

# Assignment-5

1). Find the number of unique listeners in the data set.

Map Reduce for the above task is as below:

```
package Tasks;

// imported all the built-in packages required for the task

import java.io.IOException;
import java.util.HashSet;
import java.util.Set;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.fs.Path;

public class Uniquelisteners
{
    //Mapper class
    public static class Map extends Mapper <Object ,/*Input key
Type */
Text,                               /*Input value Type*/
IntWritable,                        /*Output key Type*/
IntWritable>                        /*Output value Type*/
    {
        //Map function
        public void map(Object key, Text value,
Context context) throws IOException,
InterruptedException
        {
            IntWritable trackid = new IntWritable();

            // Given data is splitted into an array based on the delimiter
            // present in the data.

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

            trackid.set(Integer.parseInt(parts[1]));

            //The value(song id) in the second position of parts is set to
            //the variable (trackid)and it sent to the output as key. The value to
            //the key in the output of map passed as "1".

            context.write(trackid, new IntWritable(1));
```

```

        System.out.println("trackId and userId" + trackid + userid);
    }
}

//Reducer class (Received output from the mapper as IntWritable
and intWritable)
public static class Reduce extends Reducer< IntWritable,
IntWritable, IntWritable, IntWritable >
{
    //Reduce function
    public void reduce( IntWritable trackid,
Iterable<IntWritable> values,Context context) throws IOException,
InterruptedException
    {
        int sum =0;
        //the below step will check for the each value of a key.

        for(IntWritable val:values) {

            //each value will be added to the variable "sum"

            sum += val.get();
        }
        System.out.println("size of userid" + sum);

        context.write( trackid, new IntWritable(sum));

        //Now the output will be with key (trackid) and value (count)

    }
}

//Main function
public static void main(String args[])throws Exception
{
    Configuration conf = new Configuration();

    Job job = Job.getInstance(conf, "wordcount");

    job.setJarByClass(Uniquelisters.class);
    job.setOutputKeyClass(IntWritable.class);
    job.setOutputValueClass(IntWritable.class);
    job.setMapperClass(Map.class);
    job.setCombinerClass(Reduce.class);
    job.setReducerClass(Reduce.class);
    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);

    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    Path outputPath = new Path(args[1]);
    outputPath.getFileSystem(conf).delete(outputPath, true);

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

```

```
}
}
```

Executed the code **UniqueListeners.jar** with input file **musicdata.txt**

```
acadgild@localhost:~
File Edit View Search Terminal Tabs Help
acadgild@localhost:~
[acadgild@localhost ~]$ hadoop jar /home/acadgild/UniqueListeners.jar /musicdata.txt /myoutput
18/09/11 22:25:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/09/11 22:25:56 INFO client.RMPProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/09/11 22:25:57 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool in interface and execute your application with ToolRunner to remedy this.
18/09/11 22:25:57 INFO input.FileInputFormat: Total input paths to process : 1
18/09/11 22:25:58 INFO mapreduce.JobSubmitter: number of splits:1
18/09/11 22:25:58 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1536683231111_0004
18/09/11 22:25:58 INFO impl.YarnClientImpl: Submitted application application_1536683231111_0004
18/09/11 22:25:58 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1536683231111_0004/
18/09/11 22:25:58 INFO mapreduce.Job: Running job: job_1536683231111_0004
18/09/11 22:26:09 INFO mapreduce.Job: Job job_1536683231111_0004 running in uber mode : false
18/09/11 22:26:09 INFO mapreduce.Job:  map 0% reduce 0%
18/09/11 22:26:18 INFO mapreduce.Job:  map 100% reduce 0%
18/09/11 22:26:26 INFO mapreduce.Job:  map 100% reduce 100%
18/09/11 22:26:27 INFO mapreduce.Job: Job job_1536683231111_0004 completed successfully
18/09/11 22:26:27 INFO mapreduce.Job: Counters: 49
    File System Counters
        FILE: Number of bytes read=36
        FILE: Number of bytes written=215719
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=168
        HDFS: Number of bytes written=18
        HDFS: Number of read operations=6
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=2
    Job Counters
        Launched map tasks=1
        Launched reduce tasks=1
        Data-local map tasks=1
        Total time spent by all maps in occupied slots (ms)=6549
        Total time spent by all reduces in occupied slots (ms)=5869
        Total time spent by all map tasks (ms)=6549
        Total time spent by all reduce tasks (ms)=5869
```

Got the required output as shown in the below screenshot.

```
acadgild@localhost:~
File Edit View Search Terminal Tabs Help
acadgild@localhost:~
[acadgild@localhost ~]$ hadoop fs -cat /myoutput/part-r-00000
18/09/11 22:26:47 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
222      1
223      1
225      2
[acadgild@localhost ~]$
```

2). What are the number of times a song was heard fully.

