Introduction

Parallel Computing
CIS 410/510
Department of Computer and Information Science
Outline

- Course Overview
  - What is CIS 410/510?
  - What is expected of you?
  - What will you learn in CIS 410/510?

- Parallel Computing
  - What is it?
  - What motivates it?
  - Trends that shape the field
  - Large-scale problems and high-performance
  - Parallel architecture types
  - Scalable parallel computing and performance
Course Logistics

- Lecture time
  - Tuesday, Thursday: 12:00 - 13:20 pm, 208 Deady Hall

- Final schedule
  - Friday, June 13, 08:00-10:00, 208 Deady Hall

- Course webpage
  - http://www.cs.uoregon.edu/Classes/14S/cis410parallel

- Undergraduate course prerequisite
  - CIS 330 (C/C++ and Unix)

- Parallel programming laboratory
  - Neuroinformatics Center, 470 Streisinger Hall
  - Wednesday and Friday, 1 hour (times to be determined)
  - 2 sessions each day
How did the idea for CIS 410/510 originate?

- There has never been an undergraduate course in parallel computing in the CIS Department at UO
- Only 1 course taught at the graduate level (CIS 631)
- Goal is to bring parallel computing education in CIS undergraduate curriculum, start at senior level
  - CIS 410/510 (Spring 2014, “experimental” course)
  - CIS 431/531 (Spring 2015, “new” course)
- CIS 607 – Parallel Computing Course Development
  - Winter 2014 seminar to plan undergraduate course
  - Develop 410/510 materials, exercises, labs, …
- Intel gave a generous donation ($100K) to the effort
- NSF and IEEE are spearheading a curriculum initiative for undergraduate education in parallel processing

http://www.cs.gsu.edu/~tcpp/curriculum/
Who’s involved?

- Instructor
  - Allen D. Malony
    - scalable parallel computing
    - parallel performance analysis
    - taught CIS 631 for the last 10 years

- Faculty colleagues and course co-designers
  - Boyana Norris
    - Automated software analysis and transformation
    - Performance analysis and optimization
  - Hank Childs
    - Large-scale, parallel scientific visualization
    - Visualization of large data sets

- Intel scientists
  - Michael McCool, James Reinders, Bob MacKay

- Graduate students doing research in parallel computing
Intel Partners

- James Reinders
  - Director, Software Products
  - Multi-core Evangelist
- Michael McCool
  - Software architect
  - Former Chief scientist, RapidMind
  - Adjunct Assoc. Professor, University of Waterloo
- Arch Robison
  - Architect of Threading Building Blocks
  - Former lead developers of KAI C++
- David MacKay
  - Manager of software product consulting team
CIS 410/510 Graduate Assistants

- Daniel Ellsworth
  - 3rd year Ph.D. student
  - Research advisor (Prof. Malony)
  - Large-scale online system introspection

- David Poliakoff
  - 2nd year Ph.D. student
  - Research advisor (Prof. Malony)
  - Compiler-based performance analysis

- Brandon Hildreth
  - 1st year Ph.D. student
  - Research advisor (Prof. Malony)
  - Automated performance experimentation
Required Course Book


- Presents parallel programming from a point of view of patterns relevant to parallel computation
  - Map, Collectives, Data reorganization, Stencil and recurrence, Fork-Join, Pipeline

- Focuses on the use of shared memory parallel programming languages and environments
  - Intel Thread Building Blocks (TBB)
  - Intel Cilk Plus
Reference Textbooks

- **Introduction to Parallel Computing**, A. Grama, A. Gupta, G. Karypis, V. Kumar, Addison Wesley, 2nd Ed., 2003
  - Lecture slides from authors online
  - Excellent reference list at end
  - Used for CIS 631 before
  - Getting old for latest hardware

- **Designing and Building Parallel Programs**, Ian Foster, Addison Wesley, 1995.
  - Entire book is online!!!
  - Historical book, but very informative

- **Patterns for Parallel Programming**, T. Mattson, B. Sanders, B. Massingill, Addison Wesley, 2005.
  - Targets parallel programming
  - Pattern language approach to parallel program design and development
  - Excellent references
What do you mean by experimental course?

- Given that this is the first offering of parallel computing in the undergraduate curriculum, we want to evaluate how well it worked.
- We would like to receive feedback from students throughout the course:
  - Lecture content and understanding
  - Parallel programming learning experience
  - Book and other materials
- Your experiences will help to update the course for its debut offering in (hopefully) Spring 2015.
Course Plan

- Organize the course so that cover main areas of parallel computing in the lectures
  - Architecture (1 week)
  - Performance models and analysis (1 week)
  - Programming patterns (paradigms) (3 weeks)
  - Algorithms (2 weeks)
  - Tools (1 week)
  - Applications (1 week)
  - Special topics (1 week)

- Augment lecture with a programming lab
  - Students will take the lab with the course
    - graded assignments and term project will be posted
  - Targeted specifically to shared memory parallelism
Lectures

- Book and online materials are your main sources for broader and deeper background in parallel computing.
- Lectures should be more interactive:
  - Supplement other sources of information
  - Covers topics of more priority
  - Intended to give you some of my perspective
  - Will provide online access to lecture slides
- Lectures will complement programming component, but intended to cover other parallel computing aspects.
- Try to arrange a guest lecture or 2 during quarter.
Parallel Programming Lab

- Set up in the Neuroinformatics Center
  - 470 Streisinger Hall
  - Daniel Ellsworth and David Poliakoff leading the lab

