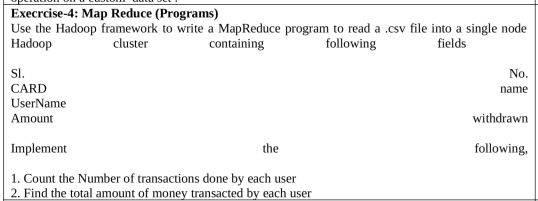
P.SESHA HARSHINI

1NT19IS107

C1 BATCH



package Harshini;

import java.io.IOException;

import java.util.\*;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.\*;

import org.apache.hadoop.mapred.\*;

public class TransactionCount {

//MAPPER CODE

public static class Map extends MapReduceBase implements

Mapper<LongWritable, Text, Text, IntWritable> {

private final static IntWritable one = new IntWritable(1);

//private Text word = new Text();

public void map(LongWritable key, Text value, OutputCollector<Text,

IntWritable> output, Reporter reporter) throws IOException {

String myString = value.toString();

String[] userCount = myString.split(",");

output.collect(new Text(userCount[3]), one);

}

}

//REDUCER CODE

public static class Reduce extends MapReduceBase implements

Reducer<Text, IntWritable, Text, IntWritable> {

public void reduce(Text key, Iterator<IntWritable> values,

OutputCollector<Text, IntWritable> output, Reporter reporter) throws

IOException { //{little: {1,1}}

int finaluserCount = 0 ;

Text mykey = key ;

while(values.hasNext()) {

IntWritable value = values.next();

finaluserCount += value.get();

}

output.collect(mykey, new IntWritable(finaluserCount));

}

}

//DRIVER CODE

public static void main(String[] args) throws Exception {

JobConf conf = new JobConf(TransactionCount.class);

conf.setJobName("wordcount");

conf.setOutputKeyClass(Text.class);

conf.setOutputValueClass(IntWritable.class);

conf.setMapperClass(Map.class);

conf.setCombinerClass(Reduce.class);

conf.setReducerClass(Reduce.class);

conf.setInputFormat(TextInputFormat.class);

conf.setOutputFormat(TextOutputFormat.class); // hadoop jar

//jarname classpath inputfolder outputfolder

FileInputFormat.setInputPaths(conf, new Path(args[0]));

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

JobClient.runJob(conf);

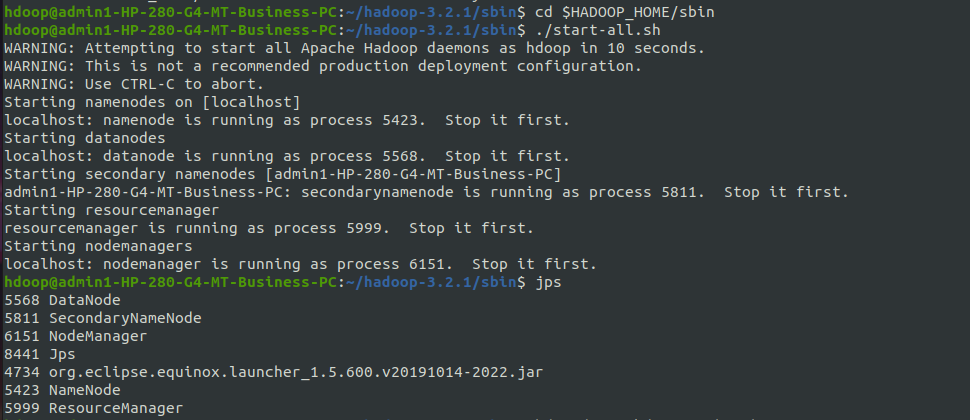
}

}

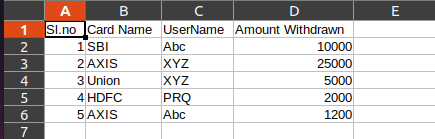
cd $HADOOP\_HOME/sbin

./start-all.sh

jps



Create a .csv file



Save the file in .csv format

Copying the 1nt19is107.csv file from local to hdfs.

