Roll No: 20BCE204

Course Name and Course Code: 2CS702 Big Data Analytics

Practical No: 6

Aim: Configure the number of mappers and reducers to execute practical 4 and

5. Evaluate the results and prepare a report on performance.

Code:

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* IntSum {

*public* *static* *class* TokenizerMapper

*extends* Mapper<Object, Text, Text, IntWritable>{

*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()) {

String s = itr.nextToken();

word.set("SUM");

IntWritable one = new IntWritable(Integer.parseInt(s));

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 conf = new Configuration();

Job job = Job.getInstance(conf, "word count");

job.setJarByClass(IntSum.class);

job.setMapperClass(TokenizerMapper.class);

job.setCombinerClass(IntSumReducer.class);

job.setReducerClass(IntSumReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

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

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

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

}

}

Steps in terminal:

hadoop fs -mkdir /Practical6

hadoop fs -mkdir /Practical6/Input

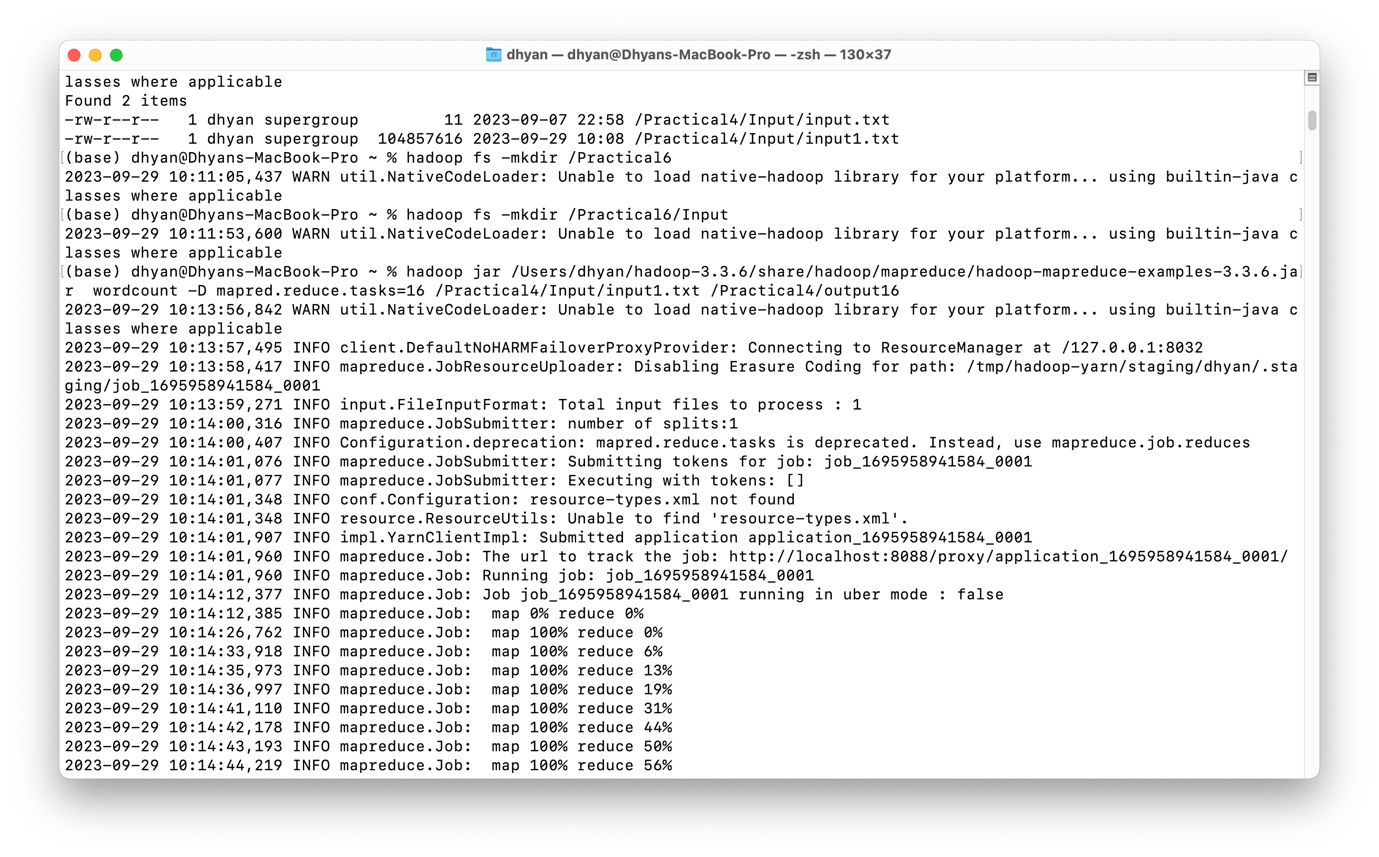
hadoop jar /Users/dhyan/hadoop-3.3.6/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar wordcount -D mapred.reduce.tasks=16 /Practical4/Input/input1.txt /Practical4/output16