- Shared memory parallel programming (everyone)
  - Cilk Plus (http://www.cilkplus.org/)
    - extension to the C and C++ languages to support data and task parallelism
  - Thread Building Blocks (TBB) (https://www.threadingbuildingblocks.org/)
    - C++ template library for task parallelism
  - OpenMP (http://openmp.org/wp/)
    - C/C++ and Fortran directive-based parallelism

- Distributed memory message passing (graduate)
  - MPI (http://en.wikipedia.org/wiki/Message_Passing_Interface)
    - library for message communication on scalable parallel systems
WOPR Programming Lab Schedule

- **Wednesday**
  - 10-11am
  - 1-2pm
  - 5-6pm

- **Friday**
  - 10-11am
  - 2-3pm
  - 4-5pm
WOPR (What Operational Parallel Resource)

- WOPR rose from whole cloth out of the depths of a programming lab crisis
  - 2 weeks ago it did not exist!
- Built Next Unit of Computing (NUC) cluster with Intel funds
  - 16x Intel NUC
    - Haswell i5 CPU (2 cores, hyperthreading)
    - Intel HD 4000 GPU (OpenCL programmable)
    - 1 GigE, 16 GB memory, 240 GB mSATA
    - 16x Logitech keyboard and mouse
  - 16x ViewSonic 22” monitor
  - Dell Edge GigE switch
  - Dell head node
Mist Cluster

- Distributed memory cluster
- 16 8-core nodes
  - 2x quad-core Pentium Xeon (2.33 GHz)
  - 16 Gbyte memory
  - 160 Gbyte disk
- Dual Gigabit ethernet adaptors
- Master node (same specs)
- Gigabit ethernet switch
- mist.nic.uoregon.edu
ACISS Cluster

- Applied Computational Instrument for Scientific Synthesis
  - NSF MRI R² award (2010)
- Basic nodes (1,536 total cores)
  - 128 ProLiant SL390 G7
  - Two Intel X5650 2.66 GHz 6-core CPUs per node
  - 72GB DDR3 RAM per basic node
- Fat nodes (512 total cores)
  - 16 ProLiant DL 580 G7
  - Four Intel X7560 2.266 GHz 8-core CPUs per node
  - 384GB DDR3 per fat node
- GPU nodes (624 total cores, 156 GPUs)
  - 52 ProLiant SL390 G7 nodes, 3 NVidia M2070 GPUs (156 total GPUs)
  - Two Intel X5650 2.66 GHz 6-core CPUs per node (624 total cores)
  - 72GB DDR3 per GPU node
- ACISS has 2672 total cores
- ACISS is located in the UO Computing Center
Course Assignments

- **Homework**
  - Exercises primarily to prepare for midterm

- **Parallel programming lab**
  - Exercises for parallel programming patterns
  - Develop programs using Cilk Plus, Thread Building Blocks, OpenMP
  - Graduate students will also do assignments with MPI

- **Team term project**
  - Programming, presentation, paper
  - Graduate students distributed across teams

- **Research summary paper (graduate students)**

- **Midterm exam later in the 7th week of the quarter**

- **No final exam**
  - Team project presentations during final period
Parallel Programming Term Project

- Major programming project for the course
  - Non-trivial parallel application
  - Include performance analysis
  - Use NUC cluster and possibly Mist and ACISS clusters

- Project teams
  - 5 person teams, 6 teams (depending on enrollment)
  - Will try our best to balance skills
  - Have 1 graduate student per team

- Project dates
  - Proposal: due April 25 (end of 4\textsuperscript{th} week)
  - Project talk: June19
  - Project report: June19

- Need to get system accounts!!!
Term Paper (for graduate students)

- Investigate parallel computing topic of interest
  - More in depth review
  - Individual choice
  - Summary of major points

- Requires minimum of ten references
  - Book and other references has a large bibliography
  - Google Scholar, Keywords: parallel computing
  - NEC CiteSeer Scientific Literature Digital Library

- Paper abstract and references due Thursday, April 24
- Final term paper due Friday, June 6
- Individual work
Grading

- **Undergraduate**
  - 5% homework
  - 10% pattern programming labs
  - 20% programming assignments
  - 30% midterm exam
  - 35% project

- **Graduate**
  - 15% programming assignments
  - 30% midterm exam
  - 35% project
  - 20% research paper
Overview

- Broad/Old field of computer science concerned with:
  - Architecture, HW/SW systems, languages, programming paradigms, algorithms, and theoretical models
  - Computing in parallel

- Performance is the *raison d’être* for parallelism
  - High-performance computing
  - Drives computational science revolution

- Topics of study
  - Parallel architectures
  - Parallel programming
  - Parallel algorithms
  - Parallel performance models and tools
  - Parallel applications
What will you get out of CIS 410/510?

- In-depth understanding of parallel computer design
- Knowledge of how to program parallel computer systems
- Understanding of pattern-based parallel programming
- Exposure to different forms parallel algorithms
- Practical experience using a parallel cluster
- Background on parallel performance modeling
- Techniques for empirical performance analysis
- Fun and new friends
Parallel Processing – What is it?

