

School of Computing, Engineering and Mathematics (CEM)

Faculty of Engineering, Environment and Computing (EEC)

**5011CEM BIG DATA PROGEAMMING PROJECT** | 2022

**CLIENT REPORT**

**NAME: YISONG LI**

**SID: 11384841**

**GitHub Link：https://github.coventry.ac.uk/5011CEM-2122JanMay/5011CEM2122\_liy332**

**BIG DATA report**

**About how to improve analytics data time& Project Client Report**

**author: Yisong Li**

**report time: 19/04/2022**

**word count: 2090 words**

CONTENTS

[Introduction: 4](#_Toc101281928)

[1.How did overcome loading nearly 9TB in memory? 5](#_Toc101281929)

[2.Sequential processing VS parallel processing 5](#_Toc101281930)

[2.1 Introduce Parallel and sequential 5](#_Toc101281931)

[2.1.1 Sequential processing 6](#_Toc101281932)

[2.1.2 parallel proceesing 6](#_Toc101281933)

[2.2Test results for Sequential and Paralle 8](#_Toc101281934)

[3.Testing Code 9](#_Toc101281935)

[3.1 Text error 9](#_Toc101281936)

[3.2 NaN test 10](#_Toc101281937)

[3.3Log file 10](#_Toc101281938)

[4.Automated testing 11](#_Toc101281939)

[4.1Introduction to Automated Tests 11](#_Toc101281940)

[5.Estimate the number of processors 12](#_Toc101281941)

[5.1Preliminary Prediction of Outcome 12](#_Toc101281942)

[5.2Estimates of processors using functions 12](#_Toc101281943)

[5.3More efficient predictive processor cores 13](#_Toc101281944)

[6.Conclusion 13](#_Toc101281945)

[7.Appendix 14](#_Toc101281946)

[8.Reference: 18](#_Toc101281947)

# Introduction:

With the advancement of technology and the development of society, more and more data are flooding our daily life. But most of the data is useless, such as the detection of electromagnetic information in space, most of the data is useless. In this regard, how to extract useful information from massive data is a common goal of mankind. With the implementation of new big data analysis technologies, the valuable data in big data will be better explored and mined. The project uses big data methods to analyse atmospheric science and climate model data provided by the European Centre for Medium-Range Weather Forecasts (ECMWF). However, after testing the technology applied on the client side, it took much longer than expected to analyze the data. In this report, I will show how to reduce the analysis time of massive data to an acceptable range (target efficiency: less than two hours of analysis per day, 25 hours of data, about 250MB of data) and according to customer requirements, code Analysis improvements.

The report will contain three main elements:

1. Comparison of sequential and parallel techniques

2. About common data errors encountered in using matlab big data analysis data

3. According to data analysis, the number of processors expected to achieve the goal

This report will revolve around the use of MATLAB, using big data data analysis methods, to analyze a 3D grid of atmospheric composition over Europe over multiple climate models, containing 100 chemical species at each location. Its total dataset is >10TB. In this data analysis, only the "Total Ozone Column" will be analyzed. This data will show the specifics of ozone at each location in 2D. So the amount of data will be greatly reduced. Although the amount of data is reduced, by using appropriate processing methods, the analysis efficiency can be greatly improved and the required target efficiency can be achieved.

# 1.How did overcome loading nearly 9TB in memory?

In our preliminary investigation, we found that the total size of the data analyzed by the target reached a staggering 9TB. In modern computers, however, we know that a computer's CPU is often inseparable from its memory. But the memory capacity is small (not enough to hold 9TB of data at once). For this, we narrowed down the analysis data (using only the 1-hour load data to infer the total data in the task) and used the built-in math functions in MATLAB, so that the analysis could be implemented using a PC, making the test Has better versatility.

# 2.Sequential processing VS parallel processing

## 2.1 Introduce Parallel and sequential

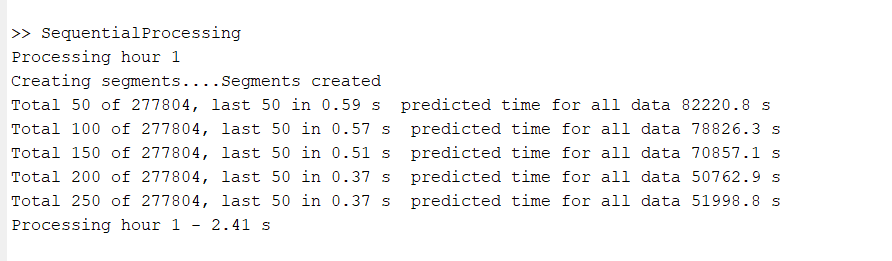
When we actually process the data, we can find that when the amount of data analyzed is a certain amount, the speed of analyzing the data largely depends on the speed of our CPU. Although we cannot change the speed at the physical level, we can change the analysis speed through different analysis methods.