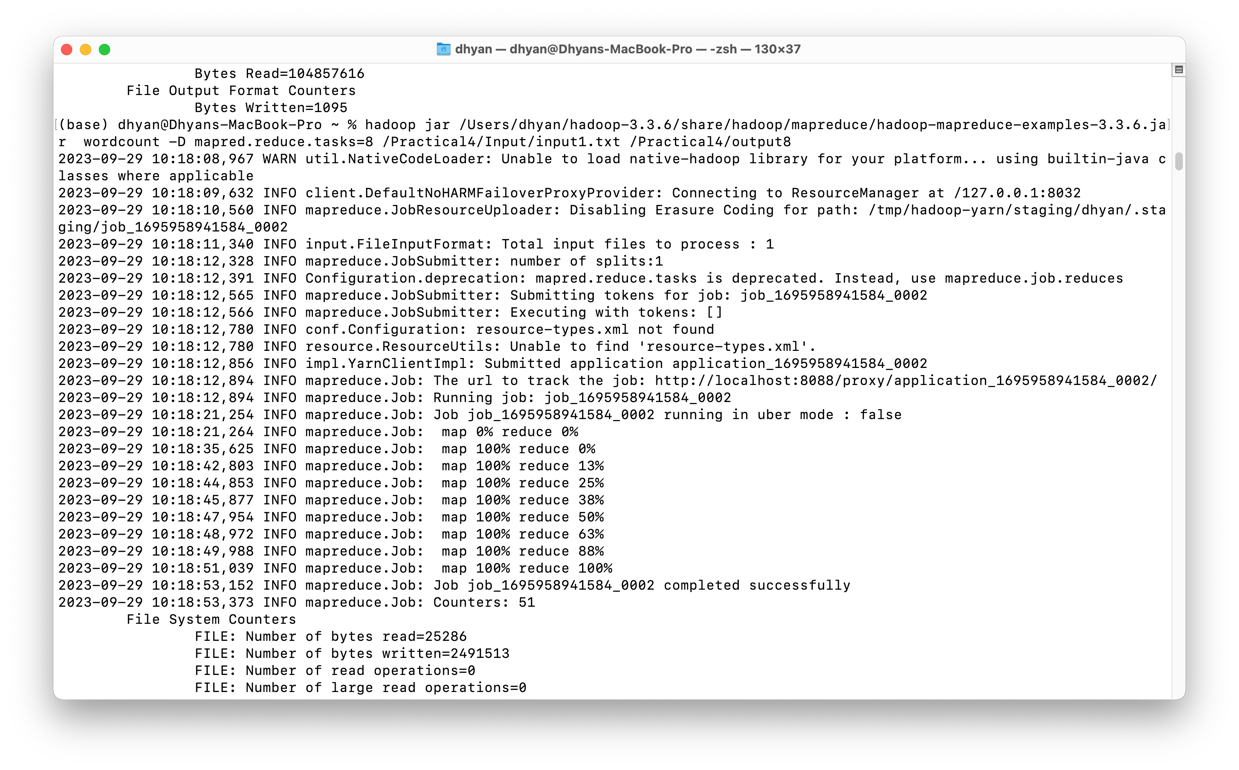
hadoop jar /Users/dhyan/hadoop-3.3.6/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar wordcount -D mapred.reduce.tasks=8 /Practical4/Input/input1.txt /Practical4/output8

