# EX 2 IMPLEMENT WORD COUNT PROGRAMS USING MAPREDUCE

## Aim:

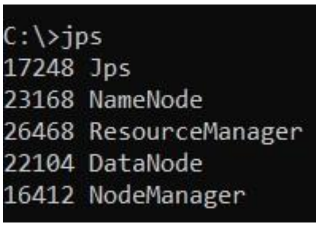
To implement word count/frequency program using mapReduce in Hadoop.

## Procedure:

1. Start the Hadoop namenode and datanode using the command **start-dfs.cmd**

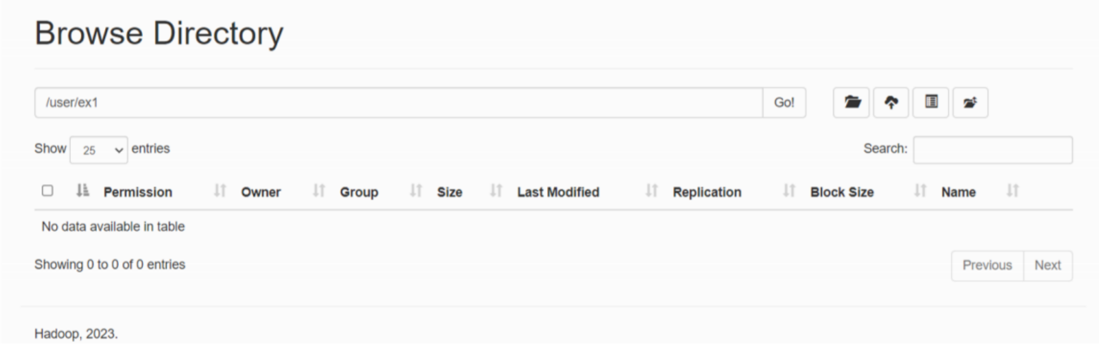
   **start-yarn.cmd**

   Check if namenode and datanode are running using the comman **jps**



```
C:\>jps
17248 Jps
23168 NameNode
26468 ResourceManager
22104 DataNode
16412 NodeManager
```

2. Create a directory in the Hadoop filesystem using the command **hadoop fs -mkdir /user/ex1**



Browse Directory

/user/ex1

Show 25 entries Search:

| ☐ | Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|---|---|---|---|---|---|---|---|---|
| No data available in table | | | | | | | | |

Showing 0 to 0 of 0 entries

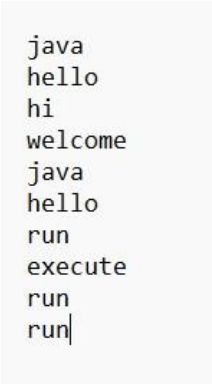
Hadoop, 2023.

Empty directory is created.

3. Insert the input file into the directory using the command **hadoop fs -put**

   **C:\Users\jawah\OneDrive\Desktop\LathikaDA\input.txt /user/ex1**

*//input.txt*



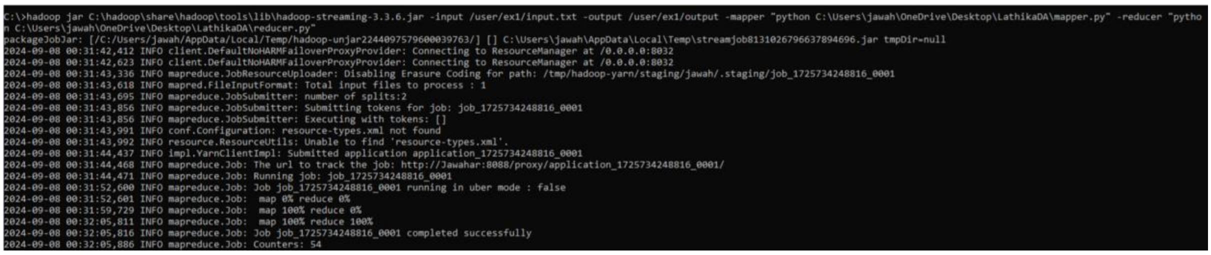
```
java
hello
hi
welcome
java
hello
run
execute
run
run
```

