

COP4634 – Systems & Networks I

Fall 2017 – November 18, 2017

Project 3 – Large Arrays

Runtime Experiment

Ali Al-Senan

Benjamin Linam

Introduction

The program *matrix.exe* performs several functions on a global 2D array of size 20480 rows, and 4096 columns. The functions are the following:

- Write Row – Writes to the matrix in row major order.
- Write Column – Writes to the matrix in column major order.
- Read Row – Read from the matrix in row major order.
- Read Column – Read from the matrix in column major order.

Row major order goes through all the elements in a row before moving to the next row, whereas, in column major order, it goes through all the elements in a column before moving to the next column.

Experiment

The time taken to run each function is measured and compared to observe the differences with iterating through an array in row or column major order. We ran the program *matrix.exe* on the UWF SSH server, the program ran through four functions ten times each in one run, in the following order:

1. Write Row
2. Write Column
3. Read Row
4. Read Column

The functions run a nested loop, one to increment the row, and one to increment the column. Functions that use row major order have the initial loop variable used for the row, and in functions that use column major order have the initial loop variable used for the column. For the purpose of the experiment, each function is executed ten times and the time taken is measured using the *clock_gettime()* function and is stored. After running the functions ten times, the average time is calculated and outputted onto the console.

Equations:

$$t_{avg} = \frac{\sum_{n=0}^{10} t_n}{10}$$

$$\sigma = \sqrt{\frac{\sum_{n=0}^{10} (t_n - t_{avg})^2}{10 - 1}}$$

$$\sigma_{avg} = \frac{\sigma}{\sqrt{10}}$$

Results

| | WRITE ROW | WRITE COLUMN | READ ROW | READ COLUMN |
|---------------|-------------|--------------|-------------|-------------|
| TIME 1 (S) | 0.3092592 | 4.141760577 | 0.304448 | 2.650339871 |
| TIME 2 (S) | 0.2850867 | 4.126293718 | 0.3217358 | 2.695580582 |
| TIME 3 (S) | 0.2901595 | 4.218093728 | 0.2957774 | 2.60620875 |
| TIME 4 (S) | 0.2794576 | 4.115353607 | 0.3115896 | 2.657013276 |
| TIME 5 (S) | 0.3168567 | 4.136293863 | 0.296624 | 2.660730788 |
| TIME 6 (S) | 0.278637 | 4.156550468 | 0.2962605 | 2.671144225 |
| TIME 7 (S) | 0.3010205 | 4.177484438 | 0.2962287 | 2.678144237 |
| TIME 8 (S) | 0.2962871 | 4.152007459 | 0.3187118 | 2.684975542 |
| TIME 9 (S) | 0.3093221 | 4.116407709 | 0.3135096 | 2.687176748 |
| TIME 10 (S) | 0.2937311 | 4.131431686 | 0.302508 | 2.699571797 |
| AVERAGE | 0.2959817 | 4.147167725 | 0.3057393 | 2.669088582 |
| STD DEVIATION | 0.0131109 | 0.031358504 | 0.0099639 | 0.027523552 |
| STD ERROR | 0.004146021 | 0.00991643 | 0.003150861 | 0.008703711 |

Table 1: Results from running the program with standard deviation/error calculations.

