CSCI 415 and 515 – Parallel Computation
Spring 2025

Time: MTWF 11 AM

Place: CF 420

Instructor: Phil Nelson

Office: CF467

Office Hours: MTWF 3:00PM-4:00PM, others by appointment

E-mail: phil@wwu.edu is my e-mail address. Please put "CSCI 415" or "CSCI 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.

Text: Grama, Gupta, Karypis, Kumar, *Introduction to Parallel Computing*, 2nd Edition, Addison Wesley, 2003.

Supplementary Text: Matloff, *Programming on Parallel Machines*, self published at `https://heather.c`

Class Web: `https://facultyweb.cs.wwu.edu/~phil/classes/s25/515`.

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 cluster.

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 programing 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 23, 2025 The second test is schedule for Friday, May 16, 2025 The final is scheduled for Monday, June 9 at 8:00AM.

Reading : CSCI 515 students will choose a paper from the current literature (back to 2015 or so), 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 2, 2025. 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, June 6, 2025.

Late work: Work is due at 11pm on the day due. Late work will be accepted up to two class days late. (For example, if the assignment is due on Wednesday, the next "class day" is Friday and the second "class day" would be the following Monday on which the late work is due.) Late work is worth 75% of the original value. A late final assignment is worth 75% of the original up until the start of the final.

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%

D: 64% – 50%

Collaboration: Each student *MUST* do their own programming. Original work is required. You should not see the source code of any other student, current or past, on this project. You may discuss problems using diagrams on scratch paper, but you should not see source code. You should not work together in designing solutions even with pseudo code. Even helping a fellow student debug their program so that source code is seen should be avoided. Programming at the same time "sitting" next to another student (either in person or virtually) while actively writing code together also should not be

done. Using generative AI to do your work is not original work. Students having problems should e-mail me or visit me in my office. (See above.)

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. If you give your password to your friend or allow access to your files or a machine on which your sources are stored, this can be considered knowingly involved. I 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. *NOTE:* Making your source files for this class available for public, unprotected access will be considered cheating and may even get you an F for this class *AFTER* a passing grade has been reported and you have graduated. (The University's policies and procedures regarding academic honesty are published in the catalog, Appendix D.)

Western Syllabus Policies: For generic syllabus policies of Western, visit `https://syllabi.wwu.edu`.

Topic Outline: An approximate outline of topics: GPU programming may be moved to earlier in the quarter and some of the later topics may not be covered.

- Brief History and Introduction (Ch 1)
- Parallel, Distributed vs. Threaded
- Intro to Parallel
- Introduction to MPI (Ch 6)
- Parallel Architecture (Ch 2)
- Classes of Machines
- Communication Networks
- Parallel Algorithms and Design (Ch 3)
- Finding Parallelism
- Generic techniques
- Parallel Algorithm Models/Paradigms
- Communication Operations (Ch 4)
- Analysis of a parallel algorithm (Ch 5)

- Corollary of Modest Potential
- Dense Matrix Algorithms (Ch 8)
- Addition
- Matrix/Vector multiplication
- Matrix/Matrix multiplication
- Message Passing Interface (Ch 6)
- Sorting (Ch 9)
- Graph Algorithms (Ch 10)
- Pipelined Algorithms
- Dynamic Programming (Ch 12)
- Fast Fourier Transform (Ch 13)
- Chapel (Parallel Languages)
- GPU programming (Supplementary text)