- A parallel computer is a computer system that uses multiple processing elements simultaneously in a cooperative manner to solve a computational problem.
- Parallel processing includes techniques and technologies that make it possible to compute in parallel:
  - Hardware, networks, operating systems, parallel libraries, languages, compilers, algorithms, tools, …
- Parallel computing is an evolution of serial computing:
  - Parallelism is natural
  - Computing problems differ in level / type of parallelism
- Parallelism is all about performance! Really?
Concurrency

- Consider multiple tasks to be executed in a computer
  - Tasks are concurrent with respect to each if
    - They can execute at the same time (*concurrent execution*)
    - Implies that there are no dependencies between the tasks

- Dependencies
  - If a task requires results produced by other tasks in order to execute correctly, the task’s execution is *dependent*
  - If two tasks are dependent, they are not concurrent
  - Some form of synchronization must be used to enforce (satisfy) dependencies

- Concurrency is fundamental to computer science
  - Operating systems, databases, networking, …
Concurrent and Parallelism

- Concurrent is not the same as parallel! Why?
- Parallel execution
  - Concurrent tasks *actually* execute at the same time
  - Multiple (processing) resources *have* to be available
- **Parallelism = concurrency + “parallel” hardware**
  - Both are required
  - Find concurrent execution opportunities
  - Develop application to execute in parallel
  - Run application on parallel hardware
- Is a parallel application a concurrent application?
- Is a parallel application run with one processor parallel? Why or why not?
Parallelism

- There are granularities of parallelism (parallel execution) in programs
  - Processes, threads, routines, statements, instructions, …
  - Think about what are the software elements that execute concurrently
- These must be supported by hardware resources
  - Processors, cores, … (execution of instructions)
  - Memory, DMA, networks, … (other associated operations)
  - All aspects of computer architecture offer opportunities for parallel hardware execution
- Concurrency is a necessary condition for parallelism
  - Where can you find concurrency?
  - How is concurrency expressed to exploit parallel systems?
Why use parallel processing?

- Two primary reasons (both performance related)
  - Faster time to solution (response time)
  - Solve bigger computing problems (in same time)
- Other factors motivate parallel processing
  - Effective use of machine resources
  - Cost efficiencies
  - Overcoming memory constraints
- Serial machines have inherent limitations
  - Processor speed, memory bottlenecks, …
- Parallelism has become the future of computing
- Performance is still the driving concern
- Parallelism = concurrency + parallel HW + performance
Perspectives on Parallel Processing

- Parallel computer architecture
  - Hardware needed for parallel execution?
  - Computer system design
- (Parallel) Operating system
  - How to manage systems aspects in a parallel computer
- Parallel programming
  - Libraries (low-level, high-level)
  - Languages
  - Software development environments
- Parallel algorithms
- Parallel performance evaluation
- Parallel tools
  - Performance, analytics, visualization, …
Why study parallel computing today?

- Computing architecture
  - Innovations often drive to novel programming models

- Technological convergence
  - The “killer micro” is ubiquitous
  - Laptops and supercomputers are fundamentally similar!
  - Trends cause diverse approaches to converge

- Technological trends make parallel computing inevitable
  - Multi-core processors are here to stay!
  - Practically every computing system is operating in parallel

- Understand fundamental principles and design tradeoffs
  - Programming, systems support, communication, memory, …
  - Performance

- Parallelism is the future of computing
Inevitability of Parallel Computing

- Application demands
  - Insatiable need for computing cycles
- Technology trends
  - Processor and memory
- Architecture trends
- Economics
- Current trends:
  - Today’s microprocessors have multiprocessor support
  - Servers and workstations available as multiprocessors
  - Tomorrow’s microprocessors are multiprocessors
  - Multi-core is here to stay and #cores/processor is growing
  - Accelerators (GPUs, gaming systems)
Application Characteristics

- Application performance demands hardware advances
- Hardware advances generate new applications
- New applications have greater performance demands
  - Exponential increase in microprocessor performance
  - Innovations in parallel architecture and integration

- Range of performance requirements
  - System performance must also improve as a whole
  - Performance requirements require computer engineering
  - Costs addressed through technology advancements
Broad Parallel Architecture Issues

- Resource allocation
  - How many processing elements?
  - How powerful are the elements?
  - How much memory?

- Data access, communication, and synchronization
  - How do the elements cooperate and communicate?
  - How are data transmitted between processors?
  - What are the abstractions and primitives for cooperation?

- Performance and scalability
  - How does it all translate into performance?
  - How does it scale?
Leveraging Moore’s Law

- More transistors = more parallelism opportunities
- Microprocessors
  - Implicit parallelism
    - pipelining
    - multiple functional units
    - superscalar
  - Explicit parallelism
    - SIMD instructions
    - long instruction works
What’s Driving Parallel Computing Architecture?

Processor-DRAM Memory Gap (latency)

“Moore’s Law”

Processor-Memory Performance Gap:
(grows 50% / year)

von Neumann bottleneck!!
(memory wall)
Microprocessor Transistor Counts (1971-2011)

Data from Kunle Olukotun, Lance Hammond, Herb Sutter, Burton Smith, Chris Batten, and Krste Asanović
Slide from Kathy Yelick
What has happened in the last several years?

- Processing chip manufacturers increased processor performance by increasing CPU clock frequency
  - Riding Moore’s law

- Until the chips got too hot!
  - Greater clock frequency $\Rightarrow$ greater electrical power
  - Pentium 4 heat sink
  - Frying an egg on a Pentium 4

- Add multiple cores to add performance
  - Keep clock frequency same or reduced
  - Keep lid on power requirements
**Power Density Growth**

Figure courtesy of Pat Gelsinger, Intel Developer Forum, Spring 2004
What’s Driving Parallel Computing Architecture?

15 Years of exponential growth ~2x year has ended

Data from Kunle Olukotun, Lance Hammond, Herb Sutter, Burton Smith, Chris Batten, and Krste Asanović
Slide from Kathy Yelick
What’s Driving Parallel Computing Architecture?

