CANS Master, Spring 2010
Visiting Professors' Seminars

SEMINAR 1

Issues in Computer Architecture and Microarchitecture for Future Computing Machines (3 ECTS)

Prof. Yale N. Patt (University of Texas at Austin, USA)

DATES: to be provided

Description:

This course identifies topics that are both fundamental to computer architecture and relevant to the design of microprocessors of the future. The emphasis is always on insights that will be useful to the graduate student, whether he/she goes on for a PhD or joins a microprocessor design team.

In the course, we will deal with principles, tradeoffs, and implementation details. We will cover mechanisms like trace caches, helper threads, branch prediction, bandwidth, etc. We will discuss the implications of multi-core on future microprocessor designs. We will examine current state of the art microprocessors.

In addition to the class lectures/discussions, students will carry out an out-of-class project and make a presentation on what they learn to the class. There will be a final examination.

SEMINAR 3

Profiling and Optimization (3 ECTS)
Prof David Kaeli (Northeastern University, Boston (USA))

DATES: to be provided

Description:

Overview:
This PhD-level course will discuss how to capture program static and dynamic characteristics of program execution using both hardware and software techniques. The course will also discuss how to utilize these characteristics in the areas of architectural modeling, program optimization, binary translation, power modeling, software fault recovery, security, software testing, and runtime monitoring. This year's course will also focus on issues of security, and will run experiments on X86 and embedded processors.

The subjects that will be covered in the course include:

a.) A historical perspective of profiling and instrumentation

b.) Instrumentation

Hardware techniques

- Performance counters
- ICE and pods
- Trace arrays
- Bus analyzers
- LLATTs

Software techniques

- source modifications
- object modifications
- executable modifications
- emulators

c.) Applications

- Performance
  - Hardware/architecture
  - Software
- Recovery
  - Trace buffers
  - Dumps
- Security
  - Anti-virus scanning
  - Trojan horses
  - Spam filtering
- Testing
  - Clear-box testing
- Power modeling
  - Instruction-based modeling

- Monitoring
- JITs/Binary Translators

d.) Optimization
- Static
- Dynamic

e.) Selected topics
- Hot-cold code layout
- Heap layout
- Cloning and Specialization
- Aspect-oriented programming/profiling
- Power/energy efficiency
- Anti-virus studies
- Def-use testing
- Cachability

f.) software/hardware interface - how does software interface to hardware
g.) program debugging and testing - tools and strategies

Course Details

Homework:
There will be one lab assignment that will ask you to run an experiment on a profiling system. You will have the choice of using either the Pin toolset or the ATOM toolset. The goal is to use a tool to obtain interesting runtime information about a program.

Paper Review:
You need to submit a review of a selected paper during the course. The review will consist of a 1-page summary of the material presented in the selected paper, followed by 2-3 pages of critical opinion of the material presented. Your review should rigorously address the following questions:

1.What is the problem being studied?
2.Is this an important problem?
3.What are the main points or results presented?
4.What quantitative methods were used in this paper? Were these the best methods to be used?
5.What did you learn from this paper?
6.Compare this paper to similar papers on the subject? (you should find and read one or two related papers, and use them to compare to this paper)

In the last class, you will be asked to stand up and discuss your paper review with the class.

Grading:
Homework: 40%
Paper review: 40%
Class discussion: 20%

Topics to be covered:
Class 1a: Introduction

Course overview
Definitions of common terminology
Historical perspective
Issues with profiling: overhead, completeness and correctness

Class 1b: Tools
Papers – ATOM, Simics, Pin, Liberty, DCPI
Review of tools and techniques
Hardware Tools, Performance Counters

Class 2: Workload Characteristics
Papers – OO-Calder, Values, Uhlig, AS/400
Instruction Profiling, Memory Profiling, VisualDSP++

Class 3: Sampling Technology, AOP
Papers – Profile-Me, Trace Stitching,
Sampling theory
Sampled workloads during simulation
Incomplete profiles

Class 4: Profile-guided Compilation
Papers – PLDI97, WMPI01, PLTO, IOICS03
Program graphs (PCG, TRG)
Code layout, Data layout
Instruction layout

