通过五大视频网站数据，分析统计节目受欢迎度

## 1. TvPlayBean

package hdfs.tvplay;  
  
import org.apache.hadoop.io.WritableComparable;  
  
import java.io.DataInput;  
import java.io.DataOutput;  
import java.io.IOException;  
  
/\*\*  
 \* Created by dishui on 2017/9/8.  
 \*/  
public class TvPlayBean implements WritableComparable<Object> {  
  
 private String name; //名称  
 private int relation; //来源  
 private int playCount; //播放量  
 private int collectCount; //收藏数  
 private int commentCount; //评论数  
 private int stampCount; //踩数  
 private int approveCount; //赞数  
  
 public TvPlayBean() {  
 }  
  
 public TvPlayBean(String name, int relation, int playCount, int collectCount, int commentCount, int stampCount, int approveCount) {  
 this.name = name;  
 this.relation = relation;  
 this.playCount = playCount;  
 this.collectCount = collectCount;  
 this.commentCount = commentCount;  
 this.stampCount = stampCount;  
 this.approveCount = approveCount;  
 }  
  
 public void set(String name, int relation, int playCount, int collectCount, int commentCount, int stampCount, int approveCount) {  
 this.name = name;  
 this.relation = relation;  
 this.playCount = playCount;  
 this.collectCount = collectCount;  
 this.commentCount = commentCount;  
 this.stampCount = stampCount;  
 this.approveCount = approveCount;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public int getRelation() {  
 return relation;  
 }  
  
 public int getPlayCount() {  
 return playCount;  
 }  
  
 public int getCollectCount() {  
 return collectCount;  
 }  
  
 public int getCommentCount() {  
 return commentCount;  
 }  
  
 public int getStampCount() {  
 return stampCount;  
 }  
  
 public int getApproveCount() {  
 return approveCount;  
 }  
  
 public int compareTo(Object o) {  
 return 0;  
 }  
  
 public void write(DataOutput out) throws IOException {  
 out.writeUTF(name); //字符串 (2)  
 out.writeInt(relation);  
 out.writeInt(playCount);  
 out.writeInt(collectCount);  
 out.writeInt(commentCount);  
 out.writeInt(stampCount);  
 out.writeInt(approveCount);  
 }  
  
 public void readFields(DataInput in) throws IOException {  
 //TODO 读取字符串要用 readUTF  
 name = in.readUTF(); //字符串 (1)  
 relation = in.readInt();  
 playCount = in.readInt();  
 collectCount = in.readInt();  
 commentCount = in.readInt();  
 stampCount = in.readInt();  
 approveCount = in.readInt();  
 }  
}

|  |  |
| --- | --- |
| **1** | 如果用 name = in.readLine(); |
| **2** | 如果用 name = out.writeChars(name); |
|  | 报错! |
|  | C:\Users\dishui\Pictures\大讲堂\QQ截图20170909110634.png |

## 2. TvPlayInputFormat

package hdfs.tvplay;  
  
import org.apache.hadoop.conf.Configuration;  
import org.apache.hadoop.fs.FSDataInputStream;  
import org.apache.hadoop.fs.FileSystem;  
import org.apache.hadoop.fs.Path;  
import org.apache.hadoop.io.Text;  
import org.apache.hadoop.mapreduce.InputSplit;  
import org.apache.hadoop.mapreduce.JobContext;  
import org.apache.hadoop.mapreduce.RecordReader;  
import org.apache.hadoop.mapreduce.TaskAttemptContext;  
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;  
import org.apache.hadoop.mapreduce.lib.input.FileSplit;  
import org.apache.hadoop.util.LineReader;  
  
import java.io.IOException;  
  
/\*\*  
 \* Created by dishui on 2017/9/8.  
 \*/  
public class TvPlayInputFormat extends FileInputFormat<Text,TvPlayBean> {  
 @Override  
 public RecordReader<Text, TvPlayBean> createRecordReader(InputSplit inputSplit, TaskAttemptContext taskAttemptContext) throws IOException, InterruptedException {  
 return new TvPlayRecordReader();  
 }  
  
 @Override  
 protected boolean isSplitable(JobContext context, Path filename) {  
 return false;  
 }  
  
 private class TvPlayRecordReader extends RecordReader<Text, TvPlayBean> {  
  
 public LineReader in;//行读取器  
 public Text lineKey;//自定义key类型  
 public TvPlayBean lineValue;//自定义value类型  
 public Text line;//每行数据类型  
  
 @Override  
 public void initialize(InputSplit input, TaskAttemptContext context) throws IOException, InterruptedException {  
 FileSplit split=(FileSplit)input;  
 Configuration job=context.getConfiguration();  
 Path file=split.getPath();  
 FileSystem fs=file.getFileSystem(job);  
  
