

Assignment No. 5

Implement Secondary Sorting. (Write hadoop code to implement Item Sort Program)

```
-----Main class-----
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.Job;

public class testdriver {
    public static void main(String[] args) throws Exception {
        if (args.length != 2) {
            System.out.printf("Usage: WordCount <input dir> <output
dir>\n");
            System.exit(-1);
        }
        Job job = new Job();

        job.setJarByClass(testdriver.class);
        job.setJobName("Word Count");
        FileInputFormat.setInputPaths(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        job.setMapperClass(testmap.class);
        job.setReducerClass(testreduce.class);

        job.setMapOutputKeyClass(IntWritable.class);
        job.setMapOutputValueClass(IntWritable.class);

        job.setOutputKeyClass(IntWritable.class);
        job.setOutputValueClass(IntWritable.class);

        boolean success = job.waitForCompletion(true);
        System.exit(success ? 0 : 1);
    }
}
```

```
-----Mapper class-----

import java.io.IOException;
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 testmap extends Mapper<LongWritable, Text, IntWritable,
IntWritable> {
    @Override
    public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException {
        String line = value.toString();
        String[] tokens = line.split(","); // This is the delimiter
between
        int keypart = Integer.parseInt(tokens[0]);
        int valuePart = Integer.parseInt(tokens[1]);
    }
}
```

```
        context.write(new IntWritable(valuePart), new
IntWritable(keypart));
```

```
    }
}
```

-----Reducer class-----

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.mapreduce.Reducer;

public class testreduce extends Reducer<IntWritable, IntWritable,
IntWritable, IntWritable> {
@Override
    public void reduce(IntWritable key, Iterable<IntWritable>
values,
Context context) throws IOException, InterruptedException {

        for (IntWritable value : values) {

            context.write(value,key);

        }

    }
}
```

Step 1: Export Java Eclipse Project Jar File to Cloudera
Step 2. Make Sort.txt file vi editor ->Write data
Step 3: Perform Below commands on terminal

Command Map Reduce Code

1) Transfer all local file to hadoop

```
Hdfs dfs -put sort.txt /user/cloudera
```

```
Hdfs dfs -put Sorting.jar /user/cloudera
```

2) Run Java Jar File for Map Reduce Operation

```
hadoop jar Sorting.jar testdriver sort.txt outputsort
```

3) List outputfile

```
hdfs dfs -ls /user/cloudera/outputsort
```

4) Show outputfile

```
hdfs dfs -cat /user/cloudera/outputsort /part-r-00000
```