CS-425 Parallel Programming and Algorithms

The CS-425 course introduces you to the foundations of parallel computing including the principles of parallel algorithm design, analytical modeling of parallel programs, programming models for shared- and distributed-memory systems, parallel computer architectures, along with numerical and non-numerical algorithms for parallel systems. The course will include material on emerging multicore hardware, shared-memory programming models, message passing programming models used for cluster computing (suggested by my mentor John), data-parallel programming models for GPUs, and problem-solving on large-scale clusters using MapReduce. A key aim of the course is for you to gain a hands-on knowledge of the fundamentals of parallel programming by writing efficient parallel programs using some of the programming models that you learn in class.

Prerequisites: CS-401 Programming Methods for Data Science
CS-231 Programming Paradigms

Class Schedule: 240 minutes of lecture

Textbooks

- Multicore and GPU Programming: An Integrated Approach 2nd Edition, Gerassimos Barlas (2022)
- Introduction to Parallel Computing, 2nd Edition, Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar (2003)
- Parallel Programming in C with MPI and OpenMP by M.J.Quinn, McGraw-Hill Science/Engineering/Math, 1st Edition, 2003, ISBN: 0072822562
- Using OpenMP: Portable Shared Memory Parallel Programming - Barbara Chapman, Gabriele Jost, Ruud van der Pas (2008)
- Using MPI: Portable Parallel Programming with the Message-Passing Interface, 3rd Ed - William Gropp, Ewing Lusk, Anthony Skjellum (2014)
- Programming Massively Parallel Processors: A Hands-on Approach, 3rd Ed. - David B. Kirk, Wen-mei W. Hwu (2016)

Topics

1. Introduction to Parallel Computing
2. Parallel Programming Platforms
3. Principles of Parallel Algorithm Design
4. Basic Communication Operations
5. Analytical Modeling of Parallel Programs
6. Programming Using the Message Passing Paradigm, e.g., Message-Passing Interface (MPI)
7. Programming Shared Address Space Platforms
8. Dense Matrix, Sorting, Searching, and Graph Algorithms
9. Graphics Processing Units (GPUs)
10. Compute Unified Device Architecture (CUDA)

Course Learning Objectives
Each student who receives credit for CS-425 will have demonstrated the ability to do the following tasks:

- Students will be able to apply the concepts of a Parallel Computer Architecture by creating a parallel program that will maximize the performance of the parallel program.
- Students will be able to apply the concepts of Message Passing to the creation of a program that executes efficiently on the parallel computer architecture.
- Students will be able to apply the concepts of CUDA to the creation of a program that executes efficiently on the parallel computer architecture.
- Students can apply the concepts of Performance to the analysis of computer performance problems.
- Students can apply the concepts of Performance Counters to the analysis of parallel program performance.

Administrative Information:
Lecture Time & Location: TBD
Lab Time & Location: TBD
Instructor: Rui Zhu, Ph.D.
Email: rzhu@kettering.edu
Office: 2300C at the Academic Building
Student (Office) Hours: Wednesdays 10:05am-12:05pm, or by appointment via email
Contact Number: (810) 762-7927

Course Requirements and Grading:

Course Breakdown:
Attendance and In Class Activities: 10%
Quizzes: 10%
Midterm: 25%
Final: 25%
Lab/Programming Assignments: 30%

Exams and Quizzes:

A midterm examination, worth 25% of the final course grade. A final examination, worth 25% of the final course grade, will be held during the final examination period. The date will be determined later in an upcoming announcement.

Five quizzes, each worth 2% of the final course grade. Examinations and quizzes will be open book with allowance for notes. Except for emergencies, makeups must be arranged in advance of the announced examination date. Naturally, answers produced should be entirely your own work. Examinations and quizzes will cover material presented in lecture and announced sections of the textbooks. You are responsible for all material presented in class.

Lab/Programming Assignments:

There will be ten programming assignments; cumulatively, programming assignments will account for 30% of the final course grade. Individual due dates for programming assignments will be announced. Unless otherwise stated in the assignment, programming assignments should be entirely your own work. You are permitted to discuss programming assignments with fellow classmates in a general nature; you are not permitted to jointly develop algorithms and/or code. You are not permitted to copy algorithms and/or code from other students or from outside sources, or to give or loan code to other students. You are not permitted to use code-generating programs to assist you in writing your code. Please avoid the code content on the web. Most of that code is spaghetti code or invalid. You will get more partial points for your own code.

Late programming assignments will be assessed a 20% penalty for the first day or portion thereof after the specified due date, and an additional 10% penalty per day or portion thereof afterwards.

Attendance and In-Class Activities:

Many classes will include some sort of in-class programming exercise. Individual requirements and weights will be announced. Cumulatively, in-class activities will account for 10% of the final course grade.

Course Grades and Grading Policies:

Midterm course grades will be projected based upon performance on all completed assessments prior to that date. Estimated course grades will be updated as the term progresses.

Final course grades will initially be computed using the relative weights described above.

A class curve will be applied. Individual grades may be adjusted to reward improvement over the course of the semester. The official interpretations of the Kettering University Grading Scheme in the Kettering University Catalog will be used to guide the above adjustments. Questions regarding individual scores should be addressed to the instructor within one week of receiving the score.

All suspected cases of academic dishonesty will be handled in strict accordance with Kettering University policy. Any questions regarding appropriate behavior should be cleared with the instructor in advance.

Attendance:

Attendance at all lectures and laboratory sessions is strongly encouraged.

University policy requires instructors to report students who appear to have stopped attending class for administrative withdrawal from the course. Consistent failure to complete homework or exams, without making contact with the instructor, may result in administrative withdrawal.

Changes to Syllabus:

This syllabus provides a general plan for the course; deviations may be necessary.