Power is the root cause of all this

A hardware issue just became a software problem

power wall
Classifying Parallel Systems – Flynn’s Taxonomy

- Distinguishes multi-processor computer architectures along the two independent dimensions
  - *Instruction* and *Data*
  - Each dimension can have one state: *Single* or *Multiple*

- **SISD**: Single Instruction, Single Data
  - Serial (non-parallel) machine

- **SIMD**: Single Instruction, Multiple Data
  - Processor arrays and vector machines

- **MISD**: Multiple Instruction, Single Data (weird)

- **MIMD**: Multiple Instruction, Multiple Data
  - Most common parallel computer systems
Parallel Architecture Types

- Instruction-Level Parallelism
  - Parallelism captured in instruction processing

- Vector processors
  - Operations on multiple data stored in vector registers

- Shared-memory Multiprocessor (SMP)
  - Multiple processors sharing memory
  - Symmetric Multiprocessor (SMP)

- Multicomputer
  - Multiple computer connect via network
  - Distributed-memory cluster

- Massively Parallel Processor (MPP)
Phases of Supercomputing (Parallel) Architecture

- Phase 1 (1950s): sequential instruction execution
- Phase 2 (1960s): sequential instruction issue
  - Pipeline execution, reservations stations
  - Instruction Level Parallelism (ILP)
- Phase 3 (1970s): vector processors
  - Pipelined arithmetic units
  - Registers, multi-bank (parallel) memory systems
- Phase 4 (1980s): SIMD and SMPs
- Phase 5 (1990s): MPPs and clusters
  - Communicating sequential processors
- Phase 6 (>2000): many cores, accelerators, scale, …
Performance Expectations

- If each processor is rated at k MFLOPS and there are p processors, we should expect to see k*p MFLOPS performance? Correct?

- If it takes 100 seconds on 1 processor, it should take 10 seconds on 10 processors? Correct?

- Several causes affect performance
  - Each must be understood separately
  - But they interact with each other in complex ways
    - solution to one problem may create another
    - one problem may mask another

- Scaling (system, problem size) can change conditions

- Need to understand performance space
Scalability

- A program can scale up to use many processors
  - What does that mean?
- How do you evaluate scalability?
- How do you evaluate scalability goodness?
- Comparative evaluation
  - If double the number of processors, what to expect?
  - Is scalability linear?
- Use parallel efficiency measure
  - Is efficiency retained as problem size increases?
- Apply performance metrics
Top 500 Benchmarking Methodology

- Listing of the world’s 500 most powerful computers
- Yardstick for high-performance computing (HPC)
  - $R_{\text{max}}$: maximal performance Linpack benchmark
    - dense linear system of equations ($Ax = b$)
  - $R_{\text{peak}}$: theoretical peak performance
  - $N_{\text{max}}$: problem size needed to achieve $R_{\text{max}}$
  - $N_{1/2}$: problem size needed to achieve 1/2 of $R_{\text{max}}$
  - Manufacturer and computer type
  - Installation site, location, and year
- Updated twice a year at SC and ISC conferences
### Top 10 (November 2013)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Site</th>
<th>System</th>
<th>Cores</th>
<th>Rmax (TFlop/s)</th>
<th>Rpeak (TFlop/s)</th>
<th>Power (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>National Super Computer Center in Guangzhou, China</td>
<td>Tianhe-2 (MilkyWay-2) - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT</td>
<td>3,120,000</td>
<td>33,862.7</td>
<td>54,902.4</td>
<td>17,808</td>
</tr>
<tr>
<td>2</td>
<td>DOE/SC/Oak Ridge National Laboratory, United States</td>
<td>Titan - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.</td>
<td>560,640</td>
<td>17,590.0</td>
<td>27,112.5</td>
<td>8,209</td>
</tr>
<tr>
<td>3</td>
<td>DOE/NNSA/LLNL, United States</td>
<td>Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM</td>
<td>1,572,864</td>
<td>17,173.2</td>
<td>20,132.7</td>
<td>7,890</td>
</tr>
<tr>
<td>4</td>
<td>RIKEN Advanced Institute for Computational Science (AICS), Japan</td>
<td>K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu</td>
<td>705,024</td>
<td>10,510.0</td>
<td>11,280.4</td>
<td>12,660</td>
</tr>
<tr>
<td>5</td>
<td>DOE/SC/Argonne National Laboratory, United States</td>
<td>Mira - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM</td>
<td>786,432</td>
<td>8,586.6</td>
<td>10,066.3</td>
<td>3,945</td>
</tr>
<tr>
<td>6</td>
<td>Swiss National Supercomputing Centre (CSCS), Switzerland</td>
<td>Piz Daint - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect, NVIDIA K20x Cray Inc.</td>
<td>115,984</td>
<td>6,271.0</td>
<td>7,788.9</td>
<td>2,325</td>
</tr>
<tr>
<td>7</td>
<td>Texas Advanced Computing Center/Univ. of Texas, United States</td>
<td>Stampede - PowerEdge C8220, Xeon E5-2680 8C 2.700GHz, Infiniband FDR, Intel Xeon Phi SE10P Dell</td>
<td>462,462</td>
<td>5,168.1</td>
<td>8,520.1</td>
<td>4,510</td>
</tr>
<tr>
<td>8</td>
<td>Forschungszentrum Juelich (FZJ), Germany</td>
<td>JUQUEEN - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM</td>
<td>458,752</td>
<td>5,008.9</td>
<td>5,872.0</td>
<td>2,301</td>
</tr>
<tr>
<td>9</td>
<td>DOE/NNSA/LLNL, United States</td>
<td>Vulcan - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM</td>
<td>393,216</td>
<td>4,293.3</td>
<td>5,033.2</td>
<td>1,972</td>
</tr>
<tr>
<td>10</td>
<td>Leibniz Rechenzentrum, Germany</td>
<td>SuperMUC - iDataPlex DX360M4, Xeon E5-2680 8C 2.700GHz, Infiniband FDR IBM</td>
<td>147,456</td>
<td>2,807.0</td>
<td>3,185.1</td>
<td>3,423</td>
</tr>
</tbody>
</table>
Top 500 – Performance (November 2013)
#1: NUDT Tiahne-2 (Milkyway-2)

