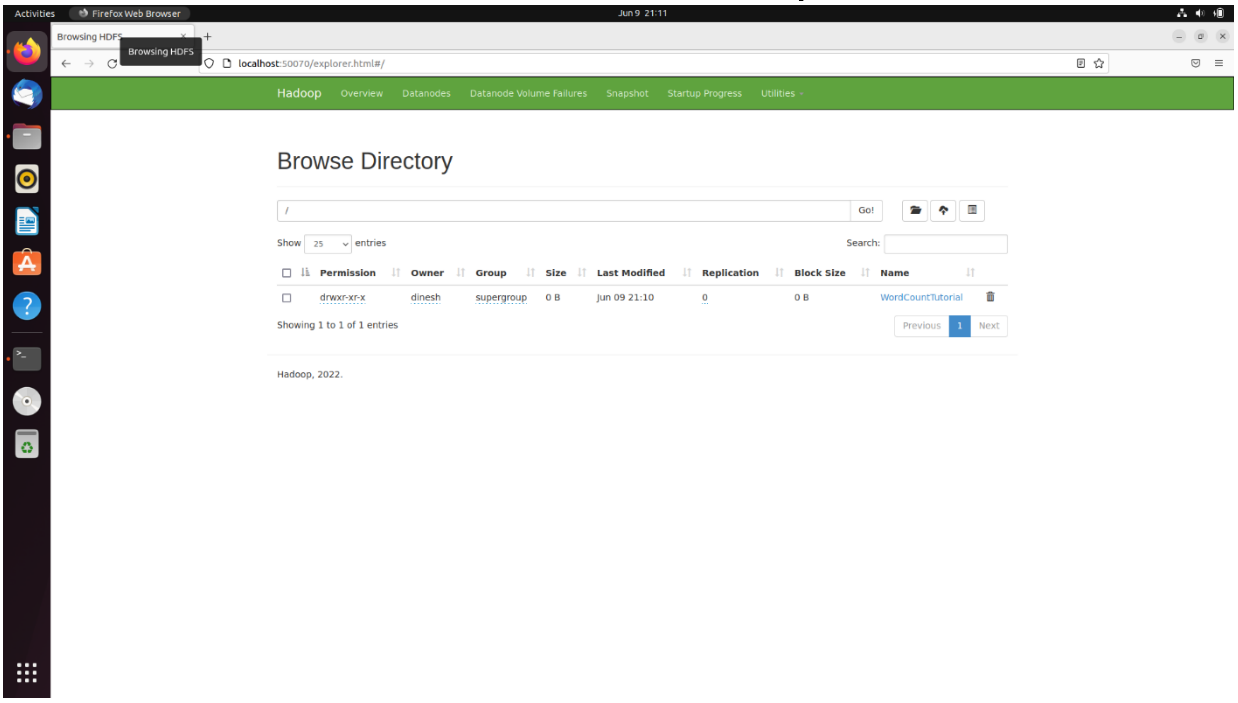
EX NO:13 Implementation of Map Reduce Program in Hadoop

Aim:

To study the implementation of MapReduce Program in Hadoop.

Procedure:

1. Install Hadoop in Ubuntu
2. Post installation of Hadoop in Ubuntu, the Hadoop will be running in the same machine, and when we navigate to localhost:50070, the filesystem will be available which we have created already



Browse Directory

/ Go!

