|  |
| --- |
|  |

|  |
| --- |
| **HADOOP MAP REDUCE CENSUS DATA** |
|  |
|  |
| This project processes USA census data from 2009-2013 using Apache Hadoop Map Reduce |
|  |
| Submitted By: |
| **Akshay Sarkar - 1001506793 Remesh Sreemoolam Venkitachalam – 1001414827** |
|  |

**Table of Contents**

Table of Contents

**Overall Status2**

Set up2

Map Reduce………………………………………………………………………………………………………………………………………...3

Salary Comparisons and Analysis …………………………………………………………………………………………………….….4

Age Group Comparison and Analysis……………………………………………………………………………………………………6

Performance Analysis…………………………………………………………………………………………………………………………..7

Screenshots………………………………………………………………………………………………………………………………………….9

**Project Difficulty11**

**File Description 11**

**Division of Labor11**

**Encountered Errors 11**

**Map Reduce of USA Census Data using Apache Hadoop**

**Overall Status**

Hadoop Map Reduce is a hot topic in the current software industry due to its ability to process vast amounts of data in a very short time, using its map and reduce concepts. The introduction to this topic made us excited to do the project. We have successfully completed, and processed the census data to obtain two different sets of output as required in the problem description.

The first problem gives a list of Average Salaries of Males and Females of a particular state every year for all the five years of data collected in the input.

The second problem finds the overall number of people in various age groups for these years, in the USA.

**SETUP**

Prerequisites:

• Latest java JDK version

• hadoop-2.7.3.tar.gz

Setup :

1) Set up the Java Home as a system variable in your path .

2) Extract Hadoop zip file, place it in a folder and setup the HADOOP\_HOME by updating system variable with path up onto the /sbin folder inside Hadoop.

3) Then, we have followed the documentation from [link](http://www.eaiesb.com/blogs/?p=319) to configure the various set up.

4) Updated core-site.xml, hdfs-site.xml, mapred-site.xml and yarn-site.xml files as per the document provided.

5) Create namenode and datanode folders explicitly as Windows do not auto generate required folders while running.

6) Run start-all.cmd/stop-all.cmd with admin privileges.

7) Change the property “dfs.blocksize” in hdfs-site.xml to obtain block sizes of your wish.

**MAP-REDUCE**

Problem 1

Mapper : The Mapper was used to obtain the required data from the data set line by line and split them to obtain year, state, gender and salary of an individual. From these we create a <key, value> pair with (year, state, gender) as key, and salary as value.

Reducer: The reducer obtains the <key, value> pair from the mapper, and adds all the salaries in a particular key, along with taking the count of each. The average salary for the particular state in a particular year, for a particular gender is found by dividing the sum with count. Reducer throws a <key, value> pair with the same key as before, but the average for value. This is then displayed into a text file. Our key is ordered on basis of year.

Problem 2

Mapper: The Mapper was used to obtain the required data from the data set line by line and split them to obtain year and ages of the individual. This was separated in various age groups like 0-9, 10-19, and so on. We create a key value pair with (year, age group) as key, and the count as value which is 1.

Reducer:

We collect the output from the mapper and reduce it to add up the counts of the number of people in different age group in different years. We output a <key,value> pair with (year, age group) as the key, and the sum of individual counts as the value.

**SALARY COMPARISON**

**COMPARISONS 1 (INDUSTRIAL STATES)**

**ANALYSIS**

In the first graph we have collected the 5-year data for average salaries of both men and women from three very industrialized states in USA. California, Texas and Connecticut (due its proximity to New York) are highly job oriented states with heavy paying jobs like in IT, Engineering and Finance.

We observe a very similar trend in all these data. Men’s averages were good in 2009, and then took a minor dip in 2010 and 2011, before it started reviving and also in a good pace in 2012 and 2013. The dip can be attributed to the slump in the market created by the recession of 2008, where many people especially in finances and IT lost their jobs. But the markets survived and grew back with greater speed. Women’s salaries haven’t varied much in these years for all these states. It indicates that for one, women are less in competitive fields like finance and IT, and have more stable jobs. We can also observe that women are paid much lesser than their male counterparts.

