IMPLEMENT WORD COUNT/FREQUENCY PROGRAMS USING MAPREDUCE

AIM:

To implement the python mapper and reducer programs using MapReduce to count the words in a text file using Hadoop.

PROCEDURE:

1. Open command prompt as administrator and start the Hadoop by using the command:

```
C:\Windows\System32>cd C:/hadoop/sbin

C:\hadoop\sbin>start-all
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\hadoop\sbin>jps
20756 DataNode
20008 ResourceManager
21944 NodeManager
23996 Jps
6476 NameNode
```

2. Create a new directory in the Hadoop file systems using the command:

   Upload the input text file into the wordCount directory using the

   command:

```
C:\hadoop\sbin>hadoop fs -mkdir /wordCount

C:\hadoop\sbin>hadoop fs -put D:\Data_Analytics\word_count\input.txt /wordCount
```

3. Create the mapper and reducer files.

4. To execute the files with Hadoop streaming run the following command:

```
C:\hadoop\sbin>hadoop jar C:\hadoop\share\hadoop\tools\lib\hadoop-streaming-3.3.6.jar ^
More?     -files "file:///D:/Data_Analytics/word_count/mapper.py,file:///D:/Data_Analytics/word_count/reducer.py" ^
More?     -input /wordCount/input.txt ^
More?     -output /wordCount/output ^
More?     -mapper "python D:/Data_Analytics/word_count/mapper.py" ^
More?     -reducer "python D:/Data_Analytics/word_count/reducer.py"
packageJobJar: [/C:/Users/OVIYA/AppData/Local/Temp/hadoop-unjar999494471873406757/] [] C:\Users\OVIYA\AppData\Local\Temp
\streamjob7892667070847918172.jar tmpDir=null
2024-08-30 17:16:39,279 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-08-30 17:16:39,466 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-08-30 17:16:44,821 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/
OVIYA/.staging/job_1725018253317_0001
2024-08-30 17:16:45,689 INFO mapred.FileInputFormat: Total input files to process : 1
2024-08-30 17:16:45,763 INFO mapreduce.JobSubmitter: number of splits:2
2024-08-30 17:16:45,879 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1725018253317_0001
2024-08-30 17:16:45,879 INFO mapreduce.JobSubmitter: Executing with tokens: []
```

