EECS 594 Spring 2012
Lecture 1:
Overview of High-Performance Computing

Jack Dongarra
Electrical Engineering and Computer Science Department
University of Tennessee

1

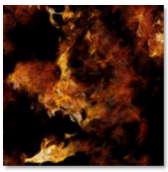
Simulation: The Third Pillar of Science

- Traditional scientific and engineering paradigm:
  1) Do theory or paper design.
  2) Perform experiments or build system.
- Limitations:
  - Too difficult -- build large wind tunnels.
  - Too expensive -- build a throw-away passenger jet.
  - Too slow -- wait for climate or galactic evolution.
  - Too dangerous -- weapons, drug design, climate experimentation.
- Computational science paradigm:
  3) Use high performance computer systems to simulate the phenomenon
     - » Base on known physical laws and efficient numerical methods.

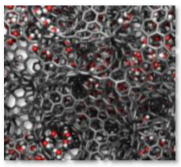
2

Computational Science

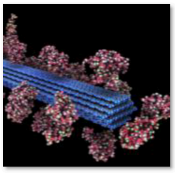
Applications to Energy



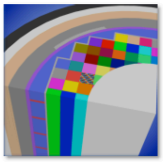
Turbulence
Understanding the statistical geometry of turbulent dispersion of pollutants in the environment.



Energy Storage
Understanding the storage and flow of energy in next-generation nanostructured carbon tube supercapacitors

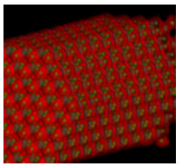


Biofuels
A comprehensive simulation model of lignocellulosic biomass to understand the bottleneck to sustainable and economical ethanol production.



Nuclear Energy
High-fidelity predictive simulation tools for the design of next-generation nuclear reactors to safely increase operating margins.

Smart Truck
Aerodynamic forces account for ~53% of long haul truck fuel use. ORNL's Jaguar predicted 12% drag reduction and yielded EPA-certified 6.9% increase in fuel efficiency.

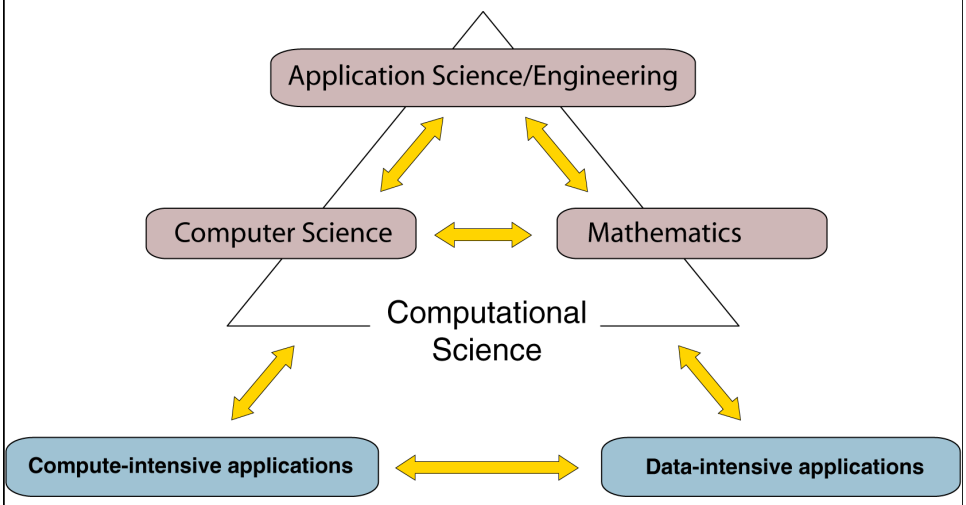


Nano Science
Understanding the atomic and electronic properties of nanostructures in next-generation photovoltaic solar cell materials.