Map Reduce for the above task is as below:

```
package Tasks;
```

```
// imported all the built-in packages required for the task
```

```
import java.io.IOException;
import java.util.ArrayList;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import
org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import
org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.fs.Path;

    public class FullSong
    {
        //Mapper class
        public static class Map extends Mapper <LongWritable
,/*Input key Type */
        Text,                                /*Input value Type*/
        IntWritable,                        /*Output key Type*/
        IntWritable>                        /*Output value Type*/
        {
            //Map function
            public void map(LongWritable key, Text value,
                Context context) throws IOException,
InterruptedException
            {
                IntWritable trackid = new IntWritable();
                IntWritable fullSong = new IntWritable();

                // Given data is splitted into an array based on the delimiter
present in the data.

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

                if(!parts[4].equals("0")) {
```

**//If the Full song status is not equals "0" satisfies then it will perform the following steps.**

```
        trackid.set(Integer.parseInt(parts[1]));
        fullSong.set(Integer.parseInt(parts[4]));
//Trackid and full song is passed as key and value to the output respectively.
        context.write(trackid, fullSong);
        System.out.println("trackId and fullSong" +
trackid + fullSong);

    }
}
}
```

**//Reducer class (Received output from the mapper as IntWritable and intWritable)**

```
public static class Reduce extends Reducer<
IntWritable, IntWritable, IntWritable, IntWritable >
{
    //Reduce function
    public void reduce( IntWritable trackid,
Iterable<IntWritable> fullSongs,Context context) throws
IOException, InterruptedException
    {
        int sum =0;
```

**//the below step will check for the each value of a key.**

```
        for(IntWritable fullsong:fullSongs) {

//each value will be added to the variable "sum"
            sum +=fullsong.get();
        }
        System.out.println("size of userid" + sum);
        context.write( trackid, new IntWritable(sum));
//Now the output will be with key (trackid) and value (count)
    }
}
```

**//Main function**

```
public static void main(String args[])throws Exception
{
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "wordcount");

    job.setJarByClass(FullSong.class);
    job.setOutputKeyClass(IntWritable.class);
    job.setOutputValueClass(IntWritable.class);
    job.setMapperClass(Map.class);
    job.setCombinerClass(Reduce.class);
```

```

        job.setReducerClass(Reduce.class);
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        FileInputFormat.addInputPath(job, new
Path(args[0]));
        FileOutputFormat.setOutputPath(job, new
Path(args[1]));
        Path outputPath = new Path(args[1]);
        outputPath.getFileSystem(conf).delete(outputPath,
true);
        System.exit(job.waitForCompletion(true)? 0 :1);
    }
}

```

Executed the code **FullSong.jar** with input file **musicdata.txt**

```

acadgild@localhost:~
File Edit View Search Terminal Tabs Help
acadgild@localhost:~
[acadgild@localhost ~]$ hadoop jar /home/acadgild/FullSong.jar /musicdata.txt /myoutput
18/09/09 19:26:44 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
18/09/09 19:26:46 INFO client.RMPProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/09/09 19:26:48 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool in
terface and execute your application with ToolRunner to remedy this.
18/09/09 19:26:49 INFO input.FileInputFormat: Total input paths to process : 1
18/09/09 19:26:49 INFO mapreduce.JobSubmitter: number of splits:1
18/09/09 19:26:49 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1536493456864_0013
18/09/09 19:26:50 INFO impl.YarnClientImpl: Submitted application application_1536493456864_0013
18/09/09 19:26:50 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1536493456864_0013/
18/09/09 19:26:50 INFO mapreduce.Job: Running job: job_1536493456864_0013
18/09/09 19:27:07 INFO mapreduce.Job: Job job_1536493456864_0013 running in uber mode : false
18/09/09 19:27:07 INFO mapreduce.Job:  map 0% reduce 0%
18/09/09 19:27:23 INFO mapreduce.Job:  map 100% reduce 0%
18/09/09 19:27:38 INFO mapreduce.Job:  map 100% reduce 100%
18/09/09 19:27:38 INFO mapreduce.Job: Job job_1536493456864_0013 completed successfully
18/09/09 19:27:38 INFO mapreduce.Job: Counters: 49
    File System Counters
        FILE: Number of bytes read=16
        FILE: Number of bytes written=215637
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=168
        HDFS: Number of bytes written=6
        HDFS: Number of read operations=6
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=2
    Job Counters
        Launched map tasks=1
        Launched reduce tasks=1
        Data-local map tasks=1
        Total time spent by all maps in occupied slots (ms)=13643
        Total time spent by all reduces in occupied slots (ms)=11514
        Total time spent by all map tasks (ms)=13643
        Total time spent by all reduce tasks (ms)=11514

```

Got the required output as shown in the below screenshot.

```

acadgild@localhost:~
File Edit View Search Terminal Tabs Help
acadgild@localhost:~
[acadgild@localhost ~]$ hadoop fs -cat /myoutput/part-r-00000
18/09/09 19:31:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
223      1
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$

```

3). What are the number of times a song was shared.

