CS315a/EE382b: Parallel Computer Architecture and Programming
Information Sheet
http://eeclass.stanford.edu/cs315a

Time & Location:		
Monday & Wednesday	9:30-10:45 a.m.	McCullough 115
	Broadcast T,Th 8:00-9:15 a.m. on channel E2	
Problem Session:		
Fridays	4:15pm-5:05pm	Gates B01
	Broadcast live on E4	

Instructor:	Kunle Olukotun	
EMail:	kunle at stanford dot edu	
Location:	Gates 302	
Telephone:	650-725-3713	
Office Hours:	MW 10:45am-11:45am	
	or by appointment	
Teaching Assistants:	Austen McDonald	Mike Houston
EMail:	austenmc at cs dot stanford dot edu	mhouston at cs dot stanford dot edu
Location:	Gates 326	
Office Hours:	T 1-3pm, Th 12-2pm	
Course Support:	Darlene Hadding	
EMail:	darleneh at stanford dot edu	
Location:	Gates 408	
Telephone:	650-723-1430	

****Paper handouts are available in Gates Bldg. 3rd floor lobby filing cabinet.****

Handouts are also available on the web

Mailing List: A class mailing list that will be used for important or late-breaking announcements. Sign up on eeclass for this mailing list as soon as possible.

Grading: Final grades will be computed approximately as follows:

Class participation	5%
Paper Reviews	5%
Problem Sets:	10 %
Programming Assignments:	35 %
Midterm:	20 %
Final:	25 %

Units: 3 units

Prerequisites: All students are expected to have developed a background in computer architecture by taking EE282 or an equivalent course. Programming assignments will assume that students have a working knowledge of C/C++, and are comfortable with writing reasonable-sized applications (a few hundred lines of code, at least), preferably in a UNIX-based environment.

Reviews: The review of each paper is due by the 9:30am on the day that the paper is covered in class. Reviews should be emailed to the TA cs315a-spr0506-tas at lists dot stanford dot edu with:

- Subject: cs315a review [?]

The body of the message should be 20 to 50 lines with:

- a paragraph summarizing the problem attacked or goal of the paper,
- a paragraph summarizing the paper's methods (if any) and results, and
- a paragraph giving your opinion of what is good and bad about the paper.

Participation: You are expected to participate in the class discussion, in fact 5% of your grade depends on it. To facilitate discussion, I will call upon you at any time during a lecture to answer a specific question. Please come with a name tag that you can prop up on your desk to enable me to call you by name. Make sure I can read your name from the front of the class. SCPD students have the option of attending three of the lectures in which we discuss papers or writing a one page review of one of the paper discussions. If you are an SCPD student, let me know you are in class at the beginning of the lecture.

Readings: The required texts for this course are: 1) *Computer Architecture: A Quantitative Approach*, 3nd Edition, by John L. Hennessy and David A. Patterson, Morgan-Kaufmann, 2003. ISBN: 1-55860-596-7. The book is available at the Stanford Bookstore, although most students should already have a copy from EE282. 2) *Introduction to Parallel Computing*, 2nd Edition, by Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, Addison-Wesley, 2003. ISBN: 0-20164-865-2. This is also available at the Stanford Bookstore. In addition, we will discuss the contents of several papers that are available on the course website: (Reviews are due for papers with a *)

Parallel Applications

[1] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 Programs: Characterization and Methodological Considerations," *Proc. 22nd International Symposium on Computer Architecture*, Santa Margherita Ligure, Italy, June 1995.

[2] L. Barroso, K. Gharachorloo, and E. Bugnion, "Memory System Characterization of Commercial Workloads," *Proc. 25th Annual International Symposium on Computer Architecture (ISCA'98)*, Barcelona, Spain, pp. 3-14, June 1998.

Locking and Memory Consistency

[3] (Optional) S. V. Adve and K. Gharachorloo, "Shared memory consistency models: a tutorial," *IEEE Computer*, vol. 29, no. 12, pp. 66–76, Dec. 1996.

[4] M. D. Hill, "Multiprocessors should support simple memory consistency models," *IEEE Computer*, vol. 31, no. 8, pp. 28–34, Aug. 1998.

Chip-Multiprocessors (CMPs)

[5] K. Olukotun and L. Hammond, "The future of microprocessors," *ACM Queue*, vol. 3, no. 7, pp. 26–34, September 2005.