To increase the speed of this operation, we refer to sequential and parallel techniques to address how to increase the speed.

Due to historical reasons, in the era when computers were not so developed, due to the limitation of physical hardware such as CPU, when we want to analyze data with computers, we can only use sequential methods for data analysis. Sequential processing means that the program executes each instruction in sequence, and there is only one running context during the entire program running process. That is, a call stack and a heap. There are no multiple run contexts. Simply put, when there are multiple tasks to be processed, each program can only be processed in turn by "queuing". This processing method not only wastes time, but also greatly affects work efficiency.

### 2.1.1 Sequential processing

Although it is well known that parallel processes are superior in principle to sequential processes, sequential process code is still created in order to compare the principle and actual speed difference between the two. The sequential process first analyzes the data sequentially by importing the parameters entered by the customer, and then through the sequential method.

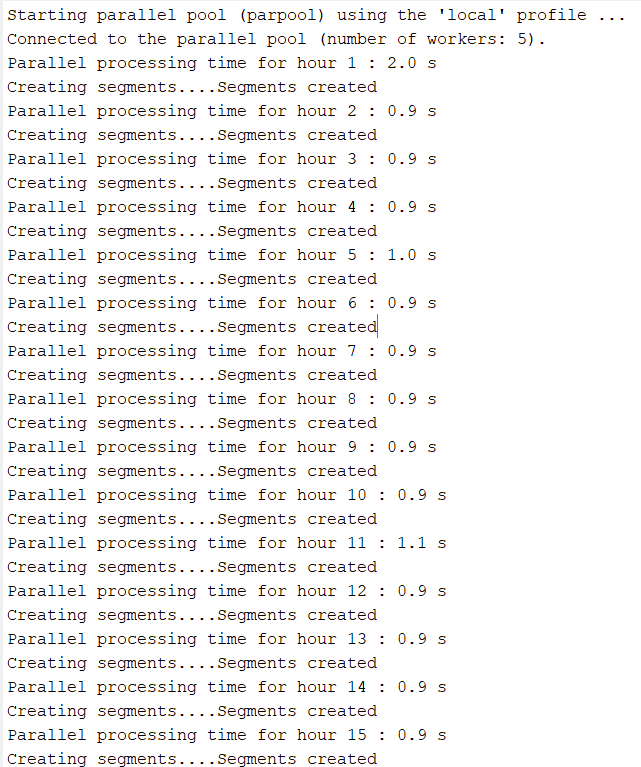
In the sequential code, first set the code of the required start time period and end time period, and analyze it in a single-core manner. Allow clients to better understand and understand current time estimates during the analysis process. Analyze 50 data as a stage and give the current run time in the output, as well as the estimated time to analyze all the code. The output screenshot is as follows: 

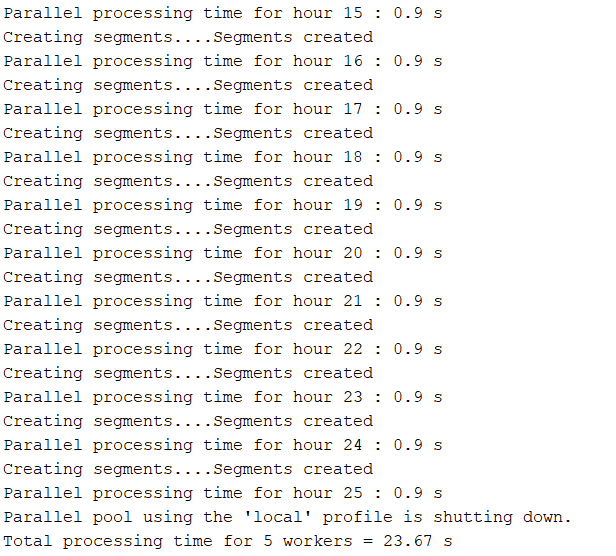
(Example of output)

### 2.1.2 parallel proceesing

Parallel processing is a better choice than sequential processing because modern cpus (most of which have multi-core, parallel processing technologies) are more efficient. But what is Parallel processing? Parallel processing means that on a multiprocessor, several tasks can run simultaneously because there are multiple execution units. This way of allowing multiple tasks to run physically simultaneously is called parallelism. Simply put, multitasking can be done simultaneously, making processing faster and saving time.

