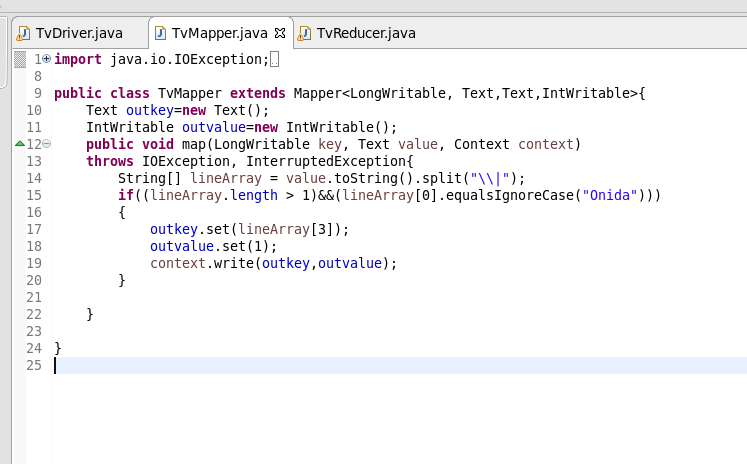
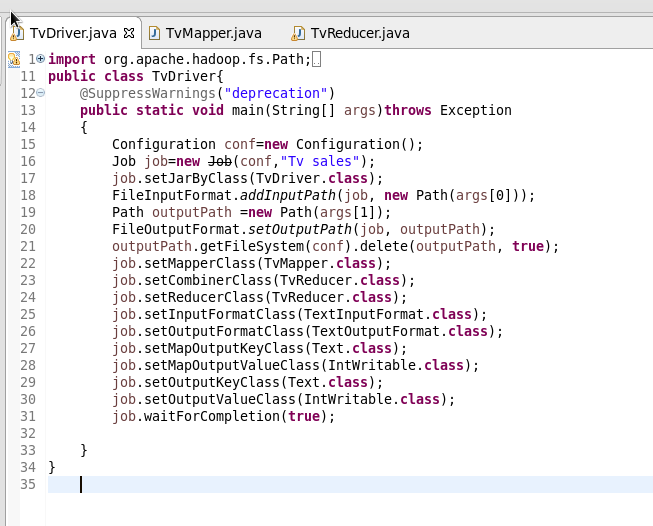
**Assignment 7.2**

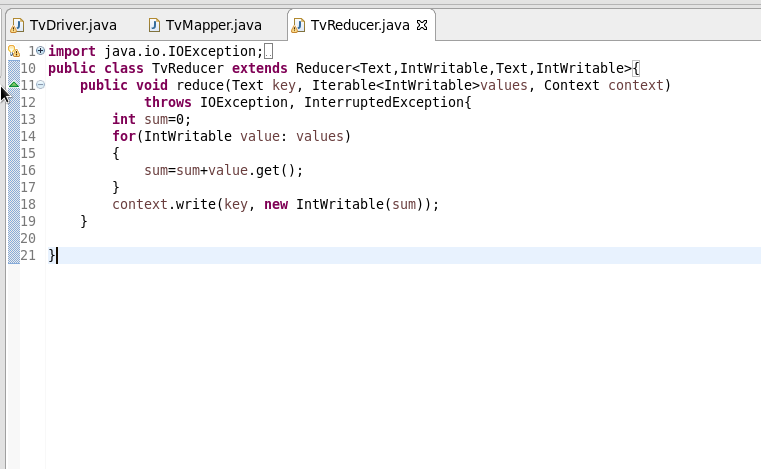
**Question:** Modify Sales of different TV Task 3 (refer session 5, assignment 2) to take advantage of Combiner.

