# PROJECT – PART 1-5

# Part 1: Set up a single node cluster and optionally an eclipse development environment to create and test your programs

1. Setup environment on Virtual box, Docker and Google Cloud

Ref Single Hadoop with GoogleCloud Instance.docx

1. Get Cloudera

Ref Single Hadoop with GoogleCloud Instance.docx

1. WordCount

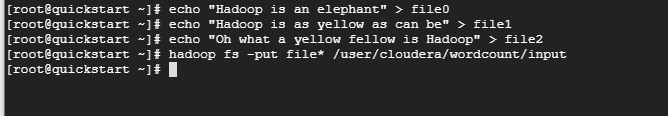
*Prepare data*

echo "Hadoop is an elephant" > file0

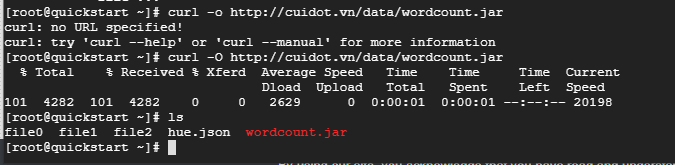
echo "Hadoop is as yellow as can be" > file1

echo "Oh what a yellow fellow is Hadoop" > file2

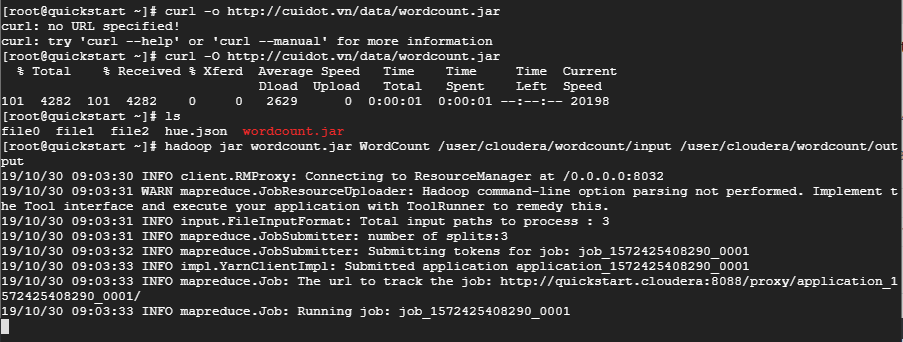
hadoop fs -put file\* /user/cloudera/wordcount/input