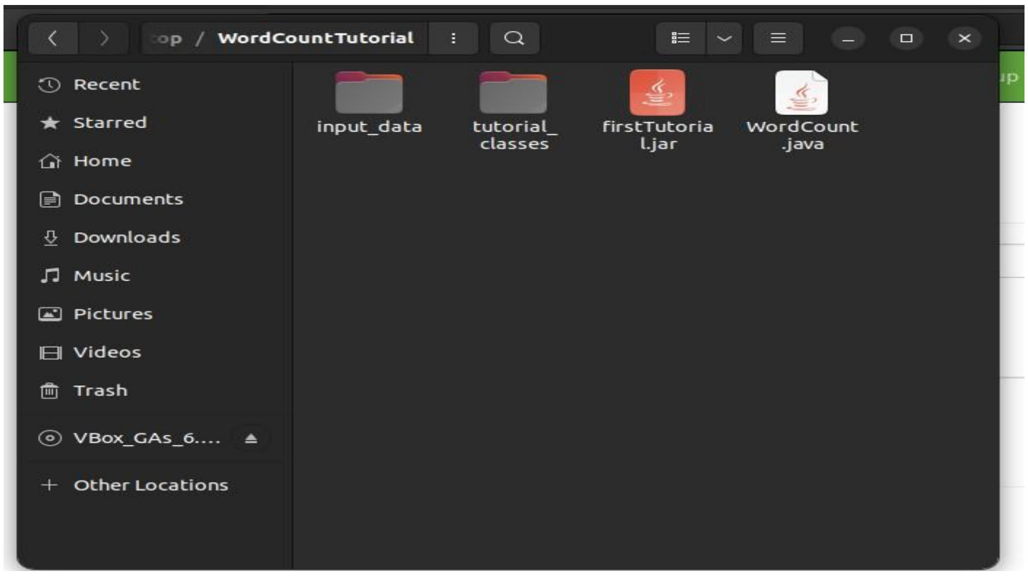
Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	dinesh	supergroup	0 B	Jun 09 21:10	0	0 B	WordCountTutorial

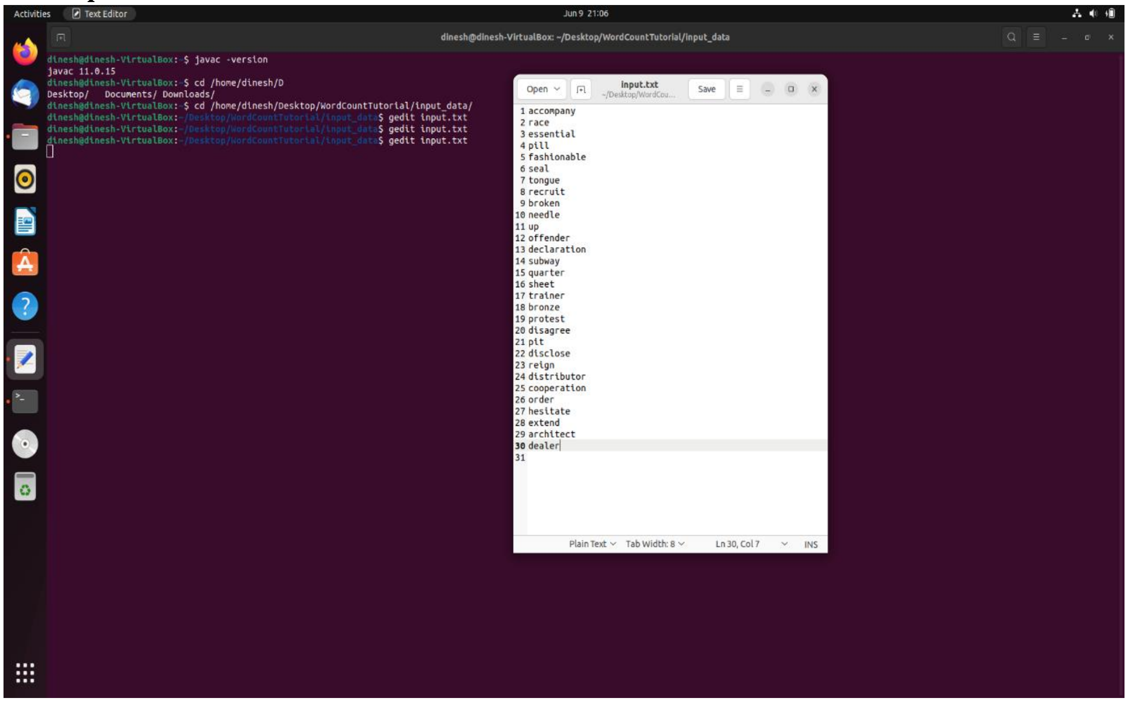
Showing 1 to 1 of 1 entries

Previous 1 Next

Hadoop, 2022.



