

Podcasts Analytics via MapReduce*

Montasser AKERMI

Last Update: March, 2025

*The course material is hosted at <https://akermi.org/c2/>.

Learning objectives

A podcast company hosts a platform where users can listen to various podcasts. The data were collected as shown below. Write a MapReduce application that measures:

- The number of unique listeners of each podcast.
- The number of times each podcast was skipped.
- The total number of times each podcast was listened to.

Input data

UserId;PodcastName;Skip

```
1;Foo;False
2;Bar;True
3;Baz;False
2;Foo;True
1;Foo;True
1;Baz;False
2;Bar;False
1;Baz;False
1;Foo;True
```

Output data

PodcastName;UniqueListeners;TimesSkipped;TotalListened

```
Foo;2;3;4
Bar;1;1;2
Baz;2;0;3
```

Solution

Listing 1: Mapper class

```
1 import java.io.IOException;
2
3 import org.apache.hadoop.io.Text;
4 import org.apache.hadoop.mapreduce.Mapper;
5
6 public class PodcastMapper extends Mapper<Object, Text, Text, Text> {
7     private static final int USER_ID = 0;
8     private static final int PODCAST_NAME = 1;
9     private static final int SKIP = 2;
10
11     private Text userId = new Text();
12     private Text podcastName = new Text();
13     private Text skip = new Text();
14
15     public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
16         String[] data = value.toString().split(";");
17         userId.set(data[USER_ID]);
18         podcastName.set(data[PODCAST_NAME]);
```

```

19     skip.set(data[SKIP]);
20
21     if (data.length == 3) {
22         context.write(podcastName, new Text(userId.toString() + ";" + skip.toString());
23     } else {
24         context.getCounter(COUNTERS.INVALID_RECORD_COUNT).increment(1L);
25     }
26 }
27 }

```

Listing 2: Reducer class

```

1  import java.io.IOException;
2  import java.util.HashSet;
3  import java.util.Set;
4
5  import org.apache.hadoop.io.Text;
6  import org.apache.hadoop.mapreduce.Reducer;
7
8  public class PodcastReducer extends Reducer<Text, Text, Text, Text> {
9      private static final int USER_ID = 0;
10     private static final int SKIP = 1;
11
12     public void reduce(Text key, Iterable<Text> value, Context context)
13         throws IOException, InterruptedException {
14         int total = 0;
15         int skipped = 0;
16         Set<Integer> userSet = new HashSet<Integer>();
17
18         for (Text record : value) {
19             String[] data = record.toString().split(";");
20             total++;
21             userSet.add(Integer.parseInt(data[USER_ID]));
22             if (Boolean.parseBoolean(data[SKIP])) {
23                 skipped++;
24             }
25         }
26         context.write(key, new Text(userSet.size() + ";" + skipped + ";" + total));
27     }
28 }

```

Listing 3: PodcastAnalytics.java

```

1  import java.io.IOException;
2
3  import org.apache.hadoop.conf.Configuration;
4  import org.apache.hadoop.fs.Path;
5  import org.apache.hadoop.io.Text;
6  import org.apache.hadoop.mapreduce.Job;
7  import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
8  import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
9
10 public class PodcastAnalytics {
11
12     public static void main(String[] args)

```

```

13     throws IOException, ClassNotFoundException, InterruptedException {
14     Configuration conf = new Configuration();
15     conf.set("mapred.textoutputformat.separator", ";");
16     Job job = Job.getInstance(conf, "Podcasts Analytics");
17     job.setJarByClass(PodcastAnalytics.class);
18     job.setMapperClass(PodcastMapper.class);
19     job.setReducerClass(PodcastReducer.class);
20     job.setOutputKeyClass(Text.class);
21     job.setOutputValueClass(Text.class);
22     FileInputFormat.addInputPath(job, new Path(args[0]));
23     FileOutputFormat.setOutputPath(job, new Path(args[1]));
24     System.exit(job.waitForCompletion(true) ? 0 : 1);
25 }
26 }

```

Listing 4: COUNTERS.java

```

1 public enum COUNTERS {
2     INVALID_RECORD_COUNT
3 }

```

Note: It is highly recommended to manually write the code instead of copy/pasting.