 FSDataInputStream filein=fs.open(file);  
 in=new LineReader(filein,job);  
 line=new Text();  
 lineKey=new Text();  
 lineValue = new TvPlayBean();  
 }  
  
 @Override  
 public boolean nextKeyValue() throws IOException, InterruptedException {  
 int linesize=in.readLine(line);//每行数据  
 if(linesize==0) return false;  
 String[] pieces = line.toString().split("\\t");//解析每行数据  
 if(pieces.length != 7){  
 throw new IOException("Invalid record received");  
 }  
 //将数据转换为 int 类型  
 String name = pieces[0];  
 int relation,playCount,collectCount,commentCount,stampCount,approveCount;  
 try{  
 relation = Integer.parseInt(pieces[1]);  
 playCount = Integer.parseInt(pieces[2]);  
 collectCount = Integer.parseInt(pieces[3]);  
 commentCount = Integer.parseInt(pieces[4]);  
 stampCount = Integer.parseInt(pieces[5]);  
 approveCount = Integer.parseInt(pieces[6]);  
 }catch(NumberFormatException nfe){  
 throw new IOException("Error parsing floating poing value in record");  
 }  
  
 switch (relation){  
 case 1:  
 lineKey.set(name+"@"+"youku");  
 break;  
 case 2:  
 lineKey.set(name+"@"+"suhu");  
 break;  
 case 3:  
 lineKey.set(name+"@"+"tudou");  
 break;  
 case 4:  
 lineKey.set(name+"@"+"aiqiyi");  
 break;  
 case 5:  
 lineKey.set(name+"@"+"xlkk");  
 break;  
 }  
// lineKey.set(name+"@"+relation);//完成自定义key数据  
 lineValue.set(name,relation, playCount, collectCount, commentCount, stampCount,approveCount);//封装自定义value数据  
 return true;  
 }  
  
 @Override  
 public Text getCurrentKey() throws IOException, InterruptedException {  
 return lineKey;  
 }  
  
 @Override  
 public TvPlayBean getCurrentValue() throws IOException, InterruptedException {  
 return lineValue;  
 }  
  
 @Override  
 public float getProgress() throws IOException, InterruptedException {  
 return 0;  
 }  
  
 @Override  
 public void close() throws IOException {  
 if(in !=null){  
 in.close();  
 }  
 }  
 }  
}

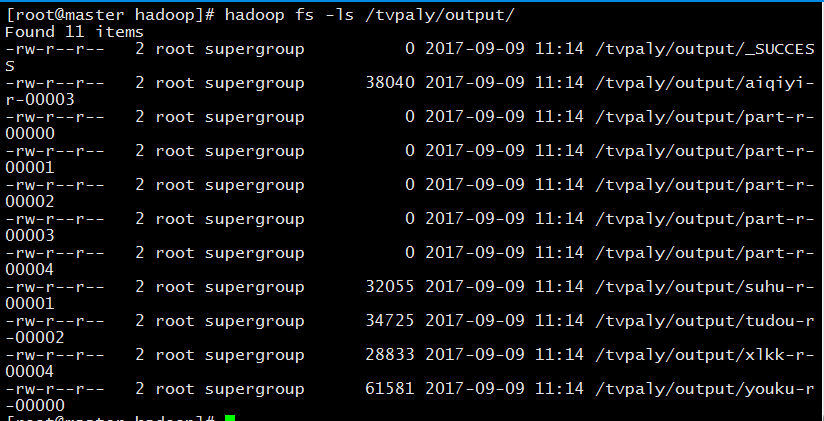
## 3. TvPlayPartitioner

package hdfs.tvplay;  
  
import org.apache.hadoop.io.Text;  
import org.apache.hadoop.mapreduce.Partitioner;  
  
public class TvPlayPartitioner extends Partitioner<Text,TvPlayBean> {  
 @Override  
 public int getPartition(Text text, TvPlayBean tvPlayBean, int i) {  
 int relation = tvPlayBean.getRelation();  
 return relation-1;  
 }  
}

