ECE/CS 757: Advanced Computer Architecture II

Spring Semester 2017, MWF 1:00-2:15pm EH 2534

Instructor: Prof. Mikko Lipasti, mikko@engr.wisc.edu, EH3621

http://ece757.ece.wisc.edu

Course Description

This course covers parallelism and the design of parallel computers. Historically, parallel computers have been designed for the sole purpose of quickly solving large-scale computational problems like weather forecasting or molecular modeling (to name just two examples). These problems are usually expressed as a series of floating-point computations of large data sets stored in multidimensional arrays, and can usually be partitioned across multiple processors to achieve large-scale parallelism. However, within the last fifteen years, new applications for parallel computers have eclipsed these traditional numeric codes, and are the driving force behind the tremendous volume and revenue growth in the marketplace for parallel computers. These applications span all the way from commercial server workloads that run in managed datacenters, to heavily-threaded games and web browsers running on PCs, laptops, and phones, to massively data-parallel applications like graphics rendering. In other words, parallelism in applications and in hardware has become pervasive in our industry.

This course will study the nature of parallelism across these application domains, as well as the hardware required to support parallel execution. We will investigate techniques for detecting, increasing, and exploiting parallelism across this spectrum of workloads, and will study in detail the design of various components of parallel computer systems. The discussion will rely heavily on examples of real or proposed parallel designs.Prerequisites: ECE 552 (or equivalent) and CS 537 (not strictly enforced). **NOTE: ECE 752 is not a prerequisite for this course.**

Course Textbook

There is no required textbook for this course. Instead, we will utilize online resources and a significant number of readings from the literature.

Recommended: Michel Dubois, Murali Annavaram, and Per Stenström. Parallel Computer Organization and Design. Cambridge University Press, 2012.

Lectures & Readings

It is very important that you attend lecture faithfully. Much of the material will be covered only in lecture, as the book does not fully cover all of the material and the readings are by definition out of date. Lecture slots are overscheduled; we will meet more often than necessary in the first half of the semester to free up time in the second half for project work. Many lecture times will be devoted to discussing the readings in detail. You are expected to be prepared to present detailed summaries, views, and opinions on the assigned readings (refer to course schedule and check website for updates).

Homework

Homework will be assigned but not all of it will be collected or graded; it's purpose is to help you learn the material and prepare for the midterm exams.

Paper Reviews and Quizzes

You must submit reviews for a subset of the papers on the reading list using the learn@uw dropbox. The detailed course schedule (on the web) indicates which ones and when they are due. There will be several unannounced in-class quizzes throughout the semester.

Project

The default course project is to do some original research in a group of four students. Some alternatives for original research are: you could examine a modest extension to a paper studied in class or simply revalidate the data in some paper by writing your own simulator. Projects will include a written report. Project work will be presented orally to the rest of the class at the end of the semester.

Examinations

There will be two in-class midterm exams held on 3/5 and 4/16. There is no final exam.

Grading

Quizzes and Paper Reviews	20%	Project	30%
Midterm 1	25%	Midterm 2	25%

Communications Channels

I strongly encourage you to meet with me during my office hours, or call me or send e-mail. Introducing yourself to me, expressing concerns, offering suggestions, and seeking advice are among the welcome topics. Make sure you monitor the web site for this course which contains course information, lecture notes, pointers to project resources, and the latest announcements.

Office Hours

Prof. Lipasti: EH3621, TBD, or by appointment

Tentative Course Outline

Week	Dates	Topics	Readings
1	1/18, 1/20	Introduction 752 review	Skim [1], Read [3], [4]
2	1/23, 1/25 1/27	In-class tutorial for gem5 Cores, multithreading, multicore	Review [2] Skim [5], Read [6], [7]
3	1/30, 2/1, 2/3	MP Software MP Memory Systems	Skim [8], Read [9], Review [10] Skim [11]
4	2/6, 2/8 2/10	Class cancelled MP Memory Systems	 Read [12] Ch. 2, Review [13]
5	2/13, 2/15, 2/17	Coherence & consistency	Read [12] Ch. 6-8
6	2/20, 2/22, 2/24	Coherence & consistency cont'd	Read [14], Review [15]
7	2/27, 3/1 3/3	Catch up / midterm review **Midterm 1 in class on 3/3**	
8	3/6 3/8, 3/10	Class cancelled Transactional Memory	 Read [16], Review [17]
9	3/13, 3/15, 3/17	Interconnection Networks **Project proposal due 3/17**	Read [18], Review [19]
N/A	3/20, 3/22, 3/24	Spring break	
10	3/27, 3/29, 3/31	SIMD MPP	Read [20] Review [21]
11	4/3, 4/5 4/7	Clusters, GPGPUs Class cancelled	Read [22], [23]
12	4/10, 4/12, 4/14	Catch up and review	
13	4/17 4/19, 4/21	**Midterm 2 in class 4/17** No lecture; project work **Project status report due 4/21**	
14	4/24, 4/26, 4/28	No lecture; project work	
15	5/1, 5/3	Project talks, course Evaluation	--
16	5/8	**No final exam** **Project reports due 5/8**	