Financial industrial states like Connecticut have huge salary averages, than a laid back states like Alabama or South Dakota. Connecticut seems to have much larger salaries than Texas or California too, but here we need to understand that California or Texas are much larger in size than Connecticut, and therefore includes larger population and a bigger variety of job opportunities, some which may not be as well paying as others.

**COMPARISONS 2 (FARMING STATES)**

**ANALYSIS**

Now considering some country side states which do not have too many industries, but have more seasonal jobs like farming, we can see that generally the average is much lower than other states in the country. The averages do not show a typical pattern and are much more of random variation. This is majorly because the work and pay is more seasonal and it is affected by weather and natural interferences.

Here too, men do have a better pay than women, but the averages are slightly more closer than in the more modern societies.

**AGE GROUP COMPARISON**

The age group comparisons are quite straight-forward for all the five years. We see that there are more number of middle aged people between ages 40-60 in USA for these years. The number of people in 50-59 range is increasing by a good margin.

We also see that the number of people between 0-9 is slightly decreasing over the years. Lesser number of families are having lesser kids as years progresses, while teenagers and youth have remained mostly on the same level throughout.

Without surprise, it is clear that the number of people decrease drastically as age increases. Very few people make it beyond 70, as life spans are decreasing. The number gets cut in half for the next age group and by even bigger percentage make it to 90.

**Performance Analysis**

On x-axis we have the various block sizes and map and reduce functions running on it, while on the y-axis we have the total time spent by the mappers, which is the summation of runtime of all map tasks.

Trying out the same program in Hadoop by specifying different block sizes gave different kinds of performance. As from the chart, the bigger the block sizes the faster the programs ran. We do not know to what extent can this trend continues, or if a single mapper with one block would be the fastest.

The causes of the delay with increasing number of blocks could be the extra time required in shuffling and sorting the mapper <key, value> pairs. As less number of blocks come out of the mappers, less shuffle is required. But we can see from the chart that the increase is not linear. The difference in time saved by using a 256MB block is slightly lesser than the difference in time between 64MB and 128MB blocks. So as the block sizes increases, the runtime differences could slowly start diminishing.

The time shown is the collective time taken by all maps or all reducers for each run.

So, on an average the ~ 64MB block sized run on the average salary mapper created 131 splits Therefore the average per mapper time is 3460/131 = 26.41s/map.

128MB block sized average salary mapper with 63 splits would run on 2389/63 = 37.92s/map.

