CS546

- Parallel and Distributing Processing
  – Using blackboard (http://blackboard.iit.edu)
  – Class: 11:25am-12:40pm Tu. & Th., SB #111

- Instructor: Zhiling Lan
  – Email: lan@iit.edu, Phone: (312)5675710
  – Office: SB #226D
  – Office hours: 10:20am-11:20am(Tu. & Th.) or by appointment

- TA: Xu Yang
  – Email: xyang56@hawk.iit.edu
  – Office: SB #012
  – Office hours: 10:00am-11:30am (Mon. & Wed.) or by appointment

Self Intro

- Research interests: Parallel and Distributed Systems
  - Current projects:
    1. SPEaR: Scheduling for Performance, Energy, and Resilience efficiency
    2. COTA: a COoperative Framework for Topology Awareness
    3. HPC Analytics for Extreme Scale Computing
    4. Intelligent Storage Resource Services for the RAINS Project
    5. CUDA Teaching and Research Center
    6. Research Experiences for Undergraduates
  - Research website: **http://www.cs.iit.edu/~lan**

- Interested in joining my research group?
  - Core courses: CS546 & CS550 & CS451
  - Group seminars

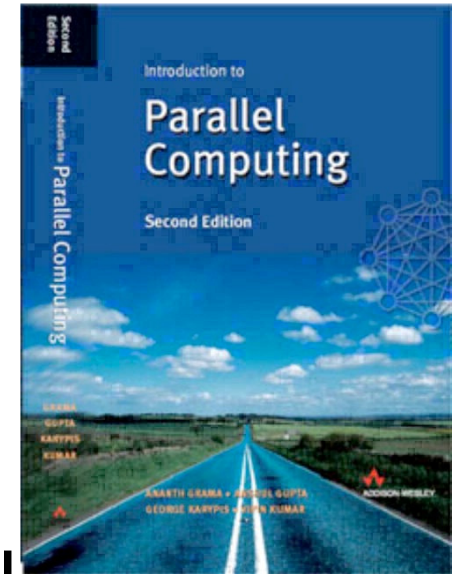
Course Objectives

- Fundamentals of parallel and distributed computing (an extensive and broad topic)
  - Parallel & distributed architectures
  - Parallel programming approaches, paradigms, tools, …
  - Principles of parallel algorithm design
- Parallel programming skills
  1. Shared memory programming: openMP and Pthreads
  2. Message passing interface: MPI
  3. GPU programming: CUDA
  4. Data parallelism: HPF
- Performance analysis and tuning
  - Performance metrics & evaluation techniques
  - Machine-independent optimization
  - Machine-dependent optimization

Prerequisite

- CS450 “Operating Systems”
- Familiar with
  - Programming in C/C++
  - Linux tools and development environment
    - Command
    - Editors, compilers (gcc), makefiles (GNU make)
  - Basic concepts of computer architecture
- Computing platforms:
  - **Jarvis** at IIT: our 19-node GPU-enabled cluster
  - **Stampede** at XSEDE (https://portal.xsede.org/tacc-stampede)

Course Materials



- Required:
  - lecture notes will be put online
- Recommended textbooks:
  1. **A. Grama, V. Kumar et al., "Introduction to Parallel Computing", Addison Wesley, 2003. ISBN 0-201-64865-2**
  2. I. Foster, "Design and Building Parallel Programs", Addison Wesley, 1995. ISBN 0-201-57594-9
     - http://www.mcs.anl.gov/dbpp
  3. W. Gropp et al, "Using MPI: Portable Parallel Programming with the Message Passing Interface", MIT Press, 1994. ISBN 0-262-57132-3
     - http://www.mcs.anl.gov/mpi/index.html
  4. David Kirk and Wen-Wei Hwu, "Programming Massively Parallel Processors", Morgan Kaufmann (2nd Edition), 2012.
     - Several books are available for loan

Course Slides

- Available on Blackboard (CS546)
  - Adapted slides from Manish Parashar (Rutgers)
  - Adapted slides from Ananth Grama
  - Adapted slides from Kathy Yelick (Berkeley)
  - Adapted slides from LLNL's tutorial
  - Adapted slides from John Mellor-Crummey (Rice)
  - Adapted slides from Jack Dongarra (Tennessee)

