

CMSC 257
Assignment 3
Due Thursday, November 9th, 2016

You are to write some variations of matrix multiplications which will examine the effects of cache and memory utilization. Your submission will be a single tar file.

1) Normal Multiplication vs Multiplying by the Transpose

The first test will be comparison of matrix multiplication where both input matrices are indexed as [row][column] versus matrix multiplication where the second matrix is transposed (meaning [column][row] accesses). I want a report as part of the assignment and a comparison of these 2 tests will be a part of the report.

2) Normal Multiplication vs Blocked Multiplication

For this comparison you need to write a version of the code which uses 6 nested loops rather than 3. The 3 outer loops will increment through blocks and the 3 inner loops will perform the multiplication of 1 block vs another adding the results to the proper location in the output matrix. Your analysis should try to determine a good block size to speed up the computation. This analysis also needs to be part of your report.

3) Normal Multiplication vs Threaded Block Multiplication

You are to write, test and report on a program to perform blocked matrix multiplication using “fork” to spawn subprocesses.

The basic goal is to allow multiple processes to work on blocks on the output matrix simultaneously. You will probably find the files shmtest.cpp, shmtest2.cpp and shmtest3.cpp from the Blackboard to be useful.

You will use the fork system call to create child processes. Note that fork is not a normal Windows system call. Don't bother trying this under Windows.

You will definitely need to use shm_open and mmap to handle the output matrix. Using a single semaphore for the entire output matrix is not optimal in terms of speed; ideally you want to assign a semaphore for each element of the matrix.

Deliverables:

- (i) turn in one code file having normal, transpose, blocked and threaded block multiplications implemented as separate functions. Turn in a report comparing the performance of these functions for large sized arrays (25,000 x 25,000).

(ii) Create a tarball of these two files and submit through blackboard.

Note: Late assignments will lose 5 points per day upto a maximum of 3 days. Code must be submitted in the prescribed format.

For questions on grading, contact the TA: Syed Khajamoinuddin <lnusk@mymail.vcu.edu>