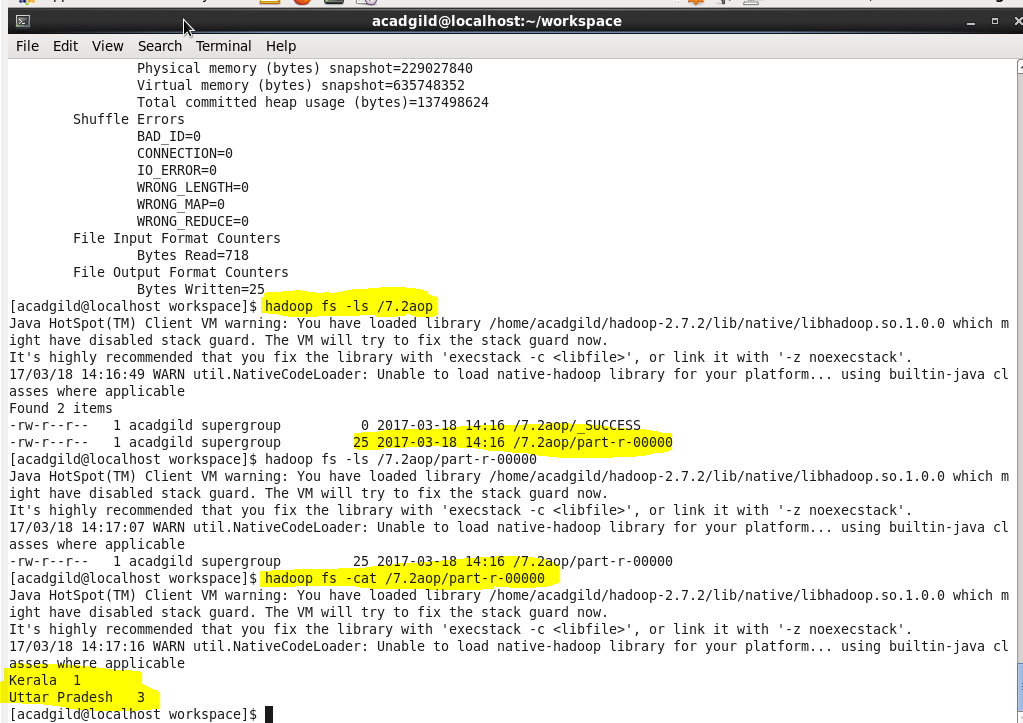
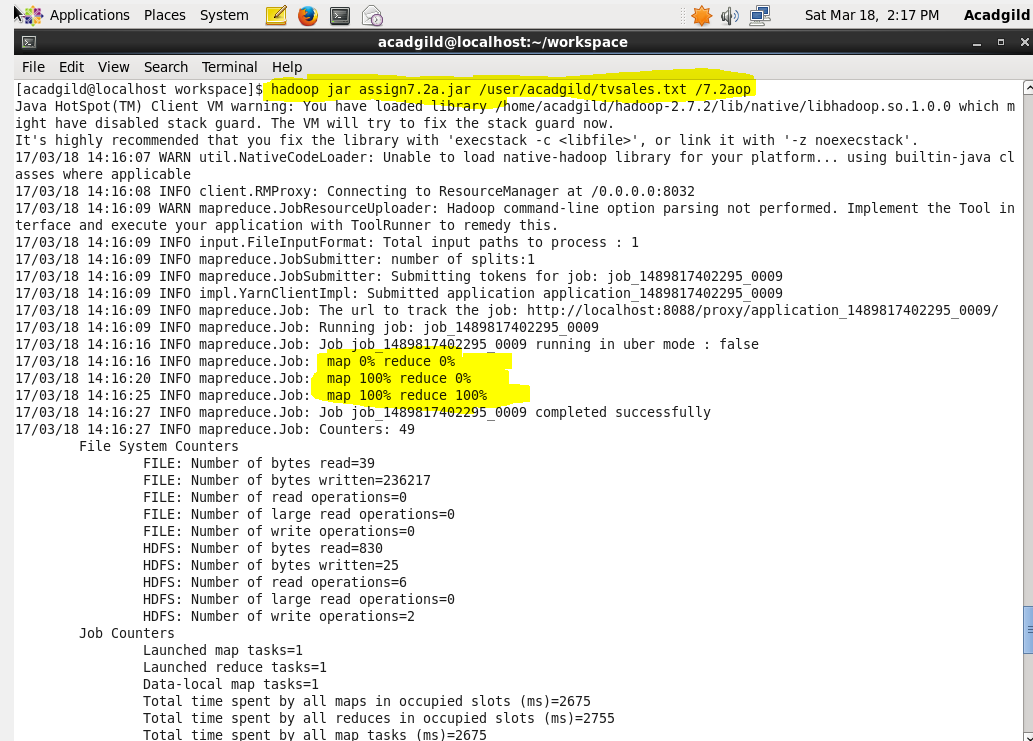
import org.apache.hadoop.fs.Path;  
import org.apache.hadoop.conf.\*;  
import org.apache.hadoop.mapreduce.Job;  
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;  
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;  
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;  
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;  
import org.apache.hadoop.io.IntWritable;  
import org.apache.hadoop.io.NullWritable;  
import org.apache.hadoop.io.Text;  
public class TvDriver{  
    @SuppressWarnings("deprecation")  
    public static void main(String[] args)throws Exception  
    {  
        Configuration conf=new Configuration();  
        Job job=new Job(conf,"Tv sales");  
        job.setJarByClass(TvDriver.class);  
        FileInputFormat.addInputPath(job, new Path(args[0]));  
        Path outputPath =new Path(args[1]);  
        FileOutputFormat.setOutputPath(job, outputPath);  
        outputPath.getFileSystem(conf).delete(outputPath, true);  
        job.setMapperClass(TvMapper.class);  
        job.setCombinerClass(TvReducer.class);  
        job.setReducerClass(TvReducer.class);  
        job.setInputFormatClass(TextInputFormat.class);  
        job.setOutputFormatClass(TextOutputFormat.class);  
        job.setMapOutputKeyClass(Text.class);  
        job.setMapOutputValueClass(IntWritable.class);  
        job.setOutputKeyClass(Text.class);  
        job.setOutputValueClass(IntWritable.class);  
        job.waitForCompletion(true);  
  