3. Let us create a folder named wordcount tutorial and have a input data file, and the same will contain the list of words which is to be fed as input to our Hadoop dfs



```
dinesh@dinesh-VirtualBox:~$ javac -version
javac 11.0.15
dinesh@dinesh-VirtualBox:~$ cd /home/dinesh/D
Desktop/   Documents/ Downloads/
dinesh@dinesh-VirtualBox:~$ cd /home/dinesh/Desktop/WordCountTutorial/input_data/
dinesh@dinesh-VirtualBox:~/Desktop/WordCountTutorial/input_data$ gedit input.txt
dinesh@dinesh-VirtualBox:~/Desktop/WordCountTutorial/input_data$ gedit input.txt
dinesh@dinesh-VirtualBox:~/Desktop/WordCountTutorial/input_data$ gedit input.txt
```

```
accompany
race
essential
pill
fashionable
seal
tongue
recruit
broken
needle
up
offender
declaration
subway
quarter
sheet
trainer
bronze
protest
disagree
pit
disclose
reign
distributor
cooperation
order
hesitate
extend
architect
dealer
```

4. Wordcount.java is the file containing the source code for the core logic of map reduce

WordCount.java

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

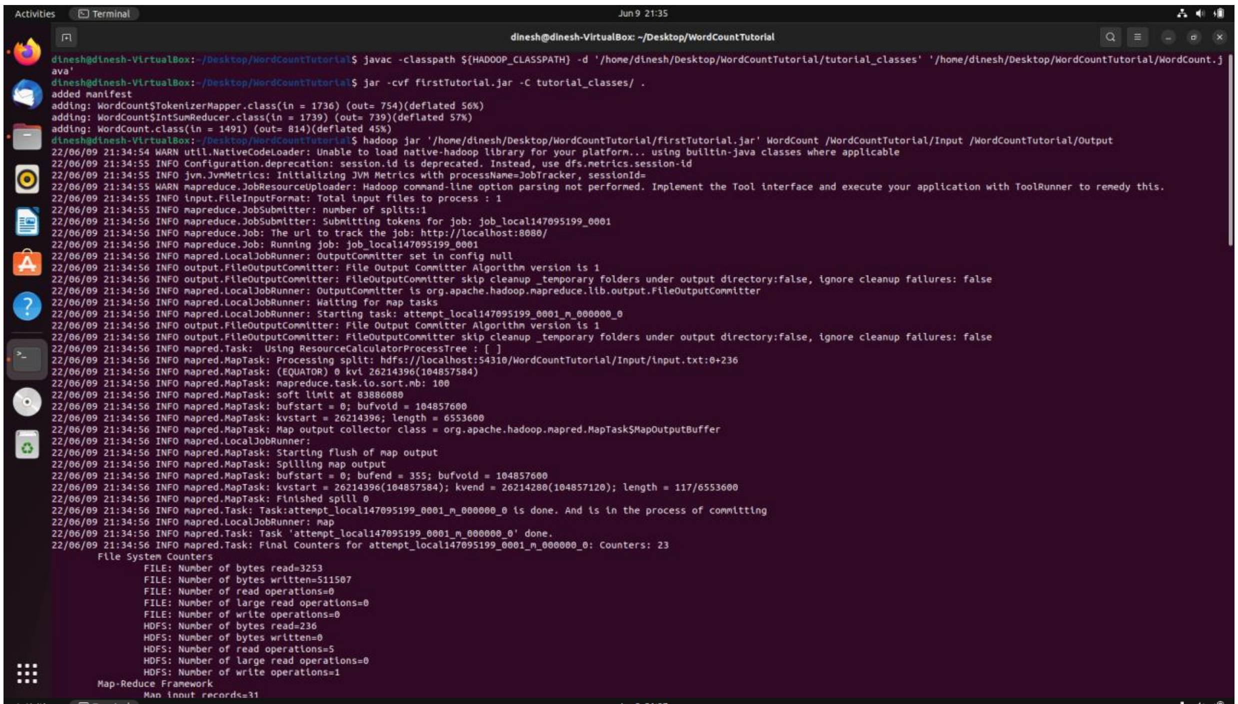
import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