Source: Steven E. Koonin, DOE

Computational Science Fuses Three Distinct Elements:

4



Computational Science As An Emerging Academic Pursuit

5

- Many Programs in Computational Science
  - College for Computing
    - Georgia Tech; NJIT; CMU; …
  - Degrees
    - Rice, Utah, UCSB; …
  - Minor
    - Penn State, U Wisc, SUNY Brockport
  - Certificate
    - Old Dominion, U of Georgia, Boston U, …
  - Concentration
    - Cornell, Northeastern, Colorado State, …
  - Courses

At the University of Tennessee

6

- A few years ago there was a discussion to create a program in Computational Science
- This program evolved out of a set of meetings and discussions with faculty, students, and administration.
- Modeled on a similar minor degree program in the Statistics Department on campus.
- Appeared in the 2007-2008 Graduate Catalog

Graduate Minor in Computational Science

7

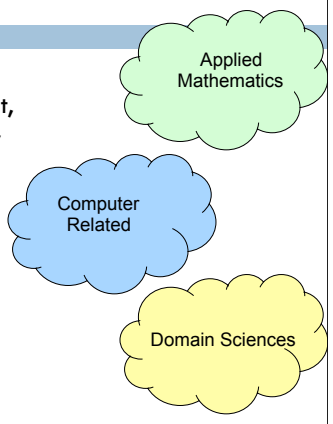
- Students in one of the three general areas of Computational Science;
  - Applied Mathematics,
  - Computer related, or
  - a Domain Science

  will become exposed to and better versed in the other two areas that are currently outside their "home" area.

- A pool of courses which deals with each of the three main areas has been put together by participating department for students to select from.
- **Interdisciplinary Graduate Minor in Computational Science (IGMCS)**

IGMCS: Requirements

- The Minor requires a combination of course work from three disciplines - Computer related, Mathematics/Stat, and a participating Science/Engineering domain (e.g., Chemical Engineering, Chemistry, Physics).
- At the Masters level a minor in Computational Science will require 9 hours (3 courses) from the pools.
  - At least 6 hours (2 courses) must be taken outside the student's home area.
  - Students must take at least 3 hours (1 course) from each of the 2 non-home areas
- At the Doctoral level a minor in computation science will require 15 hours (5 courses) from the pools.
  - At least 9 hours (3 courses) must be taken outside the student's home area.
  - Students must take at least 3 hours (1 course) from each of the 2 non-home areas



IGMCS Process for Students

1. A student, with guidance from their faculty advisor, lays out a program of courses
2. Next, discussion with department's IGMCS liaison
3. Form generated with courses to be taken
4. Form is submitted for approval by the IGMCS Program Committee

IGMCS Participating Departments

Department	IGMCS Liaison	Email
Biochem & Cell and Mole Bio	Dr. Cynthia Peterson	cbpeters@utk.edu
Chemical Engineering	Dr. David Keffer	dkeffer@utk.edu
Chemistry	Dr. Robert Hinde	rhinde@utk.edu
Civil & Envir. Eng.	Dr. Joshua Fu	jsfu@utk.edu
Earth & Planetary Sci	Dr. Edmund Perfect	eperfect@utk.edu
Ecology & Evol. Biology	Dr. Paul Armsworth	p.armsworth@utk.edu
EECS	Dr. Jack Dongarra	dongarra@eecs.utk.edu
Genome Sci & Tech	Dr. Cynthia Peterson	cbpeters@utk.edu
Geography	Dr. Bruce Ralston	bralston@utk.edu
Information Science	Dr. Devendra Potnis	dpotnis@utk.edu
Material Science & Eng	Dr. James Morris	Morrisj@ornl.gov
Mathematics	Dr. Tim Schulze	schulze@math.utk.edu
Mech, Aero & Biomed Eng	Dr. A.J. Baker	ajbaker@utk.edu
Physics	Dr. Thomas Papenbrock	tpapenbr@utk.edu
Statistics	Dr. Hamparsum Bozdogan	bozdogan@utk.edu

Students in Departments Not Participating in the IGMCS Program

- A student in such a situation can still participate.
  - Student and advisor should submit to the Chair of the IGMCS Program Committee the courses to be taken.
  - Requirement is still the same:
    - Minor requires a combination of course work from three disciplines - Computer Science related, Mathematics/Stat, and a participating Science/Engineering domain (e.g., Chemical Engineering, Chemistry, Physics).
- Student's department should be encouraged to participate in the IGMCS program.
  - Easy to do, needs approved set of courses and a liaison