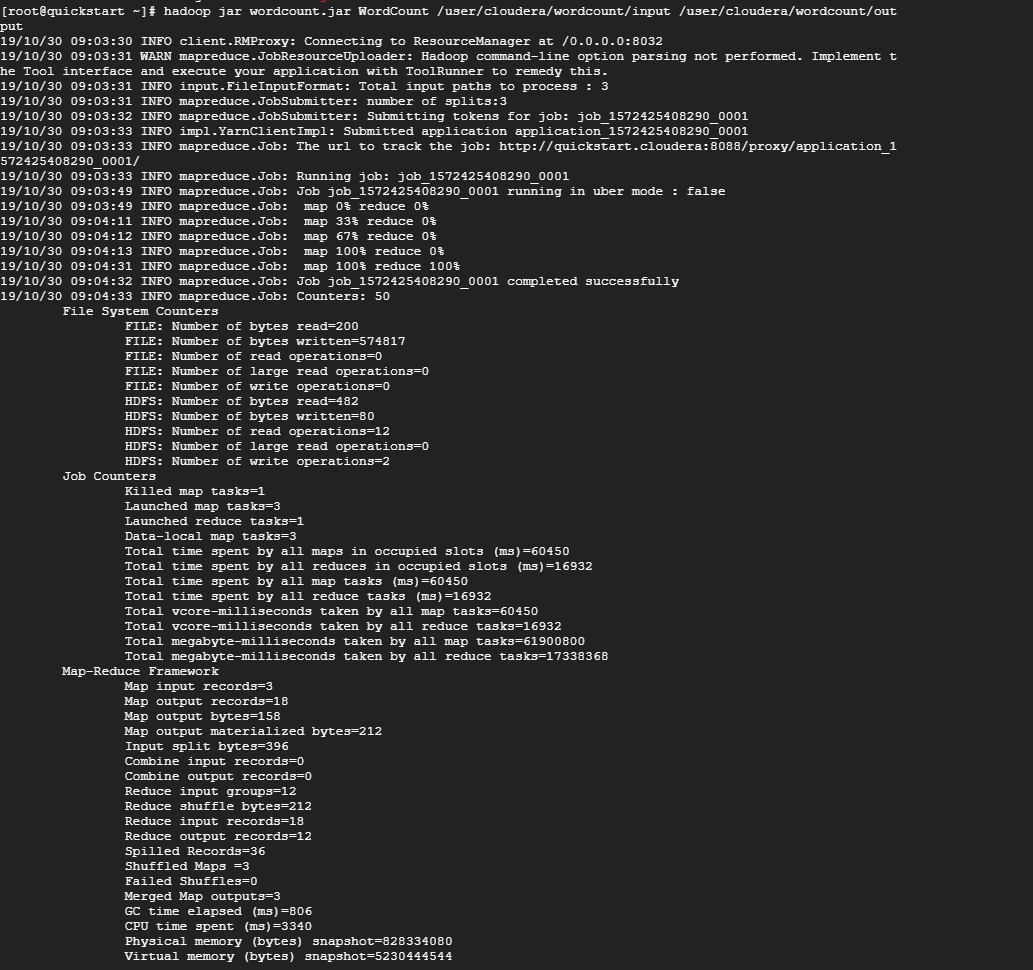
*Load jar file into cloudera*



*Run wordcount*

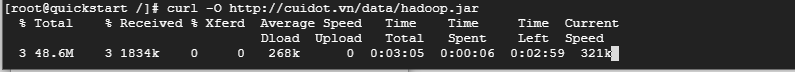


*Wordcount result:*



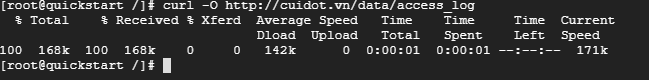
1. InMapperWordCount

*Load jar file into cloudera*

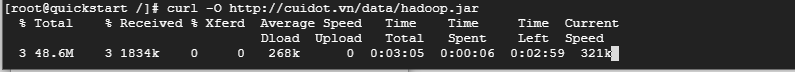


1. Average Computation Algorithm

*Load data into cloudera*



*Load jar file into cloudera*

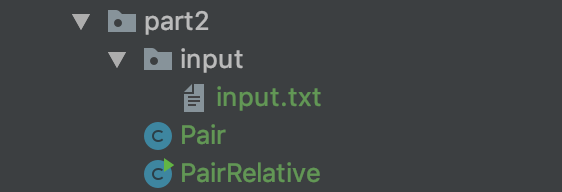


Copy file into hdfs

1. InMapper Average Computation Algorithm

# Part 2: Implement Pairs algorithm to compute relative frequencies.

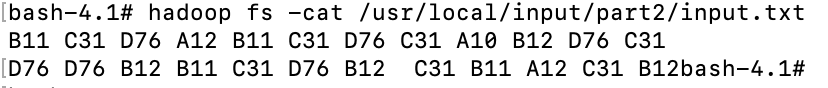
There are the .java files for the program.