    }  
}

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.\*;  
public class TvMapper extends Mapper<LongWritable, Text,Text,IntWritable>{  
    Text outkey=new Text();  
    IntWritable outvalue=new IntWritable();  
    public void map(LongWritable key, Text value, Context context)  
    throws IOException, InterruptedException{  
        String[] lineArray = value.toString().split("\\|");  
        if((lineArray.length > 1)&&(lineArray[0].equalsIgnoreCase("Onida")))  
        {  
            outkey.set(lineArray[3]);  
            outvalue.set(1);  
            context.write(outkey,outvalue);  
        }  
    }  
}

import java.io.IOException;  
import java.util.StringTokenizer;  
import org.apache.hadoop.io.IntWritable;  
import org.apache.hadoop.io.LongWritable;  
import org.apache.hadoop.io.NullWritable;  
import org.apache.hadoop.io.Text;  
import org.apache.hadoop.mapreduce.\*;  
import org.apache.hadoop.mapreduce.Mapper.Context;  
public class TvReducer extends Reducer<Text,IntWritable,Text,IntWritable>{  
    public void reduce(Text key, Iterable<IntWritable>values, Context context)  
            throws IOException, InterruptedException{  
        int sum=0;  
        for(IntWritable value: values)  
        {  
            sum=sum+value.get();  
        }  
        context.write(key, new IntWritable(sum));  
    }  
}







**Question:** Write a Mapreduce program to view the total sales for each product for every Company corresponding to each size. Make sure that all records for a single company goes to a single reducer and inside every reducer, keys must be sorted in descending order of the size. You may write a custom WritableComparable for this purpose. "

import org.apache.hadoop.fs.Path;  
import org.apache.hadoop.conf.\*;  
import org.apache.hadoop.mapreduce.Job;  
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;  
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;  
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;  
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;  
import org.apache.hadoop.io.IntWritable;  
import org.apache.hadoop.io.NullWritable;  
import org.apache.hadoop.io.Text;  
public class TvDriver{  
    @SuppressWarnings("deprecation")  
    public static void main(String[] args)throws Exception  
    {  
        Configuration conf=new Configuration();  
        Job job=new Job(conf,"Tv sales");  
        job.setJarByClass(TvDriver.class);  
        FileInputFormat.addInputPath(job, new Path(args[0]));  
        Path outputPath =new Path(args[1]);  
        FileOutputFormat.setOutputPath(job, outputPath);  
        outputPath.getFileSystem(conf).delete(outputPath, true);  
        job.setMapperClass(TvMapper.class);  
        job.setReducerClass(TvReducer.class);  
        job.setNumReduceTasks(6);  
        job.setPartitionerClass(TvPartitioner.class);  
        job.setInputFormatClass(TextInputFormat.class);  
        job.setOutputFormatClass(TextOutputFormat.class);  
        job.setMapOutputKeyClass(TvCW.class);  
        job.setMapOutputValueClass(IntWritable.class);  
        job.setOutputKeyClass(TvCW.class);  
        job.setOutputValueClass(IntWritable.class);  
        job.waitForCompletion(true);  
  