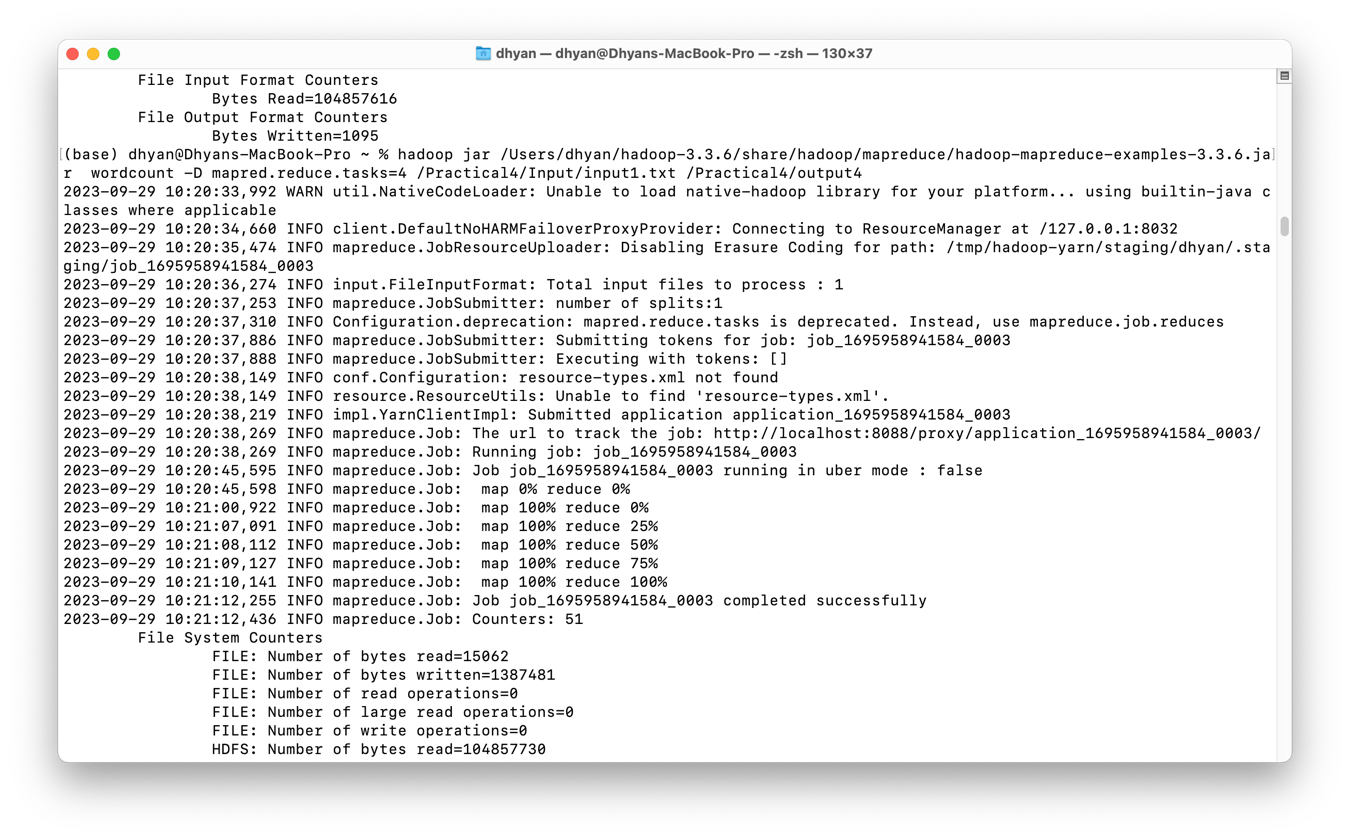
hadoop jar /Users/dhyan/hadoop-3.3.6/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar wordcount -D mapred.reduce.tasks=4 /Practical4/Input/input1.txt /Practical4/output4

