

Parallel Computing

COMP 633

Fall Semester 2011

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

Meeting Times:

Tue, Thu 12:30 – 1:45 in FB 007 (Tue Aug 23 – Tue Dec 6)

Instructor:

Jan Prins (FB 334, Tel: 962-1913, prins@cs.unc.edu)

Office hours: TBD and when door is open

Overview

This is an introductory graduate course covering several aspects of high-performance computing, primarily focusing on parallel computing. Upon completion, you 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 experience with the implementation of 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!

Text and Readings

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

Grading

Grades will be based on approximately three or four written assignments ($\approx 1/3$ weight), three programming assignments performed individually or in groups of two ($\approx 1/3$ weight), and two exams ($\approx 1/3$ weight). All assignments may be freely discussed with your COMP 633 classmates, but your submissions must be written individually (or within your group, for programming assignments).

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.

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. Its focus is algorithms, programming models, architectures, and performance analysis.

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

We will use the BASS computer (62 nodes, 452 cores, 180 gpus) for all programming assignments. Bass supports all programming models we will consider this semester, but provides limited scaling for shared-memory models.

Additional resources, when appropriate, are available through campus research computing (Kill Devil, Kure, and Emerald clusters) and through RENCi.

In all cases you will be making use of shared resources that are heavily subscribed. Please observe the usage guidelines and reservation policies for all systems. Use common sense and monitor your program's consumption of resources when performing large-scale runs.

Syllabus

The material in this course is organized around various models of parallel computing. With each model we will develop and analyze sample algorithms, and study some practical issues such as hardware implementation, algorithm design, and performance evaluation. Example algorithms include sorting, graph algorithms, linear algebra operations, and key algorithms from scientific computing (FFT, fast summation).

1. COURSE INTRODUCTION (1)
2. SHARED MEMORY MODELS (16)
 - 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. (2)
 - Multi-level parallel models: GPU architecture and CUDA. (2)

- Thread-Level Parallelism: abstractions for exclusion and synchronization: locks, monitors and conditions. (2)
 - Memory coherence and consistency, implementation of synchronization and mutual exclusion operations in cache-coherent multiprocessors. (2)
3. DISTRIBUTED MEMORY MODELS (9)
- Bulk Synchronous Processing Model: algorithm design, communication cost measures, performance prediction and measurement. (3)
 - Message Passing Model: SPMD programming, Message Passing Interface (MPI), collective communication. (2)
 - Partitioned Global Address Space Model: one-sided communication, data-distribution, UPC.(2)
 - Interconnection Networks: topology and performance metrics, routing, and flow control; implementation of collective communication operations. (2)
4. DATA-INTENSIVE AND CLOUD COMPUTING MODELS (2)
- Server architecture, google web search, distributed file systems and programming models: BigTable, MapReduce.