**Ex no: 2**

# Implement word count/frequency programs using MapReduce

## Aim:

To implement Word count program to count the number of words in a text file in Python using MapReduce.

## Procedure:

1. Open Command prompt and run as administrator and start the Hadoop service.
2. Check if the namenode is empty, type "hdfs namenode -format" if it is empty else skip this step.
3. Create a new directory in the Hadoop file system using the command "hdfs dfs -mkdir /directory_name".
4. Upload the input text file to the new directory using the command "hdfs dfs -put C:\input.txt /word_count"
5. Create the mapper and reducer python programs separately for this program.
6. For Hadoop Streaming execute the following command
   hadoop jar C:\hadoop-3.3.6\share\hadoop\tools\lib\hadoop-streaming-3.3.6.jar ^
   -mapper "python C:/mapper.py" ^ -reducer "python C:/reducer.py" ^ -input
   /word_count/input.txt ^ -output /word_count/hadoop_output/output

## Program:

mapper.py

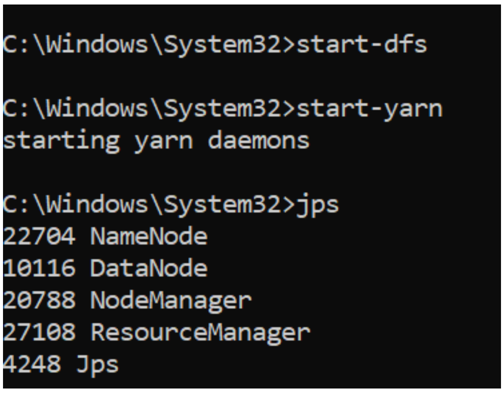
```
#! /usr/bin/env python
import sys
for line in sys.stdin:
  line = line.strip()
  words = line.split()
  for word in words:
    print('%s\t%s'%(word,1))
```

reducer.py

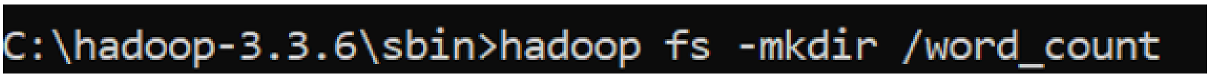
```
#! /usr/bin/env python
```

```
import sys
prev_word = None
prev_count = 0
for line in sys.stdin:
    line = line.strip()
    word, count = line.split('\t')
    count = int(count)
    if prev_word == word:
        prev_count += count
    else:
        if prev_word:
            print('%s\t%s' %(prev_word, prev_count))
        prev_count = count
        prev_word = word
if prev_word == word:
    print('%s\t%s' %(prev_word, prev_count))
```

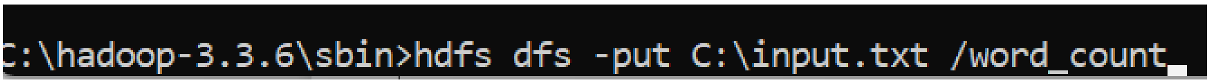
## Output Screenshots:



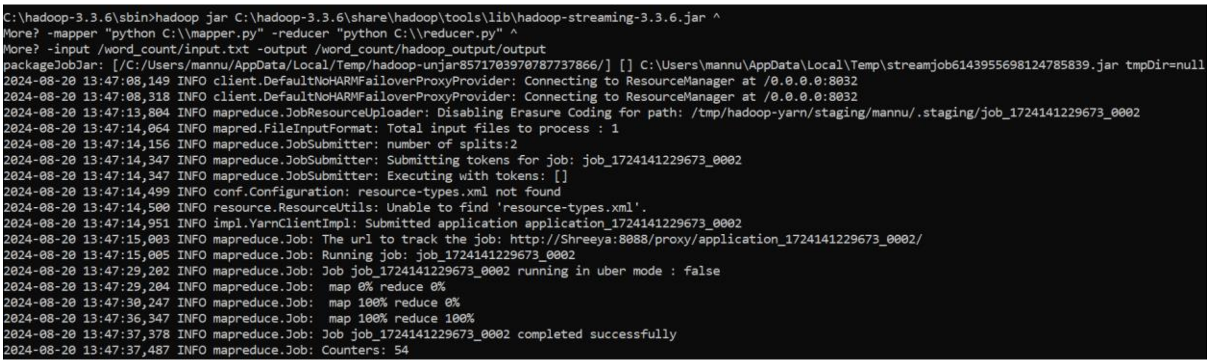
```
C:\Windows\System32>start-dfs

C:\Windows\System32>start-yarn
starting yarn daemons

C:\Windows\System32>jps
22704 NameNode
10116 DataNode
20788 NodeManager
27108 ResourceManager
4248 Jps
```



