Parallel Computing

COMP 633

Fall Semester 2018

http://www.cs.unc.edu/~prins/Classes/633/

Meeting Times:

Tue, Thu 11:00 – 12:15 in FB 007 (Tue Aug 21 – Tue Dec 4)

Instructor:

Jan Prins (FB 334, prins@cs.unc.edu, 919-590-6213)
Office hours: Mon 2-3 PM and by appt.

Course Description

This is an introductory graduate course on parallel computing, organized around various parallel computing models (e.g. shared memory, distributed memory). Within each model, we develop and analyze sample algorithms, study practical issues such as programming language and hardware support, and undertake performance prediction and experimental performance analysis. Example algorithms include sorting, graph algorithms, linear algebra operations, and key algorithms from scientific computing (e.g. N-body, FFT).

Target Audience

The target audience is graduate or advanced undergraduate computer science students. The course is also appropriate for graduate students in mathematics or sciences using computational methods (provided they have a strong background in computing).

Prerequisites

Undergraduate-level familiarity with the design and analysis of sequential algorithms (e.g. COMP 550), elementary operating systems concepts (e.g. COMP 530), and knowledge of basic computer organization (e.g. COMP 411) are expected.

Goals and Key Learning Objectives

Upon completion, the student should

(a) be able to design and analyze parallel algorithms for a variety of problems and computational models,

(b) be familiar with the fundamentals of the architecture and systems software of high-performance parallel computing systems, and

(c) have implemented parallel applications on high-performance computing systems, and be able to measure, tune, and report on their performance.

Course Announcements and Information

The definitive source for course announcements, reading assignments, reference materials, and class handouts is the course web page at the top of this handout. Please consult it regularly! We will utilize Piazza for online discussion about the course outside of class time.

Text and Readings

There is no single text that adequately covers the material in this class. Course readings are drawn from articles in the technical literature, textbook chapters, and instructor notes. A bibliography and access to the readings is maintained on the course web page.

Grading

Grades will be based on three written assignments (33% total weight), two exams (33% total weight), two programming assignments performed individually or by two students together (32% total weight), and participation in class and Piazza online discussions (2% total weight). Class attendance is not counted, but is strongly recommended, since most material in the course will not be available any other way.

Key dates

The midterm is scheduled to be held in class Thursday, October 11. The final exam, scheduled by the registrar, will be held Thursday December 13th, noon - 3pm in our classroom.

Honor Code

Assignments may be discussed with your COMP 633 classmates only (and the instructor) to gain insights into the problem and possible solution strategies, but in all cases your submissions must be written individually (or within your group, for programming assignments completed as a team of size two). Soliciting help outside of Piazza and permitted discussions is specifically prohibited. In exams you may access any materials disseminated to the class, and your class notes. If you use a computer to access these materials, any other use, such as communication with others, or search, is prohibited.

The Honor Code and the Campus Code are in effect for this course. I am committed to treating Honor Code violations seriously and urge all students to become familiar with its terms as set out at http://instrument.unc.edu. If you have questions, it is your responsibility to ask me about the Code's application. All exams, written work, and programming projects must be submitted with a statement that you complied with the requirements of the Honor Code in all aspects of the submitted work.

Related Courses

There are several courses on parallel and distributed computing offered in our department.

COMP 633 (this course) is concerned with the design and implementation of scalable parallel computations, i.e. a single problem solved using multiple processors operating simultaneously to decrease time to completion. Its focus is algorithms, programming models, architectures, and performance analysis.

COMP 734 (Distributed Systems) is concerned with the provision of ongoing reliable services to geographically dispersed users. The focus is networks, server architecture, protocols, security, resiliency, and scalability.

COMP 735 (Distributed and Concurrent Algorithms) is concerned with the specification and proof of safety and liveness properties of key algorithms used in concurrent systems such as mutual exclusion. Its focus is the application of formal techniques.

Computer Usage

The Phaedra machine in the CS department will be dedicated for use by the class and supports shared-memory and accelerator programming. The campus research computing clusters (longleaf and dogwood), may also be used and are the only way to work with the MPI programming model. Access to these clusters requires an account and must be discussed with the instructor. In all cases you will be making use of shared resources. Please observe the usage guidelines and reservation policies for all systems. Use common sense and monitor your program's consumption of resources when performing runs.

SYLLABUS

1. COURSE INTRODUCTION (1)
2. SHARED MEMORY MODELS (12)
   - PRAM and Work-Time Models: algorithm design and analysis techniques, relative power and limitations of PRAM models. (4)
   - Memory Models: parallel memory-hierarchy and locality, UMA, NUMA and CC-NUMA shared memory architectures. (2)
   - Loop-Level Parallelism: loop iteration distribution in OpenMP, performance measurement and tuning. (2)
   - Task-Level Parallelism: run-time task scheduling and load-balancing in Cilk and OpenMP 3.0. Nested parallelism. (2)
   - Memory coherence and consistency, implementation of synchronization and mutual exclusion operations in cache-coherent multiprocessors. (2)
3. HETEROGENEOUS PROCESSING MODELS (4)
   - Computational accelerators: Nvidia GPU and Intel Xeon Phi architecture, programming, and performance. (4)
4. DISTRIBUTED MEMORY MODELS (8)
   - Bulk Synchronous Processing Model: algorithm design, communication cost measures, performance prediction and measurement. (3)
   - Partitioned Global Address Space Model: one-sided communication, data-distribution, UPC. (1)
   - Message Passing Model: SPMD programming, Message Passing Interface (MPI), collective communication. (1)
   - Interconnection Networks: topology and performance metrics, routing, and flow control; implementation of collective communication operations.(2)
   - Distributed storage systems and programming models: Hadoop, Spark; google web search. (1)

Disclaimer

"The instructor reserves to right to make changes to the syllabus, including project due dates and test dates. These changes will be announced as early as possible."