- Compute Nodes have 3.432 Tflop/s per node
  - 16,000 nodes
  - 32000 Intel Xeon CPU
  - 48000 Intel Xeon Phi
- Operations Nodes
  - 4096 FT CPUs
- Proprietary interconnect
  - TH2 express
- 1PB memory
  - Host memory only
- Global shared parallel storage is 12.4 PB
- Cabinets: 125+13+24 =162
  - Compute, communication, storage
  - ~750 m2
#2: ORNL Titan Hybrid System (Cray XK7)

- Peak performance of 27.1 PF
  - 24.5 GPU + 2.6 CPU
- 18,688 Compute Nodes each with:
  - 16-Core AMD Opteron CPU
  - NVIDIA Tesla “K20x” GPU
  - 32 + 6 GB memory
- 512 Service and I/O nodes
- 200 Cabinets
- 710 TB total system memory
- Cray Gemini 3D Torus Interconnect
- 8.9 MW peak power

4,352 ft²
#3: LLNL Sequoia (IBM BG/Q)

- **Compute card**
  - 16-core PowerPC A2 processor
  - 16 GB DDR3

- **Compute node has** 98,304 cards

- **Total system size:**
  - 1,572,864 processing cores
  - 1.5 PB memory

- **5-dimensional torus interconnection network**

- **Area of 3,000 ft²**
#4: RIKEN K Computer

- 80,000 CPUs
  - SPARC64 VIIIfx
  - 640,000 cores
- 800 water-cooled racks
- 5D mesh/torus interconnect (Tofu)
  - 12 links between node
  - 12x higher scalability than 3D torus
### Contemporary HPC Architectures

<table>
<thead>
<tr>
<th>Date</th>
<th>System</th>
<th>Location</th>
<th>Comp</th>
<th>Comm</th>
<th>Peak (PF)</th>
<th>Power (MW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>Jaguar; Cray XT5</td>
<td>ORNL</td>
<td>AMD 6c</td>
<td>Seastar2</td>
<td>2.3</td>
<td>7.0</td>
</tr>
<tr>
<td>2010</td>
<td>Tianhe-1A</td>
<td>NSC Tianjin</td>
<td>Intel + NVIDIA</td>
<td>Proprietary</td>
<td>4.7</td>
<td>4.0</td>
</tr>
<tr>
<td>2010</td>
<td>Nebulae</td>
<td>NSCS Shenzhen</td>
<td>Intel + NVIDIA</td>
<td>IB</td>
<td>2.9</td>
<td>2.6</td>
</tr>
<tr>
<td>2010</td>
<td>Tsubame 2</td>
<td>TiTech</td>
<td>Intel + NVIDIA</td>
<td>IB</td>
<td>2.4</td>
<td>1.4</td>
</tr>
<tr>
<td>2011</td>
<td>K Computer</td>
<td>RIKEN/Kobe</td>
<td>SPARC64 VIIIfx</td>
<td>Tofu</td>
<td>10.5</td>
<td>12.7</td>
</tr>
<tr>
<td>2012</td>
<td>Titan; Cray XK6</td>
<td>ORNL</td>
<td>AMD + NVIDIA</td>
<td>Gemini</td>
<td>27</td>
<td>9</td>
</tr>
<tr>
<td>2012</td>
<td>Mira; BlueGeneQ</td>
<td>ANL</td>
<td>SoC</td>
<td>Proprietary</td>
<td>10</td>
<td>3.9</td>
</tr>
<tr>
<td>2012</td>
<td>Sequoia; BlueGeneQ</td>
<td>LLNL</td>
<td>SoC</td>
<td>Proprietary</td>
<td>20</td>
<td>7.9</td>
</tr>
<tr>
<td>2012</td>
<td>Blue Waters; Cray</td>
<td>NCSA/UIUC</td>
<td>AMD + (partial)</td>
<td>Gemini</td>
<td>11.6</td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Stampede</td>
<td>TACC</td>
<td>Intel + MIC</td>
<td>IB</td>
<td>9.5</td>
<td>5</td>
</tr>
<tr>
<td>2013</td>
<td>Tianhe-2</td>
<td>NSCC-GZ (Guangzhou)</td>
<td>Intel + MIC</td>
<td>Proprietary</td>
<td>54</td>
<td>~20</td>
</tr>
</tbody>
</table>
## Top 10 (Top500 List, June 2011)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Site</th>
<th>Computer</th>
<th>Country</th>
<th>Cores</th>
<th>Rmax [Pflops]</th>
<th>% of Peak</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RIKEN Advanced Inst for Comp Sci</td>
<td>K Computer Fujitsu SPARC64 VIII fx + custom</td>
<td>Japan</td>
<td>548,352</td>
<td>8.16</td>
<td>93</td>
</tr>
<tr>
<td>2</td>
<td>Nat. SuperComputer Center in Tianjin</td>
<td>Tianhe-1A, NUDT Intel + Nvidia GPU + custom</td>
<td>China</td>
<td>186,368</td>
<td>2.57</td>
<td>55</td>
</tr>
<tr>
<td>3</td>
<td>DOE / OS Oak Ridge Nat Lab</td>
<td>Jaguar, Cray AMD + custom</td>
<td>USA</td>
<td>224,162</td>
<td>1.76</td>
<td>75</td>
</tr>
<tr>
<td>4</td>
<td>Nat. Supercomputer Center in Shenzhen</td>
<td>Nebulea, Dawning Intel + Nvidia GPU + IB</td>
<td>China</td>
<td>120,640</td>
<td>1.27</td>
<td>43</td>
</tr>
<tr>
<td>5</td>
<td>GSIC Center, Tokyo Institute of Technology</td>
<td>Tusbame 2.0, HP Intel + Nvidia GPU + IB</td>
<td>Japan</td>
<td>73,278</td>
<td>1.19</td>
<td>52</td>
</tr>
<tr>
<td>6</td>
<td>DOE / NNSA LANL &amp; SNL</td>
<td>Cielo, Cray AMD + custom</td>
<td>USA</td>
<td>142,272</td>
<td>1.11</td>
<td>81</td>
</tr>
<tr>
<td>7</td>
<td>NASA Ames Research Center/NAS</td>
<td>Pleades SGI Altix ICE 8200EX/8400EX + IB</td>
<td>USA</td>
<td>111,104</td>
<td>1.09</td>
<td>83</td>
</tr>
<tr>
<td>8</td>
<td>DOE / OS Lawrence Berkeley Nat Lab</td>
<td>Hopper, Cray AMD + custom</td>
<td>USA</td>
<td>153,408</td>
<td>1.054</td>
<td>82</td>
</tr>
<tr>
<td>9</td>
<td>Commissariat a l’Energie Atomique (CEA)</td>
<td>Tera-10, Bull Intel + IB</td>
<td>France</td>
<td>138,368</td>
<td>1.050</td>
<td>84</td>
</tr>
<tr>
<td>10</td>
<td>DOE / NNSA Los Alamos Nat Lab</td>
<td>Roadrunner, IBM AMD + Cell GPU + IB</td>
<td>USA</td>
<td>122,400</td>
<td>1.04</td>
<td>76</td>
</tr>
</tbody>
</table>

