Lecture 1

Introduction

Course Overview
Welcome to CSE 260!

• Your instructor is Scott Baden
  baden@ucsd.edu

• Office: room 3244 in EBU3B

• Office hours
  – Week 1: Today (after class), Tuesday (after class)
  – Remainder of the quarter: TBD

• The class home page is
  http://www.cse.ucsd.edu/classes/fa09/cse260

• Web board - TBD

• Enrollment
Text and readings


  Be sure to get the 2nd edition.

• Assigned class readings will include handouts and on-line material

• Be prepared to discuss the readings in class

• Lecture slides

  [http://www.cse.ucsd.edu/classes/fa09/cse260/Lectures](http://www.cse.ucsd.edu/classes/fa09/cse260/Lectures)
Course Requirements

• Programming assignments: 45%
  – Teams of 2 or 3

• Class participation: 10%

• Project: 45%
  – Project teams of 2 or 3
Policies

• Academic Integrity
  – Do you own work
  – Plagiarism and cheating will not be tolerated

• By taking this course, you implicitly agree to abide by the following the course polices:
  http://www.cse.ucsd.edu/classes/fa09/cse260/Policies.html
Scheduling

• CSE 260 Symposium, project presentations
  – 9th and 10th weeks

• Some lectures will be rescheduled
  – Thursday 10/1 → Weds10/7 @ 11AM
  – Tuesday 11/17 → Fri 11/20  Time TBD
  – Thursday 11/19
  – Tuesday 11/24
Programming Assignments and projects

• Three tracks
  – Multicore (SDSC) - openMP, pthreads
  – Nvidia Tesla (NCSA) - CUDA
  – MPI on a cluster (SDSC/NCSA) - MPI

• Let me know which track you are interested in, see assignment #1

• Find a partner

• Propose a project on 10/15, teams of 3 permitted
  http://cseweb.ucsd.edu/classes/fa09/cse260/Projects/
Course overview and background

• How to solve computationally intensive problems on parallel computers
  – Software techniques
  – Performance tradeoffs

• Background
  – Graduate standing
  – Recommended: computer architecture (CSE 240A)
  – Students outside CSE are welcome
  – See me if you are unsure about your background

• Prior experience
  – Parallel computation?
  – Numerical analysis?
Background Markers

- C/C++    Java    Fortran?
- Navier Stokes Equations
- Sparse factorization
- TLB misses
- Multithreading
- MPI
- GPUs
- RPC
- Abstract base class

\[ \nabla \cdot u = 0 \]

\[ \frac{D\rho}{Dt} + \rho(\nabla \cdot v) = 0 \]

\[ f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \ldots \]
Syllabus

• Fundamentals
  Motivation, system organization, hardware execution models, limits to performance, program execution models, theoretical models

• Software and programming
  – Programming models and techniques: message passing, multithreading
  – Architectural considerations: GPUs and multicore
  – Higher level run time models, language support

• Parallel algorithm design and implementation
  – Case studies to develop a repertoire of problem solving techniques: discretization, sorting, linear algebra, irregular problems
  – Data structures and their efficient implementation: load balancing and performance
  – Performance tradeoffs, evaluation, and tuning
What is parallel computation?

• Simultaneous computation over separate resources to increase capacity or speed
• Resources may be virtual or physical
• Multiple processors co-operate to solve a related set of tasks comprising a “single” problem
• Reliable, tightly coupled interactions
• Two view points
  – Speedup: 100 processors run $\times 100$ faster than one
  – Opportunity: Tackle a larger problem, more accurately
Why is parallelism inevitable?

- Physical limitations on processor clock speed, memory bandwidth, and memory density

Average CPU clock speeds (via Bill Gropp)  http://www.pcpitstop.com/research/cpu.asp
The impact of technology
Today’s laptop would have been yesterday’s supercomputer

- Cray-1 Supercomputer
- 80 MHz processor
- 8 Megabytes memory
- Water cooled
- 1.8m H x 2.2m W
- 4 tons
- Over $10M in 1976

- MacBook
- 2.53GHz Intel Core 2 Duo
- 4 Gigabytes memory, 3 Megabytes shared cache
- NVIDIA GeForce 9400M + 256MB shared DDR3 SDRAM
- Wireless Networking
- Air cooled
- ~ 2.4 x 36 x 24 cm. 2.5 kg
- $1499 in Sept. 2009
Technological disruption

• New capabilities → increased knowledge through improvements in computer modelling

• Changes in the common wisdom for solving a problem including the implementation

Cray-1, 1976, 240 Megaflops

ASCI Red, 1997, 1Tflop

Beowulf cluster, late 1990s

Intel Teraflop on a chip 2007

Nvidia Tesla, 4.14 Tflops*

Sony Playstation, 150 Gflops
Technological disruption: the MPP

- Until the early to mid 90’s vector computers were the reigning architecture
- An Intel Paragon parallel computer had 1/6 the hardware cost of the Cray C90 supercomputer located at the San Diego Supercomputer Center
- Computational Materials Science: Solution to the Local Density Approximation [Baden Bylaska Kohn Ong Weare 1995]
- Clever algorithms played a role, avoiding costly “brute force” methods
Scaling trends in Top 500 supercomputers

www.top500.org
The age of the multi-core processor

- On chip parallel computer
- IBM Power4 (2001), many others follow (Intel, AMD)
- First dual core laptops (32005-6)
- GPUs (nVidia, ATI): supercomputer on a desktop
The impact

- A renaissance in parallel computation
- Parallelism is no longer restricted to machine rooms, it is relevant to everyone
- In a few years, everyone will have an historically unprecedented amount of parallelism at their disposal
  - Don’t need to know they are using one
  - Non HPC users