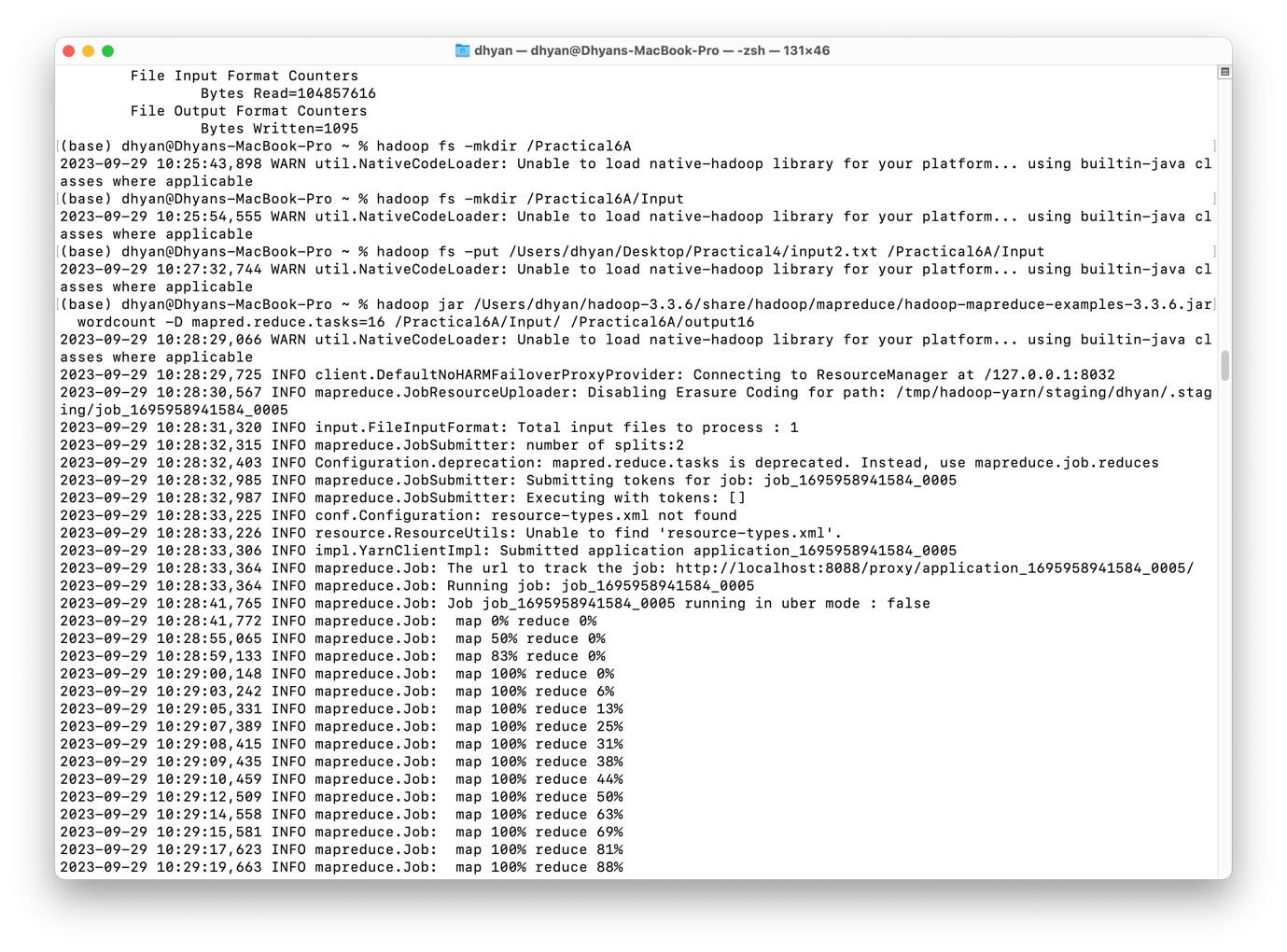
hadoop jar /Users/dhyan/hadoop-3.3.6/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar wordcount -D mapred.reduce.tasks=2 /Practical4/Input/ /Practical6A/output2