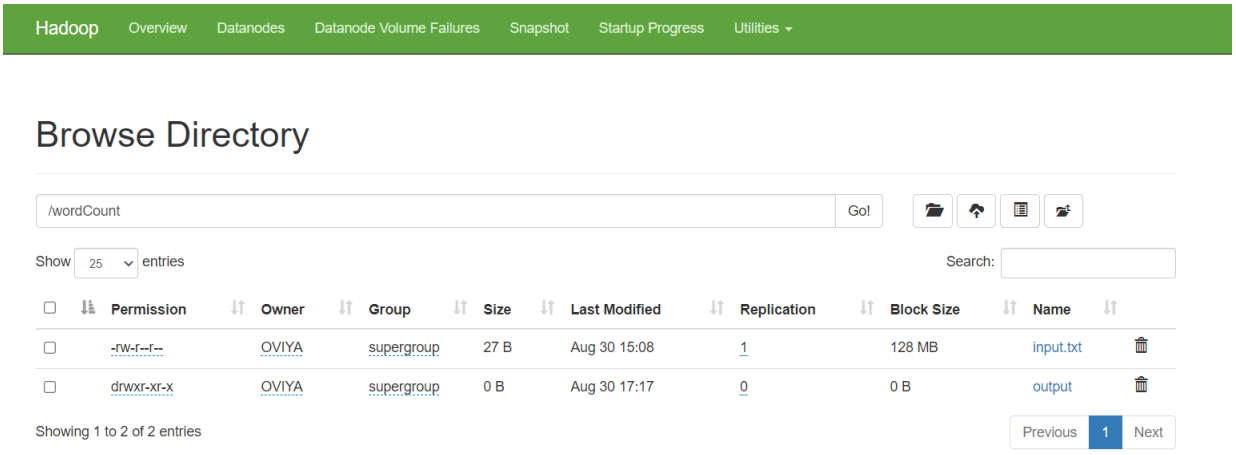
MAPPER.PY

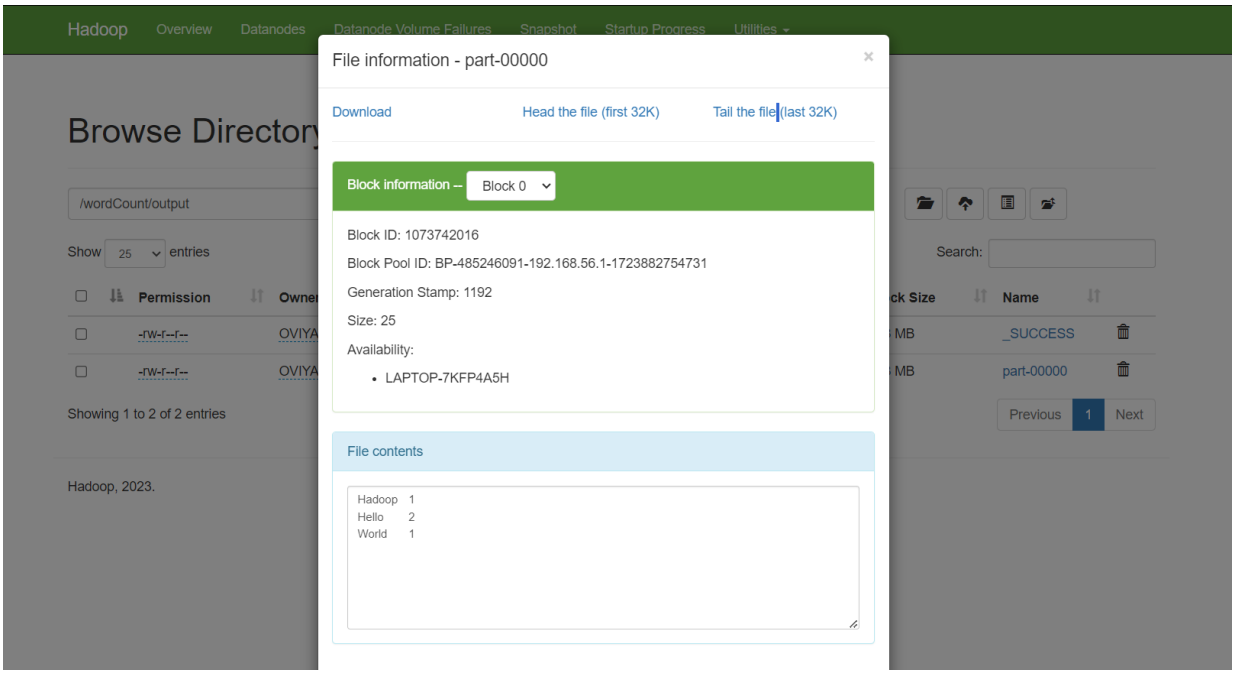
```
#!/usr/bin/env python3
import sys
for line in sys.stdin:
    line = line.strip()
    words = line.split()
    for word in words:
        print(f'{word}\t1')
```

REDUCER.PY

```
#!/usr/bin/env python3
import sys
current_word = None
current_count = 0
for line in sys.stdin:
    line = line.strip()
    word, count = line.split('\t', 1)
    count = int(count)
    if current_word == word:
        current_count += count
    else:
        if current_word:
            print(f'{current_word}\t{current_count}')
        current_word = word
        current_count = count
if current_word == word:
    print(f'{current_word}\t{current_count}')
```

OUTPUT:





RESULT:

Thus the implementation of the python mapper and reducer programs using MapReduce to count the words in a text file using Hadoop is executed successfully.