In order to realize the parallel method, the code of parallel processing is also constructed. Parallel processing allows users to set what they think is the most efficient way by setting the number of cores per CPU run and the amount of data sets per day for each analysis.





(out put example; 500data; 1-25hour data; 5workers)

## 2.2Test results for Sequential and Paralle

In the actual testing process, we found that parallel is faster than sequential. In the test, two sets are used: 250 data set and 5000 data set for testing.

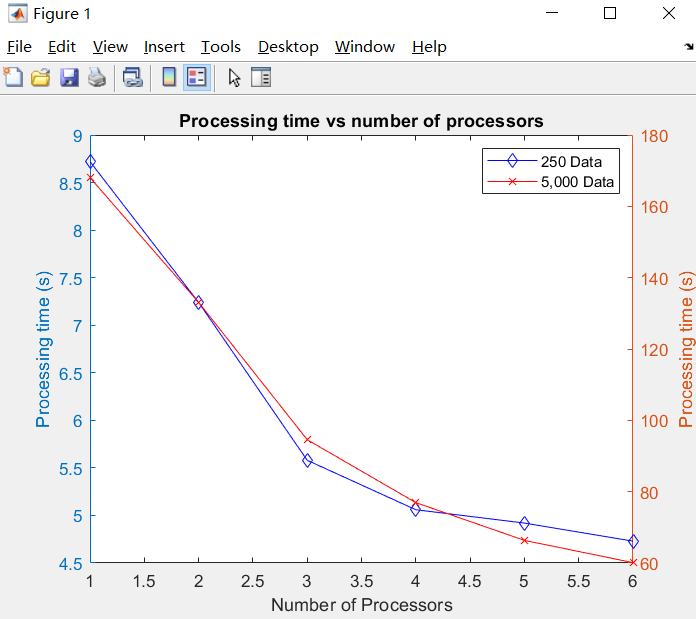
In the 250 data set, the test time for sequential (only single-core processing) is 8.72s. Since my cpu has only six cores, the test time from two cores to six cores (parallel process) is 7.24s, 5.58s, 5.06s, 4.92s, 4.73s.

In the 5000 data set, the test time for sequential (only single-core processing) is 168.04s, and the time for parallel processing (two-core to six-core) is, 133.08s, 94.57s, 76.93s, 66.34s, 60.11s.

Through matlab With the mapping software, we can get the time comparison chart.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **core number/data set** | **1(sequential)** | **2** | **3** | **4** | **5** | **6** |
| |  | | --- | | **250 data** | | 8.72s | 7.24s | 5.58s | 5.06s | 4.92s | 4.73s |
| **5000data** | 168.04s | 133.08s | 94.57s | 76.93s | 66.34s | 60.11s |

It is clear from this that parallelism is more efficient than parallelism when performing the same tasks. In particular, from the line graph analysis that know that as the cpu has more processors to do the analysis, it takes less time. So parallelism has an absolute advantage over sequentiality.



# 3.Testing Code

After analyzing the data using the parallel approach, the code and data must be tested. Because when acquiring data, the system cannot judge whether the data is wrong. If the wrong data is used for analysis, the program may crash, or the analysis data may be wrong. Likewise, the test code of the project needs to be tested to ensure that it runs successfully without any errors in the tests. Therefore, three parts will be tested:

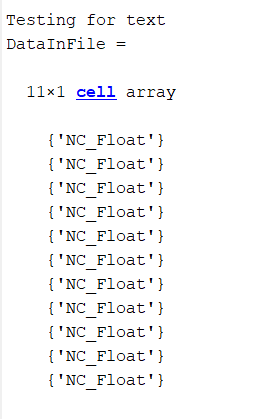
1. Text tests

2. NaN test

3. Log files

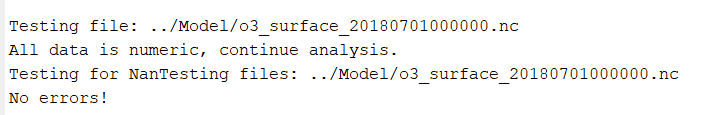
## 3.1 Text error

Text errors occur frequently, and text testing is essential. Although all the files and code in this report are bugs and bugs can be found for this. However, there is no guarantee that customers will encounter unforeseen problems when using this program for future analysis (user-defined test files). For this text test, the CreateTestData\_Text script will be used and the test file testtest.nc will be generated by analyzing the data files. During the test, the test file is extracted from the file data file, data type analysis is performed, and it is saved in an array. This test fully met expectations and achieved the expected results. The test pictures are as follows.