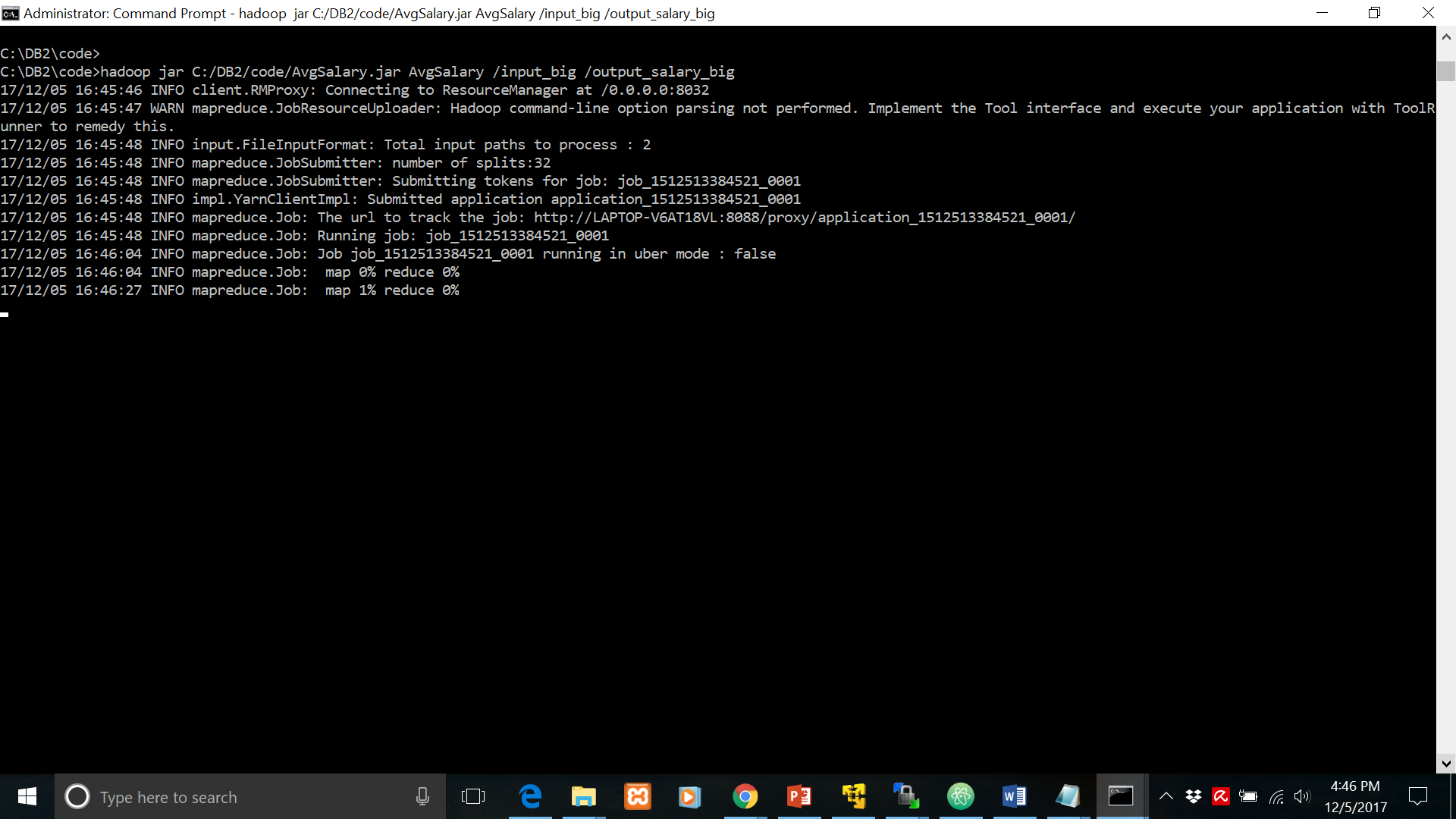
256MB block sized average salary mapper with 32 splits would run on 1972/32 = 61.63s/map