Figure credit: http://www.netlib.org/utk/people/JackDongarra/SLIDES/korea-2011.pdf
Japanese K Computer (#1 in June 2011)

<table>
<thead>
<tr>
<th>K computer Specifications</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong> (SPARC64 Vll/x)</td>
<td><strong>Topology</strong> 6D Mesh/Torus</td>
</tr>
<tr>
<td>Cores/Node</td>
<td>8 cores (@2GHz)</td>
</tr>
<tr>
<td>Performance</td>
<td>128GFlops</td>
</tr>
<tr>
<td>Architecture</td>
<td>SPARC V9 + HPC extension</td>
</tr>
<tr>
<td>Cache</td>
<td>L1(I/D) Cache : 32KB/32KB</td>
</tr>
<tr>
<td></td>
<td>L2 Cache : 6MB</td>
</tr>
<tr>
<td>Power</td>
<td>58W (typ. 30 C)</td>
</tr>
<tr>
<td>Mem. bandwidth</td>
<td>64GB/s.</td>
</tr>
<tr>
<td><strong>Node</strong></td>
<td><strong>Performance</strong> 5GB/s. for each link</td>
</tr>
<tr>
<td>Configuration</td>
<td>1 CPU / Node</td>
</tr>
<tr>
<td>Memory capacity</td>
<td>16GB (2GB/core)</td>
</tr>
<tr>
<td><strong>System board(SB)</strong></td>
<td><strong>No. of link</strong> 10 links/ node</td>
</tr>
<tr>
<td>No. of nodes</td>
<td>4 nodes /SB</td>
</tr>
<tr>
<td><strong>Rack</strong></td>
<td><strong>Additional feature</strong> H/W barrier, reduction</td>
</tr>
<tr>
<td>No. of SB</td>
<td>24 SBs/rack</td>
</tr>
<tr>
<td><strong>System</strong></td>
<td><strong>Architecture</strong> Routing chip structure (no outside switch box)</td>
</tr>
<tr>
<td>Nodes/system</td>
<td><strong>Cooling</strong> CPU, ICC*</td>
</tr>
<tr>
<td></td>
<td>Direct water cooling</td>
</tr>
<tr>
<td></td>
<td><strong>Other parts</strong> Air cooling</td>
</tr>
</tbody>
</table>

