Unique Word Count with MapReduce*

Montasser AKERMI

Last Update: March, 2025

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

Learning objectives

Write a MapReduce application that counts the number unique words in a corpus of text of one or multiple files.

Input data

```
my dreams my dreams
what has become of their sweetness
what indeed has become of my youth
```

Output data

```
Unique words  10
```

Solution

Listing 1: UniqueWordCount Class

```
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 UniqueWordCount {

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

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context
    ) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString());
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
      }
    }
  }

  public static class LineMapper
      extends Mapper<Object, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1);
    private final static Text word = new Text(”Unique words”);

    public void map(Object key, Text value, Context context
    ) throws IOException, InterruptedException {
      context.write(word, one);
    }
  }

  public static class IntSumReducer
      extends Reducer<Text, IntWritable, Text, IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values,
                       Context context
    ) throws IOException, InterruptedException {
      int sum = 0;
      for (IntWritable val : values) {
        sum += val.get();
      }
      result.set(sum);
      context.write(key, result);
    }
  }

  public static void main(String[] args) throws Exception {
    Configuration conf1 = new Configuration();
    Job job1 = Job.getInstance(conf1, "Distinct word count - step 1”);
    job1.setJarByClass(UniqueWordCount.class);
    job1.setMapperClass(TokenizerMapper.class);
    job1.setCombinerClass(IntSumReducer.class);
    job1.setReducerClass(IntSumReducer.class);
    job1.setOutputKeyClass(Text.class);
    job1.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job1, new Path(args[0]));
    FileOutputFormat.setOutputPath(job1, new Path(args[1]));
    job1.waitForCompletion(true);

    Configuration conf2 = new Configuration();
    Job job2 = Job.getInstance(conf2, "Distinct word count - step 2”);
    job2.setJarByClass(UniqueWordCount.class);
    job2.setMapperClass(LineMapper.class);
    job2.setCombinerClass(IntSumReducer.class);
    job2.setReducerClass(IntSumReducer.class);
    job2.setOutputKeyClass(Text.class);
    job2.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job2, new Path(args[1]));
    FileOutputFormat.setOutputPath(job2, new Path(args[2]));
    System.exit(job2.waitForCompletion(true) ? 0 : 1);
  }
}
```

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