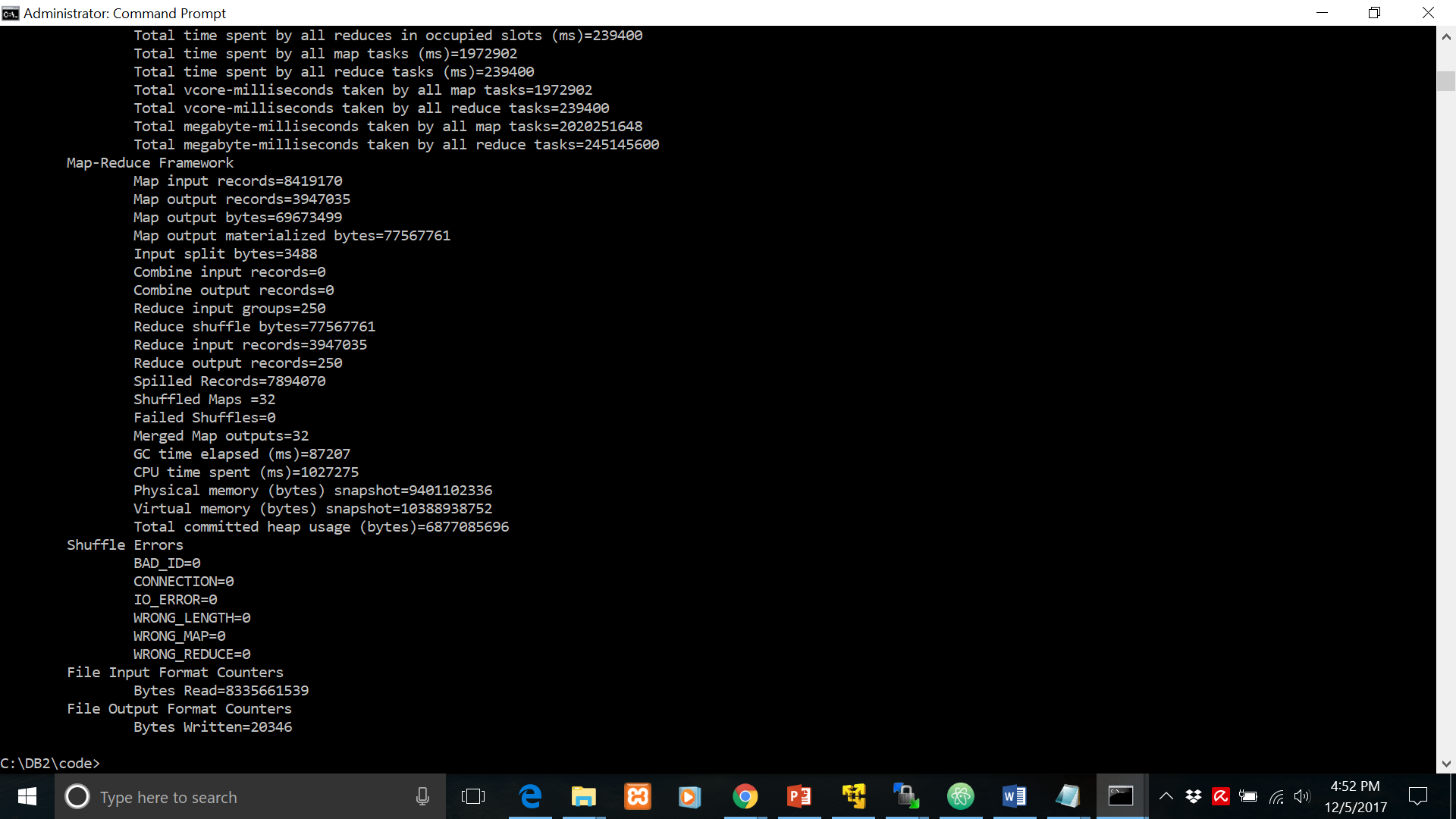
So while we can see that generally the total time taken is decreasing, the time to perform one individual map job is increasing as block size increases. The reduce jobs took only one task for all block sizes, reduce tasks = 1. Therefore the time for reduce tasks is the same as in last chart.

Per map running time is represented in the chart below.