*Run in Hadoop Docker*

# **Show input data content**

hadoop fs -cat /usr/local/input/part2/input.txt

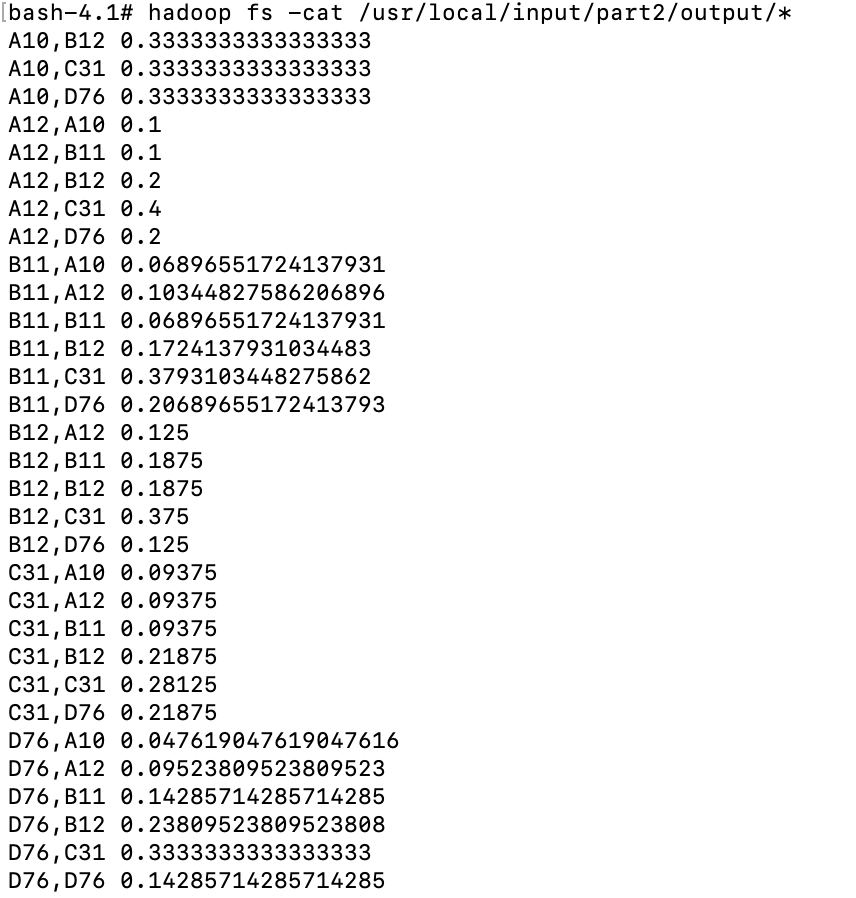


# **Execute the MapReduce program**

# hadoop jar project.jar edu.mum.bigdata.part2.PairRelative /usr/local/input/part2/input.txt /usr/local/input/part2/output

# **Program output**

**hadoop fs -cat /usr/local/input/part2/output/\***



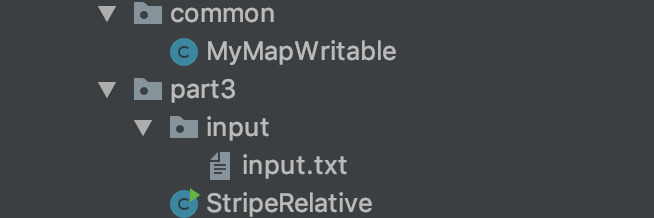
# **Run in Cloudera environment**

**# ./runAll.sh part2/ edu.mum.bigdata.part2.PairRelative**

# Part 3: Implement Stripes algorithm to compute relative frequencies.

# ***Run in Hadoop Docker***

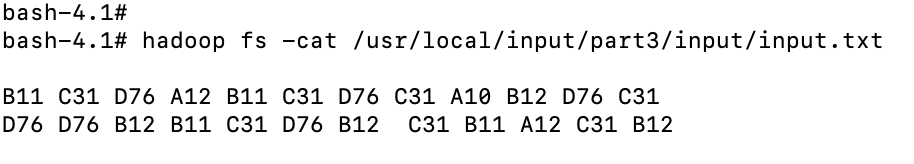
There are the .java files of program.



Build the whole project into .jar file and copy it into Hadoop docket and execute these commands:

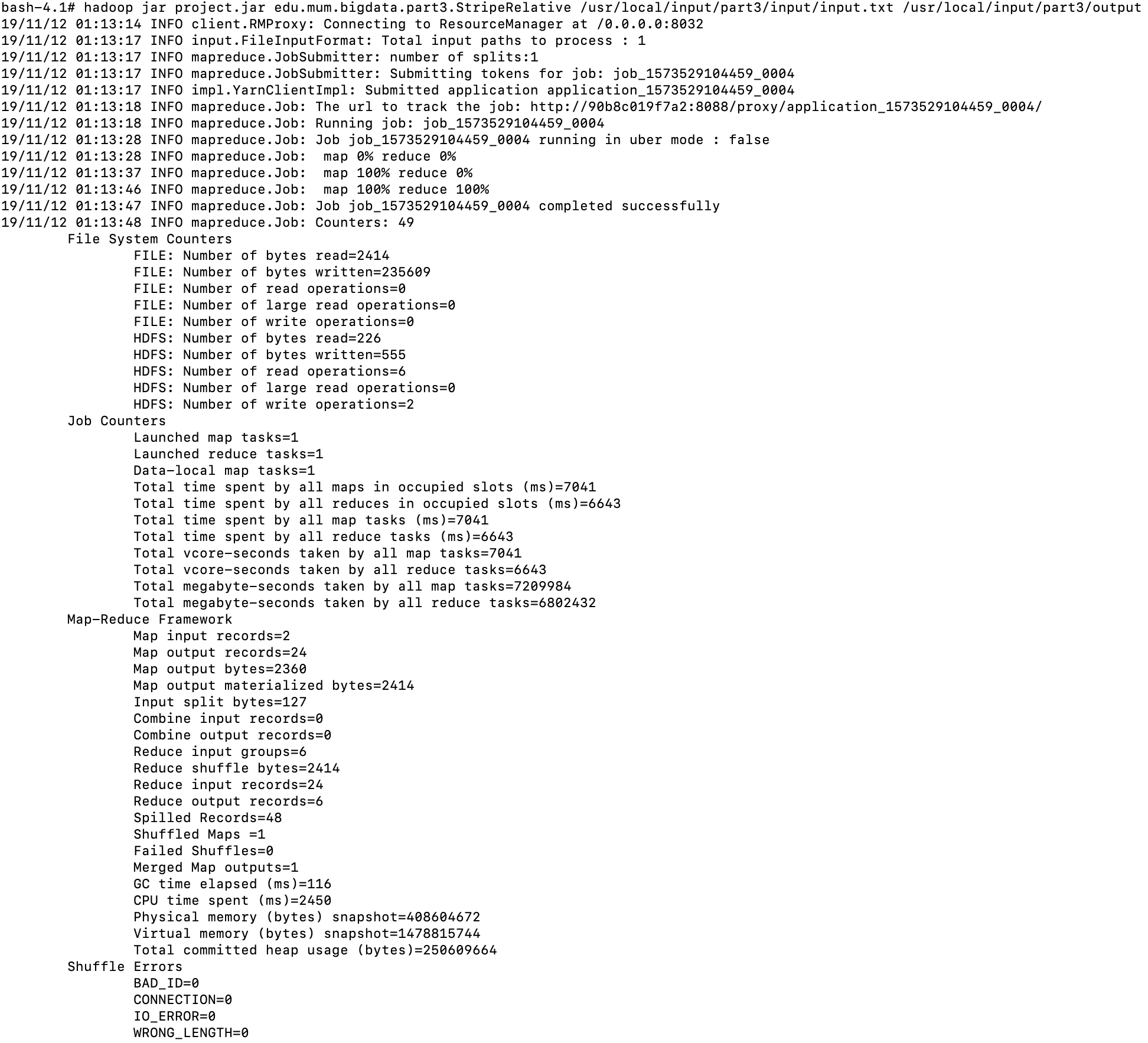
# **Show the input content**

hadoop fs -cat /usr/local/input/part3/input/input.txt



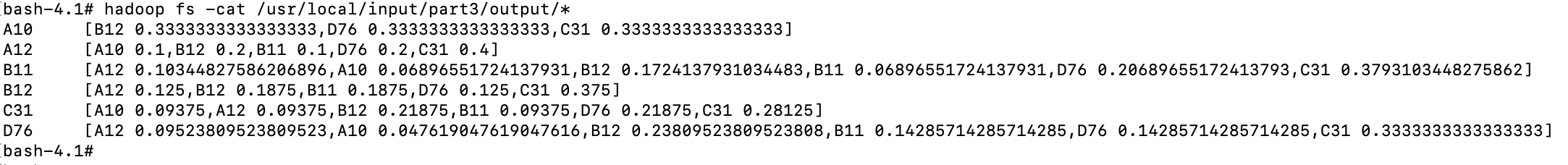
# **Execute the MapReduce program**

hadoop jar project.jar edu.mum.bigdata.part3.StripeRelative /usr/local/input/part3/input/input.txt /usr/local/input/part3/output



