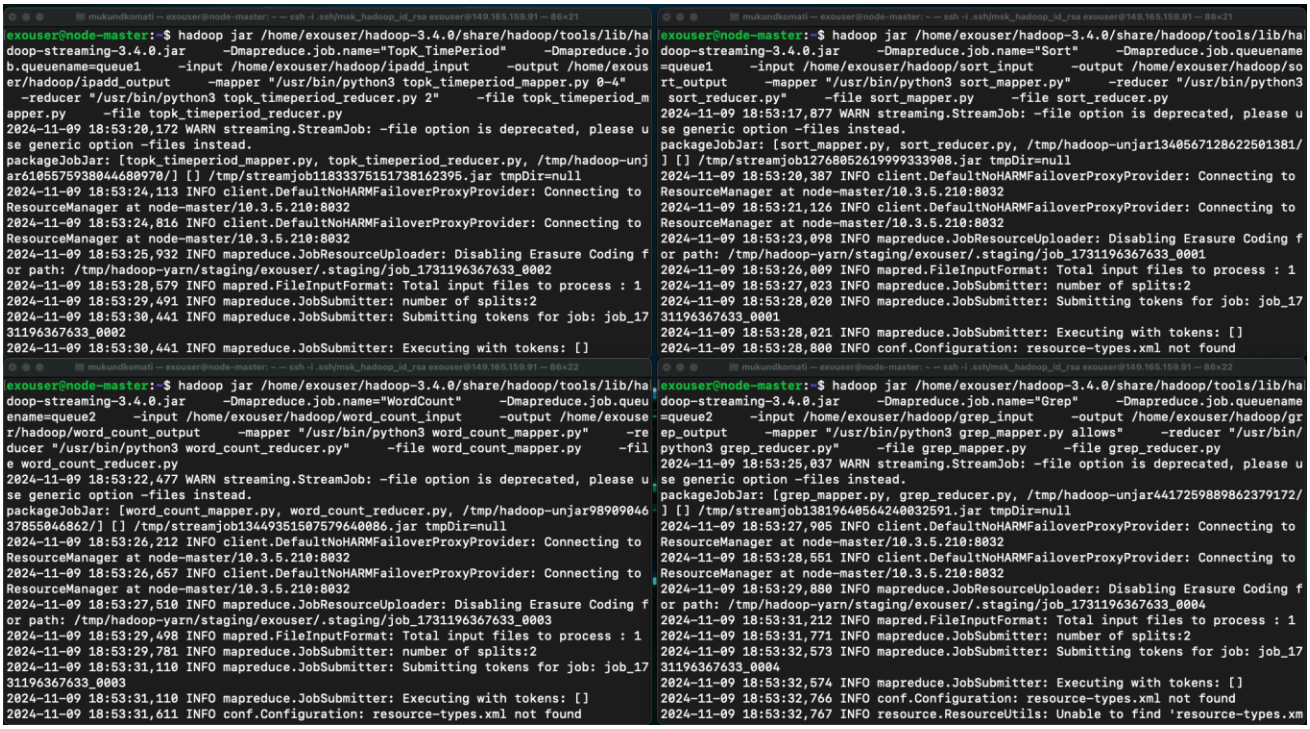
Restart all services and refresh queues

```
exouser@node-master:~$ ~/hadoop-3.4.0/sbin/stop-all.sh
```

```
exouser@node-master:~$ ~/hadoop-3.4.0/sbin/start-all.sh
```

```
exouser@node-master:~$ yarn rmadmin -refreshQueues
2024-11-09 18:39:45,698 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to Resou
rceManager at node-master/10.3.5.210:8033
```