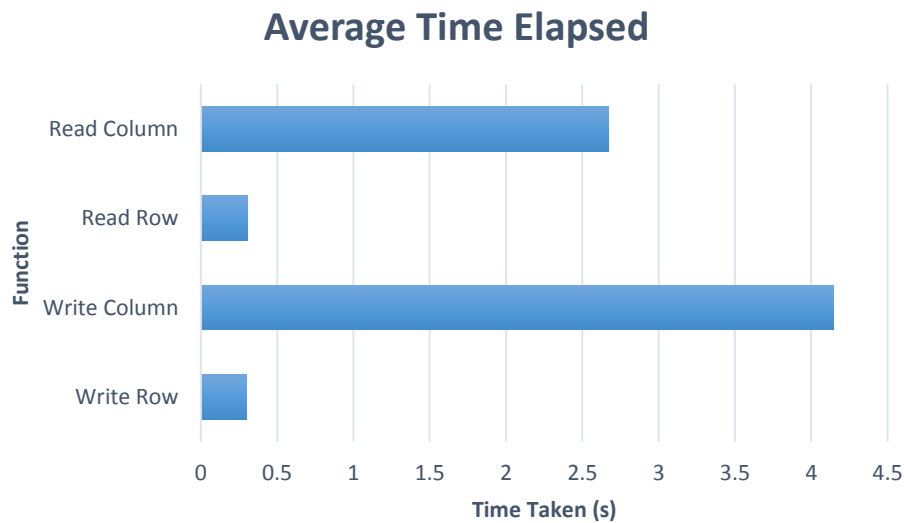


Figure 1: Average time taken for each of the four functions.

From the results acquired (Figure 1) we can see a considerable difference in time taken when using functions that use row or column major order, the functions that use row major order take considerably less time to execute than the column major order functions. A logical conclusion is that when accessing data using row major order, the data would be lined up sequentially in memory so it should take less time to execute, and on the other hand, in column major order the data is not lined up in memory so the program has to calculate the new memory locations for every array element, which should cause the function to take a longer time to execute.

The standard deviation/error values were used to plot the normal distribution and the distribution of the mean for each of the four functions.

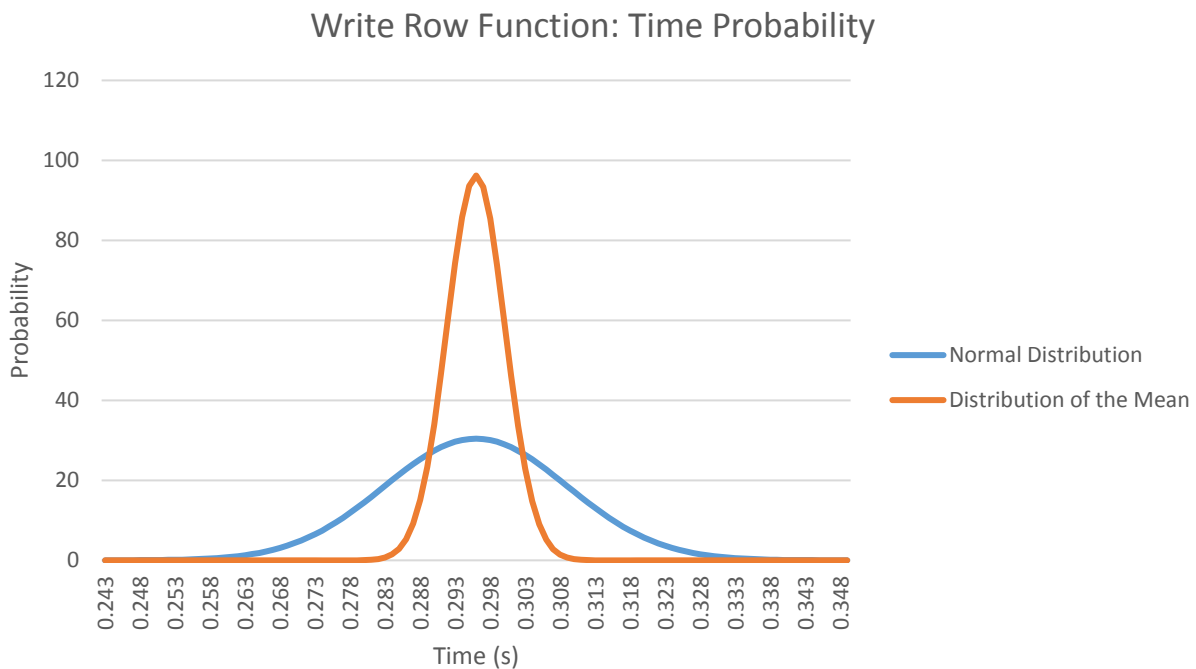


Figure 2: Distribution plot for the Write Row function.

