

EEL6763 Parallel Computer Architecture, Spring 2017

Homework #3 Assignment

Due: March 17th (Friday) by 5PM

### Summary

This homework contains two sections with various requirements for each section. Please read all material carefully.

### Submission Guidelines

- Submit (1) a single PDF containing copies of all code and answers to the discussion questions and (2) source code for each section in separate text-based .c source files.
- Sub-divide the document clearly delineating and labeling subsections of the assignment.
- Include source code by copy-pasting the complete source code from your .c files.
- Please ensure source code (in .c files and in the PDF) is consistently indented and commented.
- All code and answers MUST be your own original work and NOT copied from the Internet or elsewhere.

### Part 1: Matrix Multiply in OpenMP

Write an original program to compute matrix multiplication in C and parallelize the code using the OpenMP work-sharing directive. Report speedup after parallelization using four threads and double-precision, floating-point data type.

Your code should be submitted as a text file named hw3\_part1\_yourlastname.c (replacing <yourlastname> as appropriate) and copied appropriately into the PDF submission.

### Part 2: Matrix Multiply in MPI and OpenMP

Repeat problem 1 (above) using both MPI and OpenMP. The data from the matrix must be shared with multiple nodes (i.e. MPI tasks) (using MPI). Each node (i.e. MPI task) must have four threads running on them using OpenMP.

Submit this version of your code as a text file named hw3\_part2\_yourlastname.c (replacing <yourlastname> as appropriate) and copied appropriately into the PDF submission.

Comment on the performance of both programs from Parts 1 and 2 in the PDF file. Execute each version of the code with several different matrix sizes and comment on the changes observed (e.g. answer received, runtime, etc.) to determine parallelization efficiency. Additionally, vary the system size (number of MPI tasks/cores from 2 to 32 and OpenMP threads per core from 1 to 4) and comment on any changes observed.

### Part 3: Sobel Filter with OpenMP

Write an original program to compute a Sobel 3x3 filter on an image in C and parallelize the code using OpenMP. Report speedup after parallelization using four threads. Your input image should be of resolution 3840 x 2160 (2160p) and 8-bit precision for pixels

Submit your code as a text file named hw3\_part3\_yourlastname.c (replacing <yourlastname> as appropriate) and copied appropriately into the PDF submission.