

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:

```
start-all.cmd
```

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

```
hadoop fs -mkdir /wordCount
```

3. Upload the input text file into the wordCount directory using the command:

```
hadoop fs -put C:/Users/mercy/OneDrive/Documents/DataAnalytics/input.txt /wordcount
```

4. Create the mapper and reducer files.

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

```
hadoop jar C:/hadoop-3.3.6/share/hadoop/tools/lib/hadoop-streaming-3.3.6.jar ^ -file  
C:/Users/mercy/Documents/DataAnalytics/mapper.py ^ -file  
C:/Users/mercy/Documents/DataAnalytics/reducer.py ^ -input /wordCount/input.txt ^ -output  
/user/output ^ -mapper "python mapper.py" ^ -reducer "python reducer.py"
```

MAPPER.PY

```
#!/C:/ProgramData/chocolatey/bin/python3.exe
```

```
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

```
#!/C:/ProgramData/chocolatey/bin/python3.exe
```

```
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:

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities ▾

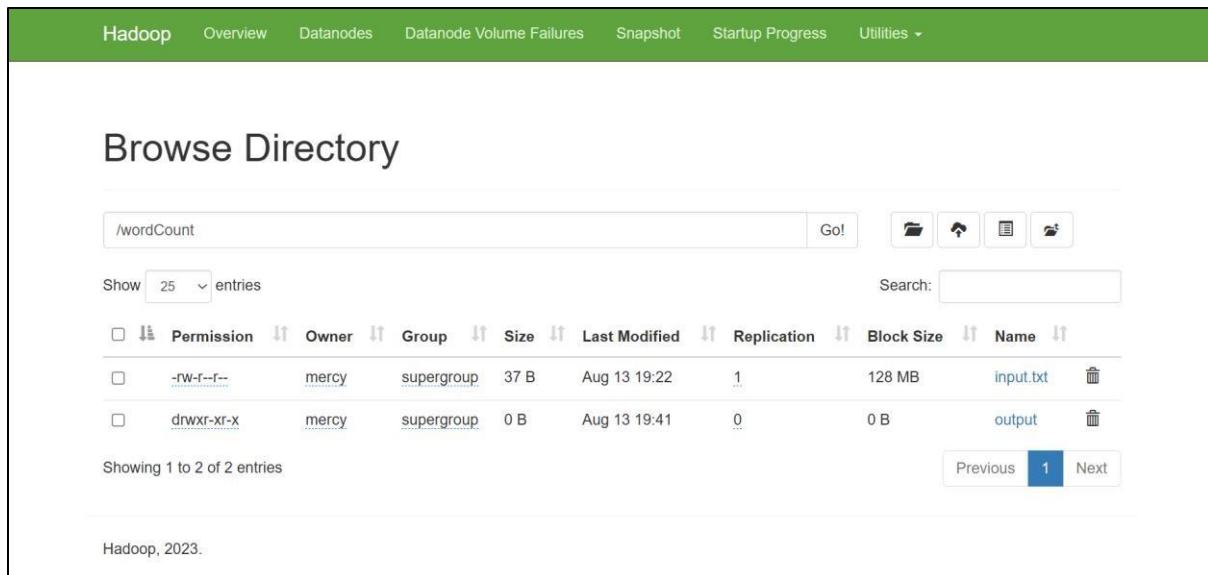
Browse Directory

/ Go!    

Show 25 ▾ entries Search:

<input type="checkbox"/>	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
<input type="checkbox"/>	drwxr-xr-x	mercy	supergroup	0 B	Aug 19 09:01	0	0 B	tmp	
<input type="checkbox"/>	drwxr-xr-x	mercy	supergroup	0 B	Aug 18 21:18	0	0 B	weather	
<input type="checkbox"/>	drwxr-xr-x	mercy	supergroup	0 B	Aug 13 19:41	0	0 B	wordCount	

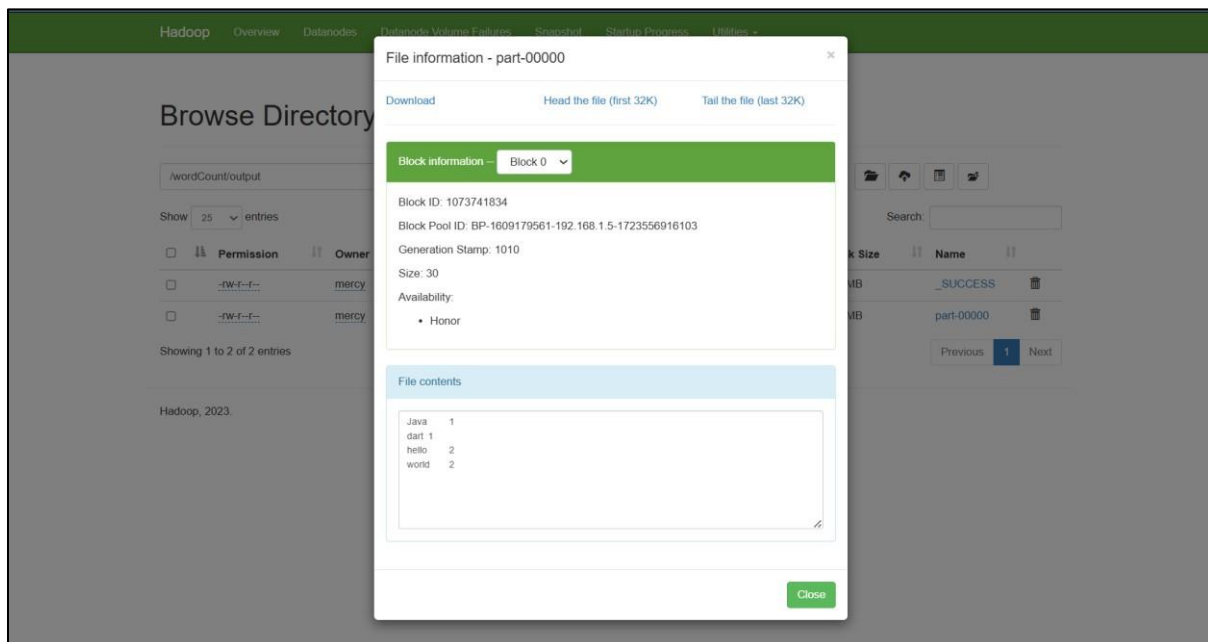
Showing 1 to 3 of 3 entries Previous **1** Next



The screenshot shows the Hadoop web interface's 'Browse Directory' page. The breadcrumb path is '/wordCount'. Below the path bar, there are icons for file operations. A table lists the directory contents:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	mercy	supergroup	37 B	Aug 13 19:22	1	128 MB	input.txt
drwxr-xr-x	mercy	supergroup	0 B	Aug 13 19:41	0	0 B	output

At the bottom, it says 'Showing 1 to 2 of 2 entries' and 'Hadoop, 2023.'.



This screenshot shows the same Hadoop interface but with a modal window open for 'File information - part-00000'. The modal contains the following details:

- Block information: Block 0
- Block ID: 1073741834
- Block Pool ID: BP-1609179561-192.168.1.5-1723556916103
- Generation Stamp: 1010
- Size: 30
- Availability: Honor

The 'File contents' section shows:

```
Java 1
dart 1
hello 2
world 2
```

The background shows the 'Browse Directory' page for '/wordCount/output' with a table listing files like '_SUCCESS' and 'part-00000'.

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.