15-495: Parallel Computer Architecture and Programming
Spring 2002
Syllabus

1 Course Details at a Glance

Lectures:	Tuesdays & Thursdays, 9:00-10:20 a.m., NSH 1305
Instructor:	Todd C. Mowry, WeH 8123, 268-3725, `tcm@cs.cmu.edu`, Thu 10:30-11:30am
TA:	Mahim Mishra, WeH 8203, 268-3562, `mahim@cs.cmu.edu`, Tue 1:30-2:30pm
Class Admin:	Maury Burgwin, WeH 8124, 268-4740, `mburgwin@cs.cmu.edu`
Web Page:	`www.cs.cmu.edu/afs/cs/academic/class/15495-s02/www/`
Newsgroup:	`cyrus.academic.cs.cs495`
Handouts:	*Electronic:* `/afs/cs.cmu.edu/academic/class/15495-s02/public` *Hardcopies:* In bins outside WeH 8124.

2 Textbook

The following textbook is required for the course:

- David E. Culler and Jaswinder Pal Singh, with Anoop Gupta. *Parallel Computer Architecture: A Hardware/Software Approach*, Morgan Kaufmann, 1998. ISBN: 1558603433.

We will be following this book quite closely, and we will covering much of the material in the book.

3 Course Description

The goal of this course is to provide a deep understanding of the fundamental principles and engineering tradeoffs involved in designing modern parallel computers (aka "multiprocessors"), as well as the programming techniques to effectively utilize these machines. Parallel machines are already ubiquitous from desktops to supercomputers, and the expectation is that they will become even more commonplace in the future. However, very few people exploit the potential processing power of these machines because they do not understand how to write efficient parallel programs. Because one cannot design a good parallel program without understanding how parallel machines are built and vice-versa, this course will cover both parallel hardware and software design, as well as the impact that they have on each other.

Course topics include naming shared data, synchronizing threads, and the latency and bandwidth associated with communication. Case studies on shared-memory, message-passing, data-parallel and dataflow machines will be used to illustrate these techniques and tradeoffs. Programming assignments will be performed on one or more commercial multiprocessors, and there will be a significant course project.

This is a relatively unique course since this material is rarely offered to undergraduates. Because parallel processing has become such an important and mainstream technology, the time has come to integrate this material into the undergraduate systems curriculum. This is the first time that this course has been offered at Carnegie Mellon. Assuming that all goes well, the current plan is to offer it again this coming Fall.

4 Prerequisites

15-213 (Intro to Computer Systems) is a strict prerequisite for this course. We will build directly upon the material presented in 15-213, including memory hierarchies, memory management, basic networking, etc.

While 18-347 (Intro to Computer Architecture) would be helpful for understanding the material in this course, it is not a prerequisite.

5 Computer Accounts

To complete your programming assignments and course projects, you will be receiving accounts on machines at some of the supercomputing centers: mostly likely either the National Center for Supercomputing Applications (NCSA) or the Pittsburgh Supercomputing Center (PSC). Details will be provided later.

Important: please note that the class will be allocated a finite (and not particularly large) amount of time on these machines, so please be careful not to waste time unnecessarily.

6 Course Work

Grades will be based on homeworks, a project, two exams, and class participation.

Homeworks: There will be roughly three parallel programming assignments (which we will call "labs"), which you will work on in groups of two. (If you have difficulty locating a partner, please post a message to the class newsgroup.) Turn in a single writeup per group. In addition, we may have a few written assignments that will focus more on parallel architecture rather than programming.

Project: A major focus of this course is the project. We prefer that you work in groups of two on the project, although groups of up to three may be permitted depending on the scale of project (ask the instructor for permission before forming a group of three). A typical project would involve designing, implementing and evaluating a fairly ambitious parallel program (perhaps on more than one architecture). Some groups may choose to do projects that evaluate a new parallel architecture idea. The project must involve an experimental component—i.e. it is not simply a paper and pencil exercise. We encourage you to try to come up with your own topic for your project (subject to approval by the instructor), although we can make some suggestions if necessary. You will have roughly six weeks to work on the project. You will present your findings in a written report (the collected reports may be published as a technical report at the end of the semester), and also during a poster session during the last day of class. Start thinking about potential project ideas soon!

Exams: There will be two exams, each covering its respective half of the course material. Note that the second exam is not cumulative, and is weighted equally with the first exam. Both exams will be closed book, closed notes.

Class Participation: In general, we would like everyone to do their part to make this an enjoyable interactive experience (one-way communication is no fun). Hence in addition to attending class, we would like you to actively participate by asking questions, joining in our discussions, etc.

6.1 Grading Policy

Your overall grade is determined as follows:

Homework:	25%
Project:	25%
Exams:	40% (20% each)
Class Participation:	10%

Late assignments will not be accepted without prior arrangement.

7 Schedule

Table 1 shows the tentative schedule. Since this is the first time that this class is being taught, it is likely that there may be some significant variations.

Table 1: 15-495, Spring 2002, Tentative Schedule.

Class	Date	Day	Topic	Reading	Assignments
1	1/15	Tue	Why Study Parallel Architecture?	1.1	
2	1/17	Thu	Evolution of Parallel Architecture	1.2	
3	1/22	Tue	Fundamental Design Issues	1.3-4	
4	1/24	Thu	Parallel Programming: Overview	2.1-4	
5	1/29	Tue	Parallel Programming: Performance	3.1-4	L1 Out
6	1/31	Thu	Parallel Programming: Case Studies	3.5	
7	2/5	Tue	Implications for Programming Models	3.6-7	
8	2/7	Thu	Scaling Workloads and Machines	4.1	L1 Due, L2 Out
9	2/12	Tue	Evaluating Machines and Ideas	4.2-5	
10	2/14	Thu	Shared Memory Multiprocessors I	5.1-3	
11	2/19	Tue	Shared Memory Multiprocessors II	5.4	L2 Due, L3 Out
12	2/21	Thu	Snoop-Based Multiprocessor Design I	6.1-4	
13	2/26	Tue	Snoop-Based Multiprocessor Design II	6.5-7	
14	2/28	Thu	Scalable Distributed Memory MPs	7.1-8	L3 Due
	3/5	Tue	**Exam I**		
	3/7	Thu	*No Class: Mid-Semester Break*		
15	3/12	Tue	Directory-Based Cache Coherence I	8.1-5	
16	3/14	Thu	Directory-Based Cache Coherence II	8.6-7, 8.9-11	Project Proposal
17	3/19	Tue	Relaxed Memory Consistency Models	9.1	
18	3/21	Thu	Cache-Only Memory Architectures	9.2.2, 9.4	
19	3/26	Tue	Synchronization	5.5, 7.9, 8.8	
20	3/28	Thu	Interconnection Network Design I	10.1-4	Project Milestone 1
			Spring Break		
21	4/9	Tue	Interconnection Network Design II	10.5-10	
22	4/11	Thu	Latency Tolerance: Overview	11.1-5	
23	4/16	Tue	Latency Tolerance: Prefetching	11.6	Project Milestone 2
24	4/18	Thu	Latency Tolerance: Multithreading	11.7-9	
	4/23	Tue	**Exam II**		
25	4/25	Thu	Terascale Computing System at PSC	N/A	
26	4/30	Tue	Earthquake Simulation Case Study	N/A	Project Due
	5/2	Thu	**Project Poster Session**		