Spring 2010

ECE 569 High-Performance Computing: Technology, Architecture, and Algorithms

TTr 12:30AM – 1:45PM HARV101

Instructor:	Ali Akoglu	**Phone**:	(520) 626-5149
Office:	ECE 356B	**Email**:	akoglu@ece.arizona.edu
Office Hours:	Tuesdays 11:00 AM - 12:00 PM Thursdays 3:00 PM – 4:00 PM or by appointment		
Web page:	http://www.engr.arizona.edu/classes/ece569a/Spring10/		

Scope: ECE 569 stresses the need for and the design of high-performance computing (HPC) systems. HPC is more than just for achieving high performance - it is a compelling vision for how computation can seamlessly scale from a single processor to virtually limitless computing power. The single enabling force for HPC is the use of parallelism. The market demands general-purpose processors that deliver high single threaded performance as well as multi-core throughput for a wide variety of workloads on client, server, and high-performance computing (HPC) systems. This pressure has given us almost three decades of progress toward higher complexity and higher clock rates. This progress hasn't always been steady. Intel cancelled its "Tejas" processor, which was rumored to have a 40-stage pipeline, and later killed off the entire Pentium 4 "NetBurst" product family because of its relative inefficiency. The Pentium 4 ultimately reached a clock rate of 3.8 GHz in the 2004 "Prescott" model, a speed that Intel has been unable to match since. In the more recent Core 2 (Conroe/Penryn) and Core i7 (Nehalem) processors, Intel uses increased complexity to deliver substantial performance improvements over the Pentium 4 line, but the pace of these improvements is slowing. Each new generation of process technology requires ever more heroic measures to improve transistor characteristics; each new core microarchitecture must work disproportionately harder to find and exploit instruction-level parallelism (ILP). As these challenges became more apparent in the 1990s, CPU architects began referring to the "power wall," the "memory wall," and the "ILP wall" as obstacles to the kind of rapid progress seen up until that time. New commodity parallel computing devices, bring the originally elite high performance computing into the reach of general public. To program and accelerate applications on the new high performance computing devices, we must understand both the computational architecture and the principles of program optimization. Throughout the course we will study state of the art processor architectures such as the IBM CELL BE, Nvidia Tesla GPU, Intel Larrabee Microarchitecture and Intel Nehalem microarchitecture. We will then discuss the trends in cluster computing and cluster based systems. We will study parallel algorithm design and programming issues for such systems. We will evaluate power, memory and ILP challenges from the perspectives of Programming Model, Computational Model, Processor Architecture Model, Threading Model, Memory Model and Power Model. Therefore, this course will provide students with an in-depth analysis of these current issues in HPC systems including: (1) Parallel Computing (2) New Processor Architectures, (3) Power-Aware Computing and Communication, (4) Advanced Topics on Petascale Computing and Optical Systems. In addition, we will also study parallel models of computation such as dataflow, and demand-driven computation. While there are no specific prerequisites for the course, the students are expected to be well versed with basics of uniprocessor computer architecture and digital logic.

Textbook: No required text books. Following books are recommended and will help your projects.

[1] "Highly Parallel Computing", by George S. Almasi and Alan Gottlieb

[2] "Advanced Computer Architecture: Parallelism, Scalability, Programmability", by Kai Hwang, McGraw Hill 1993

[2] "Parallel Computer Architecture: A hardware/Software Approach", by David Culler Jaswinder Pal Singh, Morgan Kaufmann, 1999.

[3] "Scalable Parallel Computing", by Kai Hwang, McGraw Hill 1998.

[4] "Principles and Practices on Interconnection Networks", by William James Dally and Brian Towles, Morgan Kauffman 2004.

[5] GPU Gems 3 --- by Hubert Nguyen (Chapter 29 to Chapter 41)

[6] Introduction to Parallel Computing, Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, 2nd edition, Addison-Welsey, © 2003.

[7] Petascale Computing: Algorithms and Applications, David A. Bader (Ed.), Chapman & Hall/CRC Computational Science Series, © 2007.

* Handout materials will be distributed in class from recent technical meetings and journals related to the field.