hadoop fs -cat /usr/local/input/part3/output/\*

# **Program output**



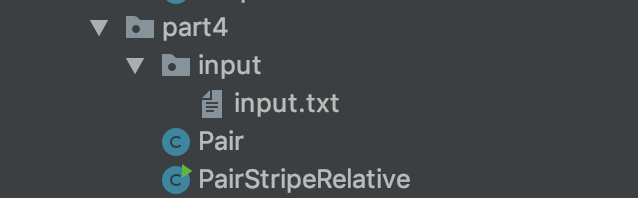
# **Run in Cloudera environment**

**# ./runAll.sh part3/ edu.mum.bigdata.part3.StripeRelative**

# Part 4: Implement Pairs in Mapper and Stripes in Reducer to compute relative frequencies (Hybrid)

# ***Run in Hadoop Docker***

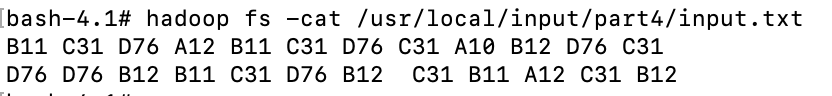
These are the .java files for the program.



Build the whole project into .jar file and copy it to Hadoop docker and execute these commands:

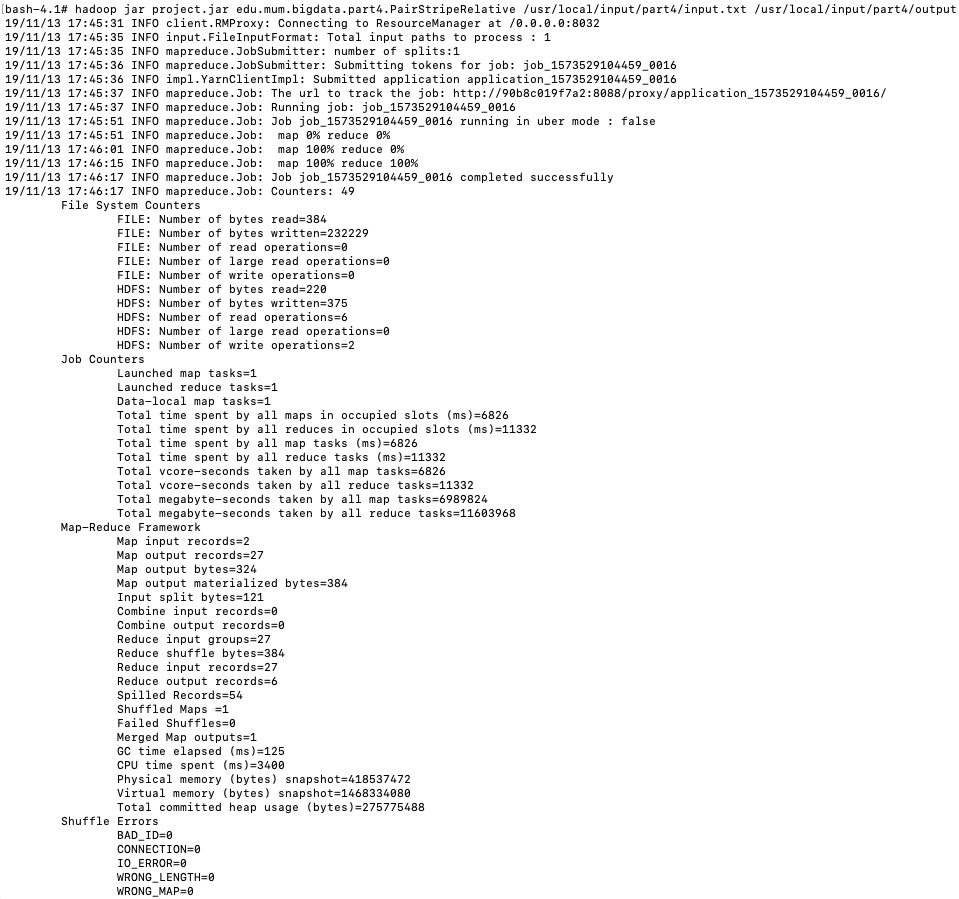
# **Show the content of input files**

hadoop fs -cat /usr/local/input/part4/input.txt



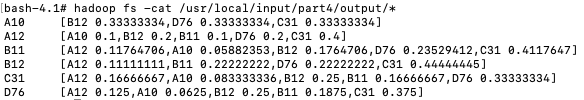
# **Execute the MapRecure program**

hadoop jar project.jar edu.mum.bigdata.part4.PairStripeRelative /usr/local/input/part4/input.txt /usr/local/input/part4/output