Readings and References

Introduction

[1] James Smith, “Chapter 1: Introduction,” textbook draft, http://ece757.ece.wisc.edu/uw-only/Chapter1.pdf

[2] N. Binkert et al.,“The gem5 simulator,” SIGARCH Comput. Archit. News 39, 2 (August 2011), 1-7. DOI=http://dx.doi.org/10.1145/2024716.2024718

[3] H. Sutter and J. Larus, Software and the Concurrency Revolution, ACM Queue, September 2005

[4] 21st Century Computer Architecture (A community white paper), May 25, 2012. http://cra.org/ccc/wp-content/uploads/sites/2/2015/05/21stcenturyarchitecturewhitepaper.pdf

Cores, Multicores, and Multithreading

[5] James Smith, “Chapter 3: Processor Cores,” textbook draft, http://ece757.ece.wisc.edu/uw-only/Chapter3.pdf

[6] K. Olukotun, et al., "The Case for a Single-Chip Multiprocessor," ASPLOS-7, October 1996.

[7] Kumar, R., et al., "Heterogeneous Chip Multiprocessors", IEEE Computer, pp. 32-38, Nov. 2005.

Multiprocessor Software and Instruction Set Architecture

[8] James Smith, “Chapter 2: Software and Instruction Set Architecture,” textbook draft, http://ece757.ece.wisc.edu/uw-only/Chapter2.pdf

[9] Michael L. Scott , “Shared-Memory Synchronization,” Sections 1.0-1.4, 4.0-4.3.3, 5.0-5.2.5, Synthesis Lectures on Computer Architecture, http://www.morganclaypool.com/doi/abs/10.2200/S00499ED1V01Y201304CAC023

[10] W. Daniel Hillis and Guy L. Steele, Data Parallel Algorithms, Communications of the ACM, December 1986, pp. 1170-1183.

Memory Systems and Cache Coherence

[11] James Smith, “Chapter 4: Multiprocessor Memory Systems,” textbook draft, http://ece757.ece.wisc.edu/uw-only/Chapter4.pdf

[12] D. Sorin, M.D. Hill, D.A. Wood, “A Primer on Memory Consistency and Cache Coherence,” Chapters 2, 6, 7, 8, Synthesis Lectures on Computer Architecture, http://www.morganclaypool.com/doi/abs/10.2200/S00346ED1V01Y201104CAC016

[13] P. Conway, N. Kalyanasundharam, G. Donley, K. Lepak, and B. Hughes. Cache hierarchy and memory subsystem of the AMD opteron processor. IEEE Micro, vol. 30, no. 2, pp. 16-29, Apr. 2010

Memory Consistency

[14] D. Sorin, M.D. Hill, D.A. Wood, “A Primer on Memory Consistency and Cache Coherence,” Chapters 3-5, Synthesis Lectures on Computer Architecture, http://www.morganclaypool.com/doi/abs/10.2200/S00346ED1V01Y201104CAC016

[15] M. D. Hill, Multiprocessors Should Support Simple Memory Consistency Models. IEEE Computer, Aug. 1998

Transactional Memory

[16] T. Harris, J. Larus, and R. Rajwar, “Transactional Memory, 2nd edition,” Synthesis Lectures on Computer Architecture, http://www.morganclaypool.com/doi/abs/10.2200/S00272ED1V01Y201006CAC011

[17] Harold W. Cain, Maged M. Michael, Brad Frey, Cathy May, Derek Williams, and Hung Le. Robust architectural support for transactional memory in the power architecture. In Proceedings of the 40th Annual International Symposium on Computer Architecture (ISCA '13), June 2013.

Interconnects

[18] N. Enright Jerger, L.-S. Pei, “On-Chip Networks,” Synthesis Lectures on Computer Architecture, http://www.morganclaypool.com/doi/abs/10.2200/S00209ED1V01Y200907CAC008

[19] D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C.-C. Miao, J. F. B. III, and A. Agarwal. On-Chip Interconnection Architecture of the Tile Processor. IEEE Micro, vol. 27, no. 5, pp.

15-31, 2007

SIMD and MPP

[20] C. Hughes, “Single-Instruction Multiple-Data Execution,” Synthesis Lectures on Computer Architecture, http://www.morganclaypool.com/doi/abs/10.2200/S00647ED1V01Y201505CAC032

[21] Steven L. Scott, Synchronization and Communication in the T3E Multiprocessor, Proceedings of International Conference on Architectural Support for Programming Languages and Operating Systems, pages 26-36, October 1996.

Clusters and GPGPUs

[22] H. Kim, R. Vuduc, S. Bahsorkhi, J. Choi, W.-M. Hwu, “Performance Analysis and Tuning for General Purpose Graphics Processing Units (GPGPU),” Synthesis Lectures on Computer Architecture, http://www.morganclaypool.com/doi/abs/10.2200/S00451ED1V01Y201209CAC020

[23] L. Barroso, J. Clidaras, U. Hölzle, “The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Second edition,” Synthesis Lectures on Computer Architecture, http://www.morganclaypool.com/doi/abs/10.2200/S00516ED2V01Y201306CAC024