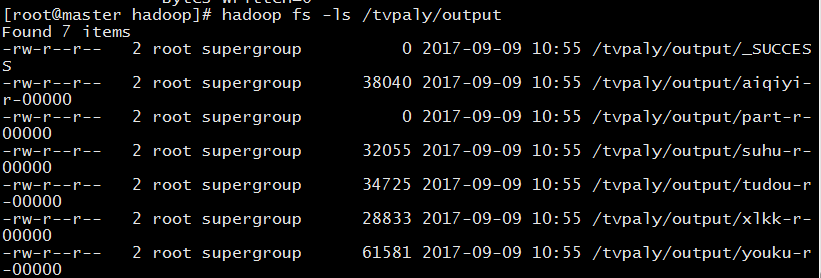
## 4. TvPlayCount

package hdfs.tvplay;  
  
import org.apache.hadoop.conf.Configuration;  
import org.apache.hadoop.conf.Configured;  
import org.apache.hadoop.fs.FileSystem;  
import org.apache.hadoop.fs.Path;  
import org.apache.hadoop.io.Text;  
import org.apache.hadoop.mapreduce.Job;  
import org.apache.hadoop.mapreduce.Mapper;  
import org.apache.hadoop.mapreduce.Reducer;  
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;  
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;  
import org.apache.hadoop.mapreduce.lib.output.MultipleOutputs;  
import org.apache.hadoop.util.Tool;  
import org.apache.hadoop.util.ToolRunner;  
  
import java.io.IOException;  
  
public class TvPlayCount extends Configured implements Tool {  
 public int run(String[] args) throws Exception {  
 Configuration conf = new Configuration();//读取配置文件  
  
 Path mypath = new Path(args[1]);  
 FileSystem hdfs = mypath.getFileSystem(conf);//创建输出路径  
 if (hdfs.isDirectory(mypath)) {  
 hdfs.delete(mypath, true);  
 }  
  
 Job job = new Job(conf, "TvPlayCount");//新建任务  
 job.setJarByClass(TvPlayCount.class);//设置主类  
  
 FileInputFormat.addInputPath(job, new Path(args[0]));// 输入路径  
 FileOutputFormat.setOutputPath(job, new Path(args[1]));// 输出路径  
  
 job.setMapperClass(TvPlayMapper.class);// Mapper  
 job.setReducerClass(TvPlayReducer.class);// Reducer  
  
 job.setPartitionerClass(TvPlayPartitioner.class); //TvPlayPartitioner (1)  
 job.setNumReduceTasks(5); (2)  
  
 job.setMapOutputKeyClass(Text.class);// Mapper key输出类型  
 job.setMapOutputValueClass(TvPlayBean.class);// Mapper value输出类型  
  
 job.setInputFormatClass(TvPlayInputFormat.class);//设置自定义输入格式  
  
 job.waitForCompletion(true);  
 return 0;  
 }  
  
 public static class TvPlayMapper extends Mapper<Text, TvPlayBean, Text, TvPlayBean> {  
 @Override  
 protected void map(Text key, TvPlayBean value, Context context) throws IOException, InterruptedException {  
 context.write(key, value);  
 }  
 }  
  
 public static class TvPlayReducer extends Reducer<Text, TvPlayBean, Text, Text> {  
 private Text text = new Text();  
 private Text text2 = new Text();  
 private MultipleOutputs<Text, Text> multipleOutputs;  
  
 @Override  
 protected void setup(Context context) throws IOException, InterruptedException {  
 multipleOutputs = new MultipleOutputs<Text, Text>(context);  
 }  
  
 @Override  
 protected void reduce(Text key, Iterable<TvPlayBean> values, Context context) throws IOException, InterruptedException {  
 String[] ss = key.toString().split("@");  
 if (ss.length == 0) {  
 return;  
 }  
 int playCountS = 0;  
 int collectCountS = 0;  
 int commentCountS = 0;  
 int stampCountS = 0;  
 int approveCountS = 0;  
 for (TvPlayBean tvPlayBean : values) {  
 playCountS += tvPlayBean.getPlayCount();  
 collectCountS += tvPlayBean.getCollectCount();  
 commentCountS += tvPlayBean.getCommentCount();  
 stampCountS += tvPlayBean.getStampCount();  
 approveCountS += tvPlayBean.getApproveCount();  
 }  
 text.set(playCountS + "\t" + collectCountS + "\t" + commentCountS + "\t" + stampCountS + "\t" + approveCountS);  
 text2.set(ss[0]);  
 multipleOutputs.write(text2, text, ss[1]);  
 }  
  
 @Override  
 protected void cleanup(Context context) throws IOException, InterruptedException {  
 multipleOutputs.close();  
 }  
 }  
  
 public static void main(String[] args) throws Exception {  
 String[] args0 = {  
 "hdfs://master:9000/tvpaly/input/tvplay.txt",  
 "hdfs://master:9000/tvpaly/output/"  
 };  
 int ec = ToolRunner.run(new Configuration(), new TvPlayCount(), args0);  
 System.exit(ec);  
 }  
  
  
}

|  |  |
| --- | --- |
| **1** | 设置 setPartitionerClass 和 setNumReduceTasks 后,会生成多余的空文件 |



2设置 setPartitionerClass 和 setNumReduceTasks 后,会出现一个多余的空文件



老师!为什么会出现这种情况?