LARGE SCALE DATA PROCESSING (CSE 3025)

WIN SEMESTER 2017-18

NAME: B.SHUBANKAR

REGNO: 15BCE1123

PROFESSOR: DR. Maheswari N

LAB-5

AIM: To create custom partitioning and changing the number of mappers using split function.

PROGRAM:

1. Custom partitioning

```
import java.io.IOException;

import java.util.*;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.conf.*;

import org.apache.hadoop.io.*;

import org.apache.hadoop.mapreduce.*;

import org.apache.hadoop.mapreduce.Reducer.Context;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

 public class Salary1 {

public static class Map extends Mapper<LongWritable, Text, Text,

IntWritable> {

 public void map(LongWritable key, Text value, Context context) throws

IOException, InterruptedException {

 String[] line =value.toString().split(",");

 int i= Integer.parseInt(line[1]);

 context.write(new Text(line[3]),new IntWritable(i));

}

}

 public static class dpart extends Partitioner<Text,IntWritable>

{

public int getPartition(Text key,IntWritable value,int nr)

{

if(value.get()<30000)

return 0;

if(value.get() < 50000)

return 1;
else
return 2;
}}
 public static class Reduce extends Reducer<Text, IntWritable, Text,
IntWritable> {
 public void reduce(Text key, IntWritable values, Context
 context)throws IOException, InterruptedException {
context.write(key,values);
}
}
public static void main(String[] args) throws Exception {
Configuration conf = new Configuration();
Job job = new Job(conf, "Salary1");
job.setJarByClass(Salary.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
job.setMapperClass(Map.class);
job.setPartitionerClass(dpart.class);
job.setNumReduceTasks(3);
job.setInputFormatClass(TextInputFormat.class);
job.setOutputFormatClass(TextOutputFormat.class);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path(args[1]));

job.waitForCompletion(true);

}

}
```

2. To create 3 mappers

```
import java.io.IOException;

import java.util.*;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.conf.*;

import org.apache.hadoop.io.*;

import org.apache.hadoop.mapreduce.*;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;


public class Salary2 {
```

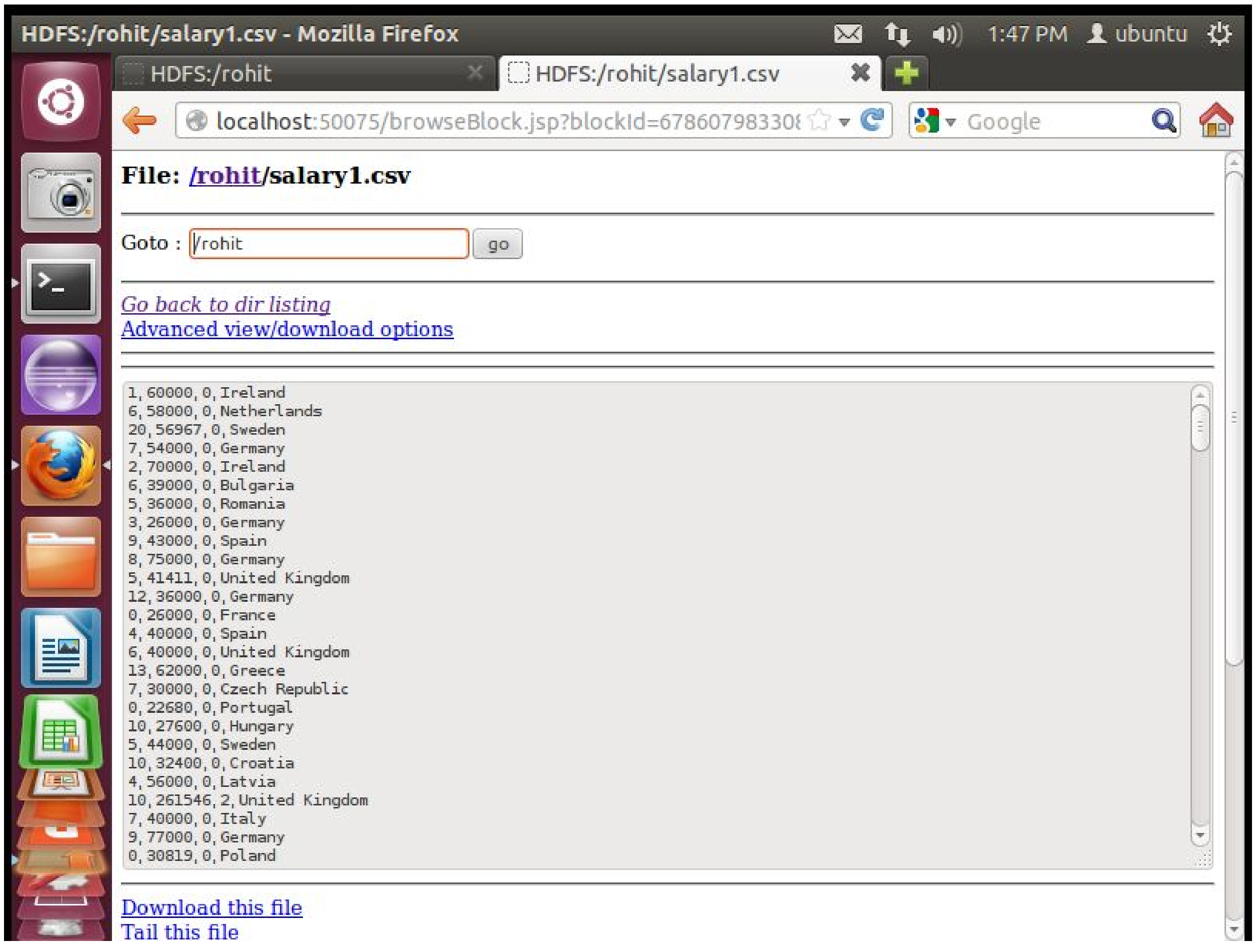
```
public static class Map extends Mapper<LongWritable, Text, Text,
IntWritable> {
public void map(LongWritable key, Text value, Context context) throws
IOException, InterruptedException {
 String[] line =value.toString().split(",");
 int i= Integer.parseInt(line[1]);
 context.write(new Text(line[3]),new IntWritable(i));
}
}
public static void main(String[] args) throws Exception {
Configuration conf = new Configuration();
conf.set("mapred.max.split.size","10000");
Job job = new Job(conf, "Salary2");
job.setJarByClass(Salary.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
job.setMapperClass(Map.class);
job.setInputFormatClass(TextInputFormat.class);
job.setOutputFormatClass(TextOutputFormat.class);
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));

job.waitForCompletion(true);

}

}
```

