

## Lab – 8

**For a given Text file, Create a Map Reduce program to sort the content in an alphabetic order listing only top 10 maximum occurrences of words. Driver-**

```
TopN.class 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);

}

public static 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, Mapper<Object, Text, Text, IntWritable>.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);

        }

    }

}

```

```

TopNCombiner.class package samples.topn; import java.io.IOException; import

org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.Text; import

org.apache.hadoop.mapreduce.Reducer; public class TopNCombiner extends Reducer<Text,

IntWritable, Text, IntWritable> { public void reduce(Text key, Iterable<IntWritable> values,

```

Reducer<Text, IntWritable, Text, IntWritable>.Context context) throws IOException,

InterruptedException { int sum = 0;

for (IntWritable val : values) sum +=

val.get(); context.write(key, new

IntWritable(sum));

}

}

**TopNMapper.class** 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, Mapper<Object, Text, Text, IntWritable>.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);

}

}

}

```

TopNReducer.class 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, Reducer<Text, IntWritable, Text,
IntWritable>.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(Reducer<Text, IntWritable, Text, IntWritable>.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));
}
}
}

```

```

C:\hadoop-3.3.0\sbin>jps
11072 DataNode
20528 Jps
5620 ResourceManager
15532 NodeManager
6140 NameNode

C:\hadoop-3.3.0\sbin>hdfs dfs -mkdir /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -ls /
Found 1 items
drwxr-xr-x - Anusree supergroup 0 2021-05-08 19:46 /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -copyFromLocal C:\input.txt /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -ls /input_dir
Found 1 items
-rw-r--r-- 1 Anusree supergroup 36 2021-05-08 19:48 /input_dir/input.txt

C:\hadoop-3.3.0\sbin>hdfs dfs -cat /input_dir/input.txt
hello
world
hello
hadoop
bye

```

```

C:\hadoop-3.3.0\sbin>hadoop jar C:\sort.jar samples.topn.TopN /input_dir/input.txt /output_dir
2021-05-08 19:54:54,582 INFO client.DefaultHadoopFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-05-08 19:54:55,291 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging/job_1620483374279_0001
2021-05-08 19:54:55,821 INFO input.FileInputFormat: Total input files to process : 1
2021-05-08 19:54:56,261 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-08 19:54:56,552 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1620483374279_0001
2021-05-08 19:54:56,552 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-08 19:54:56,843 INFO conf.Configuration: resource-types.xml not found
2021-05-08 19:54:56,843 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-08 19:54:57,387 INFO impl.YarnClientImpl: Submitted application application_1620483374279_0001
2021-05-08 19:54:57,507 INFO mapreduce.Job: The url to track the job: http://LAPTOP-1G329ESD:8088/proxy/application_1620483374279_0001/
2021-05-08 19:54:57,508 INFO mapreduce.Job: Running job: job_1620483374279_0001
2021-05-08 19:55:13,792 INFO mapreduce.Job: Job job_1620483374279_0001 running in uber mode : false
2021-05-08 19:55:13,794 INFO mapreduce.Job: map 0% reduce 0%
2021-05-08 19:55:20,020 INFO mapreduce.Job: map 100% reduce 0%
2021-05-08 19:55:27,116 INFO mapreduce.Job: map 100% reduce 100%
2021-05-08 19:55:33,199 INFO mapreduce.Job: Job job_1620483374279_0001 completed successfully
2021-05-08 19:55:33,334 INFO mapreduce.Job: Counters: 54
  File System Counters
    FILE: Number of bytes read=65
    FILE: Number of bytes written=530397
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=142
    HDFS: Number of bytes written=31
    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

```

```

C:\hadoop-3.3.0\sbin>hdfs dfs -cat /output_dir/*
hello 2
hadoop 1
world 1
bye 1

C:\hadoop-3.3.0\sbin>

```