Name : **Kartavya Mandani**
Roll No.: **20BCE120**
Course: **2CS702 Big Data Analytics**
Practical No.: **4 (HDFS MapReduce)**

Steps Followed:
1.Compilation of java file.
2.Convert class file to jar file.
3.Start Hadoop
4.Make directory in hdfs
5.Upload input file to the hdfs
6.input file format
7.run Jar file command
8.Analysis the running Procedure of mapper and reducer
9.View the Output

```
package Practical_4_Word_Count;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Counter;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.StringUtils;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.net.URI;
import java.util.*;

public class Practical4 {

   public static class TokenizerMapper extends Mapper<Object, Text, Text,
IntWritable>{
       static enum CountersEnum { INPUT_WORDS }
       private final static IntWritable one = new IntWritable(1);
       private Text word = new Text();
       private boolean caseSensitive;
```

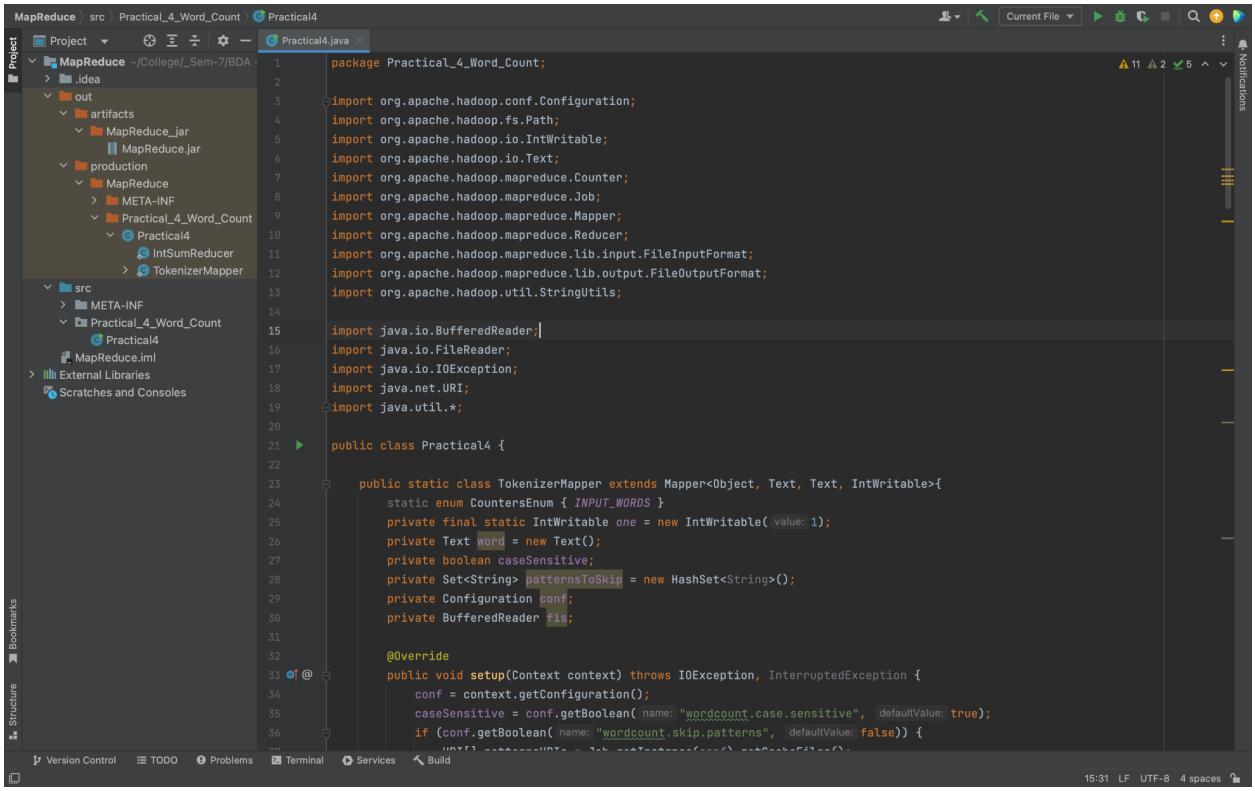
```
        private Set<String> patternsToSkip = new HashSet<String>();
        private Configuration conf;
        private BufferedReader fis;

        @Override
        public void setup(Context context) throws IOException,
InterruptedException {
            conf = context.getConfiguration();
            caseSensitive = conf.getBoolean("wordcount.case.sensitive", true);
            if (conf.getBoolean("wordcount.skip.patterns", false)) {
                URI[] patternsURIs = Job.getInstance(conf).getCacheFiles();
                for (URI patternsURI : patternsURIs) {
                    Path patternsPath = new Path(patternsURI.getPath());
                    String patternsFileName =
patternsPath.getName().toString();
                    parseSkipFile(patternsFileName);
                }
            }
        }

        private void parseSkipFile(String fileName) {
            try {
                fis = new BufferedReader(new FileReader(fileName));
                String pattern = null;
                while ((pattern = fis.readLine()) != null) {
                    patternsToSkip.add(pattern);
                }
            } catch (IOException ioe) {
                System.err.println("Caught exception while parsing the cached
file '"
                        + StringUtils.stringifyException(ioe));
            }
        }

        @Override
        public void map(Object key, Text value, Context context) throws
IOException, InterruptedException {
            String line = (caseSensitive) ?
                    value.toString() : value.toString().toLowerCase();
            for (String pattern : patternsToSkip) {
                line = line.replaceAll(pattern,
                        "");
            }
            StringTokenizer itr = new StringTokenizer(line);
            while (itr.hasMoreTokens()) {
                word.set(itr.nextToken());
                context.write(word, one);
                Counter counter =
context.getCounter(CountersEnum.class.getName(),
```

```
                    CountersEnum.INPUT_WORDS.toString());
                counter.increment(1);
            }
        }
    }

    public static class IntSumReducer extends
Reducer<Text,IntWritable,Text,IntWritable> {

        private IntWritable result = new IntWritable();
        public void reduce(Text key, Iterable<IntWritable> values, Context
context) throws IOException, InterruptedException {
            int sum = 0;
            for (IntWritable val : values) {
                sum += val.get();
            }
            result.set(sum);
            context.write(key, result);
        }
    }

    public static void main(String[] args) throws Exception {

        Configuration conf = new Configuration();
// GenericOptionsParser optionParser = new GenericOptionsParser(conf, args);
// String[] remainingArgs = optionParser.getRemainingArgs();

        if ((args.length != 2) && (args.length != 4)) {
            System.err.println("Usage: wordcount <in> <out> [-skip
skipPatternFile]");
            System.exit(2);
        }

        Job job = Job.getInstance(conf, "wordcount");
        job.setJarByClass(Practical4.class);
        job.setMapperClass(TokenizerMapper.class);
        job.setCombinerClass(IntSumReducer.class);
        job.setReducerClass(IntSumReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        List<String> otherArgs = new ArrayList<String>();
        for (int i=0; i < args.length; ++i) {
            if ("-skip".equals(args[i])) {
            job.addCacheFile(new Path(args[++i]).toUri());
            job.getConfiguration().setBoolean("wordcount.skip.patterns", true);
        } else {
            otherArgs.add(args[i]);
        }
    }
```