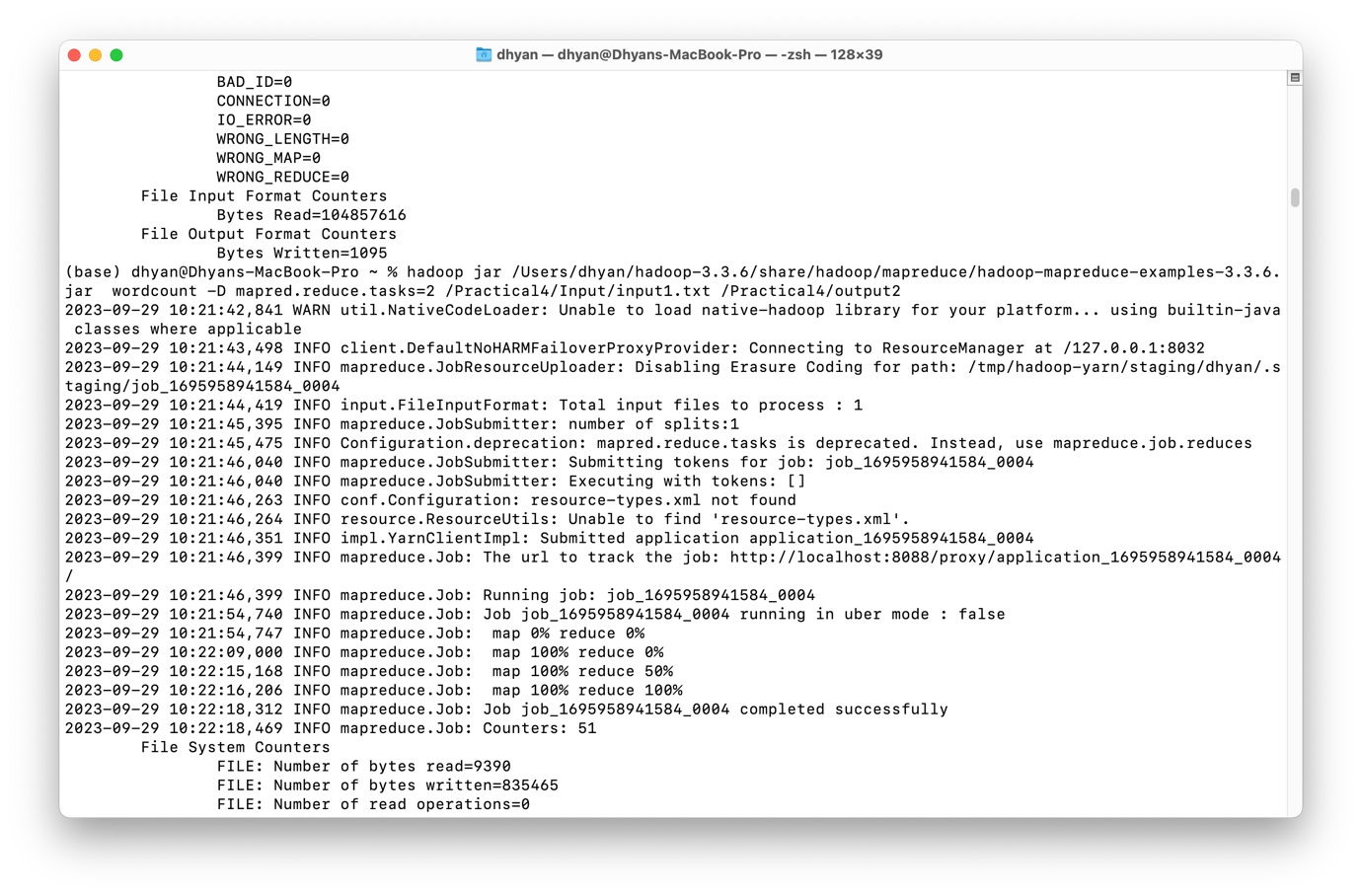
hadoop fs -cat /Practical4/output2/part-r-00000

