CSCI 415 and 515 – Parallel Computation
Spring 2017

Time: MTWF 9am
Place: CF 420
Instructor: Phil Nelson
Office: CF471 (650-3035)
Office Hours: 1pm MWF, others by appointment
E-mail: phil.nelson@wwu.edu is my e-mail address. Please put “CS 415” or “CS 515” in your subject to help your e-mail make it through my spam filters. Also, using plain text (no HTML) e-mails helps get your e-mail delivered to me.


Goals: The goal of this class is to give the student a solid understanding of the basics of parallel computation and parallel algorithm design and analysis. Also, it is expected that the student will have successfully implemented several parallel algorithms using a real parallel machine.

Course Outcomes: At the end of the class the student will have the following outcomes:

- A thorough understanding of parallel machine architecture.
- A thorough understanding of parallel programming techniques.
- A basic understanding of parallel languages.
- The ability to design and implement programs in a parallel environment.

Graded Work: 3 tests, written and programming assignments. For 515, a additional research paper.

Tests: The first two tests will be given in class. Each test will be worth 17% of your final grade. The first test is scheduled for Wednesday, April 19, 2017. The second test is scheduled for Friday, May 12, 2017. The final is scheduled for Wednesday, June 7 at 8am.

Reading (CSCI 515): You will choose a paper from the current literature (back to 2005 or so) on parallel computation, read it and write a 3 to 5 page paper reporting on the paper. This will be worth 5% of your final grade. Your paper should be selected and approved by the professor no later than May 8, 2017. In reading the paper, you may have to read other papers to get the proper background.

Written work and programming: Each week or two you will be given an assignment. The assignments will be worth 44% of your final grade. (49% for CSCI 415) The last assignment will be due on the last day of dead week.
Late work: Work is due at the beginning of class on the day due. Work will be accepted up to two class periods late. Late work is worth 75% of the original value.

Grading: Grading is usually done by a percentage of the top score. If the class does not perform as a whole to what I believe to be A work, I reserve the right to not use the top score as the reference point. If the top score is within 10% of the total possible, I will use the top score as the reference point. The following is my usual grade scale.

A: 100% – 90%
B: 89% – 80%
C: 79% – 65%

Collaboration: Each student MUST do their own assignments and programming. Original work is required. Assume you should not see the source code of any other student. You may discuss problems using diagrams on scratch paper, but you should not see source code. Even helping a fellow student debug their program so that source code is seen should be avoided. Students having problems should e-mail me or visit me in my office. (Group projects, if assigned, should see only their own group’s work.)

Cheating: Is (obviously) not allowed. If you do cheat and are caught you will receive an F as your grade for the class. This includes ALL students knowingly involved in any cheating event. Not properly protecting your source code may be considered knowingly involved. I may use mechanical means to compare student programs. These comparisons are used to raise the possibility of cheating, but all decisions about cheating will be made by me after inspecting the programs of all students involved.

Web Access: I have information for this class on the web. As they are assigned, I will put a copy of each assignment on the web. See the page http://www.cs.wwu.edu/nelson/classes/s17/515. Other information will be provided via the web.

Topic Outline: An approximate outline of topics. Topics takes from both primary text, on-line text and other sources like published papers and so forth..

- Brief History and Introduction
- Parallel, Distributed vs. Threaded
- Intro to Parallel
- Introduction to MPI
- Parallel Architecture
- Classes of Machines
- Communication Networks
- Parallel Algorithms and Design
- Finding Parallelism
- Generic techniques
- Parallel Algorithm Models/Paradigms
- Communication Operations
- Analysis of a parallel algorithm
• Corollary of Modest Potential
• Dense Matrix Algorithms
• Matrix/Vector multiplication
• Matrix/Matrix multiplication
• Message Passing Interface
• Various Algorithms: Sorting, Graph, Dynamic Programming, FFT
• Parallel Languages (Chapel)
• GPU programming (Supplementary text)