package topCustByTxn;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

import java.util.Map;

import java.util.TreeMap;

import org.apache.hadoop.io.FloatWritable;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.mapreduce.Reducer;

public class MyReduce extends Reducer<IntWritable, FloatWritable, IntWritable, FloatWritable>{

/\*Input:

\* Key: 40000740(custid)

\* Values: (102.34,111.67) i.e list of txn-amount

\*

\* Output sorted by value i.e. txn-amount:

\* key: 40000740(custid)

\* value: 111.67(txn-amnt)

\*/

private static TreeMap<Float,List<Integer>> cust\_txnMap = new TreeMap<Float,List<Integer>>();

// tree-map having key=amount & value= list of custid's

@Override

protected void reduce(IntWritable key, Iterable<FloatWritable> values,

Reducer<IntWritable, FloatWritable, IntWritable, FloatWritable>.Context context) throws IOException, InterruptedException {

float txnSumForCustID = (float) 0.0;

for(FloatWritable val:values){ //Iterate the values which is of type intwritable

txnSumForCustID += val.get(); //add the count after converting intwritable to int

}

1.

//key - c0001,

//txnSumForCustID - 10000.00

-----------------------

2.

//key - c0002,

//txnSumForCustID - 10000.00

-----------------------

3.

//key - c0003,

//txnSumForCustID - 12222.00

-----------------------

4

//key - c0004,

//txnSumForCustID - 10000.00

List<Integer> tempList;//list for customers

if(cust\_txnMap.containsKey(txnSumForCustID)){ // check for presence of amount(key) in the tree-map.

tempList = cust\_txnMap.get(txnSumForCustID) ; //get the list of cust-id corresponding to amount.

}else{

tempList = new ArrayList<Integer>();

}

tempList.add(key.get()); //add the cust-id to the list after converting to int which is converted to Interger via autoboxing

cust\_txnMap.put(txnSumForCustID, tempList) ; //update tree-map with new list having additional cust-id

}

1, //cust\_txnMap(10000.00, <c0001>)

2. //cust\_txnMap(10000.00, <c0001,c0002>)

3. //cust\_txnMap(12222.00, <C0003>)

4 //cust\_txnMap(10000.00, <c0001,c0002,c0004>)

@Override

protected void cleanup(Reducer<IntWritable, FloatWritable, IntWritable, FloatWritable>.Context context) throws IOException, InterruptedException {

for (int i=0;i<10;i++){ // top 10 txns

Map.Entry<Float,List<Integer>> entryMap = cust\_txnMap.pollLastEntry(); // get the key-value with maximum key and remove from tree-map

List<Integer> tempCustIdList = entryMap.getValue(); //get the value which is a list of cust-id's

float custTxn = entryMap.getKey(); // get the txn amount

for(int cust\_count=0; cust\_count < tempCustIdList.size(); cust\_count++){ // loop for all customers in the list for same txn amount

context.write(new IntWritable(tempCustIdList.get(cust\_count)), new FloatWritable(custTxn)); // print cust-id, txn-amount

}

} // end of loop for top 10 txns

}

}