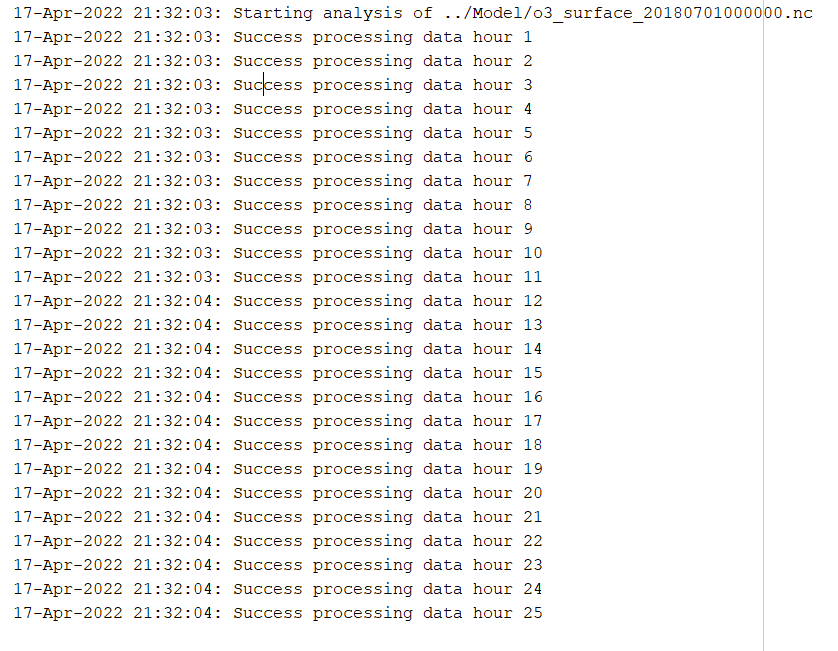
## 3.2 NaN test

NaN errors, full name not-a-number, are also A large part of the error type. In MATLAB, the common cases are: 0/0, Inf/Inf, INF-INF, Inf\*0. Each of these situations leads to uncertainty, so you get NaN. Much the same steps are used to test nan errors as to test Text. Use the createTestDatA\_nan. m script to test the testfile. nc file generated from the analysis data. The expected results of the test are consistent with the estimates. However, the data reported by NaN is not clear in the file, so when during the test, should add conditional judgment and other methods, so as to know the location of the error, which will save more time. The results are pictured below.



## 3.3Log file

Knowing that finding errors can be time-consuming and difficult to locate by querying data and code for NaNs and text errors. Therefore, you can create a log file for problems encountered during testing. In this test, TestSolutionWithLogFile will be used. M-files, dedicated to all kinds of errors, for problems and misplacements, denoted by "xx" hours, so that you can pinpoint the location of the error during testing. The result achieves the expected effect, and the test output picture is as follows.



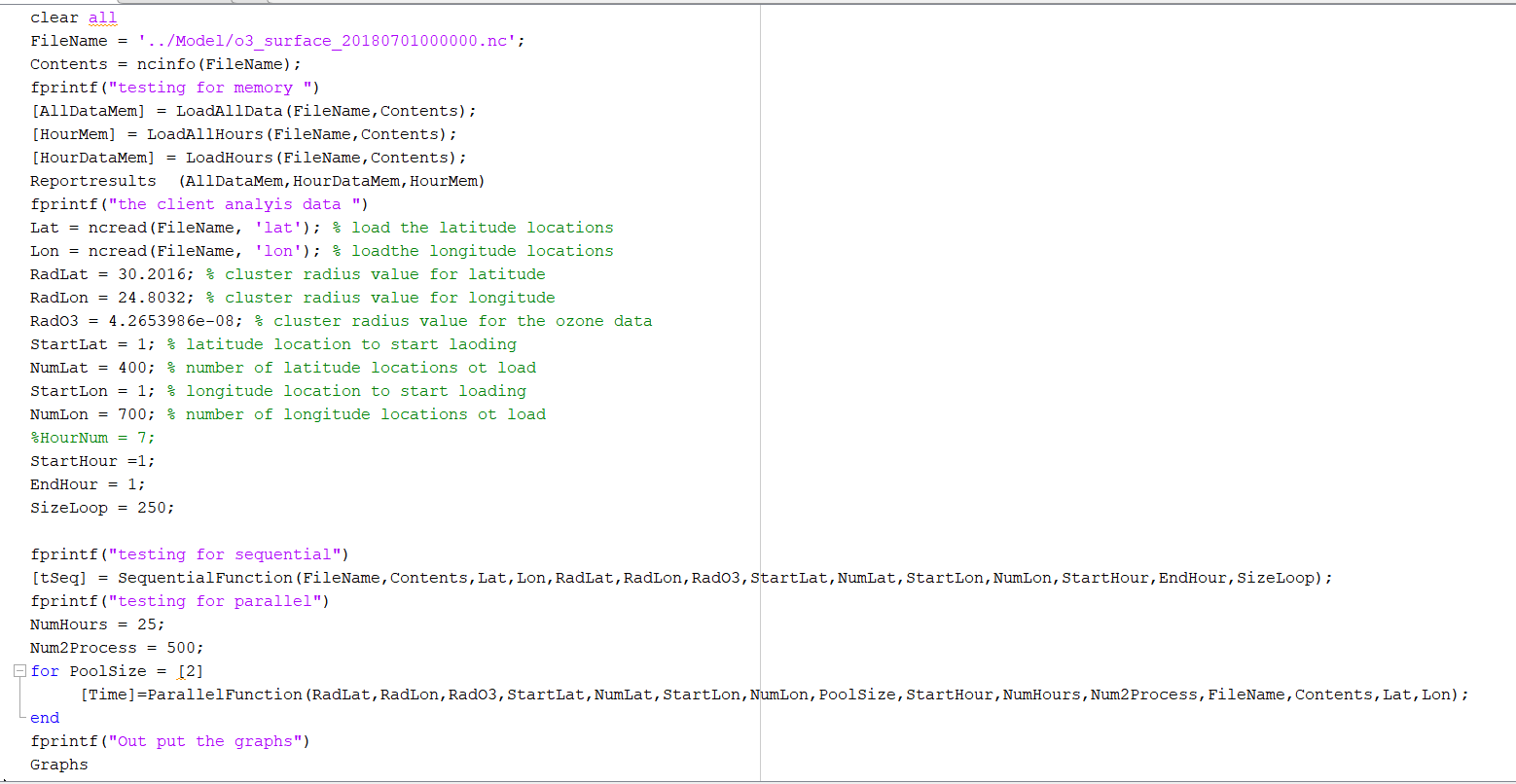
# 4.Automated testing

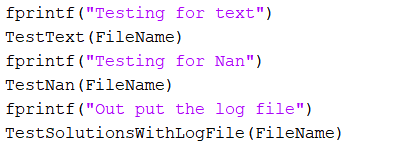
## 4.1Introduction to Automated Tests

With the code created, parallel processing can now be successfully used to analyze the target data requested by the client. In order to better provide customers with "one-stop service". A main program will be created by combining all functions. Includes line graphs comparing parallel and sequential processing, parallel and sequential efficiency. There are also text and NaN tests for test data and code, and errors can be created directly in the log, allowing customers to better check for errors. The creation of the main program is convenient for customers, allowing users to easily change the target parameters, and then "one key" to start the program and save all data, which is convenient for users to view and extract data at one time.

4.2Results of automated tests

Through the test, it is found that all the results are printed normally, and all the data appear in the workbench, and the data can be clearly presented.





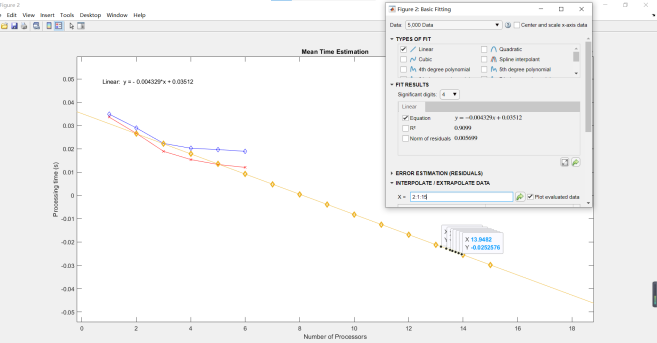
# 5.Estimate the number of processors

## 5.1Preliminary Prediction of Outcome

After our test of parallel and sequential. When the amount of data is constant, it is found that when the processor increases, the time to analyze the data will decrease. Looking at the line chart, we find that the results are close to a linear scale.

## 5.2Estimates of processors using functions

By saving the test results, it cna get the results of parallel processing. Through the extrapolation method of matlab, and using the drawing software of matlab, it will be get our line chart clearly. It was found by prediction that at least 13 cores are required to reach the predicted target (the test computer chip is an Intel i7-10750H with a maximum of six cores)



## 5.3More efficient predictive processor cores

It can be known from research that the computing power of computers is not predicted in an absolutely linear situation. The first is memory bandwidth. Memory bandwidth cannot be increased indefinitely. Because of packaging bottlenecks, it is impossible to add too many pins to support larger bandwidth. Multi-core processors will compete for bandwidth, resulting in non-linear performance growth. The second is the software bottleneck. Thread scheduling by the operating system requires overhead. Sometimes threads switch between different processor cores. cause additional overhead. Leads to non-linear growth of multi-core performance. In this regard, in order to better predict the number of target cores, The same can also be predicted using a similar exponential regression function, which can give better results

# 6.Conclusion

The report is created by comparing sequential and parallel code. Through comparison, it can be concluded that parallel is more effective than sequential and can better achieve the goals required by customers.

Recorded by parallel processing time, and by extrapolation, it can be estimated that if you want to achieve the customer's goal, at least 13 processors are required to complete it. If you want to get a more accurate value, you can use the exponential regression function and get answer

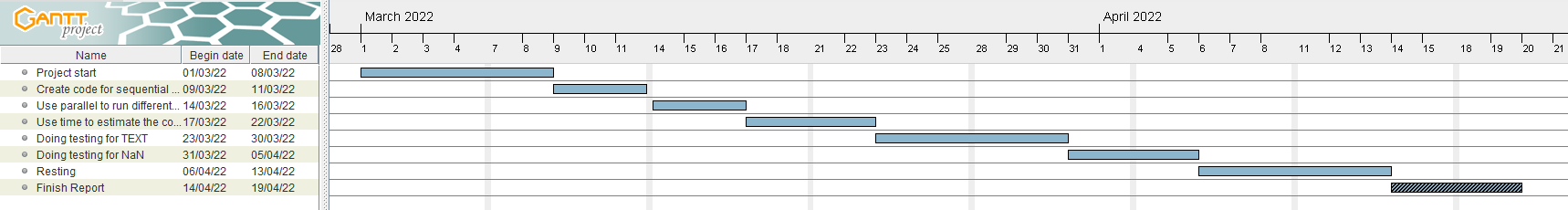
In the process of testing, by creating an automatic search for common errors (such as text; NaN errors), the location of the error can be precisely located, and the location of the error (in a certain hour data set) is stored in the log, so that customers can query and Revise.

In order to facilitate the operation of the customer, the main scrip is established, and all functions and functions are built on one screen, which is convenient for the customer to modify the parameter value, and also greatly avoids the "unintentional" damage of the test file.

Finally, through parallel processing, the customer's goal (to complete the 25-hour data set test within two hours) was successfully achieved, which greatly facilitated the tester's detection efficiency.

# 7.Appendix

Gannett chart：



Logbook:

March 1st - March 8th

The process of the project started today, I studied the needs of customers, determined the goal of completing the analysis of data sets in less than two hours a day, and conducted online data search and corresponding literature reading on how to achieve this goal. Discuss related issues, such as whether the code needs to be improved to achieve the goal. Finally, it is determined that based on the analysis, certain modification of the code is the goal.

March 9th - March 13th

Read up on sequential and parallel code, and learn about code comments. The two codes were separated from parameters and variables, and divided into two files, function and processing, and were successfully implemented. However, in the process of implementation, I encountered the information that the added parameters were incomplete, and some codes were omitted.

March 14th - March 16th

By using parallel processing, set different number of cores, test 250 data sets and 5000 data sets, and get the corresponding data. But set 500,000 data set at the beginning, found it too time-consuming and gave up. And during the test, it was found that using the parallel method to test, the sequential consumes more time, and when the data set is larger, the time difference is more obvious.

March 17th - March 22nd

By comparing the recorded time, a line graph comparison was established comparing sequential and parallel processes. First, by calculating the number of cores to guess how many cores are needed to complete the goal, and then by extrapolating the icon, the specific estimated number of cores is obtained. But along the way, a search on the website explored a better mathematical way of estimating the number of cores.

March 23rd - April 5th

The code is tested for text errors and NaN errors, respectively. and log errors. And through the Internet search, I found some specific types of errors, and recorded them in the notes.

Flowchart:

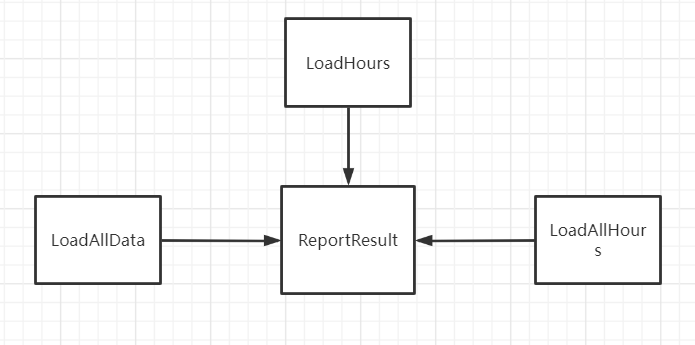
April 14th - April 15th

As required, automated testing was done. The main thing is to put all the functions in a script, so that we can test the edges by ourselves, and it is also convenient for customers to change the parameters and use them. At the beginning of the implementation, I only referenced the function, but did not save the data of each test, so through efforts, the data of each function was stored in a list and displayed on the workbench