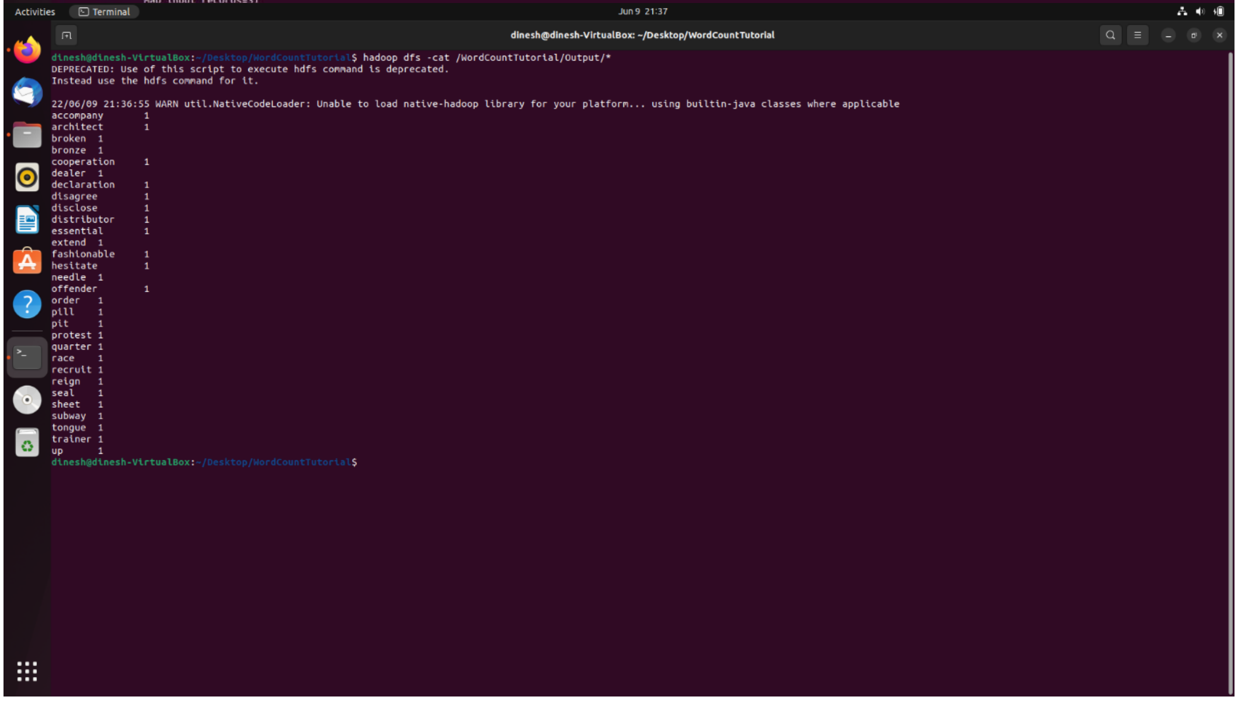
import org.apache.hadoop.mapreduce.Job;

```
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;
public class WordCount {
  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();
    public void map(Object key, Text value, Context context
                    ) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString());
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
      }

    }
  }
  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();
    Job job = Job.getInstance(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

5. After running the map reduce program with the input data specified , we will be getting the output .