Run 4 jobs at once (Topk Period, Wordcount, Sort and Grep)



```
exouser@node-master:~$ hadoop jar /home/exouser/hadoop-3.4.0/share/hadoop/tools/lib/hadoop-streaming-3.4.0.jar -Dmapreduce.job.name="TopK_TimePeriod" -Dmapreduce.job.queuename=queue1 -input /home/exouser/hadoop/ipadd_input -output /home/exouser/hadoop/ipadd_output -mapper "/usr/bin/python3 topk_timeperiod_mapper.py 0-4" -reducer "/usr/bin/python3 topk_timeperiod_reducer.py 2" -file topk_timeperiod_mapper.py -file topk_timeperiod_reducer.py
2024-11-09 18:53:20,172 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [topk_timeperiod_mapper.py, topk_timeperiod_reducer.py, /tmp/hadoop-unjar6105575938044680970/] [] /tmp/streamjob11833375151738162395.jar tmpDir=null
2024-11-09 18:53:24,113 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at node-master/10.3.5.210:8032
2024-11-09 18:53:24,816 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at node-master/10.3.5.210:8032
2024-11-09 18:53:25,932 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/exouser/.staging/job_1731196367633_0002
2024-11-09 18:53:28,579 INFO mapred.FileInputFormat: Total input files to process : 1
2024-11-09 18:53:29,491 INFO mapreduce.JobSubmitter: number of splits:2
2024-11-09 18:53:30,441 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1731196367633_0002
2024-11-09 18:53:30,441 INFO mapreduce.JobSubmitter: Executing with tokens: []
```

```
exouser@node-master:~$ hadoop jar /home/exouser/hadoop-3.4.0/share/hadoop/tools/lib/hadoop-streaming-3.4.0.jar -Dmapreduce.job.name="Sort" -Dmapreduce.job.queuename=queue1 -input /home/exouser/hadoop/sort_input -output /home/exouser/hadoop/sort_output -mapper "/usr/bin/python3 sort_mapper.py" -reducer "/usr/bin/python3 sort_reducer.py" -file sort_mapper.py -file sort_reducer.py
2024-11-09 18:53:17,877 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [sort_mapper.py, sort_reducer.py, /tmp/hadoop-unjar1340567128622501381/] [] /tmp/streamjob12768052619999333908.jar tmpDir=null
2024-11-09 18:53:20,387 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at node-master/10.3.5.210:8032
2024-11-09 18:53:21,126 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at node-master/10.3.5.210:8032
2024-11-09 18:53:23,098 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/exouser/.staging/job_1731196367633_0001
2024-11-09 18:53:26,009 INFO mapred.FileInputFormat: Total input files to process : 1
2024-11-09 18:53:27,023 INFO mapreduce.JobSubmitter: number of splits:2
2024-11-09 18:53:28,020 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1731196367633_0001
2024-11-09 18:53:28,021 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-11-09 18:53:28,800 INFO conf.Configuration: resource-types.xml not found
```

```
exouser@node-master:~$ hadoop jar /home/exouser/hadoop-3.4.0/share/hadoop/tools/lib/hadoop-streaming-3.4.0.jar -Dmapreduce.job.name="WordCount" -Dmapreduce.job.queuename=queue2 -input /home/exouser/hadoop/word_count_input -output /home/exouser/hadoop/word_count_output -mapper "/usr/bin/python3 word_count_mapper.py" -reducer "/usr/bin/python3 word_count_reducer.py" -file word_count_mapper.py -file word_count_reducer.py
2024-11-09 18:53:22,477 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [word_count_mapper.py, word_count_reducer.py, /tmp/hadoop-unjar9890904637855046862/] [] /tmp/streamjob13449351507579640086.jar tmpDir=null
2024-11-09 18:53:26,212 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at node-master/10.3.5.210:8032
2024-11-09 18:53:26,657 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at node-master/10.3.5.210:8032
2024-11-09 18:53:27,510 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/exouser/.staging/job_1731196367633_0003
2024-11-09 18:53:29,498 INFO mapred.FileInputFormat: Total input files to process : 1
2024-11-09 18:53:29,781 INFO mapreduce.JobSubmitter: number of splits:2
2024-11-09 18:53:31,110 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1731196367633_0003
2024-11-09 18:53:31,110 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-11-09 18:53:31,611 INFO conf.Configuration: resource-types.xml not found
```

```
exouser@node-master:~$ hadoop jar /home/exouser/hadoop-3.4.0/share/hadoop/tools/lib/hadoop-streaming-3.4.0.jar -Dmapreduce.job.name="Grep" -Dmapreduce.job.queuename=queue2 -input /home/exouser/hadoop/grep_input -output /home/exouser/hadoop/grep_output -mapper "/usr/bin/python3 grep_mapper.py allows" -reducer "/usr/bin/python3 grep_reducer.py" -file grep_mapper.py -file grep_reducer.py
2024-11-09 18:53:25,037 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [grep_mapper.py, grep_reducer.py, /tmp/hadoop-unjar4417259889862379172/] [] /tmp/streamjob13819640564240032591.jar tmpDir=null
2024-11-09 18:53:27,905 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at node-master/10.3.5.210:8032
2024-11-09 18:53:28,551 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at node-master/10.3.5.210:8032
2024-11-09 18:53:29,880 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/exouser/.staging/job_1731196367633_0004
2024-11-09 18:53:31,212 INFO mapred.FileInputFormat: Total input files to process : 1
2024-11-09 18:53:31,771 INFO mapreduce.JobSubmitter: number of splits:2
2024-11-09 18:53:32,573 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1731196367633_0004
2024-11-09 18:53:32,574 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-11-09 18:53:32,766 INFO conf.Configuration: resource-types.xml not found
2024-11-09 18:53:32,767 INFO resource.ResourceUtils: Unable to find 'resource-types.xm
```

