Lab 7 TopN

TopN Driver Code:

```
package samples.topn;
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.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;

public class TopN {
   public static void main(String[] args) throws Exception {
      Configuration conf = new Configuration();
      String[] otherArgs = (new GenericOptionsParser(conf, args)).getRemainingArgs();
      if (otherArgs.length != 2) {
         System.err.println("Usage: TopN <in> <out>");
         System.exit(2);
      }
      Job job = Job.getInstance(conf);
      job.setJobName("Top N");
      job.setJarByClass(TopN.class);
      job.setMapperClass(TopNMapper.class);
      job.setReducerClass(TopNReducer.class);
      job.setOutputKeyClass(Text.class);
      job.setOutputValueClass(IntWritable.class);
      FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
      FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
      System.exit(job.waitForCompletion(true) ? 0 : 1);
   }
}
```

TopN Mapper Code:

```
package samples.topn;
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
```

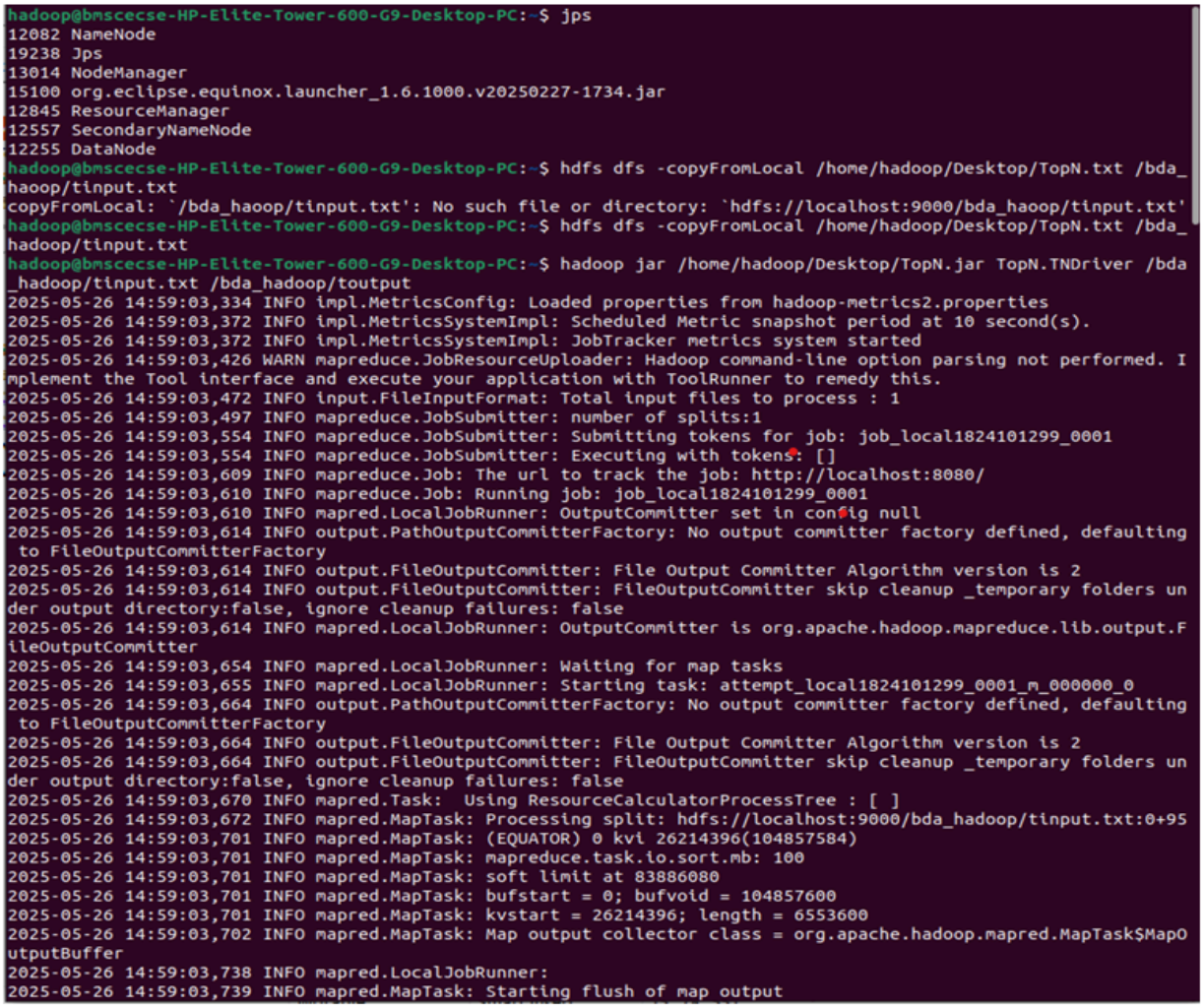
```
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class TopNMapper extends Mapper<Object, Text, Text, IntWritable> {
   private static final IntWritable one = new IntWritable(1);
   private Text word = new Text();
   private String tokens = "[_|$#<>\\^=\\[\\]\\*/\\\\,;,.\\-:()?!\"']";

   public void map(Object key, Text value, Context context) throws IOException,
InterruptedException {
      String cleanLine = value.toString().toLowerCase().replaceAll(this.tokens, " ");
      StringTokenizer itr = new StringTokenizer(cleanLine);
      while (itr.hasMoreTokens()) {
         this.word.set(itr.nextToken().trim());
         context.write(this.word, one);
      }
   }
}
```

TopN Reducer Code:

```
package samples.topn;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
import utils.MiscUtils;

public class TopNReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
   private Map<Text, IntWritable> countMap = new HashMap<>();

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

   protected void cleanup(Context context) throws IOException, InterruptedException {
      Map<Text, IntWritable> sortedMap = MiscUtils.sortByValues(this.countMap);
      int counter = 0;
```

```
        for (Text key : sortedMap.keySet()) {
            if (counter++ == 20)
                break;
            context.write(key, sortedMap.get(key));
        }
    }
}
```

Output screenshots:



```
        File System Counters
                FILE: Number of bytes read=10682
                FILE: Number of bytes written=1291808
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=190
                HDFS: Number of bytes written=40
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=3
                Map output records=15
                Map output bytes=154
                Map output materialized bytes=190
                Input split bytes=108
                Combine input records=0
                Combine output records=0
                Reduce input groups=5
                Reduce shuffle bytes=190
                Reduce input records=15
                Reduce output records=5
                Spilled Records=30
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=0
                Total committed heap usage (bytes)=1052770304
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=95
        File Output Format Counters
                Bytes Written=40
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hdfs dfs -ls /bda_hadoop/toutput
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2025-05-26 14:59 /bda_hadoop/toutput/_SUCCESS
-rw-r--r--   1 hadoop supergroup         40 2025-05-26 14:59 /bda_hadoop/toutput/part-r-00000
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hdfs dfs -cat /bda_hadoop/toutput/part-r-00000
banana  5
apple   4
fruit   3
mango   2
kiwi    1
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$
```