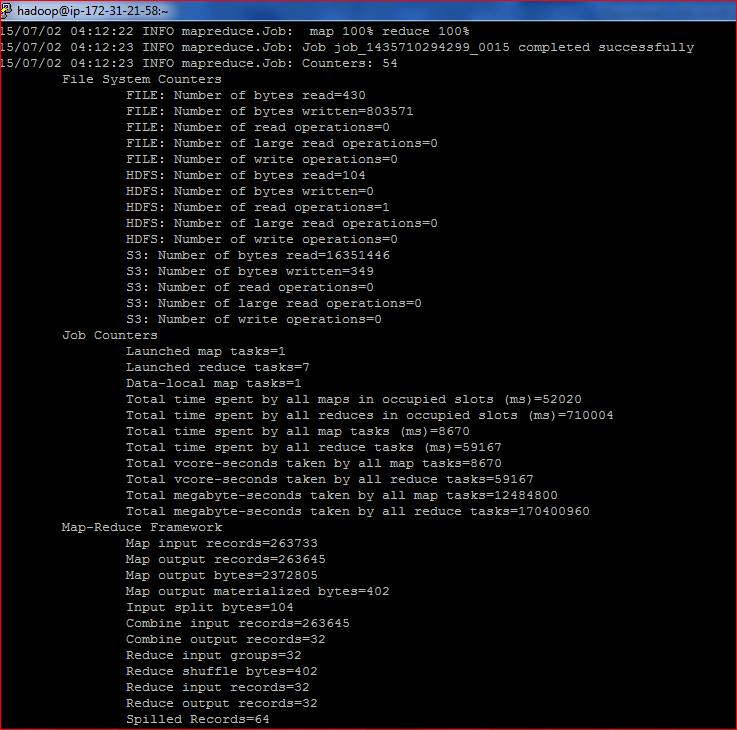
**HADOOP PROJECT**

Name: Rehana Devani

ID: 1001100807

CSE 6331: Cloud Computing-002

Screenshot of output:



I used Elastic Map Reduce for this project. Java Code was written in Eclipse. The jar was exported from Eclipse and the input file was placed in S3 bucket.

The output file was created automatically by using the following command

**hadoop jar wordcountproject.jar Worwordcountproject.jarwordcountproject.jardCount "s3://mapreduce0807/input" "s3://mapreduce0807/output9"**

The above command was entered in putty.

I used FileZilla to move JAR file from local system to Hadoop and executed commands on putty.

I authorized putty using newinstance.ppm (my keypair).

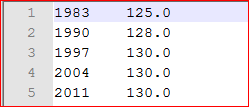
**Code for calculating average temperature annually:**

for (int i = 0; i < tokens.length; i++) {

String year = tokens[i].substring(0, 4);

Text date\_key = new Text(year);

String[] split = tokens[i].split(" ");



The sample output shows the year as the first column and the average temperature calculated annually in that year.

**Code for calculating average for Seasons:**

String line = value.toString();  
String [] path = line.split(**","**);  
String year = String.*valueOf*(Double.*parseDouble*(path[0].substring(0,4)));  
  
**double** Month= Double.*parseDouble*(path[0].substring(5, 6));  
Double temperature;  
  
  
**if**(Month<= 6)  
{  
 year=year.concat(**"Summer"**);  
}  
**else** year=year.concat(**"Winter"**);  
  
  
  
Double Tmaxtemp = Double.*parseDouble*(path[1].toString());  
Double Tmintemp = Double.*parseDouble*(path[2].toString());  
  
temperature = (Tmaxtemp+Tmintemp)/2;  
*//String quality = line.substring(92, 93);  
// && quality.matches("[01459]")***if**(Tmaxtemp != ***MISSING*** && Tmintemp != ***MISSING***) {  
 output.collect(**new** Text(year), **new** DoubleWritable(temperature));

}

Please refer the mapper and reducer class code for further details.

**TIME MEASUREMENTS:**

* + Time taken for 1 Mapper 1 reducer (1,1) : 21962 m
  + Time taken for 2 Mapper 1 reducer (2,1): 24964 ms
  + Time taken for 2 Mapper 2 reducer (2,1): 24120 ms
  + Time taken for 10 Mapper s 1 reducer (10,1): 44999 ms
  + Time taken for 10 Mapper 10 reducer (10,10) : 117030 ms