There are two queues that run concurrently in a FIFO order. Each queue processes jobs sequentially, with the next job only starting after the current job finishes.

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority	Scheduler Busy %	RM Dispatcher EventQueue Size
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:128, vCores:1>	<memory:1536, vCores:4>	0	0	0

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Allocated GPUs	Reserved CPU VCores	Reserved Memory MB	Reserved GPUs
application_1731196367633_0004	exouser	Grep	MAPREDUCE		root.queue2	0	Sat Nov 9 18:53:32 -0500 2024	N/A	N/A	ACCEPTED	UNDEFINED	0	0	0	-1	0	0	-1
application_1731196367633_0003	exouser	WordCount	MAPREDUCE		root.queue2	0	Sat Nov 9 18:53:31 -0500 2024	Sat Nov 9 18:53:32 -0500 2024	Sat Nov 9 18:54:09 -0500 2024	FINISHED	SUCCEEDED	1	1	512	N/A	0	0	N/A
application_1731196367633_0002	exouser	TopK_TimePeriod	MAPREDUCE		root.queue1	0	Sat Nov 9 18:53:30 -0500 2024	N/A	N/A	ACCEPTED	UNDEFINED	0	0	0	-1	0	0	-1
application_1731196367633_0001	exouser	Sort	MAPREDUCE		root.queue1	0	Sat Nov 9 18:53:29 -0500 2024	Sat Nov 9 18:53:32 -0500 2024	Sat Nov 9 18:54:09 -0500 2024	FINISHED	SUCCEEDED	1	1	512	N/A	0	0	N/A

Showing 1 to 4 of 4 entries

All jobs have reached FINISHED state

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Allocated GPUs	Reserved CPU VCores	Reserved Memory MB	Reserved GPUs
application_1731196367633_0004	exouser	Grep	MAPREDUCE		root.queue2	0	Sat Nov 9 18:53:32 -0500 2024	Sat Nov 9 18:54:15 -0500 2024	Sat Nov 9 18:54:50 -0500 2024	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	N/A	N/A
application_1731196367633_0003	exouser	WordCount	MAPREDUCE		root.queue2	0	Sat Nov 9 18:53:31 -0500 2024	Sat Nov 9 18:53:32 -0500 2024	Sat Nov 9 18:54:09 -0500 2024	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	N/A	N/A
application_1731196367633_0002	exouser	TopK_TimePeriod	MAPREDUCE		root.queue1	0	Sat Nov 9 18:53:30 -0500 2024	Sat Nov 9 18:54:16 -0500 2024	Sat Nov 9 18:54:50 -0500 2024	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	N/A	N/A
application_1731196367633_0001	exouser	Sort	MAPREDUCE		root.queue1	0	Sat Nov 9 18:53:29 -0500 2024	Sat Nov 9 18:53:32 -0500 2024	Sat Nov 9 18:54:09 -0500 2024	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Showing 1 to 4 of 4 entries

7. Fair Scheduler

7.1 yarn-site.xml

```
exouser@node-master:~$ vi ~/hadoop-3.4.0/etc/hadoop/yarn-site.xml
```

Change YARN configuration to use the Fair Scheduler

```
    <property>
            <name>yarn.resourcemanager.scheduler.class</name>
            <value>org.apache.hadoop.yarn.server.resourcemanager.scheduler.fair.FairScheduler</value>
      </property>
</configuration>
```

Configure the maximum allocation for vCores and Resource Manager memory to match the instance's available resources.

```
    <property>
            <name>yarn.scheduler.maximum-allocation-vcores</name>
            <value>1</value>
    </property>
```

```
    <property>
        <name>yarn.nodemanager.resource.memory-mb</name>
        <value>750</value>
    </property>

    <property>
        <name>yarn.scheduler.maximum-allocation-mb</name>
        <value>750</value>
    </property>
```