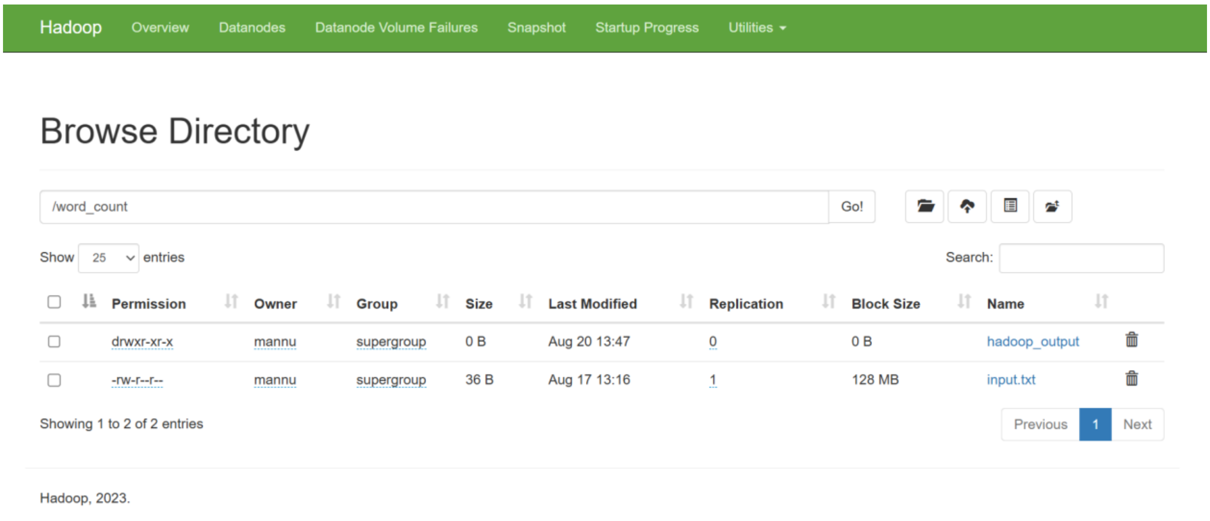
```
C:\hadoop-3.3.6\sbin>hadoop fs -mkdir /word_count
```



```
C:\hadoop-3.3.6\sbin>hdfs dfs -put C:\input.txt /word_count_
```



```
C:\hadoop-3.3.6\sbin>hadoop jar C:\hadoop-3.3.6\share\hadoop\tools\lib\hadoop-streaming-3.3.6.jar ^
More? -mapper "python C:\\mapper.py" -reducer "python C:\\reducer.py" ^
More? -input /word_count/input.txt -output /word_count/hadoop_output/output
packageJobJar: [/C:/Users/mannu/AppData/Local/Temp/hadoop-unjar8571703970787737866/] [] C:\Users\mannu\AppData\Local\Temp\streamjob6143955698124785839.jar tmpDir=null
2024-08-20 13:47:08,149 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-08-20 13:47:08,318 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-08-20 13:47:13,804 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/mannu/.staging/job_1724141229673_0002
2024-08-20 13:47:14,064 INFO mapred.FileInputFormat: Total input files to process : 1
2024-08-20 13:47:14,156 INFO mapreduce.JobSubmitter: number of splits:2
2024-08-20 13:47:14,347 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1724141229673_0002
2024-08-20 13:47:14,347 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-08-20 13:47:14,499 INFO conf.Configuration: resource-types.xml not found
2024-08-20 13:47:14,500 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-08-20 13:47:14,951 INFO impl.YarnClientImpl: Submitted application application_1724141229673_0002
2024-08-20 13:47:15,003 INFO mapreduce.Job: The url to track the job: http://Shreeya:8088/proxy/application_1724141229673_0002/
2024-08-20 13:47:15,005 INFO mapreduce.Job: Running job: job_1724141229673_0002
2024-08-20 13:47:29,202 INFO mapreduce.Job: Job job_1724141229673_0002 running in uber mode : false
2024-08-20 13:47:29,204 INFO mapreduce.Job:  map 0% reduce 0%
2024-08-20 13:47:30,247 INFO mapreduce.Job:  map 100% reduce 0%
2024-08-20 13:47:36,347 INFO mapreduce.Job:  map 100% reduce 100%
2024-08-20 13:47:37,378 INFO mapreduce.Job: Job job_1724141229673_0002 completed successfully
2024-08-20 13:47:37,487 INFO mapreduce.Job: Counters: 54
```