Tentative Schedule

- Week 1: Overview
- Week 2: Parallel platforms and programming models
- Week 3-4: Performance analysis and tuning
- Week 5-7: Shared memory architecture & programming
- Week 8: midterm
- Week 9-11: Distributed memory architecture & programming
- Week 12: GPU architecture & programming
- Week 13-14: Data parallel programming
- Week 15: final exam

Misc. Course Details

- You are expected to attend all of the lectures
- Grading
  - Two written and three programming assignments (2*8%+3*10%=46%)
  - Two exams (2*20%)
  - One final programming project (14%)
- Assignment late penalty:
  - There is a late penalty of 20% per day
- Use the course blackboard (**http://blackboard.iit.edu**)
  - Announcements
  - Lecture notes
  - Assignments
  - Discussion

Policies

- Collaboration policy
  - Encouraged for high level concepts and understanding the courses materials
  - but …..

- Cheating policy
  - Copying all or part of another student's homework
  - Allowing another student to copy all or part of your homework
  - Copying all or part of code found in a book, magazine, the Internet, or other resource

Policies

- IIT Code of Academic Honesty [link]
- All violations of academic integrity will be reported to academichonesty@iit.edu
- Sanctions for violations of academic integrity
  - **Expulsion from a course.** The student is assigned a punitive failing grade of 'E' for the course and can no longer participate in the course or receive evaluation of coursework from the instructor.
  - **Suspension.** Suspension is a status assigned for various periods of time in which a student's enrollment is interrupted. A suspended student may not attend day or evening classes, participate in student activities, or live in campus housing. A suspended student may apply for reinstatement at the end of the period of suspension. If reinstated, the student may be placed on disciplinary probation for a period of time designated by the Dean for Academic Discipline.
  - **Expulsion**. Expulsion is the complete severance of association with the University. Notation of the violation of the Code is made on the student's transcript

Work Opportunities

- Student introduction
- Research opportunities for graduate students:
  - Always look for self-motivated and hard-working grad students
  - Ph.D. students: CS597 and CS691
  - MS students: CS591 “Research and Thesis for MS Degree”
  - Take CS546 & CS550, check my research projects, send me your CV
- Research opportunities for undergrad students:
  - NSF REU (Research Experiences for Undergraduates) with Prof. Zhiling Lan
    - Various project topics, including development of scheduling simulator, analysis of system logs, ….
    - $10-$15 per hour
    - If interested, contact Prof. Lan (lan@iit.edu)

Any Questions?

Introduction

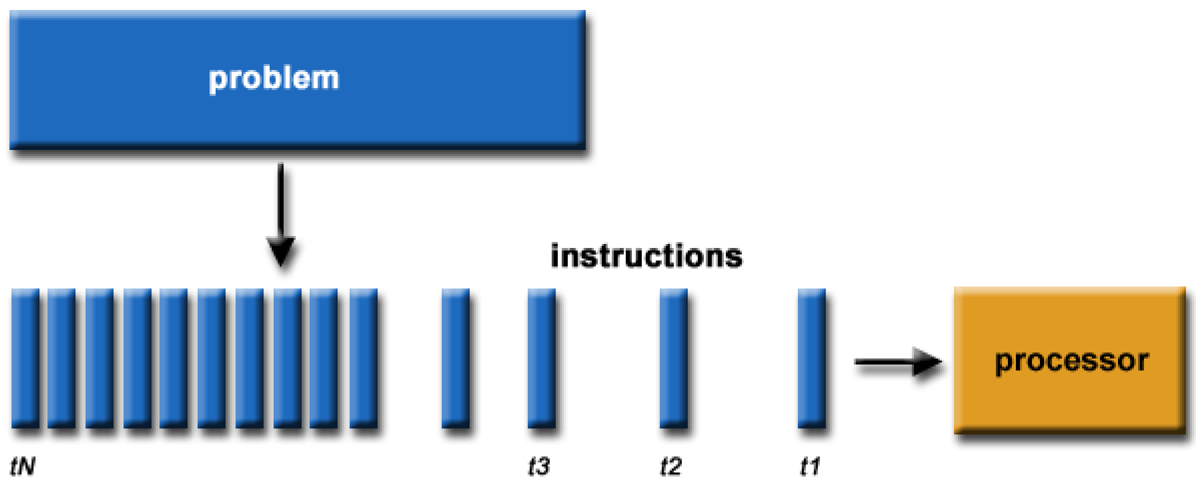
Outline

- What is parallel computing
- Why parallel computing
  - Application pull
  - Technology push
- Distributed and cloud computing