7.2 fair-scheduler.xml

```
exouser@node-master:~$ vi ~/hadoop-3.4.0/etc/hadoop/fair-scheduler.xml
```

Create a single queue for the Fair Scheduler Configuration

```
<allocations>
        <!-- Define the default queue -->
        <queue name="default">
                <weight>1.0</weight>
                <schedulingPolicy>fair</schedulingPolicy>
        </queue>
</allocations>
~
```

7.3 Run Jobs using fair scheduler

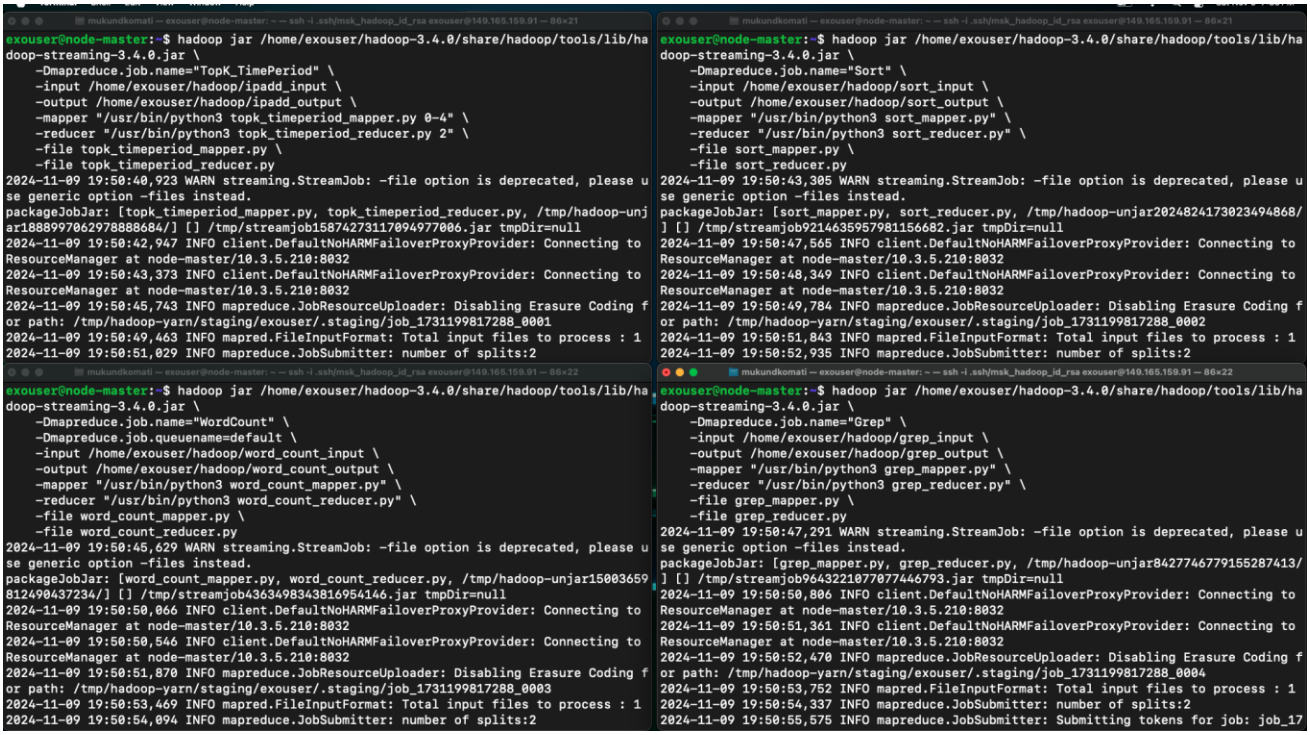
Restart all services and refresh queues

```
exouser@node-master:~$ ~/hadoop-3.4.0/sbin/stop-all.sh
```

```
exouser@node-master:~$ ~/hadoop-3.4.0/sbin/start-all.sh
```

```
exouser@node-master:~$ yarn rmadmin -refreshQueues
2024-11-09 18:39:45,698 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to Resou
rceManager at node-master/10.3.5.210:8033
```