Map Reduce code for the above task is as below:

```
package Tasks;

// imported all the built-in packages required for the task

import java.io.IOException;
import java.util.ArrayList;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import
org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import
org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.fs.Path;

public class SongShared
{
    //Mapper class
    public static class Map extends Mapper <LongWritable
,/*Input key Type */
    Text,                      /*Input value Type*/
    IntWritable,              /*Output key Type*/
    IntWritable>              /*Output value Type*/
    {
        //Map function
        public void map(LongWritable key, Text value,
            Context context) throws IOException,
InterruptedException
        {
            IntWritable trackid = new IntWritable();
            IntWritable songShared = new IntWritable();

// Given data is splitted into an array based on the delimiter
present in the data.

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

            if(!parts[2].equals("0")) {

//If the song shared status is not equals "0" satisfies then
it will perform the following steps.

```

```

        trackid.set(Integer.parseInt(parts[1]));
        songShared.set(Integer.parseInt(parts[2]));

        //Trackid and song share status is passed as key and
        value to the output respectively.

        context.write(trackid, songShared);
        System.out.println("trackId and fullSong" +
trackid + songShared);
    }
}

//Reducer class (Received output from the mapper as
IntWritable and intWritable)
public static class Reduce extends Reducer<
IntWritable, IntWritable, IntWritable, IntWritable >
{
    //Reduce function
    public void reduce( IntWritable trackid,
Iterable<IntWritable> songShare,Context context) throws
IOException, InterruptedException
    {
        int sum =0;

        //the below step will check for the each value.

        for(IntWritable songShared:songShare) {

            //each value will be added to the variable "sum"

            sum +=songShared.get();
        }
        System.out.println("size of userid" + sum);
        context.write( trackid, new IntWritable(sum));

        //Now the output will be with key (trackid) and value
(count)
    }
}

//Main function
public static void main(String args[])throws Exception
{
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "SongShared");

    job.setJarByClass(SongShared.class);
    job.setOutputKeyClass(IntWritable.class);
    job.setOutputValueClass(IntWritable.class);
}

```



```

        job.setMapperClass(Map.class);
        job.setCombinerClass(Reduce.class);
        job.setReducerClass(Reduce.class);
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        Path outputPath = new Path(args[1]);
        outputPath.getFileSystem(conf).delete(outputPath, true);

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

```

Executed the code **SongShared.jar** with input file **musicdata.txt**

```

acadgild@localhost:~
File Edit View Search Terminal Tabs Help
acadgild@localhost:~
[acadgild@localhost ~]$ hadoop jar /home/acadgild/SongShared.jar /musicdata.txt /myoutput
18/09/09 19:34:41 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
18/09/09 19:34:43 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/09/09 19:34:44 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool in
terface and execute your application with ToolRunner to remedy this.
18/09/09 19:34:44 INFO input.FileInputFormat: Total input paths to process : 1
18/09/09 19:34:45 INFO mapreduce.JobSubmitter: number of splits:1
18/09/09 19:34:45 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1536493456864_0014
18/09/09 19:34:46 INFO impl.YarnClientImpl: Submitted application application_1536493456864_0014
18/09/09 19:34:46 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1536493456864_0014/
18/09/09 19:34:46 INFO mapreduce.Job: Running job: job_1536493456864_0014
18/09/09 19:34:57 INFO mapreduce.Job: Job job_1536493456864_0014 running in uber mode : false
18/09/09 19:34:57 INFO mapreduce.Job: map 0% reduce 0%
18/09/09 19:35:07 INFO mapreduce.Job: map 100% reduce 0%
18/09/09 19:35:17 INFO mapreduce.Job: map 100% reduce 100%
18/09/09 19:35:17 INFO mapreduce.Job: Job job_1536493456864_0014 completed successfully
18/09/09 19:35:18 INFO mapreduce.Job: Counters: 49
    File System Counters
        FILE: Number of bytes read=16
        FILE: Number of bytes written=215651
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=168
        HDFS: Number of bytes written=6
        HDFS: Number of read operations=6
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=2
    Job Counters
        Launched map tasks=1
        Launched reduce tasks=1
        Data-local map tasks=1
        Total time spent by all maps in occupied slots (ms)=7863
        Total time spent by all reduces in occupied slots (ms)=6684
        Total time spent by all map tasks (ms)=7863
        Total time spent by all reduce tasks (ms)=6684

```

Got the required output as shown in the below screenshot.

```

acadgild@localhost:~
File Edit View Search Terminal Tabs Help
acadgild@localhost:~
[acadgild@localhost ~]$ hadoop fs -cat /myoutput/part-r-00000
18/09/09 19:36:08 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
225      2
[acadgild@localhost ~]$

```