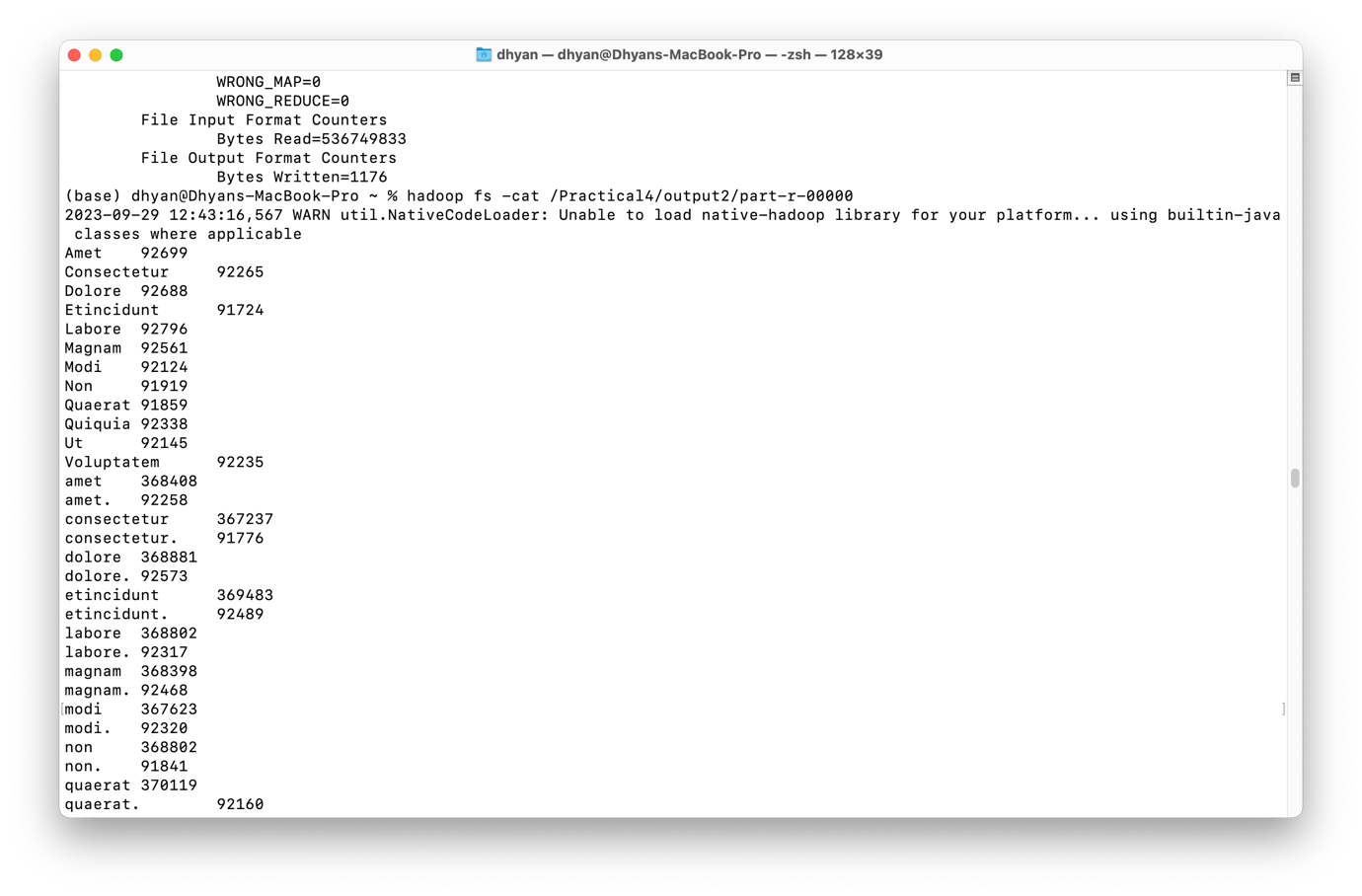












Analysis:

