ASSIGNMENT 2

MATRIX MULTIPLICATION

NAME : L.Y.T. NIRMAL

INDEX NO : 19/ENG/072

REGISTRATION NO: EN 93921

SUBMISSION DATE: 24/12/2021

In this assignment, matrix multiplication was done by using single thread and multiple thread method. This code support both integer and decimal numbers. And there are two methods for filling matrices which are two text files inputs or random number generator.

In Single threaded processes it executes instructions in a single sequence (one command processed at one time).

In multiple threaded processes it allows the execution of multiple parts of a program at the same time. In here number of multiplication threads are equal to first matrices row count. For this POSIX Pthreads are used in here.

To demonstrate the execution times between single thread and multiple threads 100 matrices were generated from size 0 to 1000000. ($0 \times 0{,}10 \times 10{,}20 \times 20{,}20 \times 30$)

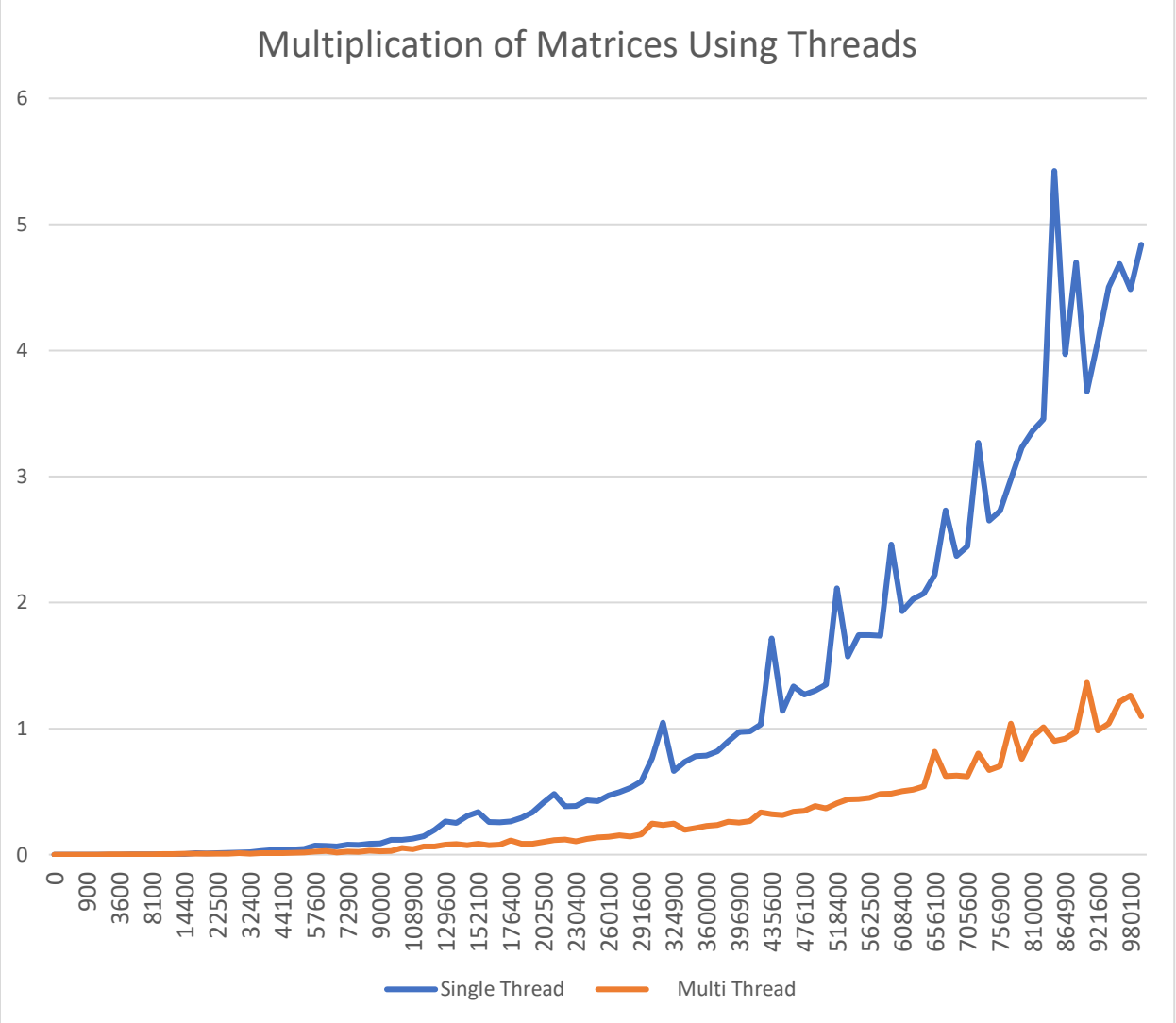

FIGURE 1 : ANALYSIS ON SINGLE THREAD AND MULTI-THREAD MATRICES MULTIPLICATION

TABLE 1 : EXECUTION TIMES FOR MULTIPLYING MATRICES USING SINGLE THREAD AND MULTI-THREADING

No. of Elements	Single Thread	Multi Thread
0	0.000	0.000
100	0.000	0.001
400	0.000	0.001
900	0.000	0.001
1600	0.001	0.001
2500	0.001	0.002
3600	0.001	0.002
4900	0.002	0.002
6400	0.002	0.003
8100	0.003	0.003
10000	0.003	0.004
12100	0.005	0.006
14400	0.005	0.007
16900	0.011	0.007
19600	0.009	0.008
22500	0.013	0.007
25600	0.015	0.008
28900	0.017	0.011
32400	0.02	0.008
36100	0.029	0.011
40000	0.037	0.011
44100	0.037	0.013
48400	0.041	0.015
52900	0.045	0.018
57600	0.071	0.023
62500	0.07	0.029
67600	0.064	0.017
72900	0.079	0.025
78400	0.078	0.022
84100	0.087	0.031
90000	0.088	0.026
96100	0.117	0.029
102400	0.118	0.053
108900	0.127	0.043
115600	0.146	0.066
122500	0.197	0.064
129600	0.264	0.08
136900	0.252	0.084
144400	0.308	0.075
152100	0.338	0.086
160000	0.258	0.074
168100	0.257	0.079
176400	0.263	0.113
184900	0.293	0.087
193600	0.335	0.087
202500	0.412	0.102
211600	0.481	0.115
220900	0.383	0.119
230400	0.386	0.106
240100	0.431	0.124
250000	0.424	0.136
260100	0.471	0.141
270400	0.497	0.153
280900	0.529	0.143
291600	0.58	0.161
302500	0.765	0.248
313600	1.047	0.235
324900	0.665	0.248
336400	0.736	0.197
348100	0.781	0.212
360000	0.787	0.227
372100	0.819	0.235
384400	0.899	0.261
396900	0.973	0.255
409600	0.978	0.266
422500	1.033	0.336
435600	1.715	0.322
448900	1.141	0.314
462400	1.334	0.341
476100	1.271	0.347
490000	1.302	0.386
504100	1.349	0.366
518400	2.114	0.408
532900	1.571	0.439
547600	1.742	0.441
562500	1.743	0.451
577600	1.738	0.481
592900	2.46	0.483

608400	1.932	0.504
624100	2.027	0.516
640000	2.072	0.541
656100	2.222	0.817
672400	2.731	0.623
688900	2.369	0.629
705600	2.446	0.62
722500	3.268	0.802
739600	2.65	0.67
756900	2.726	0.702
774400	2.979	1.041
792100	3.229	0.76
810000	3.362	0.937
828100	3.455	1.011
846400	5.425	0.9
864900	3.97	0.921
883600	4.699	0.975
902500	3.675	1.364
921600	4.073	0.984
940900	4.501	1.039
960400	4.686	1.212
980100	4.486	1.263
1000000	4.841	1.097

According to the Figure 01 and the Table 01, at the beginning of the execution, the single thread execution is more efficient than the multi thread execution. The reason is since there is small workload at the beginning it can be easily handled by using single thread. Because Multi-threading take some time to do its works like creating, initializing and joining the threads.

But when considering larger workload or when the matrices size getting bigger, the multi-threading gets better and better compared to single thread. This happens because the increasing work can be divided parallel, so the overhead is very small when comparing to the calculation time.

In this scenario, multiplication is done in row wise, and then the solutions for multiple rows will be calculated parallelly and all the solutions will be joined together. This way code can preform a big task in smaller time compared to single thread execution.

So as a conclusion, multi-threaded matrices multiplication is better for large matrices and single thread is better for small matrices.

```
C:\Users\timni\CLionProjects\untitled1\cmake-build-debug\untitled1.exe
Enter Matrix 1 size (ex. 3 4) :5 5

Enter Matrix 2 size (ex. 4 3) :5 5

[5][5] x [5][5] are Multipliable
==================================================

        1 - Fill Matrix with Text File
        2 - Fill Matrix with random values

        Select your option :1

==================================================

Text Read Matrices

-------- 1st Matrix -------

        21              26              35              60              23
        69              9               22              27              21
        5               29              60              28              73
        99              13              29              36              19
        3               25              19              21              3
```

```
-------- 2nd Matrix -------

       82              52              8               28              2
       36              89              5               28              70
       24              6               6               72              22
       28              32              25              83              20
       34              10              22              0               10


---------Multiplied Matrix---------

       5960            5766            2514            8816            4062
       7980            5595            1866            6009            2002
       6160            4827            2851            7596            4650
       10936           7821            2349            8212            2656
       2292            3197            854             3895            2624

 Matrix Size    :  25
 Time taken for Single Thread   :  0.000 s
 Time taken for Multi Thread    :  0.001 s

Process finished with exit code 0
```

Figure 02 : Input File Demo

```
Enter Matrix 1 size (ex. 3 4) :3 4

Enter Matrix 2 size (ex. 4 3) :4 3

[3][4] x [4][3] are Multipliable
==================================================

        1 - Fill Matrix with Text File
        2 - Fill Matrix with random values

        Select your option :2

==================================================

-------- 1st Matrix -------

        56              4.577776        10.986664       15.869625
        53              16              80              19.531845
        66              19              85              25.025177

-------- 2nd Matrix -------

        18              2.441481        93
        76              29              21.057772
        6.714072        3               7.324442
        5.798517        18.921476       11
```

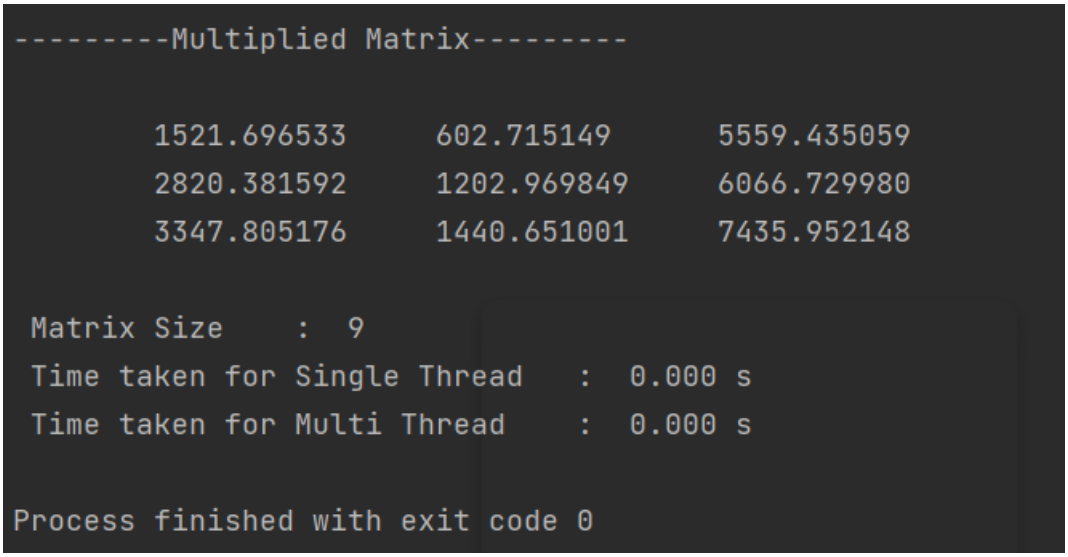

Figure 03 : Random Generated Matrix Demo