4. The MapReduce Program is written to count the frequency of word in the input file.

```
//mapper.py
#!/usr/bin/env python
import sys

# Input comes from STDIN (standard input) for
line in sys.stdin:

    # Remove leading and trailing whitespace
line = line.strip()     # Split the line into
words     words = line.split()

    # Output each word with a count of 1
for word in words:
print(f'{word}\t1') //reducer.py
#!/usr/bin/env python import sys
current_word = None current_count = 0
word = None

for line in sys.stdin:     line =
line.strip()     word, count =
line.split('\t', 1)     try:

        count      =      int(count)

except                ValueError:

continue      if current_word

== word:          current_count

+= count


    else:         if
current_word:

            print(f'{current_word}\t{current_count}')
        current_count = count
current_word = word if
current_word == word:

    print(f'{current_word}\t{current_count}')
```

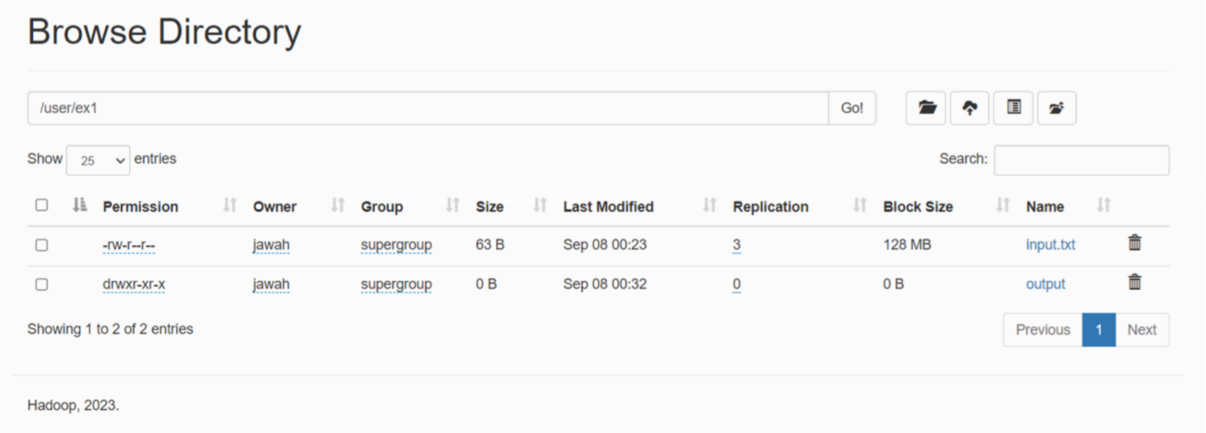
5. The mapper reducer program is executed by the following command