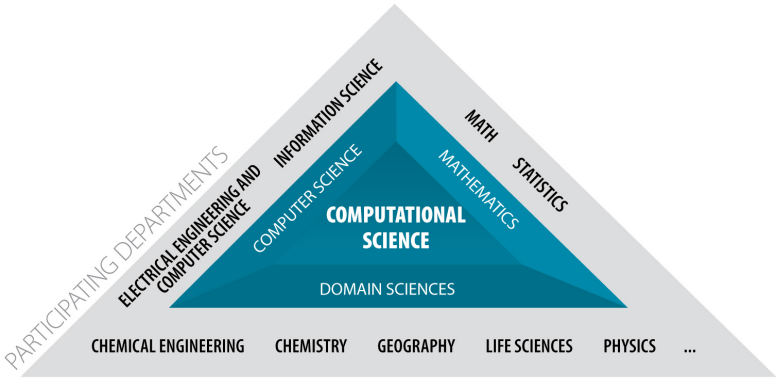
Internship

12

- This is optional but strongly encouraged.
- Students in the program can fulfill 3 hrs. of their requirement through an Internship with researchers outside the student's major.
- The internship may be taken offsite, e.g. ORNL, or on campus by working with a faculty member in another department.
- Internships must have the approval of the IGMCS Program Committee.

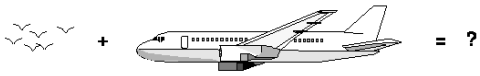
13

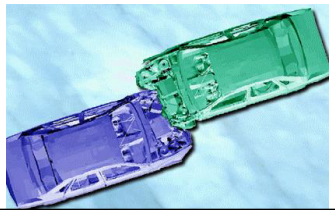
COMPUTATIONAL SCIENCE: INHERENTLY INTERDISCIPLINARY

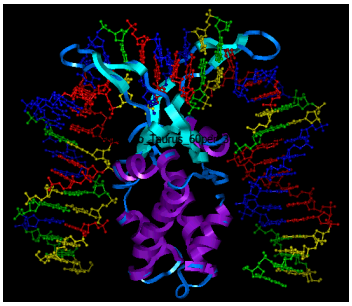


Why Turn to Simulation?

- When the problem is too . . .
  - Complex
  - Large / small
  - Expensive
  - Dangerous
- to do any other way.

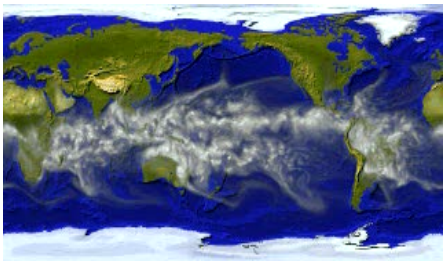




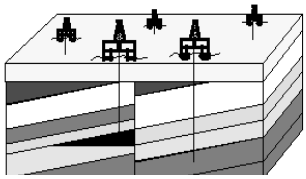


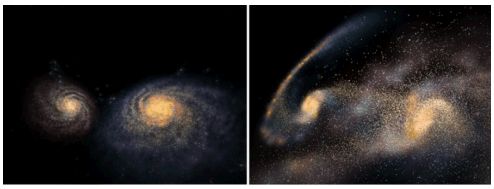
14

Why Turn to Simulation?



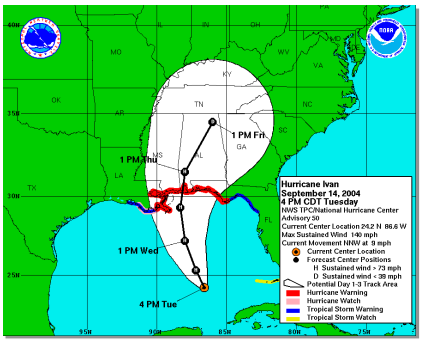
- Climate / Weather Modeling
- Data intensive problems (data-mining, oil reservoir simulation)
- Problems with large length and time scales (cosmology)





