T Krishna Prashanth

CS18BTECH11045

Subject: Operating Systems 2

22 January 2020

Assignment Report Matrix multiplication

Matrix multiplication in this assignment is done in three fashions; Single-threaded, Multi-threaded, Multi-process. Each is done as a separate function and returns the time for the calculation of speed up.

PROCEDURE

The program uses the pthread library for implementing multi-threading. Uses fork() system call for producing multiple child processes. The interactive mode expects the user to enter input for matrices whereas the non-interactive mode fills the matrices automatically with random numbers using rand(); (from stdlib.h). Memory for matrices is allocated in the heap segment for single thread and multi-thread but a shared memory segment is used for each matrix separately in the multi-process procedure. Each function returns the time it takes to complete the task at the core excluding the time for input/output/initialization of matrices. However, thread creation and process creation time are not excluded in calculating time. The gettimeofday(); (from sys/time.h) function is used to calculate time.

The Algorithm.

I used the same core algorithm for division of work among processes as well as threads. The algorithm calculates no. of threads/ processes ‘n’ based on a variable ‘k’. The algorithm allocates ‘n’ such that it divides the output matrix C into feasible k x k submatrices. Each submatrix is evaluated by a separate thread/process. If after dividing the matrix into k x k submatrices any lower-order submatrices are left, that remaining pieces are also evaluated by another thread up to the end of the matrix. This way, we achieve speed up for multi-threaded and multi-process evaluation.

This abstraction of having a separate variable k to decide the number of threads/processes ensures the feasibility of matrix multiplication for any valid dimensions.

Optimization is obtained by varying k for a particular set of dimensions for maximum speedup.

The benchmark used for this optimization is

ar = ac = br = bc = 500.

Some outputs:

Each configuration of k is tested 5 times for optimization (non-interactive).

For k = 20

```
krishna@krishna-Inspiron-7572:~/Desktop/OS$ ./a.out --ar 500 --ac 500 --br 500 --bc 500
Time taken for single threaded: 398225 us
Time taken for multi process: 233671 us
Time taken for multi threaded: 171181 us
Speedup for multi process : 1.70 x
Speedup for multi threaded : 2.33 x
krishna@krishna-Inspiron-7572:~/Desktop/OS$
```

For k = 40

```
krishna@krishna-Inspiron-7572:~/Desktop/OS$ ./a.out --ar 500 --ac 500 --br 500 --bc 500
Time taken for single threaded: 378760 us
Time taken for multi process: 172308 us
Time taken for multi threaded: 177919 us
Speedup for multi process : 2.20 x
Speedup for multi threaded : 2.13 x
krishna@krishna-Inspiron-7572:~/Desktop/OS$
```

For k = 50

```
krishna@krishna-Inspiron-7572:~/Desktop/OS$ ./a.out --ar 500 --ac 500 --br 500 --bc 500
Time taken for single threaded: 370012 us
Time taken for multi process: 165656 us
Time taken for multi threaded: 170372 us
Speedup for multi process : 2.23 x
Speedup for multi threaded : 2.17 x
krishna@krishna-Inspiron-7572:~/Desktop/OS$
```

For k = 80

```
krishna@krishna-Inspiron-7572:~/Desktop/OS$ ./a.out --ar 500 --ac 500 --br 500 --bc 500
Time taken for single threaded: 379758 us
Time taken for multi process: 156398 us
Time taken for multi threaded: 168096 us
Speedup for multi process : 2.43 x
Speedup for multi threaded : 2.26 x
krishna@krishna-Inspiron-7572:~/Desktop/OS$
```

For k = 100

```
krishna@krishna-Inspiron-7572:~/Desktop/OS$ ./a.out --ar 500 --ac 500 --br 500 --bc 500
Time taken for single threaded: 446648 us
Time taken for multi process: 149471 us
Time taken for multi threaded: 170204 us
Speedup for multi process : 2.99 x
Speedup for multi threaded : 2.62 x
krishna@krishna-Inspiron-7572:~/Desktop/OS$
```

For k =250

```
krishna@krishna-Inspiron-7572:~/Desktop/OS$ ./a.out --ar 500 --ac 500 --br 500 --bc 500
Time taken for single threaded: 383936 us
Time taken for multi process: 176502 us
Time taken for multi threaded: 171075 us
Speedup for multi process : 2.18 x
Speedup for multi threaded : 2.24 x
```

For k = 500

```
krishna@krishna-Inspiron-7572:~/Desktop/OS$ ./a.out --ar 500 --ac 500 --br 500 --bc 500
Time taken for single threaded: 371214 us
Time taken for multi process: 355939 us
Time taken for multi threaded: 356819 us
Speedup for multi process : 1.04 x
Speedup for multi threaded : 1.04 x
```

CONCLUSION

The speedup here first increases for both multi-process as well as multi-threaded as k increases and then decreases when k is further increased. This speed up differs from machine to machine based on the underlying architecture. The default value of k in my program is 20.

PLAGIARISM STATEMENT <Include it in your report>

I certify that this assignment/report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment/report has not previously been submitted for assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarised the work of other students and/or persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honour violations by other students if I become aware of it.

Name: T.Krishna Prashanth

Date: 22-01-2020

Signature: TKP