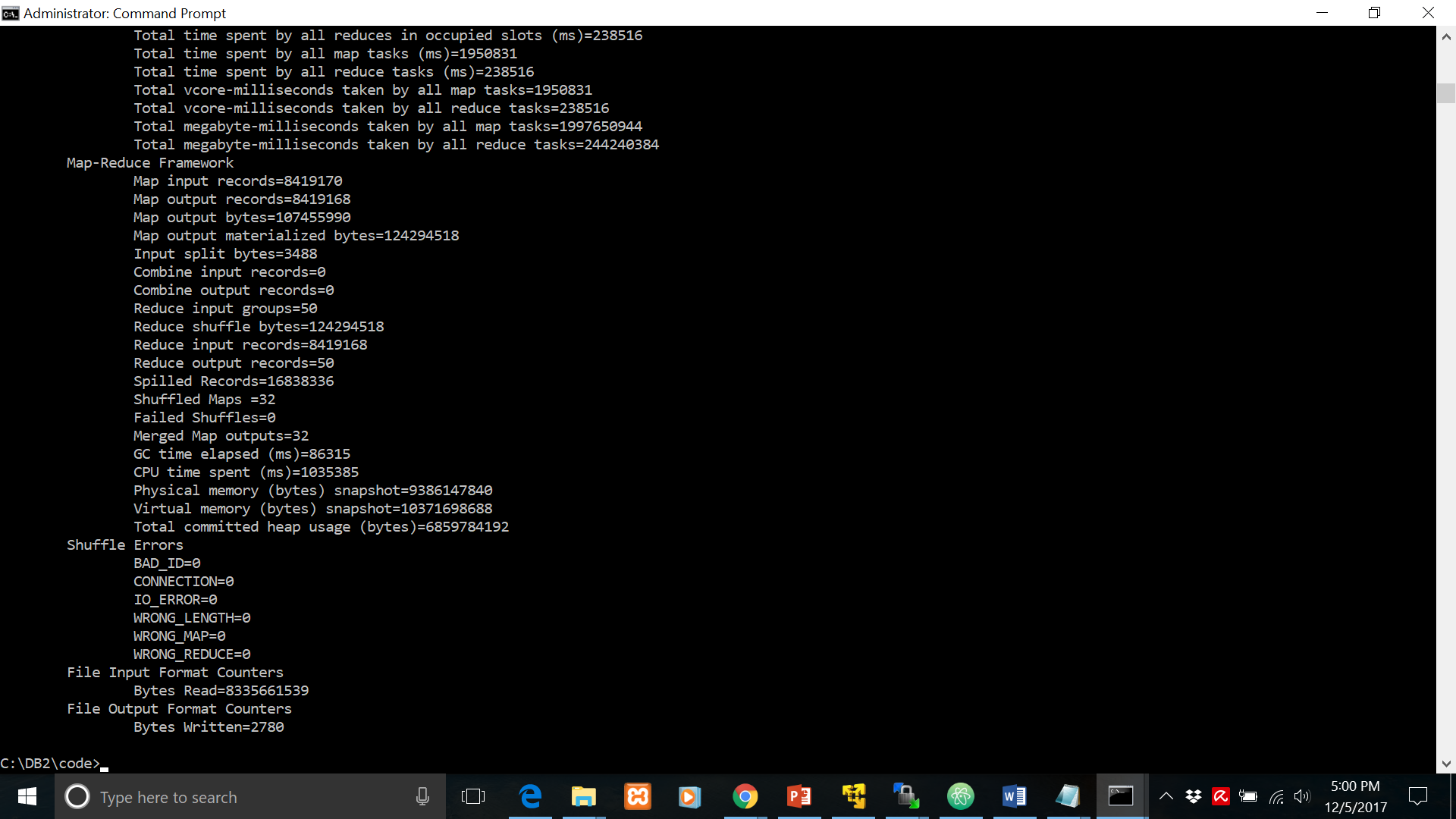
**SCREENSHOTS**

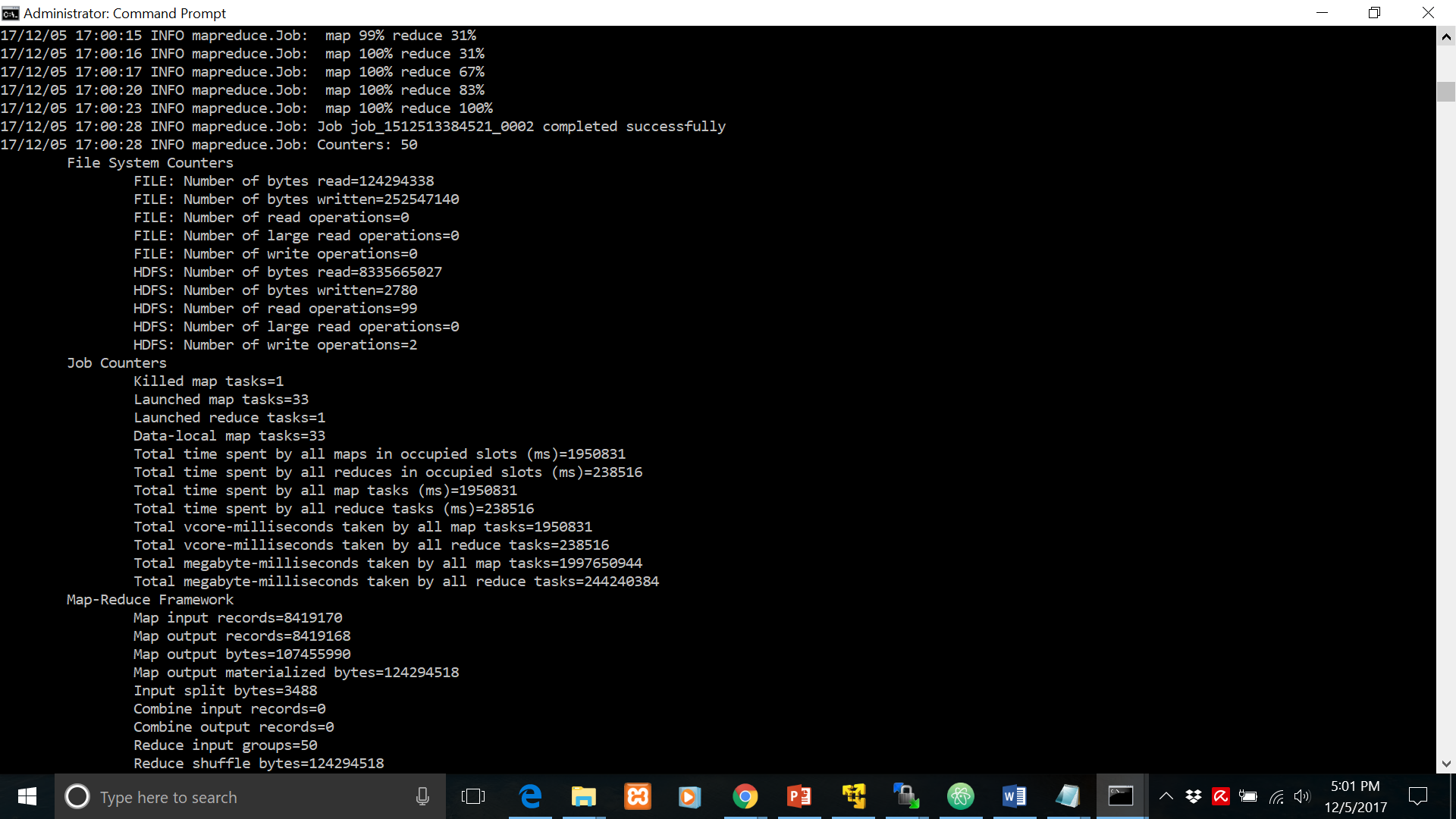
Average Salary





**Number per Age Group**





**Difficulty**

Deciding on the method of installing and using Hadoop was the most important process of this project. The decision had to be made upon the ease of installation and also the processing of huge amount of data. Due to the lack of space in our personal laptops, we had to rearrange quite a bit of our laptop drives to accommodate the huge data set.

**File Description**

We just needed to implement two different java classes for each of the different problems given in the assignment.

AvgSalary.java gives the average salary based on the year, state and gender of the data set.

CountAge.java gives the number of people in an age group for corresponding years.

**Division of Labour**

We took a couple of meetings together to decide on the installation of Hadoop and understand what goes into it. Once Hadoop was installed, we used the examples given in class to learn the process of map reduce and how it could be used to solve the problems given to us.

Once the code was written, we also spared time to learn the commands to run these codes along with the given data sets. By coding together, we were able to help each other with the debugging of the program, and that made it easier to find flaws in the code. We coded for around almost the 5 Hour with total time including the preparation 15 Hour.

Akshay worked on Average salary problem, while Remesh performed the Age count problem.

**Encountered Errors**

1. When we started coding we realized that the slots for gender and salary given in the description and its position in the document were different. This was giving wrong results on running the program. It took us some time to analyze this discrepancy and report it to TA.
2. Installation proved to be difficult at first. On first try, we used a linux platform and tried to install Hadoop in it. Due to lack of experience in Linux, we were stuck at certain places.
3. We first tried to use ‘int’ for salary, which we realized was a major mistake as it started giving negative results. This could be because the actual summation exceeds the range of ‘int’. We decided to change the data type to ‘double’ so as to get exact average to decimal precision.