Course Outline

1. Parallel Processing Concepts (Quick Overview)
    a) Levels of parallelism (instruction, transaction, task, thread, memory, function)
    b) Models (SIMD, MIMD, SIMT, SPMD, Dataflow Models, Demand-driven Computation etc)
    c) Architectures: N-wide superscalar architectures, multi-core, multi-threaded

2. Parallel Programming with CUDA
    a) Processor Architecture, Interconnect, Communication, Memory Organization, and Programming Models in high performance computing architectures: (Examples: IBM CELL BE, Nvidia Tesla GPU, Intel Larrabee Microarchitecture and Intel Nehalem microarchitecture)
    b) Memory hierarchy and transaction specific memory design
    c) Thread Organization

3. Fundamental Design Issues in Parallel Computing
    a) Synchronization
    b) Scheduling
    c) Job Allocation
    d) Job Partitioning
    e) Dependency Analysis
    f) Mapping Parallel Algorithms onto Parallel Architectures
    g) Performance Analysis of Parallel Algorithms

4. Fundamental Limitations Facing Parallel Computing
    a) Bandwidth Limitations
    b) Latency Limitations
    c) Latency Hiding/Tolerating Techniques and their limitations

5. Power-Aware Computing and Communication
    a) Power-aware Processing Techniques
    b) Power-aware Memory Design
    c) Power-aware Interconnect Design
    d) Software Power Management

6. Advanced Topics
    (a) Petascale Computing
    (b) Optics in Parallel Computing
    (c) Quantum Computers
    (d) Recent developments in Nanotechnology and its impact on HPC

Assignments

Students will gain experience with leading-edge performance analysis tools, cycle-accurate hardware simulators, and dynamic program instrumentation systems to examine the operation of next-generation applications on modern hardware. Students will have programming assignments to evaluate and compare the architectural features of the state of the art high performance commodity hardware platforms.

Project, Term Paper, Presentation

Semester project will involve 2 phases:

- During the first half of the course, students will:
    - Propose a project on a selected topic taught in class,
    - Document their survey by reporting existing solutions,
    - Tackle a problem and propose their solution,
    - Present their initial findings and solution strategy
- During the second half of the course, students will:
    - Implement their proposed approach,
    - Put together a paper quality document with experimental results,
    - Present project findings

General policies

- Course will have 2-4 assignments, 1 mid-term examination, a semester project
- No late assignments will be accepted, except under extreme non-academic circumstances discussed with the instructor at least one week before the assignment is due.
- **Make-ups** for assignments and exam *may* be arranged if a student's absence is caused by documented illness or personal emergency. A written explanation (including supporting documentation) must be submitted to your instructor; if the explanation is acceptable, an alternative to the graded activity will be arranged. When possible, make-up arrangements must be completed prior to the scheduled activity.
  Any extenuating circumstances that have an impact on your participation in the course should be discussed with your instructor as soon as those circumstances are known.
- Inquiries about graded material have to be turned in within 3 days of receiving a grade.
- Approximate weight of each assignment will be specified when the assignment is handed out. Assignments will be due in class on the due date.
- The instructor reserves the right to modify course policies, course calendar, course content, assignment values and due dates, as circumstances require.
- Students are strongly encouraged to attend the class. Lecture notes are intended to serve as a supplement and not as a substitute for attending class.
- You are encouraged to discuss the assignment specifications with your instructor and your fellow students. However, anything you submit for grading must be unique and should NOT be a duplicate of another source. The Department of Electrical and Computer Engineering expects all students to adhere to UofA's policies and procedures on Code of Academic Integrity. http://web.arizona.edu/~studpubs/policies/cacaint.htm

Evaluation

- Midterm: 15%
- Quiz: 20%
- Assignments: 15%
- Project: (40% total)
  - Presentation: 10%
  - Survey paper: 10%
  - Final paper: 20%
- Participation: 10%

Grading Policy

- Overall points >= 85 %: A
- 70 % <= Overall points < 85 %: B
- 50 % <= Overall points < 70 %: C
- Overall points < 50 %: F