- Reading:
  - Kumar – ch 1; Foster – ch 1

Serial Computing

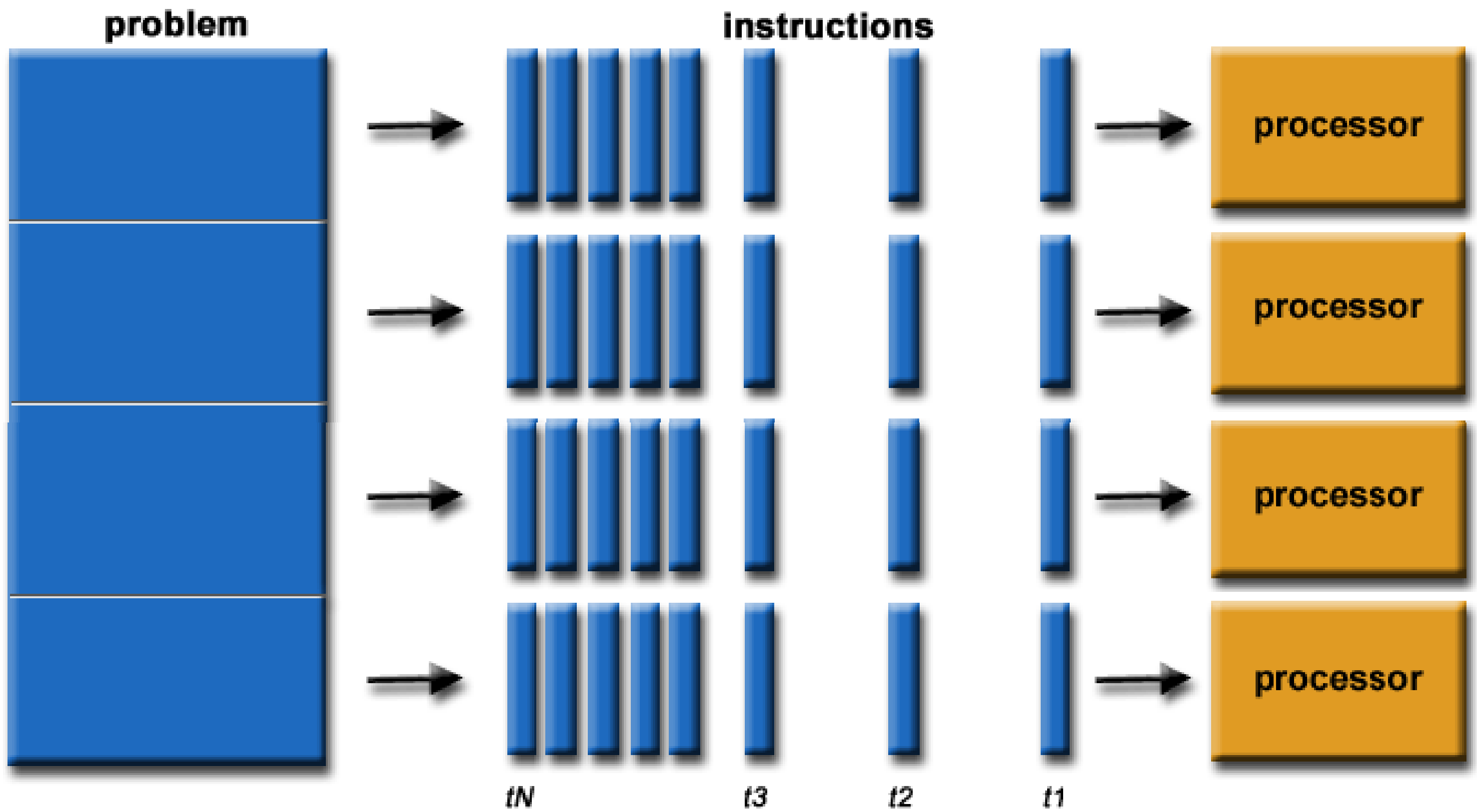
- Traditional, software has been written for serial computation
  - A problem is broken into a discrete series of inst.
  - Inst. are executed sequentially (one after another)
  - Executed on a single processor
  - Only one inst. executes at any moment in time



What is Parallel Computing

- Simultaneous use of multiple compute resources to solve a problem
  - A problem is broken into discrete parts that can be solve concurrently
  - Each part is further broken down to a series of inst.
  - Inst. from each part execution simultaneously on different processors
  - An overall control/coordination mechanism is employed
- Examples of parallel machines:
  - A cluster computer
  - A shared memory multiprocessor (SMP*)
  - A Chip Multi-Processor (CMP)
- Daily life example: stacking a set of library books