**System**
- LINPACK 10 PFlops
- over 1PB mem.
- 800 racks
- 80,000 CPUs
- 640,000 cores

**New Linpack run with 705,024 cores at 10.51 Pflop/s (88,128 CPUs)**
## Top 500 Top 10 (2006)

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Computer</th>
<th>Rmax [TF/s]</th>
<th>Installation Site</th>
<th>Country</th>
<th>Year</th>
<th>#Proc</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>BlueGene/L eServer Blue Gene</td>
<td>280.6</td>
<td>DOE/NNSA/LLNL</td>
<td>USA</td>
<td>2005</td>
<td>131,072</td>
</tr>
<tr>
<td>Sandia/Cray</td>
<td>Red Storm Cray XT3</td>
<td>101.4</td>
<td>NNSA/Sandia</td>
<td>USA</td>
<td>2006</td>
<td>26,544</td>
</tr>
<tr>
<td>IBM</td>
<td>BGW eServer Blue Gene</td>
<td>91.29</td>
<td>IBM Thomas Watson</td>
<td>USA</td>
<td>2005</td>
<td>40,960</td>
</tr>
<tr>
<td>IBM</td>
<td>ASC Purple eServer pSeries p575</td>
<td>75.76</td>
<td>DOE/NNSA/LLNL</td>
<td>USA</td>
<td>2005</td>
<td>12,208</td>
</tr>
<tr>
<td>IBM</td>
<td>MareNostrum JS21 Cluster, Myrinet</td>
<td>62.63</td>
<td>Barcelona Supercomputing Center</td>
<td>Spain</td>
<td>2006</td>
<td>12,240</td>
</tr>
<tr>
<td>Dell</td>
<td>Thunderbird PowerEdge 1850, IB</td>
<td>53.00</td>
<td>NNSA/Sandia</td>
<td>USA</td>
<td>2005</td>
<td>9,024</td>
</tr>
<tr>
<td>Bull</td>
<td>Tera-10 NovaScale 5160, Quadrics</td>
<td>52.84</td>
<td>CEA</td>
<td>France</td>
<td>2006</td>
<td>9,968</td>
</tr>
<tr>
<td>SGI</td>
<td>Columbia Altix, Infiniband</td>
<td>51.87</td>
<td>NASA Ames</td>
<td>USA</td>
<td>2004</td>
<td>10,160</td>
</tr>
<tr>
<td>NEC/Sun</td>
<td>Tsubame Fire x4600, ClearSpeed, IB</td>
<td>47.38</td>
<td>GSIC / Tokyo Institute of Technology</td>
<td>Japan</td>
<td>2006</td>
<td>11,088</td>
</tr>
<tr>
<td>Cray</td>
<td>Jaguar Cray XT3</td>
<td>43.48</td>
<td>ORNL</td>
<td>USA</td>
<td>2006</td>
<td>10,424</td>
</tr>
<tr>
<td>Computer (Full Precision)</td>
<td>Number of Processors</td>
<td>$R_{\text{max}}$ Gflop/s</td>
<td>$N_{\text{max}}$ order</td>
<td>$N_{1/2}$ order</td>
<td>$R_{\text{peak}}$ Gflop/s</td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------------</td>
<td>-------------------------</td>
<td>------------------------</td>
<td>----------------</td>
<td>------------------------</td>
<td></td>
</tr>
<tr>
<td>Earth Simulator, NEC processors****</td>
<td>esc</td>
<td>5104</td>
<td>35610</td>
<td>1041216</td>
<td>265408</td>
<td>40832</td>
</tr>
<tr>
<td>ASCI White-Pacific, IBM SP Power 3 (375 MHz)</td>
<td>llnl</td>
<td>8000</td>
<td>7226</td>
<td>518096</td>
<td>179000</td>
<td>12000</td>
</tr>
<tr>
<td>Compaq AlphaServer SC ES45/EV68 1GHz</td>
<td>psc</td>
<td>3016</td>
<td>4463</td>
<td>280000</td>
<td>85000</td>
<td>6032</td>
</tr>
<tr>
<td>Compaq AlphaServer SC ES45/EV68 1GHz</td>
<td>psc</td>
<td>3024</td>
<td>4059</td>
<td>525000</td>
<td>105000</td>
<td>6048</td>
</tr>
<tr>
<td>Compaq AlphaServer SC ES45/EV68 1GHz</td>
<td>cea</td>
<td>2560</td>
<td>3980</td>
<td>360000</td>
<td>85000</td>
<td>5120</td>
</tr>
<tr>
<td>IBM SP Power3 208 nodes 375 MHz</td>
<td>llnl</td>
<td>3328</td>
<td>3052</td>
<td>371712</td>
<td>4992</td>
<td></td>
</tr>
<tr>
<td>Compaq Alphaserver SC ES45/EV68 1GHz</td>
<td>lanl</td>
<td>2048</td>
<td>2916</td>
<td>272000</td>
<td>4096</td>
<td></td>
</tr>
<tr>
<td>IBM SP Power3 158 nodes 375 MHz</td>
<td>llnl</td>
<td>2528</td>
<td>2526</td>
<td>371712</td>
<td>102400</td>
<td>3792</td>
</tr>
<tr>
<td>ASCI Red Intel Pentium II Xeon core 333MHz</td>
<td>snl</td>
<td>9632</td>
<td>2379.6</td>
<td>362880</td>
<td>75400</td>
<td>3207</td>
</tr>
<tr>
<td>ASCI Blue-Pacific SST, IBM SP 604E(332 MHz)</td>
<td>llnl</td>
<td>5808</td>
<td>2144.4</td>
<td>431344</td>
<td>432344</td>
<td>3868</td>
</tr>
<tr>
<td>ASCI Red Intel Pentium II Xeon core 333MHz</td>
<td>snl</td>
<td>9472</td>
<td>2121.3</td>
<td>251904</td>
<td>66000</td>
<td>3154</td>
</tr>
<tr>
<td>Compaq Alphaserver SC ES45/EV68 1GHz</td>
<td>lanl</td>
<td>1520</td>
<td>2096</td>
<td>390000</td>
<td>71000</td>
<td>3040</td>
</tr>
<tr>
<td>IBM SP 112 nodes (375 MHz POWER3 High)</td>
<td>ibm</td>
<td>1792</td>
<td>1791</td>
<td>275000</td>
<td>275000</td>
<td>2688</td>
</tr>
<tr>
<td>HITACHI SR8000/MPP/1152(450MHz)</td>
<td>u tokyo</td>
<td>1152</td>
<td>1709.1</td>
<td>141000</td>
<td>16000</td>
<td>2074</td>
</tr>
<tr>
<td>HITACHI SR8000-F1/168(375MHz)</td>
<td>leibniz</td>
<td>168</td>
<td>1653</td>
<td>160000</td>
<td>19560</td>
<td>2016</td>
</tr>
<tr>
<td>ASCI Red Intel Pentium II Xeon core 333MHz</td>
<td>snl</td>
<td>6720</td>
<td>1633.3</td>
<td>306720</td>
<td>52500</td>
<td>2238</td>
</tr>
<tr>
<td>SCI ASCI Blue Mountain</td>
<td>lanl</td>
<td>5040</td>
<td>1608</td>
<td>374400</td>
<td>138000</td>
<td>2520</td>
</tr>
<tr>
<td>IBM SP 328 nodes (375 MHz POWER3 Thin)</td>
<td>noo</td>
<td>1312</td>
<td>1417</td>
<td>374000</td>
<td>374000</td>
<td>1968</td>
</tr>
<tr>
<td>Intel ASCI Option Red (200 MHz Pentium Pro)</td>
<td>snl</td>
<td>9152</td>
<td>1338</td>
<td>235000</td>
<td>63000</td>
<td>1830</td>
</tr>
<tr>
<td>NEC SX-5/128M8(3.2ns)</td>
<td>osaka</td>
<td>128</td>
<td>1192.0</td>
<td>129536</td>
<td>10240</td>
<td>1280</td>
</tr>
<tr>
<td>CRAY T3E-1200 (600 MHz)</td>
<td>us government</td>
<td>1488</td>
<td>1127</td>
<td>148800</td>
<td>28272</td>
<td>1786</td>
</tr>
<tr>
<td>HITACHI SR8000-F1/112(375MHz)</td>
<td>leibniz</td>
<td>112</td>
<td>1035.0</td>
<td>120000</td>
<td>15160</td>
<td>1344</td>
</tr>
</tbody>
</table>
Japanese Earth Simulator

