# Average WordCount:

## AvgDriver.java

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.FloatWritable; import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; import org.apache.hadoop.mapreduce.Job;

public class AvgDriver {

public static void main(String[] args) throws Exception { if (args.length != 2) {

System.out.printf("Usage: WordCount <input dir> <output

dir>\n");

System.exit(-1);

}

Job job = new Job();

job.setJarByClass(AvgDriver.class); job.setJobName("Word Count");

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

job.setMapperClass(AvgMapper.class);

job.setReducerClass(AvgReducer.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(FloatWritable.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(FloatWritable.class);

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

}

}

## AvgMapper.java

import java.io.IOException;

import org.apache.hadoop.io.FloatWritable; import org.apache.hadoop.io.LongWritable; import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

public class AvgMapper extends Mapper<LongWritable, Text, Text, FloatWritable> {

@Override

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

String line = value.toString();

for (String word : line.split("\\W+")) { if (word.length() > 0) {

context.write(new Text(word), new

FloatWritable(1));

}

}

}

}

## AvgReducer.java

import java.io.IOException;

import org.apache.hadoop.io.FloatWritable; import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

public class AvgReducer extends Reducer<Text, FloatWritable, Text, FloatWritable> {

float wordCount = 0.0f; float sum=0.0f;

@Override

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

for (FloatWritable value : values) { wordCount += value.get();

}

sum=sum+1;

context.write(key, new FloatWritable(wordCount/sum));

}

## Command:

hadoop jar WordCountAverage.jar AvgDriver sample-file.txt ResultWordCountAverage