What is Parallel Computing



Who is Using Parallel Computing

- Science and engineering
  - Historically, it has been considered to be “the high end of computing”, and has been used to model difficult problems in many areas of science and engineering
- Industrial and commercial
  - Today, commercial applications provide an equal or greater driving force in the development of faster computers
  - These applications require the processing of large amount of data in sophisticated ways

Why Parallel Computing

- Application pull
  - Desire to solve bigger and more realistic applications

- Technology push
  - Fundamental limits are being approached

Application Drivers for Change

- Continued exponential increase in computational speed $\rightarrow$ simulation is becoming third pillar of science, complementing theory and experiment
- Continued exponential increase in experimental data $\rightarrow$ techniques and technology in data analysis, visualization, analytics, networking, and collaboration tools are becoming essential in all data rich scientific applications

Application Pull

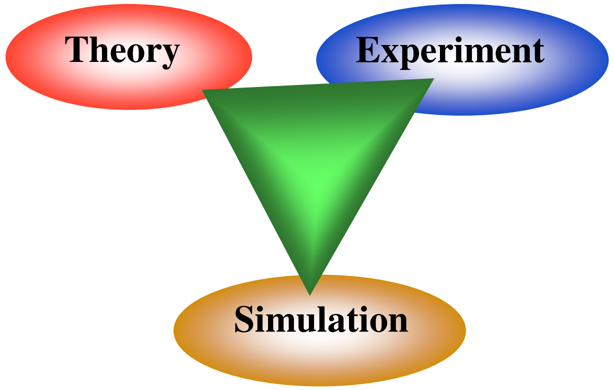
"Computational modeling and simulation are among the most significant developments in the practice of scientific inquiry in the 20th century. Within the last two decades, scientific computing has become an important contributor to all scientific disciplines.

It is particularly important for the solution of research problems that are insoluble by traditional scientific theoretical and experimental approaches, hazardous to study in the laboratory, or time consuming or expensive to solve by traditional means"

— *"Scientific Discovery through Advanced Computing" DOE Office of Science, 2000*

Simulation: 3^{rd} Pillar of Science

- Traditional scientific and engineering method:
  - (1) Do theory or paper design
  - (2) Perform experiments or build system



- Limitations:
  - –Too difficult
  - –Too expensive
  - –Too slow
  - –Too dangerous

- Computational science and engineering paradigm
  - What is it?
  - Based on known physical laws and efficient numerical methods
  - Analyze simulation results with computational tools and methods beyond what is possible manually

Challenging Computations

- Science
  - Global climate modeling
  - Biology: genomics; protein folding; drug design
  - Astrophysical modeling
  - Computational Chemistry
  - Computational Material Sciences and Nanosciences
- Engineering
  - Semiconductor design
  - Earthquake and structural modeling
  - Computation fluid dynamics (airplane design)
  - Combustion (engine design)
  - Crash simulation
- Business
  - Financial and economic modeling
  - Transaction processing, web services and search engines
- Defense
  - Nuclear weapons -- test by simulations
  - Cryptography

Parallel Computing in Web Search

- Functional parallelism: crawling, indexing, sorting
- Parallelism between queries: multiple users
- Finding information amidst junk
- Preprocessing of the web data to help find information

- General themes of sifting through large, unstructured data sets:
  - When to put white socks on sale
  - What advertisements should you receive
  - Finding medical problems in a community

Economic Impact of HPC

- Airlines:
  - System-wide logistics optimization systems on parallel systems.
  - Savings: approx. $100 million per airline per year.
- Automotive design:
  - Major automotive companies use large systems (500+ CPUs) for:
    - CAD-CAM, crash testing, structural integrity and aerodynamics.
    - One company has 500+ CPU parallel system.
  - Savings: approx. $1 billion per company per year.
- Semiconductor industry:
  - Semiconductor firms use large systems (500+ CPUs) for
    - device electronics simulation and logic validation
  - Savings: approx. $1 billion per company per year.
- Energy
  - Computational modeling improved performance of current nuclear power plants, equivalent to building two new power plants.

Summary of Application Pull

- Save time and/or money
- Solve larger or more complex problems
- Provide concurrency
- Take advantage of non-local resources
- Make better user of underlying parallel hardware