Lab 4 – Total Order Sorting Report

Net ID: tw2770

Name: Tzu-An Wang

1. TotalOrderSortMapper

```java
import java.io.IOException;
import javax.naming.Context;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;



public class TotalOrderSortMapper extends Mapper<LongWritable, Text, Text, Text> {

    @Override
    public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
        // Convert the line to a string
        String line = value.toString();

        // Check if the line is not empty and has at least ten characters
        if (line != null && line.length() >= 10) {
            // Extract the first ten bytes as the sort key
            String sortKey = line.substring(beginIndex:0, endIndex:10);

            // Output the sort key and the original line as the value
            context.write(new Text(sortKey), new Text(line.substring(beginIndex:10, line.length())));
        }
    }
}
```

2. TotalOrderSortReducer

```java
import java.io.IOException;
import javax.naming.Context;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class TotalOrderSortReducer extends Reducer<Text, Text, Text, Text> {
    @Override
    public void reduce(Text key, Iterable<Text> values, Context context)
            throws IOException, InterruptedException {
        // Iterate through all values associated with the key and write them to the context
        for (Text value : values) {
            context.write(key, value);
        }
    }
}
```

3. TotalOrderSort – Job 1

```java
public class TotalOrderSort {
    Run | Debug
    public static void main(String[] args) throws Exception {
        Path inputPath = new Path(args[0]);
        Path outputPath = new Path(args[1]);
        Path stagingPath = new Path(args[1] + "_staging");
        Path partitionFile = new Path(args[1] + "_partition");

        // Configure job to prepare for sampling
        Job sampleJob = Job.getInstance();
        sampleJob.setJarByClass(TotalOrderSort.class);

        // Use the mapper implementation with zero reduce tasks
        sampleJob.setMapperClass(TotalOrderSortMapper.class);
        sampleJob.setNumReduceTasks(0);

        sampleJob.setOutputKeyClass(Text.class);
        sampleJob.setOutputValueClass(Text.class);

        TextInputFormat.setInputPaths(sampleJob, inputPath);

        // Set the output format to a sequence file
        sampleJob.setOutputFormatClass(SequenceFileOutputFormat.class);
        SequenceFileOutputFormat.setOutputPath(sampleJob, stagingPath);

        // Submit the job
        sampleJob.waitForCompletion(true);
```

4. TotalOrderSort – Job 2

```
43          Job orderJob = Job.getInstance();
44          orderJob.setJarByClass(TotalOrderSort.class);
45
46          // Here, use the identity mapper to output the key/value pairs in the SequenceFile
47          orderJob.setReducerClass(TotalOrderSortReducer.class);
48
49          // Set the number of reduce tasks to an appropriate number for the amount of data being sorted
50          orderJob.setNumReduceTasks(10);
51
52          // Use Hadoop's TotalOrderPartitioner class
53          orderJob.setPartitionerClass(TotalOrderPartitioner.class);
54
55
56          // Set the partition file
57          TotalOrderPartitioner.setPartitionFile(orderJob.getConfiguration(), partitionFile);
58
59          orderJob.setOutputKeyClass(Text.class);
60          orderJob.setOutputValueClass(Text.class);
61
62          // Set the input to the previous job's output
63          orderJob.setInputFormatClass(SequenceFileInputFormat.class);
64          SequenceFileInputFormat.setInputPaths(orderJob, stagingPath);
65
66          // Set the output path to the command line parameter
67          TextOutputFormat.setOutputPath(orderJob, outputPath);
68
69          // Use the InputSampler to go through the output of the previous job, sample it, and create the partition file
70          InputSampler.writePartitionFile(orderJob, new InputSampler.RandomSampler(.001, 10000));
71
72          // Submit the job
73          orderJob.waitForCompletion(true);
74      }
75  }
```

5. Program runs successfully

```
tw2770_nyu_edu@nyu-dataproc-m:~$ hadoop fs -ls /user/tw2770_nyu_edu/lab4/output
Found 11 items
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu          0 2024-03-06 04:49 /user/tw2770_nyu_edu/lab4/output/_SUCCESS
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu     833200 2024-03-06 04:49 /user/tw2770_nyu_edu/lab4/output/part-r-00000
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu    1056000 2024-03-06 04:49 /user/tw2770_nyu_edu/lab4/output/part-r-00001
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu    1525700 2024-03-06 04:49 /user/tw2770_nyu_edu/lab4/output/part-r-00002
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu     838300 2024-03-06 04:49 /user/tw2770_nyu_edu/lab4/output/part-r-00003
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu     729800 2024-03-06 04:49 /user/tw2770_nyu_edu/lab4/output/part-r-00004
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu    1176500 2024-03-06 04:49 /user/tw2770_nyu_edu/lab4/output/part-r-00005
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu    1242600 2024-03-06 04:49 /user/tw2770_nyu_edu/lab4/output/part-r-00006
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu     746300 2024-03-06 04:49 /user/tw2770_nyu_edu/lab4/output/part-r-00007
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu    1143100 2024-03-06 04:49 /user/tw2770_nyu_edu/lab4/output/part-r-00008
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu     708500 2024-03-06 04:49 /user/tw2770_nyu_edu/lab4/output/part-r-00009
```