[6] L. Barroso, K. Gharachorloo, R. McNamara, A. Nowatzyk, S. Qadeer, B. Sano, S. Smith, R. Stets, and B. Verghese, "Piranha: A Scalable Architecture Based on Single-Chip Multiprocessing," *Proc. 27th Annual International Symposium on Computer Architecture (ISCA'00)*, Vancouver, British Columbia, Canada, pp. 282–293, June 2000.

[7] P. Kongetira, K. Aingaran, and K. Olukotun, "Niagara: A 32-Way multithreaded SPARC® processor," *IEEE Micro*, vol. 25, no. 2, March/April 2005.

Thread-Level Speculation (TLS)

[8] M. J. Garzaran, M. Prvulovic, J. M. Llaberia, V. Vinals, L. Rauchwerger, and J. Torrellas, " Tradeoffs in buffering memory state for thread-level speculation in multiprocessors," Proc. *9th International Symposium on High-Performance Computer Architecture (HPCA*), February 2003.

[9] L. Hammond, B. Hubbert, M. Siu, M. Prabhu, M. Chen, and K. Olukotun, "The Stanford Hydra CMP," IEEE MICRO, vol. 20, no. 2, pp. 71–83 March-April 2000.

Transactional Memory

[10] B. D. Carlstrom, J. Chung, A. McDonald, H. Chafi, C. Kozyrakis, and K. Olukotun, "The atomos transactional programming language," Proc. ACM SIGPLAN 2006 Conference on Programming Language Design and Implementation, Ottawa, Canada, June 10–16 2006.

[11] A. McDonald, J. Chung, H. Chafi, C. C. Minh, B. D. Carlstrom, C. Kozyrakis, and K. Olukotun, "Characterization of TCC on chip-multiprocessors," *Proc. 14th International Conference on Parallel Architecture and Compilation Techniques (PACT 2005)*, St. Louis, MO, Sept. 17–21 2005.

[12] K. Moore, J. Bobba, M. Moravan, M. Hill and D. Wood, "LogTM: log-based transactional memory," *International Symposium on High Performance Computer Architecture (HPCA)*, February 2006.

Distributed Shared Memory

[13] J. Laudon and D. Lenoski, "The SGI Origin: A ccNUMA Highly Scalable Server," *Proc. 24th International Symposium on Computer Architecture*, Denver, CO, pp. 241–251, June 1997.

Website: The class website is located at:

http://eeclass.stanford.edu/cs315a

All important class information including lecture notes, homework assignments and solutions, and information about the programming assignments will be posted to this site. The website also includes a Frequently Asked Questions (FAQs) section for problem sets and programming projects. **Check the website frequently** since new information and announcements will be added regularly.

Course Newsgroup su.class.cs315a

Problem Sets: There will be three problem sets during the quarter. Solving the problems is critical to learning the material and understanding the concepts presented in this course. Since there is often a

significant benefit to teamwork, students can work in groups of two, although this is not required. For groups, a single copy of the answers should be submitted with both students' names. Problem sets are due at 5 PM on the due date noted in the course schedule.

Programming Assignments: There will be three programming assignments during the quarter, which like the problem sets may be completed individually or in groups of two. The programming projects will be written in C/C++, using Pthreads and OpenMP parallel programming interfaces. Unless you have access to a fairly large parallel machine elsewhere, most will require that you run code in parallel on one or more of the three large servers available in Sweet Hall: the two, 8-processor "tree" machines and/or the 16-processor "junior" machine. Each individual or group must submit its own code *and* answers to questions associated with the programming assignments. Please note that submitting code written by another person is considered a violation of the Honor Code. Code for all programming assignments will be submitted electronically and must be submitted by 11:59 PM on the due date. Answers to questions associated with the programming assignments are also due the same day, and may be submitted electronically as text and/or PDF files along with the code (by 11:59 PM) or on paper (by 5 PM). Electronic submission instructions will be provided with the assignments.