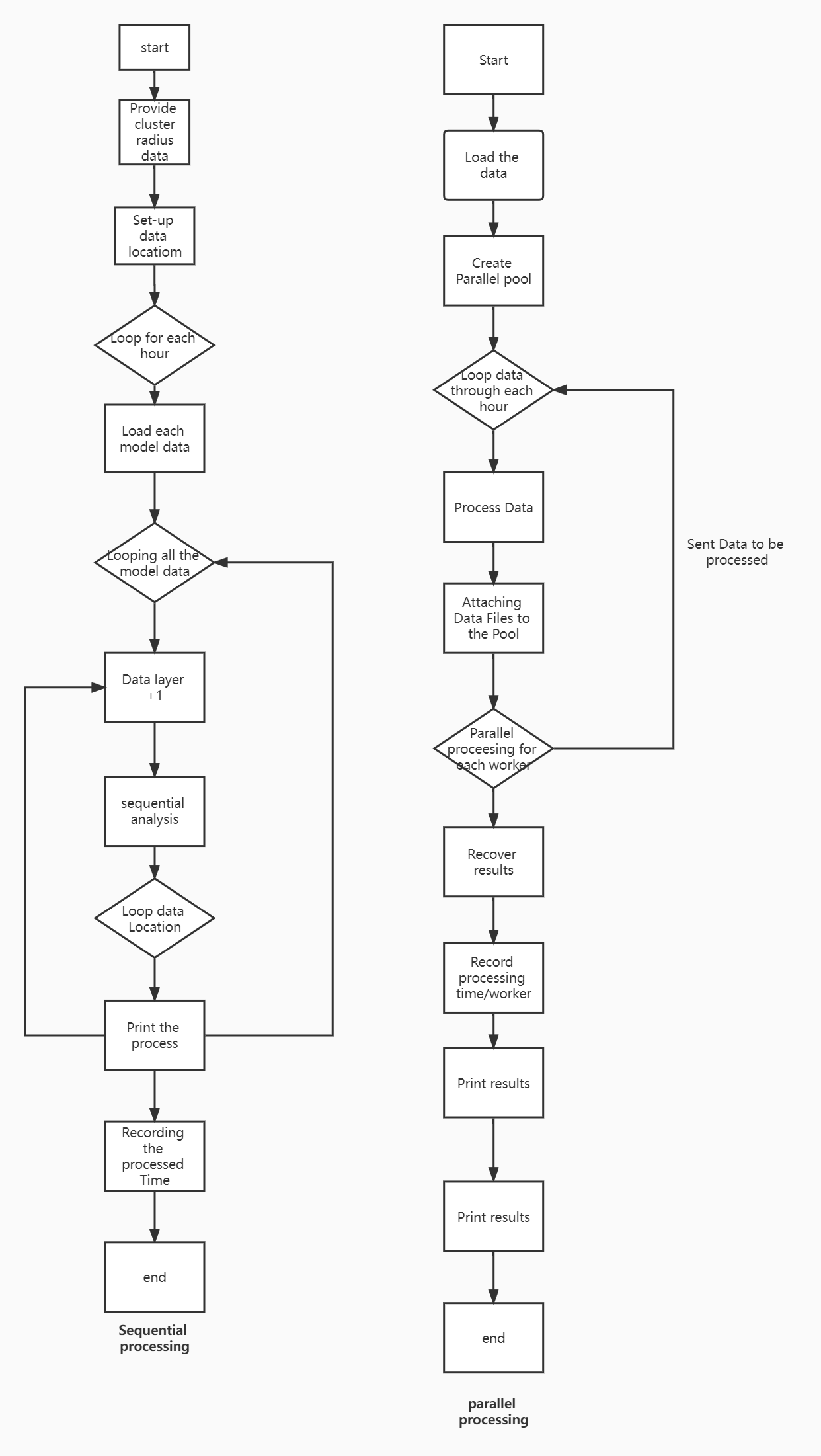
April 16th - April 19th

Finally, based on what has been done before, summarize all the things done and write a project report for the client.

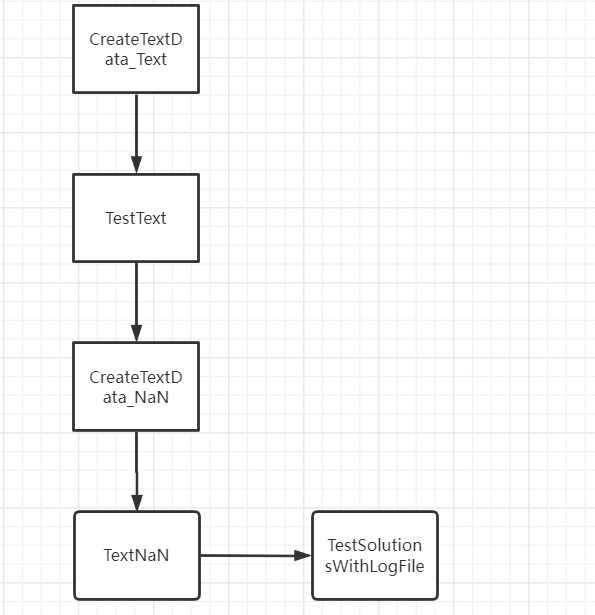
(Memory)



(sequential and parallel processing)



(Test NaN and Text)



Smart target:

S and M: requires that one day's data (25 datasets) be analyzed within one day (24 hours)

A - A dataset must be completed within two hours

R - Need to reduce the time to analyze the data to save resources

T - Your computer needs to be at x cores to analyze a day's worth of data in two hours

Summary: The goal of this project is to focus on analysis, how many cores are needed, and 25 datasets can be analyzed in two hours (one day's dataset)

Analyze data in parallel

S: Data is analyzed in parallel

M: Data can run through multiple cores

A: Data is analyzed in parallel

R: Multithreading is required through code

T: Complete parallelism, you can make line chart

Analyze data using sequential methods

S: Analyze data using sequential methods

M: Use only one core for analysis

A: The data is analyzed in a sequential manner

R: Need to implement code to "queue" data for processing

T: Complete the sequence, you can make a line chart

Comparing the efficiency of sequential and parallel

S: Comparison between sequential and parallel

M: Comparison based on time trends between sequential and parallel

A: Analysis using graphical comparisons

R: Need to record separately the time of different cores in parallel and the time of sequential for a certain amount of data

T: Complete the comparison to know which method is better

Testing against code

S: Error testing of the code

M: Error type e.g. NaN, text error

A: Implement error detection for code

R: Add conditional judgement to the test code

T: Complete the code test to know the type of error

# 8.Reference:

1. Ranshon, 2013-11-12, Sequential and parallel processing learning

https://blog.csdn.net/ranshon/article/details/15505027?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522165011135716780357235506%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request\_id=165011135716780357235506&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~baidu\_landing\_v2~default-1-15505027.142^v9^control,157^v4^control&utm\_term=sequential+parallel&spm=1018.2226.3001.4187

[2]sak2000, 2019.09.28, Understand multi-core CPU bottlenecks

<https://www.zhihu.com/question/348178453>

[3]Analyze the source of the source data

<https://www.ecmwf.int/>

[4]Matelab 2021a download

https://ww2.mathworks.cn/products/matlab.html

[5]parallel processing toolbox

https://ww2.mathworks.cn/products/parallel-computing.html