CSE 4392-601: PARALLEL SOFTWARE TOOLS

(Summer 2002: T R 1:00-2:50, Nedderman 110)

Instructor: Bob Weems, Associate Professor
Office: 344 Nedderman, 817/272-2337, weems@uta.edu
Hours: T R 3:00-5:30

GTA: Yongsheng Bai
Office: 412 Wolf Hall (yxb4544@omega.uta.edu)
Hours: T R 3:00-6:00

GTA: Zhengheng Li
Office: 412 Wolf Hall (zhli@cse.uta.edu)
Hours: M W 3:00-6:00

Prerequisite: Operating Systems (CSE 3320)

Objective: Introduction to the variety of topics necessary for developing parallel software.

Goals:

1. Ability to implement small applications on shared-memory multiprocessor using pthreads and OpenMP.
2. Ability to implement small applications in message-passing paradigm using MPI.
3. Understanding of concepts of parallel algorithms
4. Understanding of elementary topologies and communication techniques
5. Understanding of compiler concurrentization concepts

Textbooks: G.R. Andrews, *Foundations of Multithreaded, Parallel, and Distributed Programming*, Addison-Wesley, 2000.

P.S. Pacheco, *Parallel Programming with MPI*, Morgan Kaufmann, 1997. (optional)

Course notes - available in PDF form from `http://reptar.uta.edu/NOTES4351/cse4351.html`.

References: S.G. Akl, *The Design and Analysis of Parallel Algorithms*, Prentice-Hall, 1989.

Almasi and Gottlieb, *Highly Parallel Computing*, Benjamin/Cummings, 1989.

D.R. Butenhof, *Programming with POSIX Threads*, Addison-Wesley, 1997.

F.T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann, 1992.

M.J. Quinn, *Parallel Computing: Theory and Practice*, McGraw-Hill, 1994.

J.H. Reif (ed.), *Synthesis of Parallel Algorithms*, Morgan Kaufmann, 1993.

Exams: Two take-home exams (see calendar for topic coverage)

Homeworks: Homework problems are integrated with the notes. Solutions are included with the notes.

Programs: Four assignments to reinforce the algorithm ideas and to provide exposure to practical issues. Alternate project(s) may be proposed based on your interests/research.

Grade: Based on the following weights:

Exams:	40% (divided evenly among the two exams)
Programs:	60% (divided evenly among the six assignments)

Policies:

1. Faithful attendance is expected. Consult me in advance if you must miss class.

2. If you need a handout, check with me during office hours.

3. CHEATING - YOU ARE EXPECTED TO KNOW UNIVERSITY POLICIES. If you are suspected of cheating, the matter must go through university channels outside of the CSE Department.

4. Any request for special consideration must be appropriately documented <u>in advance</u>. (Special consideration does not include giving a higher grade than has been earned.)

5. Late programs are penalized according to the following schedule. LABS ARE DUE AT 3:30 PM, NOT MIDNIGHT. After the due date I will not provide assistance (nor will the grader).

Degree of lateness	Penalty
Up to 4:45 next day	10 pts
Up to 4:45 two days	30 pts
Up to 4:45 three days	60 pts

Late labs submitted on non-class days should be submitted to the CSE Dept. receptionist, who will timestamp the submission.

6. Each student will have available *one* two-day no-penalty extension that may be applied to *one* of the lab assignments. To use your extension you must send an email to `yxb4544@omega.uta.edu` *before* the due time. He will send an acknowledgement.

7. Each lab is graded as follows:

Correctness	30%
Efficiency	30%
Coding Style	15%
Analysis	25%

If your program is not working correctly, you should show what portions of your program do work. *No credit will be given for analysis of incorrect programs.*

8. Electronic mail. I try to check my mail at least twice a day, less frequently on weekends. Please include your name in the message.

9. Before logging off the Linux systems, check for orphan processes by using `ps -g`. Use `kill -9 <processid>` to remove them. Leaving active processes (or ‘‘core’’ files) will lead to suspension of account privileges.

YOU MUST BE LOGGED IN WHEN RUNNING ANY PROCESSES ON THE LINUX SYSTEMS!!!

SIMILARLY, CODE SHOULD BE DEBUGGED BEFORE RUNNING ON THE 4-PROCESSOR COMPAQ ALPHA SMP!!!

10. GTA duties:

a. Provide first-level of assistance for labs.
b. Grade programs.
c. Proctor examinations.

11. Instructor duties:

a. Lecture.
b. Guidance
c. Tests - preparation and grading.
d. Special consideration.
e. Design homework and programming assignments.

Course Contents and Readings - number to left is lecture note #, section numbers in parentheses refer to MPI book or Andrews book

1. Motivation/Introduction (MPI: 1.1-1.6, 11.1-11.9, 12.1-12.8, Andrews Chapter 1)
   History of high performance computing - government, academe, and industry
   Challenge problems
   Classic metrics
2. Data parallel programming techniques using pthreads (Andrews 4.6, 5.5, 12.1.1)
   Process management
   Shared memory
   Synchronization
   Problem decomposition- static (interleaved, contiguous) and dynamic approaches
   Examples
   Enumeration of combinatorial objects - another example of contiguous decomposition
   Multiple processor ‘‘binary’’ search/finding roots
   Multiple processor merging of ordered tables
3. Elementary message-based programming using MPI (MPI: 3.1-3.6, 4.1-4.5, 5.1-5.10, 9.1-9.7, 10.1-10.7)
   Process management
   Messages
   Examples
   MPI collective operations
4. Synchronization: shared memory and message passing (Andrews Chapters 2-6)
   Barriers
   Highly-concurrent queue
   Concurrent AVL trees
   Four forms of message passing: procedure call, process creation, asynchronous message, rendezvous
   Termination detection: Mattern’s credit-recovery technique and Dijkstra-Scholten Tree Technique

TEST 1

5. Interconnections (MPI: 2.1-2.4)
   - Linear Array/Mesh/Toruses
   - Butterfly/Fat tree
   - Benes/Hypercube Routing via perfect matching
   - Mesh routing - static and dynamic
   - Overview of other communication problems
     - Broadcasting
     - Sorting (MPI: 14.3-14.4)
6. Numerical problems - Systems of linear equations
   - Gaussian
   - LU
   - Householder
   - Iterative/Sparse
7. Task graph scheduling (function parallelism) & load balancing
8. DOACROSS style parallelism
9. PRAM and other theoretical models of parallel computing (Andrews 3.5)
   - Isoeffiency
   - Prefix sums
   - Matrix multiplication
   - Elementary list ranking
   - Euler tours and tree traversals
   - P-Completeness and its meaning to the parallel computing practitioner
10. Other problems that are difficult to parallelize
    - Discrete event simulation
    - Heuristic search
    - Two-person games

TEST 2

Calendar/Topics

Calendar - with course content numbers

	May/June			July/August			
28	Syllabus/1.	30	2.	2	5.	4	Holiday
4	2.	6	2.	9	Test 1 due	11	6.
11	3.	13	3.	16	7.	18	8.
18	4.	20	4.	23	8.	25	9.
25	5.	27	5.	30	9.	1	9.
				6	10.	8	
				13	Test 2 due		

July 23 is the last day for both undergraduate and graduate students to withdraw.

CSE 4392 Survey

(Please submit by the end of the third meeting)

Name:

Email:

Other Courses this Semester:

Special Circumstances Affecting your Performance:

CS Related Experience/Interests:

(Optional) What do you hope to gain from this course?