Browse Directory

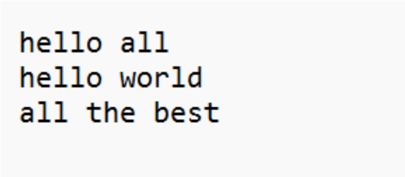
/word_count | Go!

Show 25 entries | Search:

| □ | Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |  |
|---|---|---|---|---|---|---|---|---|---|
| □ | drwxr-xr-x | mannu | supergroup | 0 B | Aug 20 13:47 | 0 | 0 B | hadoop_output | 🗑 |
| □ | -rw-r--r-- | mannu | supergroup | 36 B | Aug 17 13:16 | 1 | 128 MB | input.txt | 🗑 |

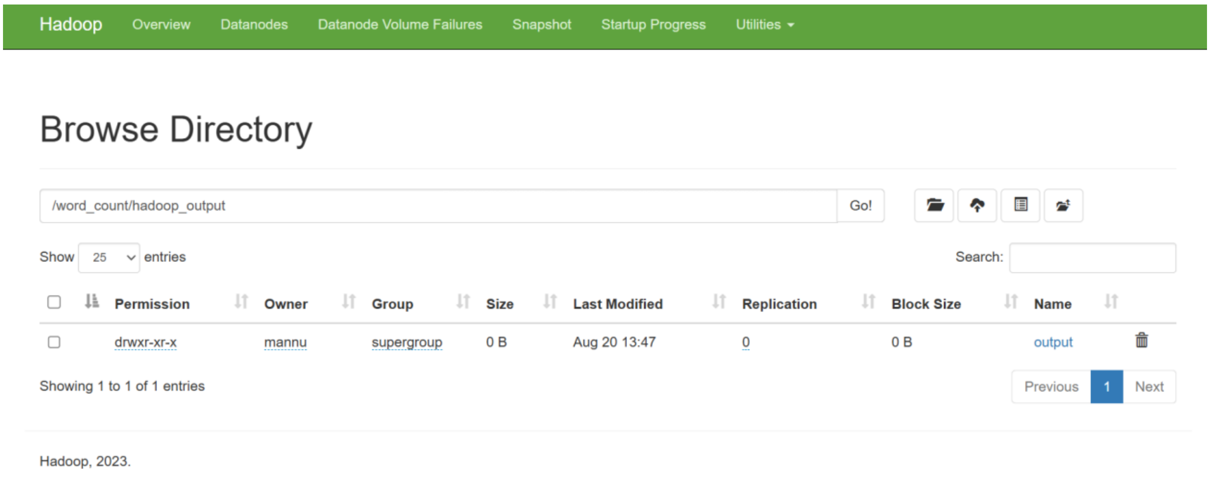
Showing 1 to 2 of 2 entries

Hadoop, 2023.



```
hello all
hello world
all the best
```

input.txt



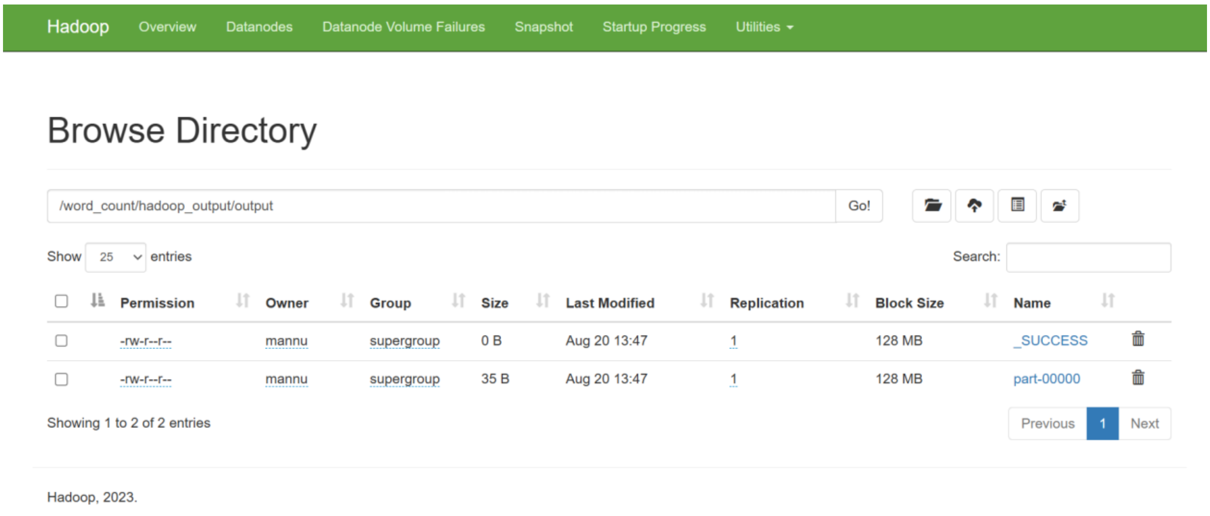
Browse Directory

/word_count/hadoop_output

| □ | Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|---|---|---|---|---|---|---|---|---|
| □ | drwxr-xr-x | mannu | supergroup | 0 B | Aug 20 13:47 | 0 | 0 B | output |