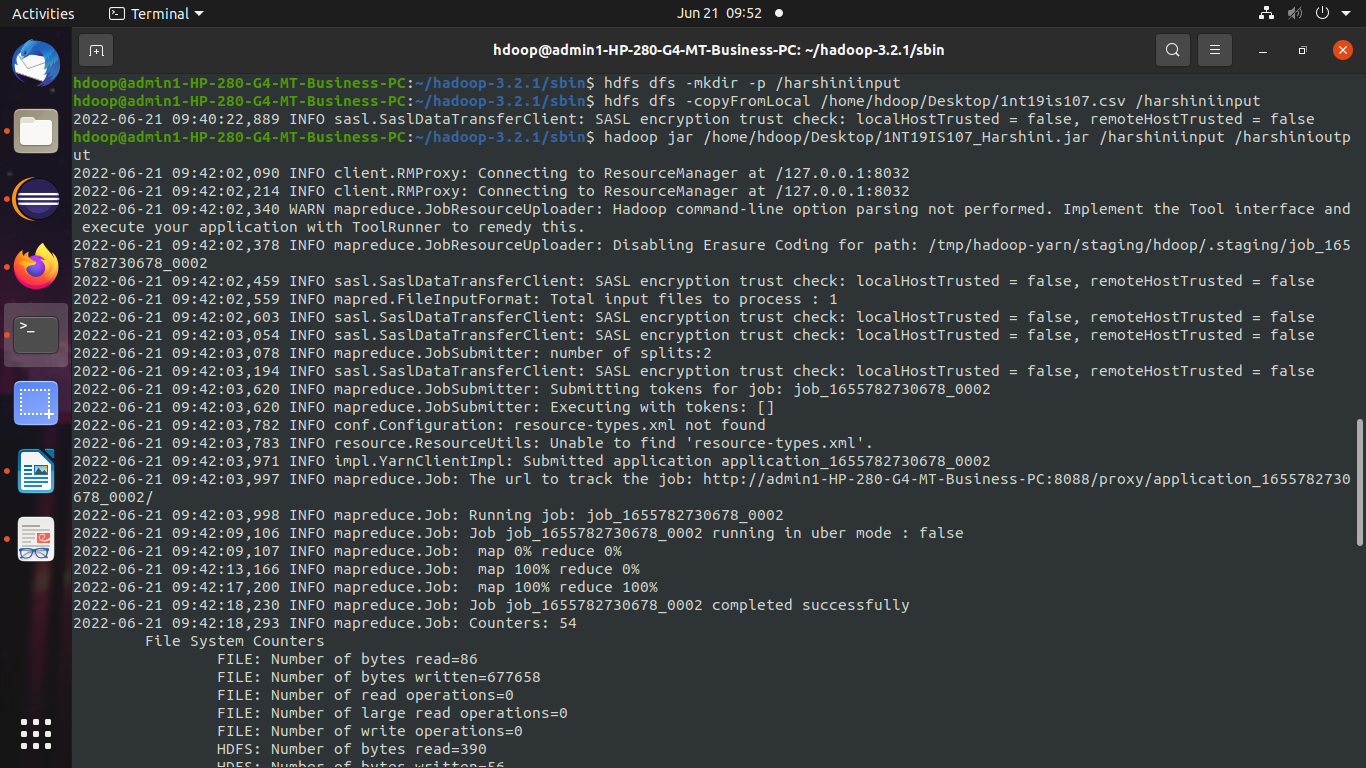
hdfs dfs -mkdir -p /harshiniinput

hdfs dfs -copyFromLocal /home/hdoop/Desktop/1nt19is107.csv /harshiniinput



hadoop jar /home/hdoop/Desktop/1NT19IS107\_Harshini.jar /harshiniinput /harshinioutput

By using the above command,we can print the class path needed to get the hadoop jar and required libraries.



To check the output

hdfs dfs -cat /harshinioutput/part\*