Class 5: Static Optimization
Papers – Ball/Larus, Static Branch Prediction
Code Layout

Class 6: Static/Dynamic Optimization
Papers – Shade, Spike, FX!32
Overhead
Optimization techniques

Class 7: Dynamic Optimization
Papers – Dynamo, JITs, replay

Class 8: Translation, Software Testing
Papers – Dixie, UQBT, Def-Use, Dataflow

Class 9: Power Profiling, Security
Paper Review Discussion

SEMINAR 8

HIPEAC Summer School Seminars (3 ECTS)
Local organizer contact: Prof. Nacho Navarro (UPC)

ACACES 2010
Sixth International Summer School on Advanced Computer Architecture and Compilation for Embedded Systems
July 11th to July 17th 2010 Terrassa (Barcelona) Spain
Organized by the HiPEAC Network of Excellence
http://www.hipeac.net/summerschool

SEMINAR 14

Multicore Systems Programming (2 ECTS)
Prof Dimitrios Nikolopoulos
(ICS-FORTH – Foundation for Research and Technology - Hellas (Greece))

DATES: to be provided

Description:

The course explores emerging topics in system software for multi-core processors. The selected topics cover:

- runtime systems and architectural support for achieving scalable parallel execution on homogeneous and heterogeneous chip multi-processors (explicit versus implicit communication; task creation, scheduling and synchronization; locality exploitation and memory management; power and thermal management)

- operating systems and virtual machine support for scalability and performance isolation (memory management; protection and sharing mechanisms; scheduling; device management; power and thermal management)

- programming models and APIs for commercial workloads (high-performance embedded systems; data centers; utility computing; mobile computing)

The course will use lectures and selected publications in the aforementioned topics. Students will be evaluated on the basis of class participation, discussion of topics covered in the cost and a technical write-up summarizing a paper on a selected topic covered in the course..

SEMINAR 19

Game Theory in Networks (2 ECTS)
Prof Jean C. Walrand (University of California Berkeley, USA)

DATES: to be provided

Description:

This course is a self-contained introduction to game theory and its applications to networks. The objective of the course is to enable students to model strategic problems in networks and to understand the literature on these topics.

Syllabus:

1.Game Theory: Noncooperative and cooperative games
2.Pricing of Services: Pricing models, service differentiation, user discrimination, net neutrality
3.Revenue Sharing: multiprovider services
4.Auctions: Theory, bandwidth on demand
5.Service Architecture: open network or stove pipes
6.Economics of Security: Regulations, insurance, liability
7.MAC games: cognitive radios; cheating wifi
8.Routing Games: Braess' paradox, hot-potato routing, pricing links
9.Transport Games: price of anarchy
10. Cooperative providers and Shapley value

References:

The course is based both on reference texts and on recent papers. Lecture notes are provided for the student.

Here are a few selected references.

- D. Acemoglu and A. Ozdaglar, "Competition and efficiency in congested markets," Mathematics of Operations Research, vol. 32, no. 1, pp. 1 – 31, February 2007.
- D. Acemoglu and A. Ozdaglar, "Competition in parlallel-serial networks," IEEE Journal on Selected Areas in Communications, vol. 25, no. 6, pp. 1180–1192, 2007.
- T. Alpcan, T. Basar, R. Srikant, and E. Altman, "CDMA uplink power control as a noncooperative game," Wireless Networks, vol. 8, pp. 659–670, 2002.
- A. Dixit and J. Stiglitz, "Monopolistic competition and optimum product diversity," The American Economic Review, vol. 67, no. 3, pp. 297–308, 1977.
- J. Feigenbaum, R. Sami, and S. Shenker, "Mechanism design for policy routing," Distributed Computing, vol. 18, pp.293–305, 2006.