# **Show the program output**

hadoop fs -cat /usr/local/input/part4/output/\*



# **Run in Cloudera environment**

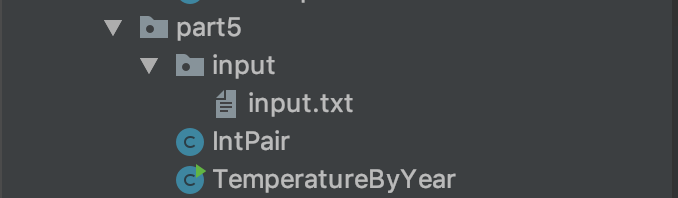
**# ./runAll.sh part4/ edu.mum.bigdata.part4.PairStripeRelative**

# Part 5: Solve a MapReduce problem of your choice!

# **Run in Hadoop Docker**

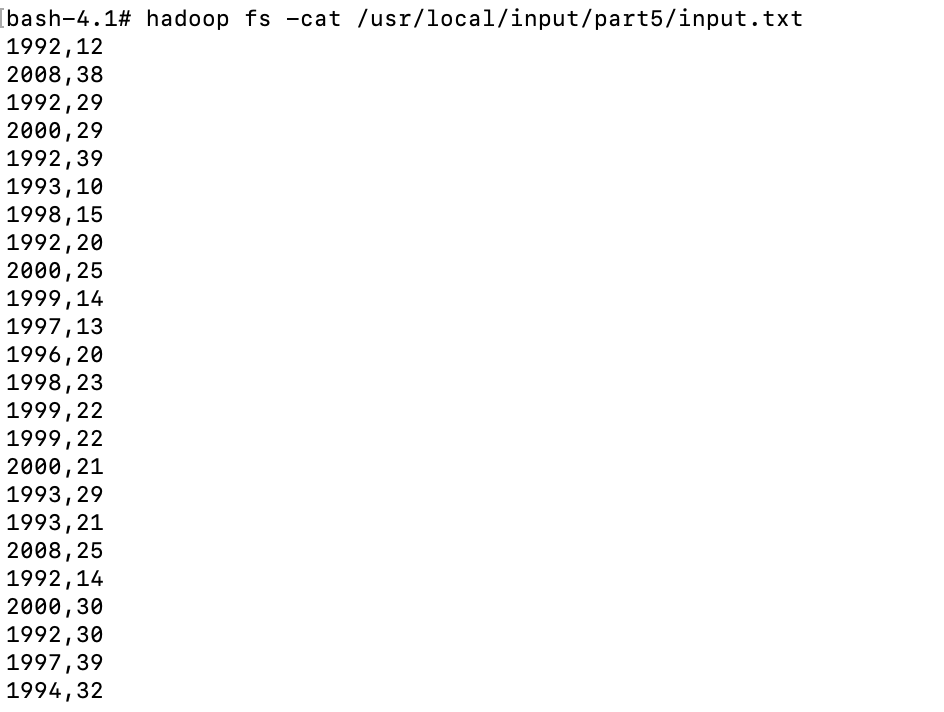
The problem is to compute the average temperature by year. Build the whole project into .jar file and copy it into Hadoop docker and execute with the follow commands.

The following is the .java files for this part and the testing input data.