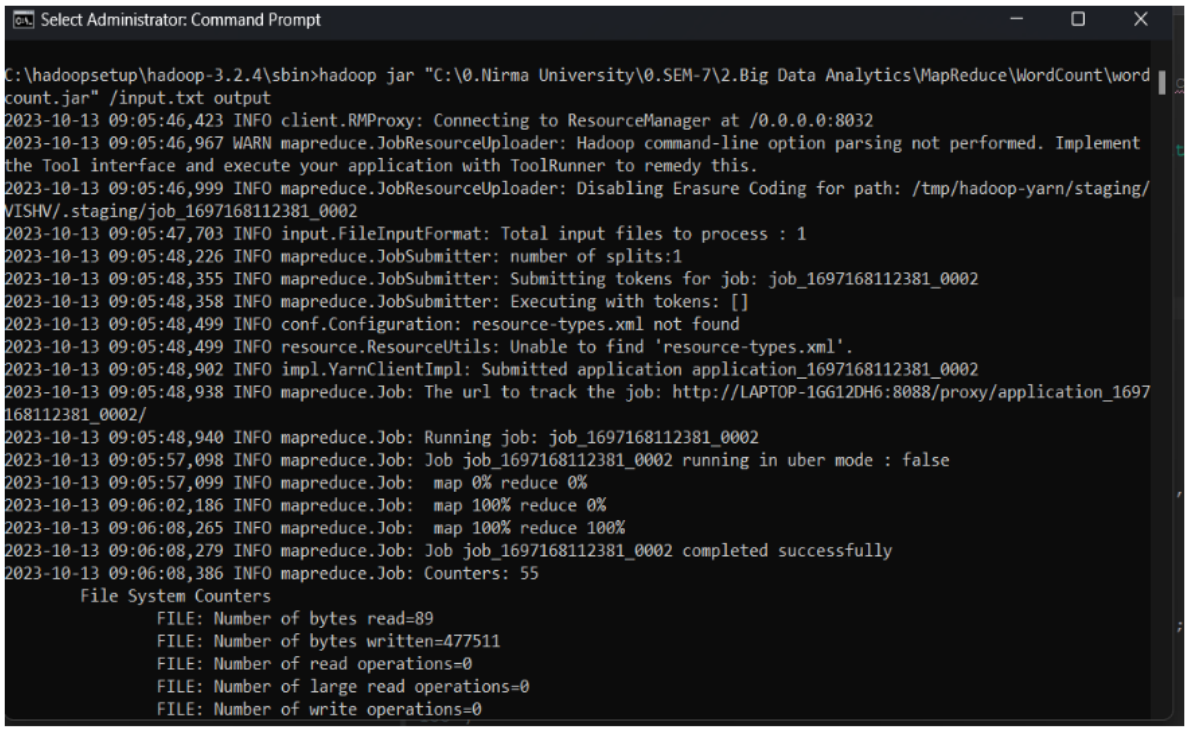
```
   FileInputFormat.addInputPath(job, new Path(otherArgs.get(0)));
   FileOutputFormat.setOutputPath(job, new Path(otherArgs.get(1)));
   System.exit(job.waitForCompletion(true) ? 0 : 1);
   }
}
//import org.apache.hadoop.conf.Configuration;
//import org.apache.hadoop.io.IntWritable;
//import org.apache.hadoop.io.LongWritable;
//import org.apache.hadoop.mapreduce.Job;
//import org.apache.hadoop.mapreduce.Mapper;
//import org.apache.hadoop.mapreduce.Reducer;
//import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
//import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
//import org.w3c.dom.Text;
//
//import java.io.IOException;
//import java.util.StringTokenizer;
//
//
//public class Practical4 {
//
//    public static class Map extends Mapper
<LongWritable,Text,Text,IntWritable> {
//        public void map(LongWritable key, Text value, Context context)
throws IOException,InterruptedException{
//            String line = value.toString();
//            StringTokenizer tokenizer = new StringTokenizer(line);
//            while (tokenizer.hasMoreTokens()) {
//                value.setData(tokenizer.nextToken());
//                context.write(value, new IntWritable(1));
//            }
//        }
//    }
//    public static class Reduce extends Reducer
<Text,IntWritable,Text,IntWritable> {
//        public void reduce(Text key, Iterable <IntWritable> values,Context
context) throws IOException,InterruptedException {
//            int sum=0;
//            for(IntWritable x: values)
//            {
//                sum+=x.get();
//            }
//            context.write(key, new IntWritable(sum));
//        }
//    }
//    public static void main(String[] args) throws Exception {
//        Configuration conf = new Configuration();
//
```