3. To run whole file in one mapper

```
import java.io.IOException;

import java.util.*;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.conf.*;

import org.apache.hadoop.io.*;

import org.apache.hadoop.mapreduce.*;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class Salary2b {

public static class Map extends Mapper<LongWritable, Text, Text,

IntWritable> {

public void map(LongWritable key, Text value, Context context) throws

IOException, InterruptedException {
 String[] line =value.toString().split(",");
 int i= Integer.parseInt(line[1]);
 context.write(new Text(line[3]),new IntWritable(i));
}
public class splitfalse extends TextInputFormat {
protected boolean isSplitable(JobContext context, Path file) {
return false;
}
}
}
public static void main(String[] args) throws Exception {
Configuration conf = new Configuration();
//conf.set("mapred.max.split.size","10000");
Job job = new Job(conf, "Salary2b");
job.setJarByClass(Salary.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
job.setMapperClass(Map.class);
job.setInputFormatClass(splitfalse.class);
job.setOutputFormatClass(TextOutputFormat.class);
```

```
FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path(args[1]));

job.waitForCompletion(true);

}

}
```

OUTPUT:



HDFS:/rohit/salary1.csv - Mozilla Firefox
1:47 PM
ubuntu
HDFS:/rohit
HDFS:/rohit/salary1.csv
localhost:50075/browseBlock.jsp?blockId=67860798330
Google

File: /rohit/salary1.csv

Goto : /rohit go

Go back to dir listing
Advanced view/download options

```
1,60000,0,Ireland
6,58000,0,Netherlands
20,56967,0,Sweden
7,54000,0,Germany
2,70000,0,Ireland
6,39000,0,Bulgaria
5,36000,0,Romania
3,26000,0,Germany
9,43000,0,Spain
8,75000,0,Germany
5,41411,0,United Kingdom
12,36000,0,Germany
0,26000,0,France
4,40000,0,Spain
6,40000,0,United Kingdom
13,62000,0,Greece
7,30000,0,Czech Republic
0,22680,0,Portugal
10,27600,0,Hungary
5,44000,0,Sweden
10,32400,0,Croatia
4,56000,0,Latvia
10,261546,2,United Kingdom
7,40000,0,Italy
9,77000,0,Germany
0,30819,0,Poland
```

Download this file
Tail this file