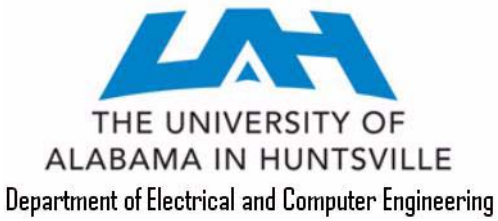

CPE 412/512-- Fall Semester 2015
Class Project Information

All CPE 412/512 students are required to complete an individual class project of their choosing that represents a non-trivial multi-core parallel/distributed processing programming assignment. This project must be an actual implementation not just a theoretical analysis.

Proposal Guidelines

(Proposal due 9/24/2015)

In proposing the project students are to write a paragraph that describes what type of problem they are planning to implement within an acceptable multicore parallel or distributed processing environment. In the proposal, students are to state what they are trying to accomplish with the parallel implementation (For example, improvement in run time performance, real time performance, fault tolerance, etc.). Also students must state how they are planning to evaluate the success of their work using an appropriate set of metrics (such as Improved Runtime, Speedup, Efficiency, etc.). Finally, students are to indicate if they are planning to use a non UAH/Alabama Supercomputer Center platform or some other application-specific parallel resource. The proposal should be submitted electronically to the instructor for approval on Canvas™ by the expected due date.

Final Reports

(Final Report due 12/8/2015)

Students are required to submit electronically an original final written report on Canvas™ by the posted due date which gives an overview of the projects purpose, major results, and conclusions along with the associated program source code if that is appropriate. Students may demonstrate the project in person to the instructor or they can supply the instructor with the necessary code, data set, and instructions to allow him to execute the demonstration on his own.

This report should include the following sections or their equivalents

Problem Introduction

Parallelization Approach Employed

(your approach to applying multi-processing and/or multi-threading improved performance or some other metric that you identified.

Empirical Method and Results

(compare your parallel results with the sequential representation using measured execution times and appropriate metrics such as speedup, efficiency, cost, etc.)

General Conclusions

(How effective were your parallel implementations? What were sources of bottlenecks? What are the limits of your approach? Do you have other ideas that may provide better results in the future? Can your parallel method be applied to other problems?)

Appendix

Source Code for your problem, example program runs

No specific style is mandated but IEEE format is recommended. Final Reports are due by the last day of class. No minimum or maximum page limit.

The final should be submitted electronically to the instructor for approval on Canvas™ by the expected due date.

CPE 412/512-- Fall Semester 2015

Previous Class Project Titles & Topics

Experimental Analysis of the cost of interprocess communication on the Amazon™ Cloud.

Parallel Computational Fluid Dynamics Simulation of two dimensional compressible flows.

Parallel processing for digital circuit test pattern generation.

A comparison of using OpenMP and Matlab to implement Data Compression Algorithm

Utilizing CSP Style Programming and ADA derived techniques in MPI

A parallel implementation of a steganography detection algorithm

Parallel Simulation of a Tether Dynamics Problem using pThreads

Parallel processing of a one dimensional decimation-in-time FFT Algorithm

Implementing a Three dimensional FFT Algorithm on a Network of Workstations

Parallel Particle-in-Cell Simulation of a plasma dynamics problem using MPI.

Parallel Particle-in-Cell Simulation of a plasma dynamics problem on a GPU.

Parallel Implementation of a bionic sorting routine

Parallel processing of a matrix operations on a heterogeneous network of workstations

Implementing a Wave-Front/Systolic Array Processing

Evaluation of the Effectiveness of employing Ordinary Differential Equations on a Parallel Computer

A Parallel Bank Transaction Processing Simulation on a Network of Workstations

Parallel Processing of the Classical Wave Equations

A Parallel Simulated Annealing Implementation Applied to the Travelling Salesman Problem

Distributed simulation of a simulated battle field

An Efficient Parallel FIR Filter implementation using OpenMP.

Parallel implementation of Particle Swarm Optimization, PSO, to solve the segmented optical mirror problem using OpenMP.

Parallel particle swarm optimization applied to an inverse heat transfer problem to determine the thermal characteristics of a material.

The application of a distributed genetic algorithm to a nonlinear optimization problem

An OpenMP Implementation of a Synthetic Aperture Radar

An pThreads Implementation of a Ray-Tracing code for Medical Imaging Application.

Solving a Branch-and-bound Optimization Problem using OpenMP