**hadoop jar C:\hadoop\share\hadoop\tools\lib\hadoop-streaming-3.3.6.jar -input /user/ex1/input.txt -output /user/ex1/output -mapper "python C:\Users\jawah\OneDrive\Desktop\LathikaDA\mapper.py" -reducer "python C:\Users\jawah\OneDrive\Desktop\LathikaDA\reducer.py"**



```
C:\>hadoop jar C:\hadoop\share\hadoop\tools\lib\hadoop-streaming-3.3.6.jar -input /user/ex1/input.txt -output /user/ex1/output -mapper "python C:\Users\jawah\OneDrive\Desktop\LathikaDA\mapper.py" -reducer "pytho
n C:\Users\jawah\OneDrive\Desktop\LathikaDA\reducer.py"
packageJobJar: [/C:/Users/jawah/AppData/Local/Temp/hadoop-unjar2244097579600039763/] [] C:\Users\jawah\AppData\Local\Temp\streamjob8131026796637894696.jar tmpDir=null
2024-09-08 00:31:42,412 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-09-08 00:31:42,623 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-09-08 00:31:43,336 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/jawah/.staging/job_1725734248816_0001
2024-09-08 00:31:43,618 INFO mapred.FileInputFormat: Total input files to process : 1
2024-09-08 00:31:43,695 INFO mapreduce.JobSubmitter: number of splits:2
2024-09-08 00:31:43,856 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1725734248816_0001
2024-09-08 00:31:43,856 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-09-08 00:31:43,991 INFO conf.Configuration: resource-types.xml not found
2024-09-08 00:31:43,992 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-09-08 00:31:44,437 INFO impl.YarnClientImpl: Submitted application application_1725734248816_0001
2024-09-08 00:31:44,468 INFO mapreduce.Job: The url to track the job: http://Jawahar:8088/proxy/application_1725734248816_0001/
2024-09-08 00:31:44,471 INFO mapreduce.Job: Running job: job_1725734248816_0001
2024-09-08 00:31:52,600 INFO mapreduce.Job: Job job_1725734248816_0001 running in uber mode : false
2024-09-08 00:31:52,601 INFO mapreduce.Job:  map 0% reduce 0%
2024-09-08 00:31:59,729 INFO mapreduce.Job:  map 100% reduce 0%
2024-09-08 00:32:05,811 INFO mapreduce.Job:  map 100% reduce 100%
2024-09-08 00:32:05,816 INFO mapreduce.Job: Job job_1725734248816_0001 completed successfully
2024-09-08 00:32:05,886 INFO mapreduce.Job: Counters: 54
```

Thus the output directory is created.



Browse Directory

/user/ex1

Show 25 entries

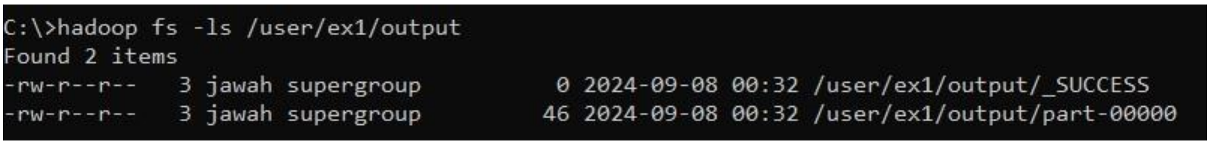
Search:

| □ | Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |  |
|---|---|---|---|---|---|---|---|---|---|
| □ | -rw-r--r-- | jawah | supergroup | 63 B | Sep 08 00:23 | 3 | 128 MB | input.txt | 🗑 |
| □ | drwxr-xr-x | jawah | supergroup | 0 B | Sep 08 00:32 | 0 | 0 B | output | 🗑 |

Showing 1 to 2 of 2 entries

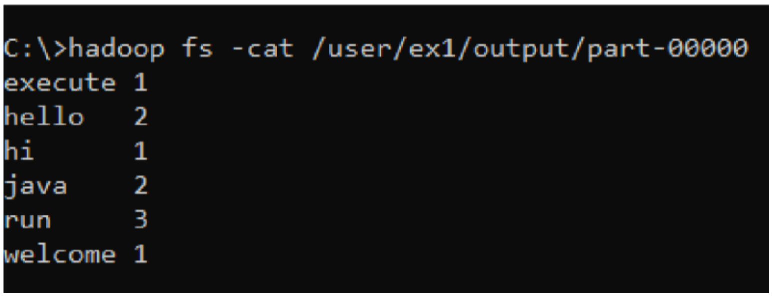
Hadoop, 2023.

6. To view the output files



```
C:\>hadoop fs -ls /user/ex1/output
Found 2 items
-rw-r--r--   3 jawah supergroup          0 2024-09-08 00:32 /user/ex1/output/_SUCCESS
-rw-r--r--   3 jawah supergroup         46 2024-09-08 00:32 /user/ex1/output/part-00000
```

**hadoop fs -cat /user/ex1/output/part-00000**



```
C:\>hadoop fs -cat /user/ex1/output/part-00000
execute 1
hello   2
hi      1
java    2
run     3
welcome 1
```

7. Stop the Hadoop namenode and datanode **stop-all.cmd** **Result:**

Thus the MapReduce Program to find the word count of a input file is completed successfully