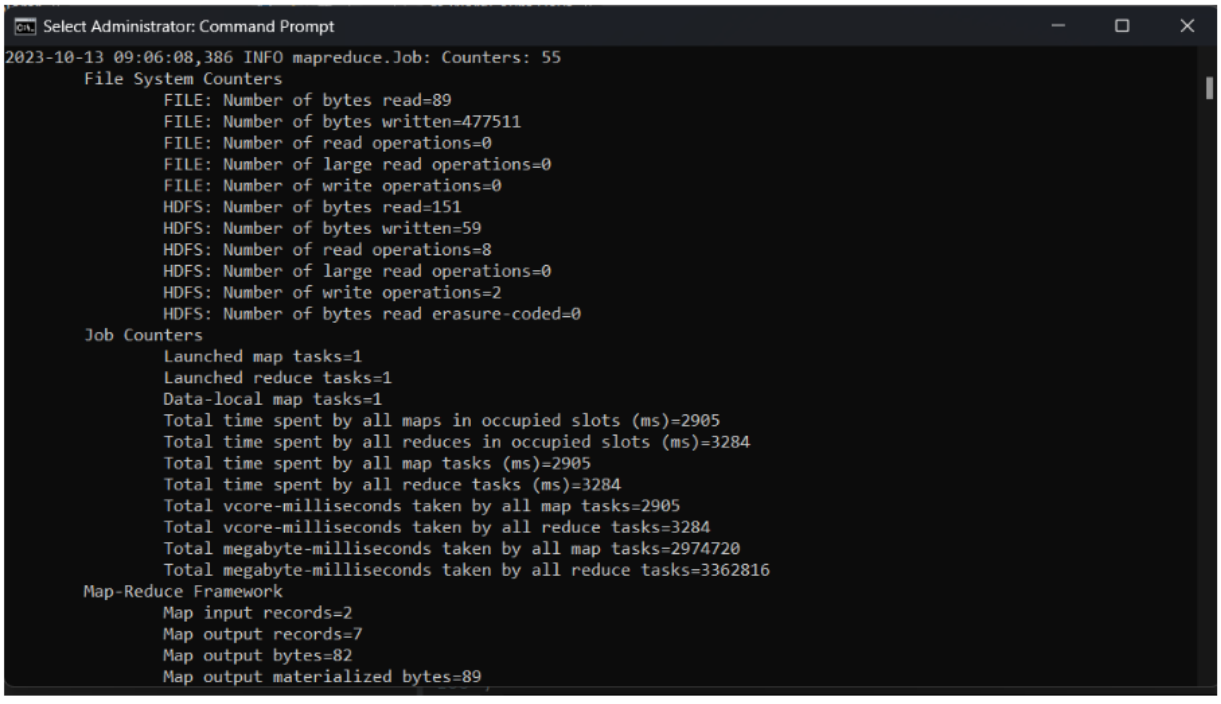
```
//         Job job = Job.getInstance(conf, "Phrase Frequency");
//         job.setJarByClass(Practical4.class);
//         job.setMapperClass(Map.class);
//         job.setReducerClass(Reduce.class);
//         job.setOutputKeyClass(Text.class);
//         job.setOutputValueClass(LongWritable.class);
//         FileInputFormat.addInputPath(job, new
org.apache.hadoop.fs.Path(args[3]));
//         FileOutputFormat.setOutputPath(job, new
org.apache.hadoop.fs.Path(args[4]));
//         System.exit(job.waitForCompletion(true) ? 0 : 1);
//     }
//}
```

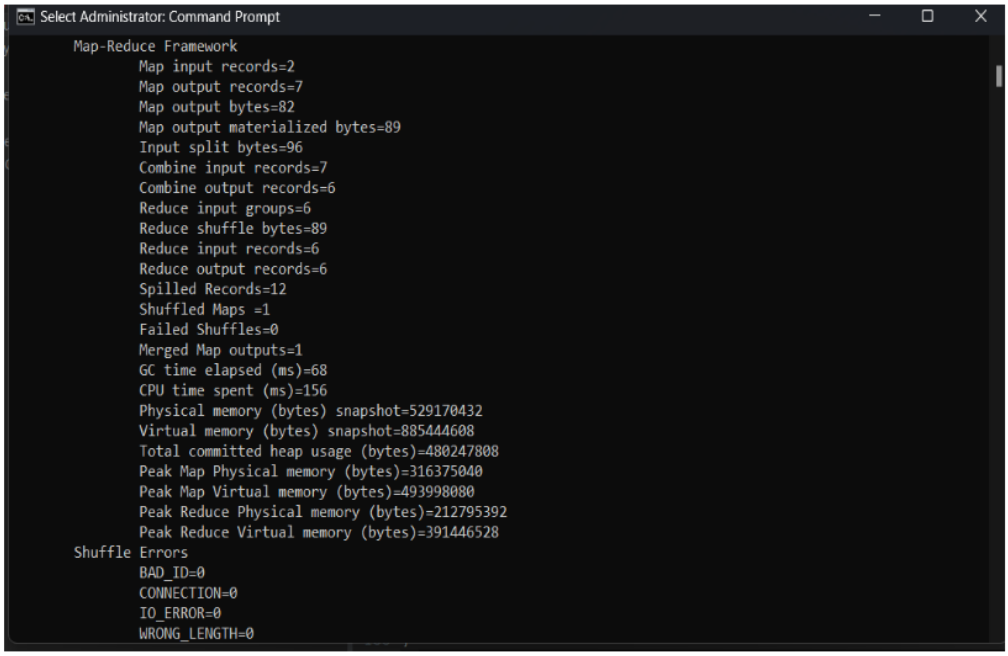
JAR File:



Output:


Select Administrator: Command Prompt
C:\hadoopsetup\hadoop-3.2.4\sbin>hadoop jar "C:\0.Nirma University\0.SEM-7\2.Big Data Analytics\MapReduce\WordCount\word
count.jar" /input.txt output
2023-10-13 09:05:46,423 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2023-10-13 09:05:46,967 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement
the Tool interface and execute your application with ToolRunner to remedy this.
2023-10-13 09:05:46,999 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/
VISHV/.staging/job_1697168112381_0002
2023-10-13 09:05:47,703 INFO input.FileInputFormat: Total input files to process : 1
2023-10-13 09:05:48,226 INFO mapreduce.JobSubmitter: number of splits:1
2023-10-13 09:05:48,355 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1697168112381_0002
2023-10-13 09:05:48,358 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-10-13 09:05:48,499 INFO conf.Configuration: resource-types.xml not found
2023-10-13 09:05:48,499 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2023-10-13 09:05:48,902 INFO impl.YarnClientImpl: Submitted application application_1697168112381_0002
2023-10-13 09:05:48,938 INFO mapreduce.Job: The url to track the job: http://LAPTOP-1GG12DH6:8088/proxy/application_1697
168112381_0002/
2023-10-13 09:05:48,940 INFO mapreduce.Job: Running job: job_1697168112381_0002
2023-10-13 09:05:57,098 INFO mapreduce.Job: Job job_1697168112381_0002 running in uber mode : false
2023-10-13 09:05:57,099 INFO mapreduce.Job: map 0% reduce 0%
2023-10-13 09:06:02,186 INFO mapreduce.Job: map 100% reduce 0%
2023-10-13 09:06:08,265 INFO mapreduce.Job: map 100% reduce 100%
2023-10-13 09:06:08,279 INFO mapreduce.Job: Job job_1697168112381_0002 completed successfully
2023-10-13 09:06:08,386 INFO mapreduce.Job: Counters: 55
File System Counters
FILE: Number of bytes read=89
FILE: Number of bytes written=477511
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0


Select Administrator: Command Prompt
2023-10-13 09:06:08,386 INFO mapreduce.Job: Counters: 55
File System Counters
FILE: Number of bytes read=89
FILE: Number of bytes written=477511
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=151
HDFS: Number of bytes written=59
HDFS: Number of read operations=8
HDFS: Number of large read operations=0
HDFS: Number of write operations=2
HDFS: Number of bytes read erasure-coded=0
Job Counters
Launched map tasks=1
Launched reduce tasks=1
Data-local map tasks=1
Total time spent by all maps in occupied slots (ms)=2905
Total time spent by all reduces in occupied slots (ms)=3284
Total time spent by all map tasks (ms)=2905
Total time spent by all reduce tasks (ms)=3284
Total vcore-milliseconds taken by all map tasks=2905
Total vcore-milliseconds taken by all reduce tasks=3284
Total megabyte-milliseconds taken by all map tasks=2974720
Total megabyte-milliseconds taken by all reduce tasks=3362816
Map-Reduce Framework
Map input records=2
Map output records=7
Map output bytes=82
Map output materialized bytes=89



Output

```
C:\hadoopsetup\hadoop-3.2.4\sbin>hdfs dfs -ls
Found 1 items
drwxr-xr-x   - VISHV supergroup          0 2023-10-13 09:06 output

C:\hadoopsetup\hadoop-3.2.4\sbin>hdfs dfs -ls /output
ls: `/output': No such file or directory

C:\hadoopsetup\hadoop-3.2.4\sbin>hdfs dfs -ls output
Found 2 items
-rw-r--r--   1 VISHV supergroup          0 2023-10-13 09:06 output/_SUCCESS
-rw-r--r--   1 VISHV supergroup         59 2023-10-13 09:06 output/part-r-00000

C:\hadoopsetup\hadoop-3.2.4\sbin>hdfs dfs -cat output/part-r-00000
Example 1
Hadoop  2
Install 1
Mapreduce       1
Run     1
Wordcount       1
```

THE END

:)