Showing 1 to 1 of 1 entries

Hadoop, 2023.



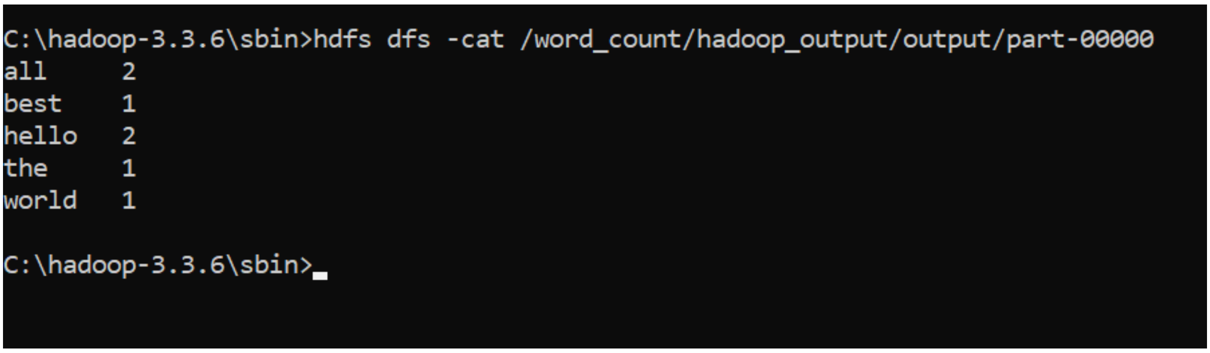
Browse Directory

/word_count/hadoop_output/output

| □ | Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|---|---|---|---|---|---|---|---|---|
| □ | -rw-r--r-- | mannu | supergroup | 0 B | Aug 20 13:47 | 1 | 128 MB | _SUCCESS |
| □ | -rw-r--r-- | mannu | supergroup | 35 B | Aug 20 13:47 | 1 | 128 MB | part-00000 |

Showing 1 to 2 of 2 entries

Hadoop, 2023.



```
C:\hadoop-3.3.6\sbin>hdfs dfs -cat /word_count/hadoop_output/output/part-00000
all     2
best    1
hello   2
the     1
world   1

C:\hadoop-3.3.6\sbin>_
```

**Result:**

Thus the implementation of word count program in python using hadoop’s mapreducer has been executed successfully.