- World’s fastest supercomputer!!! (2002)
  - 640 NEC SX-6 nodes
    - 8 vector processors
  - 5104 total processors
  - Single stage crossbar
    - ~2900 meters of cables
  - 10 TB memory
  - 700 TB disk space
  - 1.6 PB mass storage
  - 40 Tflops peak performance
  - 35.6 Tflops Linpack performance
Prof. Malony and colleagues at Japanese ES
Performance Development in Top 500

Figure credit: http://www.netlib.org/utk/people/JackDongarra/SLIDES/korea-2011.pdf
**Exascale Initiative**

- Exascale machines are targeted for 2019
- What are the potential differences and problems?

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>System peak</td>
<td>8.7 Pflop/s</td>
<td>1 Eflop/s</td>
<td>O(100)</td>
</tr>
<tr>
<td>Power</td>
<td>10 MW</td>
<td>~20 MW</td>
<td>????</td>
</tr>
<tr>
<td>System memory</td>
<td>1.6 PB</td>
<td>32 - 64 PB</td>
<td>O(10)</td>
</tr>
<tr>
<td>Node performance</td>
<td>128 GF</td>
<td>1,2 or 15TF</td>
<td>O(10) - O(100)</td>
</tr>
<tr>
<td>Node memory BW</td>
<td>64 GB/s</td>
<td>2 - 4TB/s</td>
<td>O(100)</td>
</tr>
<tr>
<td>Node concurrency</td>
<td>8</td>
<td>O(1k) or 10k</td>
<td>O(100) - O(1000)</td>
</tr>
<tr>
<td>Total Node Interconnect BW</td>
<td>20 GB/s</td>
<td>200-400GB/s</td>
<td>O(10)</td>
</tr>
<tr>
<td>System size (nodes)</td>
<td>68,544</td>
<td>O(100,000) or O(1M)</td>
<td>O(10) - O(100)</td>
</tr>
<tr>
<td>Total concurrency</td>
<td>548,352</td>
<td>O(billion)</td>
<td>O(1,000)</td>
</tr>
<tr>
<td>MTTI</td>
<td>days</td>
<td>O(1 day)</td>
<td>- O(10)</td>
</tr>
</tbody>
</table>
Major Changes to Software and Algorithms

- What were we concerned about before and now?
- Must rethink the design for exascale
  - Data movement is expensive (Why?)
  - Flops per second are cheap (Why?)
- Need to reduce communication and synchronization
- Need to develop fault-resilient algorithms
- How do we deal with massive parallelism?
- Software must adapt to the hardware (autotuning)
Supercomputing and Computational Science

- By definition, a supercomputer is of a class of computer systems that are the most powerful computing platforms at that time
- Computational science has always lived at the leading (and bleeding) edge of supercomputing technology
- “Most powerful” depends on performance criteria
  - Performance metrics related to computational algorithms
  - Benchmark “real” application codes
- Where does the performance come from?
  - More powerful processors
  - More processors (cores)
  - Better algorithms
Computational Science

- Traditional scientific methodology
  - Theoretical science
    - Formal systems and theoretical models
    - Insight through abstraction, reasoning through proofs
  - Experimental science
    - Real system and empirical models
    - Insight from observation, reasoning from experiment design

- Computational science
  - Emerging as a principal means of scientific research
  - Use of computational methods to model scientific problems
    - Numerical analysis plus simulation methods
    - Computer science tools
  - Study and application of these solution techniques
Computational Challenges

- Computational science thrives on computer power
  - Faster solutions
  - Finer resolution
  - Bigger problems
  - Improved interaction
  - BETTER SCIENCE!!!

- How to get more computer power?
  - Scalable parallel computing

- Computational science also thrives better integration
  - Couple computational resources
  - Grid computing
Scalable Parallel Computing

- Scalability in parallel architecture
  - Processor numbers
  - Memory architecture
  - Interconnection network
  - Avoid critical architecture bottlenecks
- Scalability in computational problem
  - Problem size
  - Computational algorithms
    - Computation to memory access ratio
    - Computation to communication ratio
- Parallel programming models and tools
- Performance scalability
Next Lectures

- Parallel computer architectures (Thursday)
- Parallel performance models (Tuesday next week)