```
dinesh@dinesh-VirtualBox:~/Desktop/WordCountTutorial$ javac -classpath ${HADOOP_CLASSPATH} -d '/home/dinesh/Desktop/WordCountTutorial/tutorial_classes' '/home/dinesh/Desktop/WordCountTutorial/WordCount.j
ava'
dinesh@dinesh-VirtualBox:~/Desktop/WordCountTutorial$ jar -cvf firstTutorial.jar -C tutorial_classes/ .
added manifest
adding: WordCount$TokenizerMapper.class(in = 1736) (out= 754)(deflated 56%)
adding: WordCount$IntSumReducer.class(in = 1739) (out= 739)(deflated 57%)
adding: WordCount.class(in = 1491) (out= 814)(deflated 45%)
dinesh@dinesh-VirtualBox:~/Desktop/WordCountTutorial$ hadoop jar '/home/dinesh/Desktop/WordCountTutorial/firstTutorial.jar' WordCount /WordCountTutorial/Input /WordCountTutorial/Output
22/06/09 21:34:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
22/06/09 21:34:55 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id
22/06/09 21:34:55 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=
22/06/09 21:34:55 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
22/06/09 21:34:55 INFO input.FileInputFormat: Total input files to process : 1
22/06/09 21:34:55 INFO mapreduce.JobSubmitter: number of splits:1
22/06/09 21:34:56 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local147095199_0001
22/06/09 21:34:56 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
22/06/09 21:34:56 INFO mapreduce.Job: Running job: job_local147095199_0001
22/06/09 21:34:56 INFO mapred.LocalJobRunner: OutputCommitter set in config null
22/06/09 21:34:56 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
22/06/09 21:34:56 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
22/06/09 21:34:56 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
22/06/09 21:34:56 INFO mapred.LocalJobRunner: Waiting for map tasks
22/06/09 21:34:56 INFO mapred.LocalJobRunner: Starting task: attempt_local147095199_0001_m_000000_0
22/06/09 21:34:56 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
22/06/09 21:34:56 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
22/06/09 21:34:56 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
22/06/09 21:34:56 INFO mapred.MapTask: Processing split: hdfs://localhost:54310/WordCountTutorial/Input/input.txt:0+236
22/06/09 21:34:56 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
22/06/09 21:34:56 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
22/06/09 21:34:56 INFO mapred.MapTask: soft limit at 83886080
22/06/09 21:34:56 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
22/06/09 21:34:56 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
22/06/09 21:34:56 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
22/06/09 21:34:56 INFO mapred.LocalJobRunner:
22/06/09 21:34:56 INFO mapred.MapTask: Starting flush of map output
22/06/09 21:34:56 INFO mapred.MapTask: Spilling map output
22/06/09 21:34:56 INFO mapred.MapTask: bufstart = 0; bufend = 355; bufvoid = 104857600
22/06/09 21:34:56 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26214280(104857120); length = 117/6553600
22/06/09 21:34:56 INFO mapred.MapTask: Finished spill 0
22/06/09 21:34:56 INFO mapred.Task: Task:attempt_local147095199_0001_m_000000_0 is done. And is in the process of committing
22/06/09 21:34:56 INFO mapred.LocalJobRunner: map
22/06/09 21:34:56 INFO mapred.Task: Task 'attempt_local147095199_0001_m_000000_0' done.
22/06/09 21:34:56 INFO mapred.Task: Final Counters for attempt_local147095199_0001_m_000000_0: Counters: 23
        File System Counters
                FILE: Number of bytes read=3253
                FILE: Number of bytes written=511507
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=236
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=5
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=1
        Map-Reduce Framework
                Map input records=31
```



```
dinesh@dinesh-VirtualBox:~/Desktop/WordCountTutorial$ hadoop dfs -cat /WordCountTutorial/Output/*
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

22/06/09 21:36:55 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
accompany       1
architect       1
broken  1
bronze  1
cooperation     1
dealer  1
declaration     1
disagree        1
disclose        1
distributor     1
essential       1
extend  1
fashionable     1
hesitate        1
needle  1
offender        1
order   1
pill    1
pit     1
protest 1
quarter 1
race    1
recruit 1
reign   1
seal    1
sheet   1
subway  1
tongue  1
trainer 1
up      1
dinesh@dinesh-VirtualBox:~/Desktop/WordCountTutorial$
```

RESULTS:

Thus, the implementation of Map Reduce in Hadoop has been studied

REFERENCES:

https://www.youtube.com/watch?v=6sK3LDY7Pp4

https://www.youtube.com/watch?v=CtOhsZ0Sb1E