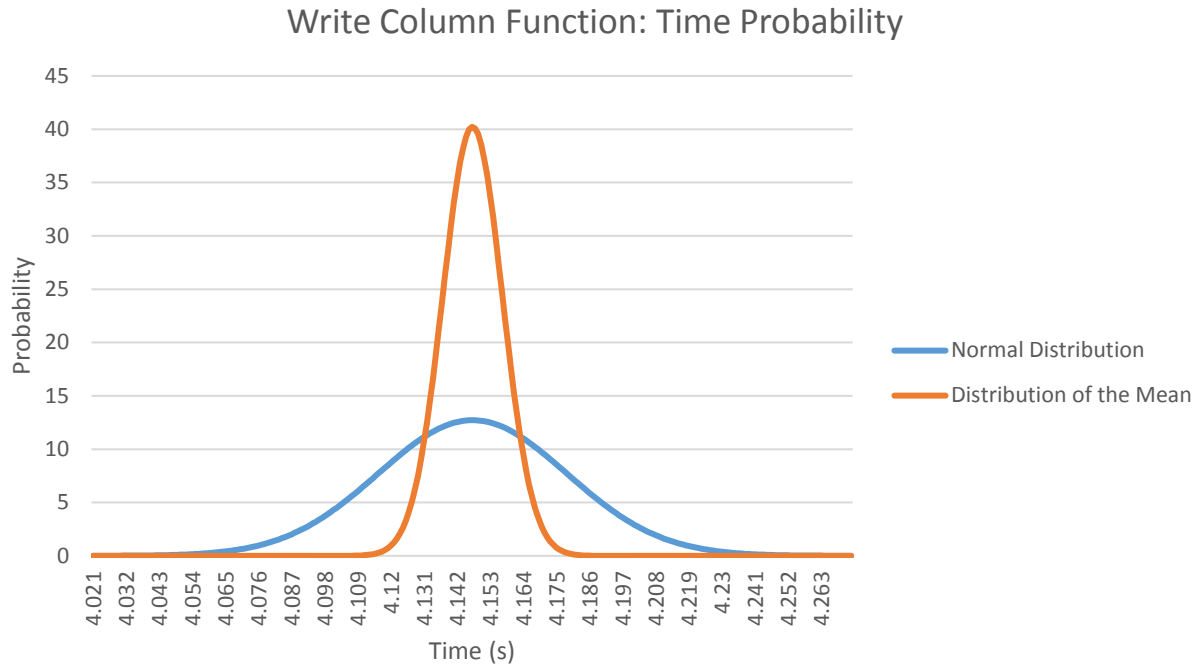


Figure 3: Distribution plot for the Write Column function.

