CSCE 5160 Parallel Processing and Algorithms

Spring 2024

Instructor:	Song Fu
Office:	F250
Email:	song.fu@unt.edu
Office Hours:	Tuesday, 2:00-3:00pm Thursday, 10:00-11:00am
Teaching Assistant:	Jubair Ibn Malik Rifat (JubairIbnMalikRifat@my.unt.edu)
Office Hours:	Monday, 11:00am-12:00pm Wednesday, 11:00am-12:00pm F296B

Breakdown of Course Grade

Homework Assignments (including programming labs)	25%
Mid-semester exam	25%
Final exam	25%
Term Project	20%
Attendance/Discussion	5%

The course is designed to introduce issues involved in parallel programming along with efficient parallel algorithms and an analysis of the algorithms. Programming exercises will involve the use of MPI, OpenMP, Cuda, OpenCL and/or Pthreads.

Prerequisites: Programming in C, C++, Linux/Unix, computer organization, Data Structures, Algorithms, complexity analysis and measuring performance.

Term Project: The goal of the project is for you to explore parallel processing to an application of interest based on your research. You should consider implementing an application using either OpenMP, MPI or CUDA or a combination. Alternatively, you can learn a new parallel programming language, libraries (e.g., MapReduce, Parallel Haskell, OpenCL), simulator, or implement some applications/algorithms from recent publications.

Academic Integrity: All students will be trusted to pursue their academic careers with honesty and integrity. Academic dishonesty includes, but not limited to, cheating on a test or other course work, plagiarism, unauthorized collaboration with other persons. Students found guilty of dishonesty will be subject to penalties that may include suspension from the university. For more information about your rights and responsibilities, visit https://vpaa.unt.edu/ss/integrity

CSCE 5160 Parallel Processing and Algorithms

Tentative Schedule

1. Introduction — 3 hours
    - Motivation
    - Multiprocessor architectures, Networking
    - Levels of parallelism
2. Performance Models — 3 hours
    - Performance and Speedup
    - Scalability models
3. Communication and Coordination — 4 hours
    - Communication models
    - Synchronization models
    - Analyzing communication overhead
    - Analyzing synchronization overhead
4. Parallel Programming — 6 hours
    - Message passing and Shared memory
    - Using MPI, OpenMP, Pthreads, CUDA, OpenCL
5. Parallel Algorithm Design — 3 hours
    - Task level and data level decomposition
6. Matrix Algorithms — 6 hours
    - Matrix inverse
    - Matrix-Vector multiplication
    - Matrix Multiplication
7. Solving Linear Systems — 4 hours
    - Iterative methods
    - Conjugate Gradient Method
8. Sorting — 6 hours
    - Parallel sorting
    - Parallel search
9. Graph Algorithms — 4 hours
    - Spanning trees
    - Shortest paths
10. Search and optimizations — 4 hours
    - Load balancing
    - Termination

Textbook:

A. Grama, A. Gupta, G. Karypis and V. Kumar. Introduction to Parallel Computing

Useful Reference Books:

1. R. van der Pas, E. Stotzer and C. Terboven: Using OpenMP-The Next Steps, MIT Press
2. W. Gropp and E. Lusk. Using MPI: Portable parallel programming with Message-Passing Interface, MIT Press

3. J.Sanders and E. Kandrot. CUDA by Example: An introduction to general purpose GPU Programming, Addison-Wesley
4. N. Wilt. The CUDA Handbook: A comprehensive guide to GPU programming, end Edition
5. B. Wilkinson and M. Allen. Parallel Programming, 2nd Edition
6. M. Herlihy and N. Shavit. The Art of Multiprocessor Programming
7. J. Dongarra (Editor) The Sourcebook of Parallel Computing
8. Michael Quinn: Parallel Programming in C with MPI and OpenMP
9. B. Chapman. Using OpenMP: Portable Shared Memory Parallel Programming