Weather and Economic Loss

- $10T U.S. economy
  - 40% is adversely affected by weather and climate
- $1M in loss to evacuate each mile of coastline
  - we now over warn by 3X!
  - average over warning is 200 miles, or $200M per event
- Improved forecasts
  - lives saved and reduced cost
- LEAD
  - Linked Environments for Atmospheric Discovery
    - » Oklahoma, Indiana, UCAR, Colorado State, Howard, Alabama, Millersville, NCSA, North Carolina



16

Source: Kelvin Droegemeier, Oklahoma

Example Science and Engineering Drivers

- Climate
- Nuclear Energy
- Combustion
- Advanced Materials
- CO_2 Sequestration
- Basic Science

- Common Needs
  - Multiscale
  - Uncertainty Quantification
  - Rare Event Statistics

17

Look at the Fastest Computers

- **Strategic importance of supercomputing**
  - Essential for scientific discovery
  - Critical for national security
  - Fundamental contributor to the economy and competitiveness through use in engineering and manufacturing
- **Supercomputers are *the tool for solving the most challenging problems through simulations***

18

Units of High Performance Computing

1 Mflop/s	1 Megaflop/s	10^6 Flop/sec
1 Gflop/s	1 Gigaflop/s	10^9 Flop/sec
1 Tflop/s	1 Teraflop/s	10^{12} Flop/sec
1 Pflop/s	1 Petaflop/s	10^{15} Flop/sec
1 MB	1 Megabyte	10^6 Bytes
1 GB	1 Gigabyte	10^9 Bytes
1 TB	1 Terabyte	10^{12} Bytes
1 PB	1 Petabyte	10^{15} Bytes

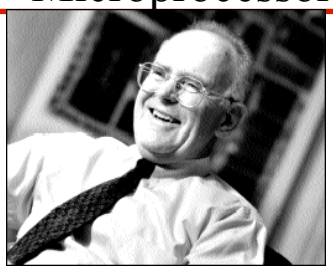
19

High-Performance Computing Today

- **In the past decade, the world has experienced one of the most exciting periods in computer development.**
- **Microprocessors have become smaller, denser, and more powerful.**
- **The result is that microprocessor-based supercomputing is rapidly becoming the technology of preference in attacking some of the most important problems of science and engineering.**

20

Technology Trends: Microprocessor Capacity

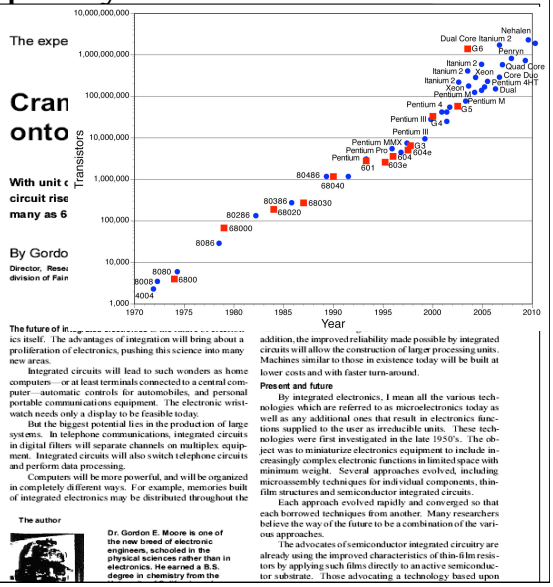


Gordon Moore (co-founder of Intel) Electronics Magazine, 1965

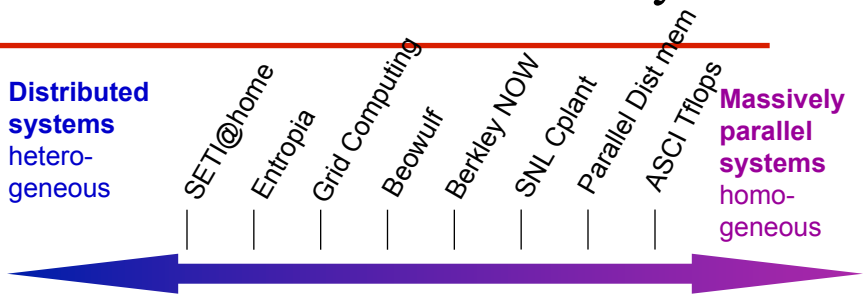
Number of devices/chip doubles every 18 months