Run 4 jobs at once (Topk Period, Wordcount, Sort and Grep)



```
exouser@node-master:~$ hadoop jar /home/exouser/hadoop-3.4.0/share/hadoop/tools/lib/ha
doop-streaming-3.4.0.jar \
    -Dmapreduce.job.name="TopK_TimePeriod" \
    -input /home/exouser/hadoop/ipadd_input \
    -output /home/exouser/hadoop/ipadd_output \
    -mapper "/usr/bin/python3 topk_timeperiod_mapper.py 0-4" \
    -reducer "/usr/bin/python3 topk_timeperiod_reducer.py 2" \
    -file topk_timeperiod_mapper.py \
    -file topk_timeperiod_reducer.py
2024-11-09 19:50:40,923 WARN streaming.StreamJob: -file option is deprecated, please u
se generic option -files instead.
packageJobJar: [topk_timeperiod_mapper.py, topk_timeperiod_reducer.py, /tmp/hadoop-unj
ar1888997062978888684/] [] /tmp/streamjob1587427311709497 7006.jar tmpDir=null
2024-11-09 19:50:42,947 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to
ResourceManager at node-master/10.3.5.210:8032
2024-11-09 19:50:43,373 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to
ResourceManager at node-master/10.3.5.210:8032
2024-11-09 19:50:45,743 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding f
or path: /tmp/hadoop-yarn/staging/exouser/.staging/job_1731199817288_0001
2024-11-09 19:50:49,463 INFO mapred.FileInputFormat: Total input files to process : 1
2024-11-09 19:50:51,029 INFO mapreduce.JobSubmitter: number of splits:2
```

```
exouser@node-master:~$ hadoop jar /home/exouser/hadoop-3.4.0/share/hadoop/tools/lib/ha
doop-streaming-3.4.0.jar \
    -Dmapreduce.job.name="Sort" \
    -input /home/exouser/hadoop/sort_input \
    -output /home/exouser/hadoop/sort_output \
    -mapper "/usr/bin/python3 sort_mapper.py" \
    -reducer "/usr/bin/python3 sort_reducer.py" \
    -file sort_mapper.py \
    -file sort_reducer.py
2024-11-09 19:50:43,305 WARN streaming.StreamJob: -file option is deprecated, please u
se generic option -files instead.
packageJobJar: [sort_mapper.py, sort_reducer.py, /tmp/hadoop-unjar2024824173023494868/
] [] /tmp/streamjob9214635957981156682.jar tmpDir=null
2024-11-09 19:50:47,565 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to
ResourceManager at node-master/10.3.5.210:8032
2024-11-09 19:50:48,349 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to
ResourceManager at node-master/10.3.5.210:8032
2024-11-09 19:50:49,784 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding f
or path: /tmp/hadoop-yarn/staging/exouser/.staging/job_1731199817288_0002
2024-11-09 19:50:51,843 INFO mapred.FileInputFormat: Total input files to process : 1
2024-11-09 19:50:52,935 INFO mapreduce.JobSubmitter: number of splits:2
```

```
exouser@node-master:~$ hadoop jar /home/exouser/hadoop-3.4.0/share/hadoop/tools/lib/ha
doop-streaming-3.4.0.jar \
    -Dmapreduce.job.name="WordCount" \
    -Dmapreduce.job.queuename=default \
    -input /home/exouser/hadoop/word_count_input \
    -output /home/exouser/hadoop/word_count_output \
    -mapper "/usr/bin/python3 word_count_mapper.py" \
    -reducer "/usr/bin/python3 word_count_reducer.py" \
    -file word_count_mapper.py \
    -file word_count_reducer.py
2024-11-09 19:50:45,629 WARN streaming.StreamJob: -file option is deprecated, please u
se generic option -files instead.
packageJobJar: [word_count_mapper.py, word_count_reducer.py, /tmp/hadoop-unjar15003659
812490437234/] [] /tmp/streamjob4363498343816954146.jar tmpDir=null
2024-11-09 19:50:50,066 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to
ResourceManager at node-master/10.3.5.210:8032
2024-11-09 19:50:50,546 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to
ResourceManager at node-master/10.3.5.210:8032
2024-11-09 19:50:51,870 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding f
or path: /tmp/hadoop-yarn/staging/exouser/.staging/job_1731199817288_0003
2024-11-09 19:50:53,469 INFO mapred.FileInputFormat: Total input files to process : 1
2024-11-09 19:50:54,094 INFO mapreduce.JobSubmitter: number of splits:2
```

```
exouser@node-master:~$ hadoop jar /home/exouser/hadoop-3.4.0/share/hadoop/tools/lib/ha
doop-streaming-3.4.0.jar \
    -Dmapreduce.job.name="Grep" \
    -input /home/exouser/hadoop/grep_input \
    -output /home/exouser/hadoop/grep_output \
    -mapper "/usr/bin/python3 grep_mapper.py" \
    -reducer "/usr/bin/python3 grep_reducer.py" \
    -file grep_mapper.py \
    -file grep_reducer.py
2024-11-09 19:50:47,291 WARN streaming.StreamJob: -file option is deprecated, please u
se generic option -files instead.
packageJobJar: [grep_mapper.py, grep_reducer.py, /tmp/hadoop-unjar8427746779155287413/
] [] /tmp/streamjob9643221077077446793.jar tmpDir=null
2024-11-09 19:50:50,806 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to
ResourceManager at node-master/10.3.5.210:8032
2024-11-09 19:50:51,361 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to
ResourceManager at node-master/10.3.5.210:8032
2024-11-09 19:50:52,470 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding f
or path: /tmp/hadoop-yarn/staging/exouser/.staging/job_1731199817288_0004
2024-11-09 19:50:53,752 INFO mapred.FileInputFormat: Total input files to process : 1
2024-11-09 19:50:54,337 INFO mapreduce.JobSubmitter: number of splits:2
2024-11-09 19:50:55,575 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_17
```