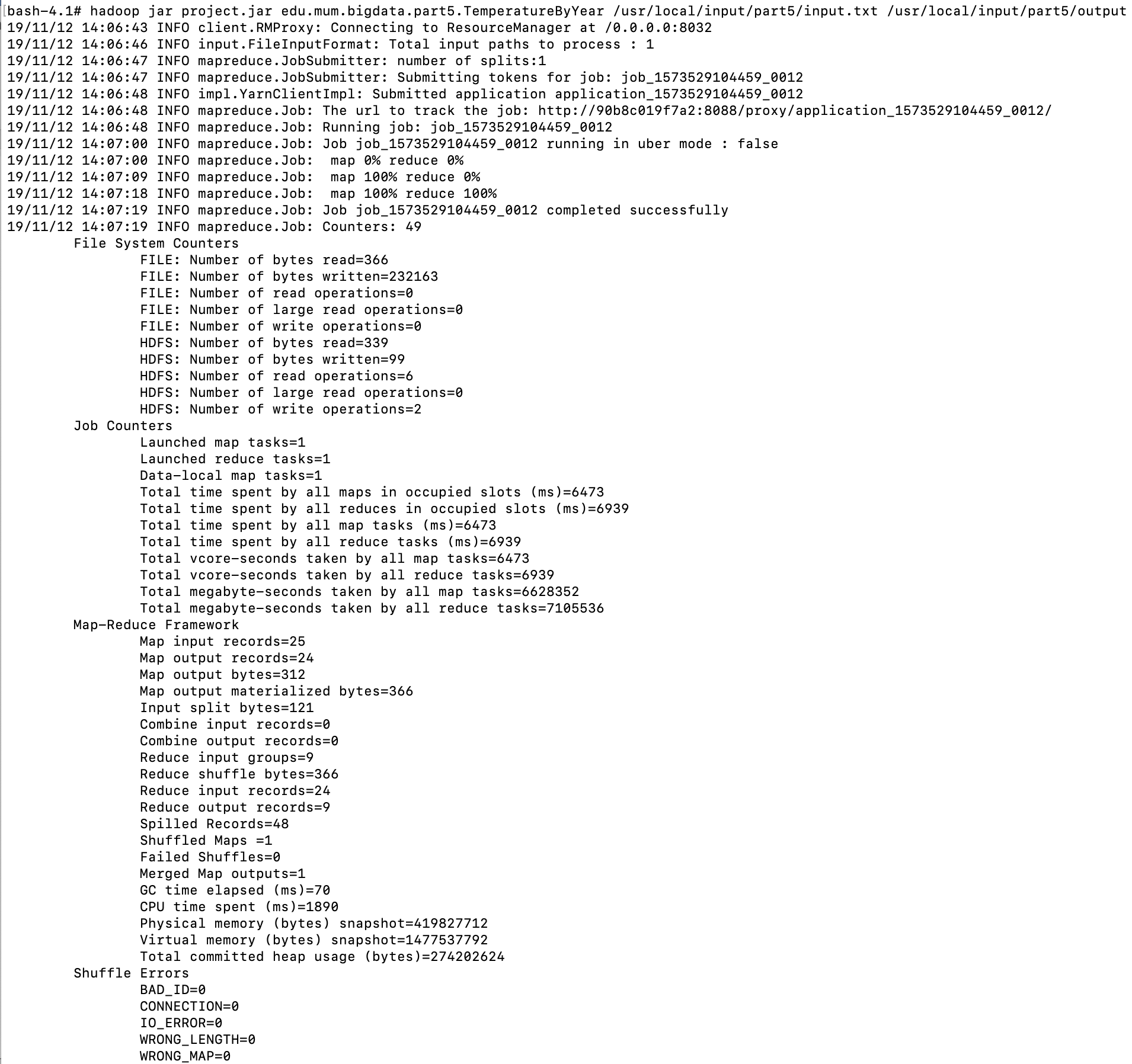
# **Show the content of input data.**

hadoop fs -cat /usr/local/input/part5/input.txt



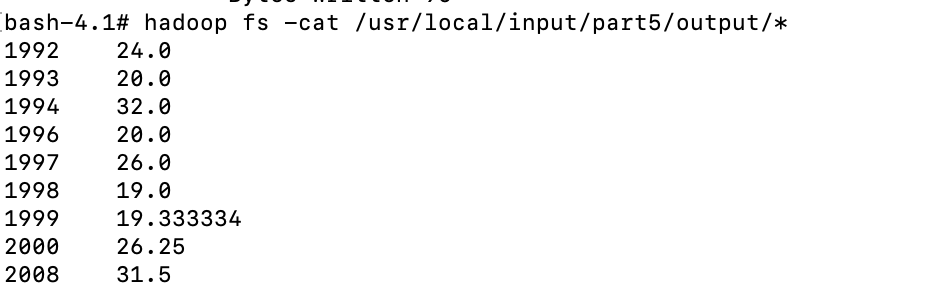
# **Execute the MapReduce program**

hadoop jar project.jar edu.mum.bigdata.part5.TemperatureByYear /usr/local/input/part5/input.txt /usr/local/input/part5/output



# **Show the output data.**

hadoop fs -cat /usr/local/input/part5/output/\*



# **Run in Cloudera environment**

**# ./runAll.sh part5/ edu.mum.bigdata.part5.TemperatureByYear**