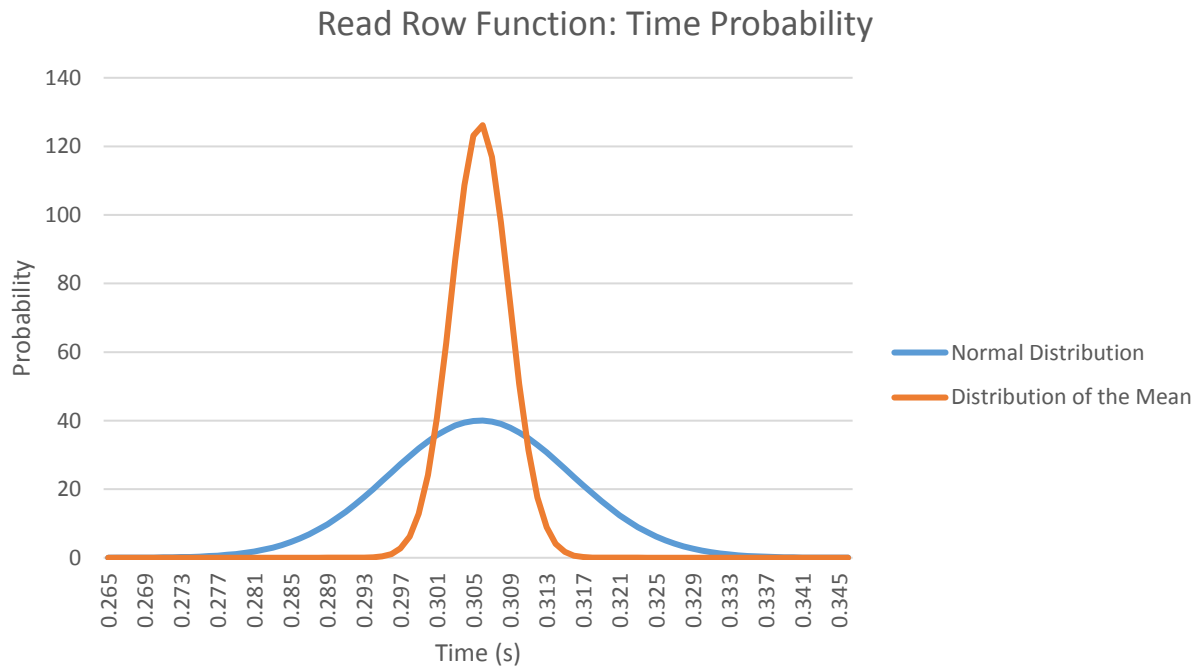


Figure 4: Distribution plot for the Read Row function.

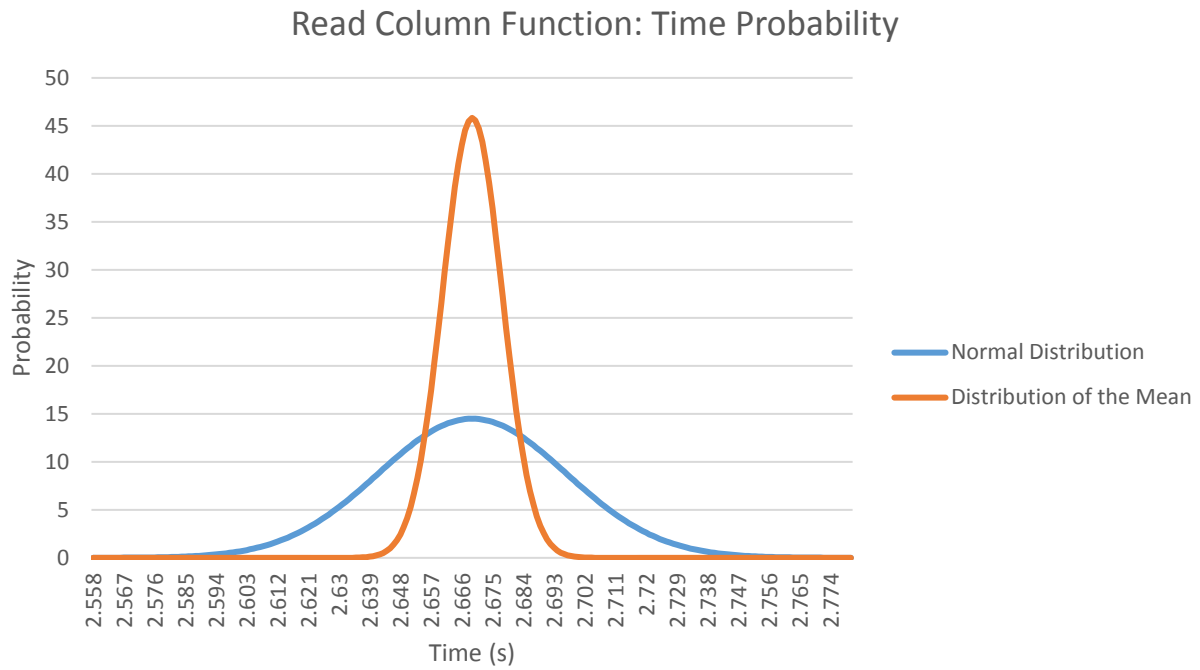


Figure 5: Distribution plot for the Read Column function.

From the distribution plots, the time needed for the function to be executed can be predicted with a certain accuracy using the standard deviation/error.