Program :

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

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;

public class MusicTrackAnalysis {

public static class TokenizerMapper extends Mapper<Object, Text, Text, Text> {

private final static Text one = new Text("1");

private Text word = new Text();

public void map(Object key, Text value, Context context) throws IOException, InterruptedException {

// Splitting the input line by comma

String[] parts = value.toString().split(",");

if (parts.length == 3) {

String track\_id = parts[0].trim();

String source = parts[1].trim();

String action = parts[2].trim();

// Filtering for tracks played on Radio

if (source.equals("Radio")) {

word.set(track\_id + ",listened");

context.write(word, one);

} else if (action.equals("skipped")) {

word.set(track\_id + ",skipped");

context.write(word, one);

}

}

}

}

public static class IntSumReducer extends Reducer<Text, Text, Text, IntWritable> {

private IntWritable result = new IntWritable();

public void reduce(Text key, Iterable<Text> values, Context context) throws IOException, InterruptedException {

int sum = 0;

for (Text val : values) {

sum += Integer.parseInt(val.toString());

}

result.set(sum);

context.write(key, result);

}

}

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

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "music track analysis");

job.setJarByClass(MusicTrackAnalysis.class);

job.setMapperClass(TokenizerMapper.class);

job.setCombinerClass(IntSumReducer.class);

job.setReducerClass(IntSumReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(Text.class);

FileInputFormat.addInputPath(job, new Path(args[0]));

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

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}

File :

track1,shared

track2,shared

track3,shared

track1,shared

track2,shared

track4,shared

track1,shared

track5,played

track3,shared

track2,shared

Output :

[cloudera@quickstart ~]$ hadoop fs -put Desktop/likely.txt likely.txtput: `likely.txt': File exists

[cloudera@quickstart ~]$ hadoop jar flight.jar pict.Music likely.txt likelyoutput

24/05/08 21:25:48 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032

24/05/08 21:25:49 WARN security.UserGroupInformation: PriviledgedActionException as:cloudera (auth:SIMPLE) cause:org.apache.hadoop.mapred.FileAlreadyExistsException: Output directory hdfs://quickstart.cloudera:8020/user/cloudera/likelyoutput already exists

Exception in thread "main" org.apache.hadoop.mapred.FileAlreadyExistsException: Output directory hdfs://quickstart.cloudera:8020/user/cloudera/likelyoutput already exists

at org.apache.hadoop.mapreduce.lib.output.FileOutputFormat.checkOutputSpecs(FileOutputFormat.java:146)

at org.apache.hadoop.mapreduce.JobSubmitter.checkSpecs(JobSubmitter.java:562)

at org.apache.hadoop.mapreduce.JobSubmitter.submitJobInternal(JobSubmitter.java:432)

at org.apache.hadoop.mapreduce.Job$10.run(Job.java:1306)

at org.apache.hadoop.mapreduce.Job$10.run(Job.java:1303)

at java.security.AccessController.doPrivileged(Native Method)

at javax.security.auth.Subject.doAs(Subject.java:415)

at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1671)

at org.apache.hadoop.mapreduce.Job.submit(Job.java:1303)

at org.apache.hadoop.mapreduce.Job.waitForCompletion(Job.java:1324)

at pict.Music.main(Music.java:57)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:606)

at org.apache.hadoop.util.RunJar.run(RunJar.java:221)

at org.apache.hadoop.util.RunJar.main(RunJar.java:136)

[cloudera@quickstart ~]$ hadoop fs -cat likelyoutput/part-r-00000track1 3

track2 3

track3 2

track4 1

[cloudera@quickstart ~]$