- A. Hayrapetyan, E. Tardos, and T. Wexler, “A network pricing game for selfish traffic,” Distributed Computing, vol. 19, no. 4, pp. 255–266, March 2007.
- B. Hermalin and M. Katz, “The economics of product-line restrictions with an application to the network neutrality controversy,” Information Economics and Policy, vol. 19, pp. 215–48, 2007.
- C. Hogendorn, “Broadband internet: Net neutrality versus open access,” International Economics and Economic Policy, vol. 4, pp. 185–208, 2007.
- Y. Jin, S. Sen, R. Guerin, K. Hosanagar, and Z.-L. Zhang, “Dynamics of competition between incumbent and emerging network technologies,” in Proceedings of the ACM NetEcon08: The Workshop on the Economics of Networks, Systems, and Computation, Seattle, WA, August 2008, pp. 49–54.
- R. Johari and J. Tsitsiklis, “A scalable network resource allocation mechanism with bounded efficiency loss,” IEEE Journal on Selected Areas in Communications, vol. 24, no. 5, pp. 992 – 99, May 2006.
- J. W. Lee, M. Chiang, and R. A. Calderbank, “Price-based distributed algorithm for optimal rate-reliability tradeoff in network utility maximization,” IEEE Journal on Selected Areas in Communications, vol. 24, no. 5, pp. 962–976, May 2006.
- J. Musacchio, G. Schwartz, and J. Walrand, “A two-sided market analysis of provider investment incentives with an application to the net neutrality issue,” Review of Network Economics, vol. 8, no. 1, pp. 22–39, March 2009.
- J. Musacchio and S. Wu, “The price of anarchy in a network pricing game,” in Proceedings of the 45th Annual Allerton Conference on Communication, Control, and Computing, Monticello, October 2007, pp. 882–891.
- J. Musacchio, J. Walrand, and S. Wu, “A game theoretic model for network upgrade decisions,” in Proceedings of the 44th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, September 2006, pp. 191–200.
- J. Musacchio, G. Schwartz, and J. Walrand, “Network economics: Neutrality, competition and service differentiation,” to appear in Next Generation Internet Architectures and Protocols, Cambridge University Press, 2009.
- A. Ozdaglar, “Price competition with elastic traffic,” Networks, vol. 52, no. 3, pp. 141 – 155, October 2008.
- A. Odlyzko, “Paris metro pricing for the internet,” in Proceedings of the 1st ACM conference on Electronic Commerce, Denver, CO, November 1999, pp. 140–147.
- A. Odlyzko, “Network neutrality, search neutrality, and the never-ending conflict between efficiency and fairness in markets,” Review of Network Economics, vol. 8, pp. 40–60, 2009.
- J.-C. Rochet and J. Tirole, “Two-sided markets: A progress report,” RAND Journal of Economics, vol. 37, no. 3, pp. 655–667, 2006.
- T. Roughgarden, “How bad is selfish routing?” Journal of ACM, vol. 49, no. 2, pp. 236–259, March 2002.
- T. Roughgarden, Y. Xi and E. M. Yeh, “Pricing, competition, and routing for selfish and strategic nodes in multi-hop networks,” in Proceedings of IEEE INFOCOM, Phoenix, AZ, 2008.
- S. Shakkottai and R. Srikant, “Economics of network pricing with multiple isps,” IEEE/ACM Transactions on Networking, vol. 14, no. 6, pp. 1233–1245, December 2006.
- H. Yaiche, R. Mazumdar, and C. Rosenberg, “A game-theoretic framework for bandwidth allocation and pricing in broadband networks,” IEEE/ACM Transactions on Networking, vol. 8, no. 5, pp. 667–678, October 2000.

SEMINAR 20

Software Support for Multicore Architectures (2 ECTS)
Prof John Cavazos (University of Delaware, USA)

DATES: to be provided

Description:

Overview:

Although researchers in industry and academia are exploring many different multicore hardware design choices, most agree that software for execution on multicore processors is the major unsolved problem. Unlike earlier generations of hardware evolution, this shift will have a major impact on how software is designed and developed. Developers will have to learn how to properly design their applications to utilize multicore parallelism. Opportunities to address the problem span multiple levels of the software stack.

This course will focus on the entire spectrum of the software stack as it applies to multicore architectures, including libraries, tools, programming languages, compilers, runtime systems, operating systems, and hypervisors. The course will include discussions of current multicore architectures and parallel programming models.