Late Policy To account for unforeseen emergencies (illness, accidents, disk drive crashes, network troubles, etc.), every student has one free problem set "late day" to use on either problem sets 1 or 3 (due to the proximity of the midterm and final, no late days are allowed on problem set 2). A late day is a no-penalty extension to the following class day and must be submitted by 5 PM. Late days are tracked individually. A 15% penalty will be applied to each subsequent late problem set. **A maximum of one late day is permitted on any given problem set.** Please contact the course staff if an extenuating circumstance arises. In addition to the one late day for problem sets, everyone also has one programming project late day. Late day programming projects are due the following class day at 11:59 PM (or 5 PM for question answers on paper). A 15% penalty will be applied to each late assignment after the first. **A maximum of one late day is permitted on any given programming assignment.**
NOTE: If you take a late day for PA3, it will be due 5 PM June 9.
NOTE: You must write "use late day" on your problem set and programming project. If you do not indicate that you are using a late day on a late assignment, a 15% penalty will be applied.

Exams: There will be one 2-hour exam during the quarter and a take-home exam given during the final week of class. The exam dates are as follows:

Midterm:	Wednesday, May 10 7:00 – 9:00 PM, Gates B03
Take home Final:	Wednesday, June 7, 5 PM
	Due Thursday, June 8, 5 PM, Gates 408

Both exams are open-book and open-notes. No electronic devices other than a calculator will be permitted for the midterm. Anything may be used (except people) for the take home. Local SCPD students are expected to come to campus for the midterm exam and remote students will receive the exam through SCPD. Remote SCPD students are expected to take the exam the same day as local students.

Section: The TA will have a weekly review session on Fridays at 4:15pm-5:05pm in Gates B01 (Live on E4). These sessions clarify topics covered during lecture and review special topics. Attendance is optional, but highly recommended. The sessions will be televised and available via Stanford Online.

Honor Code: The Honor Code is taken seriously in this course and suspected violations are referred to the Office of Judicial Affairs. Expectations for this course are covered in detail in the Honor Code handout.

CS315a/EE382b
Tentative Course Schedule
Spring 2005/06

Date	Lecture	Subject	Reading	PS /PA Assigned	PS/PA Due
Wed Apr 5	1	Introduction, course overview	[5], CA: 6.1, PC: 1, 2.3		
Mon Apr 10	2	Shared Memory Programming	PC: 7		
Wed Apr 12	3	Parallel Programming Techniques I	PC: 3.1-3, 3.6	PA1	
Mon Apr 17	4	Parallel Programming Techniques II	PC: 3.4		
Wed Apr 19	5	Parallel Debugging and Performance Optimization Analyzing Parallel Algorithms	CA: H.3, PC: 3.5	PS1	
Mon Apr 24	6	Example Applications: regular and SPLASH-2	PC: 5, *[1]	PA2	PA1
Wed Apr 26	7	Example Applications: irregular and commercial	CA: 6.2, PC: 8, *[2]	PS2	PS1
Mon May 1	8	Snoopy Cache Coherence I	CA: 6.3-4,		
Wed May 3	9	Snoopy Cache Coherence II	CA:I.1		
Mon May 8	10	Consistency & Synchronization Techniques	CA: 6.7-8, [3], *[4]		PS2 **no late day**
Wed May 10		**Midterm Exam**, 7:00-9:00 p.m., Gates B03	Lectures 1-9		
Mon May 15	11	CMPs - latency	[5],[8],*[9]	PA3	PA2
Wed May 17	12	Programming with Transactions	*[10]	PS3	
Mon May 22	13	Transactional memory (Austen)	*[11], [12]		PA3.1
Wed May 24	14	Directory-based Cache Coherence I	CA: 6.5-6		
Mon May 29		*Holiday* – Memorial Day – No Class			
Wed May 31	15	Directory-based Cache Coherence II	CA: 6.11, I.2, [13]		
Mon June 5	16	CMPs - throughput	*[6],[7]		PS3
Wed June 7	17	Wrap Up & Review, Take home final distributed @ 5 PM	Lectures 1-17	Take home Final	PA3.2 @ 5 PM
Thur June 8		Take home final due @ 5 PM, Gates 408			

Notes on PS/PA and readings
PS1 = Parallel Programming, Analyzing Parallel Algorithms
PS2 = Example Apps, Snoopy Cache Coherence
PS3 = Consistency/Synchronization, CMPs, TLS, Transactions, Directories

PA1 = Using pthreads/OpenMP & basic parallel techniques (toy benchmarks, a “programming problem set”)
PA2 = Shared memory programming of an irregular benchmark
PA3 = Architecture and programming evaluation of transactions

* = You should turn in a review for this paper ([1], [2], [4], [6], [9], [10], [11])