In summary, with one CPU core VM on Jetstream, true parallelism is not achievable. The Fair Scheduler will instead manage jobs sequentially, distributing CPU time among them fairly. This will give each job a chance to progress. The Table below shows all the jobs have run successfully

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority	Scheduler Busy %	RM Dispatcher EventQueue Size
Fair Scheduler	[memory-mb (unit=Mi), vcores]	<memory:128, vCores:1>	<memory:750, vCores:1>	0	0	0

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Allocated GPUs	Reserved CPU VCores	Reserved Memory MB	Reserved GPUs
application_1731199817288_0004	exouser	Grep	MAPREDUCE		root.exouser	0	Sat Nov 9 19:50:56 -0500 2024	Sat Nov 9 19:52:33 -0500 2024	Sat Nov 9 19:52:55 -0500 2024	FINISHED	SUCCEEDED	1	1	750	N/A	0	0	N/A
application_1731199817288_0003	exouser	WordCount	MAPREDUCE		root.exouser	0	Sat Nov 9 19:50:55 -0500 2024	Sat Nov 9 19:52:03 -0500 2024	Sat Nov 9 19:52:25 -0500 2024	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	N/A	N/A
application_1731199817288_0002	exouser	Sort	MAPREDUCE		root.exouser	0	Sat Nov 9 19:50:54 -0500 2024	Sat Nov 9 19:51:31 -0500 2024	Sat Nov 9 19:51:55 -0500 2024	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	N/A	N/A
application_1731199817288_0001	exouser	TopK_TimePeriod	MAPREDUCE		root.exouser	0	Sat Nov 9 19:50:53 -0500 2024	Sat Nov 9 19:50:56 -0500 2024	Sat Nov 9 19:51:23 -0500 2024	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Showing 1 to 4 of 4 entries

8. Outputs of the 4 jobs

Top K Job Output

```
exouser@node-master:~$ hadoop fs -cat /home/exouser/hadoop/ipadd_output/part-00000
0-4     66.111.54.249
0-4     5.211.97.39
```

Sort Job Output

```
exouser@node-master:~$ hadoop fs -cat /home/exouser/hadoop/sort_output/part-00000
apple
apple
banana
banana
banana
cherry
cherry
date
fig
grape
```

Word Count Output

```
exouser@node-master:~$ hadoop fs -cat /home/exouser/hadoop/word_count_output/part-00000
hadoop  2
hello   3
mapreduce       2
world   2
```

Grep Ouput (returns all lines that contain "Hadoop")

```
exouser@node-master:~$ hadoop fs -cat /home/exouser/hadoop/grep_output/part-00000
Hadoop Streaming allows us to use any programming language for mapper and reducer.
Hadoop can handle large amounts of data efficiently.
Hadoop is a framework for distributed storage and processing.
Python is commonly used with Hadoop for data processing.
```