    }  
}

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.\*;  
public class TvMapper extends Mapper<LongWritable, Text,TvCW,IntWritable>{  
    TvCW outkey=new TvCW();  
    IntWritable outvalue=new IntWritable();  
    public void map(LongWritable key, Text value, Context context)  
    throws IOException, InterruptedException{  
        String[] lineArray = value.toString().split("\\|");  
        if(lineArray.length>1)  
        {  
        outkey.set(lineArray[0],Integer.parseInt(lineArray[2]));  
            outvalue.set(1);  
        }  
            context.write(outkey,outvalue);  
    }

import java.io.IOException;  
import org.apache.hadoop.io.IntWritable;  
import org.apache.hadoop.io.LongWritable;  
import org.apache.hadoop.io.NullWritable;  
import org.apache.hadoop.io.Text;  
import org.apache.hadoop.mapreduce.\*;  
import org.apache.hadoop.mapreduce.Mapper.Context;  
public class TvPartitioner extends Partitioner<TvCW,IntWritable>{  
    public int getPartition(TvCW key,IntWritable value ,int numReduceTasks)  
    {  
        String tvname=key.toString();  
        String fc=(Character.toString(tvname.charAt(0)));  
        if(fc.matches("[L?i]"))  
        {  
            return 0;  
        }  
        else if(fc.matches("[A?i]"))  
        {  
            return 1;  
        }  
        else if(fc.matches("[S?i]"))  
        {  
            return 2;  
        }  
        else if(fc.matches("[O,o?i]"))  
        {  
            return 3;  
        }  
        else if(fc.matches("[Z?i]"))  
        {  
            return 4;  
        }  
        else  
        {  
            return 5;  
        }  
    }  
}

import java.io.DataInput;  
import java.io.DataOutput;  
import java.io.IOException;  
  
import org.apache.hadoop.io.WritableComparable;  
  
public class TvCW implements WritableComparable<TvCW>{  
    private String company;  
    private int size;  
  
    public String getCompany() {  
        return company;  
    }  
    public int getSize() {  
        return size;  
    }  
  @Override  
    public void readFields(DataInput input) throws IOException {  
        // TODO Auto-generated method stub  
        company=input.readUTF();  
    size=input.readInt();  
    }  
    @Override  
    public void write(DataOutput output) throws IOException {  
        output.writeUTF(company);  
        output.writeInt(size);  
    }  
    public String toString()  
    {  
        return company+ "\t" +size;  
    }  
   @Override  
    public int compareTo(TvCW TvCW) {  
        // TODO Auto-generated method stub  
        int cmp=company.compareTo(TvCW.company);  
        if(cmp!=0)  
        {  
            return cmp;  
        }  
        return (-1)\*(size - TvCW.getSize());  
    }  
    public int hashcode(){  
        return company.hashCode();  
    }  
    public boolean equals(Object o)  
    {  
        if(o instanceof TvCW)  
        {  
            TvCW TvCW =(TvCW) o;  
            return company.equalsIgnoreCase(TvCW.company);  
        }  
        return false;  
    }  
    public void set(String company, int size) {  
        // TODO Auto-generated method stub  
        this.company=company;  
        this.size=size;  
    }  
}

import java.io.IOException;  
import java.util.StringTokenizer;  
import org.apache.hadoop.io.IntWritable;  
import org.apache.hadoop.io.LongWritable;  
import org.apache.hadoop.io.NullWritable;  
import org.apache.hadoop.io.Text;  
import org.apache.hadoop.mapreduce.\*;  
import org.apache.hadoop.mapreduce.Mapper.Context;  
public class TvReducer extends Reducer<TvCW,IntWritable,TvCW,IntWritable>{  
    public void reduce(TvCW key, Iterable<IntWritable>values, Context context)  
            throws IOException, InterruptedException{  
        int sum=0;  
        for(IntWritable value: values)  
        {  
            sum=sum+value.get();  
        }  
        context.write(key, new IntWritable(sum));  
    }  
}

