CSCI 3145 INTRODUCTION TO PARALLEL PROGRAMMING
Fall 1998
Tuesday/Thursday 8:00 am to 9:20 am

Parallel computing uses multiple computers, or computers with multiple internal processors, to solve a problem at a greater computational speed than using a single computer. It also offers the opportunity to tackle larger problems, problems with more computational steps or more memory requirements. In this first course in parallel computing, students will learn parallel computing techniques and algorithms, and have practical experiences writing parallel programs for networks of workstations using either MPI (message-passing interface) or PVM (parallel virtual machine). We will also consider thread-based programming using Pthreads (an IEEE standard).

Course text:

Parallel Programming: Techniques and Application Using Networked Workstations and Parallel Computers, by Barry Wilkinson and Michael Allen, Prentice Hall Inc., 1998, ISBN 0-13-671710-1.

Material provided on how to use MPI/PVM and networked workstations at http://www.cs.uncc.edu/par_prog. Class assignments etc. posted here.

For reference:

Pacheco, P. (1997), *Parallel Programming with MPI*, Morgan Kaufmann Publishers Inc.: San Francisco, California.
Geist, A., A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam (1994), *PVM3 User's Guide and Reference Manual*, Oak Ridge National Laboratory: Tennessee.
Kumar, V., A. Grama, A. Gupta, and G. Karypis (1994), *Introduction to Parallel Computing*, Benjamin/ Cummings Publishing Company Inc.: Redwood City California.

Prerequisites:

Basic knowledge of C and UNIX essential (or willingness to learn quickly). A dedicated workstation cluster is available at UNCC for all students in the class.

Course contents: (Ch1-11 of course text)

Parallel computers: architectural types, shared memory, message passing, interconnection networks, meshes, hypercubes, etc., potential for increased speed.
Basic techniques: embarrassingly parallel computations, partitioning and divide and conquer, pipelined computations, synchronous computations, load balancing and termination detection, shared memory programming.
Algorithms and applications: sorting algorithms, searching algorithms, numerical algorithms, image processing algorithms.

Assessment:

Programming assignments	50%
Class tests (2)	30%
Final exam	20%

(The percentages may be altered.)

All submitted assignments must be your own work. Copied work or work done by more than one person unless specifically instructed will not be accepted.

Instructor: Barry Wilkinson
Department of Computer Science
University of North Carolina at Charlotte
Room: Kennedy 224 Tele: 547 4879 Email: abw@uncc.edu

Office Hours: Tuesday/Thursday: 9:30 am to 10:30 am
2:30 pm to 3:30 pm