2X transistors/Chip Every 1.5 years

Called "Moore's Law"



Distributed and Parallel Systems

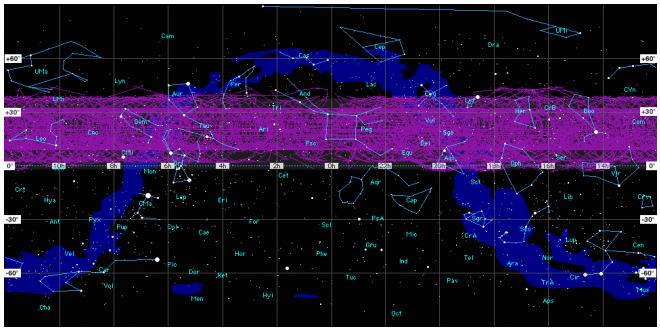


- Gather (unused) resources
- Steal cycles
- System SW manages resources
- System SW adds value
- 10% - 20% overhead is OK
- Resources drive applications
- Time to completion is not critical
- Time-shared

- Bounded set of resources
- Apps grow to consume all cycles
- Application manages resources
- System SW gets in the way
- 5% overhead is maximum
- Apps drive purchase of equipment
- Real-time constraints
- Space-shared

SETI@home: Global Distributed Computing

- Running on 500,000 PCs, ~1000 CPU Years per Day
  - 485,821 CPU Years so far
- Sophisticated Data & Signal Processing Analysis
- Distributes Datasets from Arecibo Radio Telescope

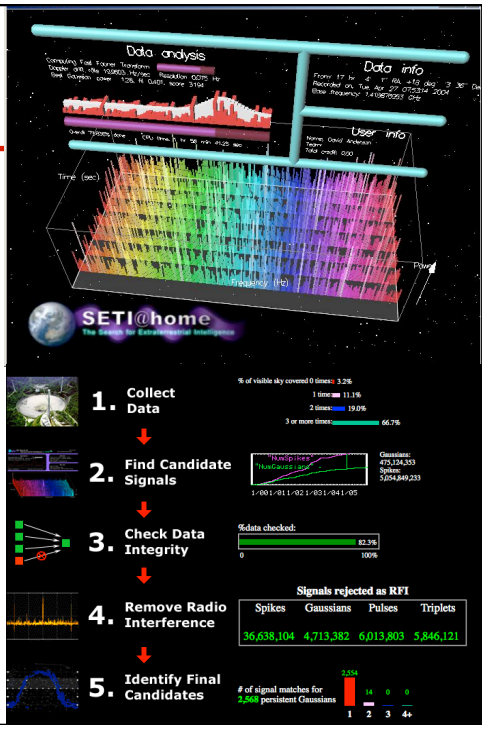


Berkeley Open Infrastructure for Network Computing

23

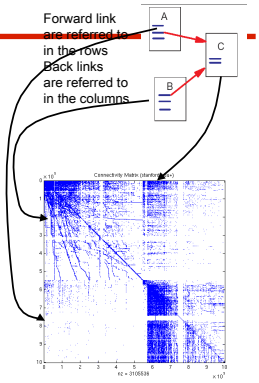
SETI@home

- Use thousands of Internet-connected PCs to help in the search for extraterrestrial intelligence.
- When their computer is idle or being wasted this software will download a 300 kilobyte chunk of data for analysis. Performs about 3 Tflops for each client in 15 hours.
- The results of this analysis are sent back to the SETI team, combined with thousands of other participants.
- Largest distributed computation project in existence
  - Potential 769 Tflop/s



Google™

- Google query attributes
  - 150M queries/day (2000/second)
  - 100 countries
  - 8.0B documents in the index
- Data centers
  - 100,000 Linux systems in data centers around the world
    - » 15 TFlop/s and 1000 TB total capability
    - » 40-80 1U/2U servers/cabinet
    - » 100 MB Ethernet switches/ cabinet with gigabit Ethernet uplink
  - growth from 4,000 systems (June 2000)
    - » 18M queries then
- Performance and operation
  - simple reissue of failed commands to new servers
  - no performance debugging
    - » problems are not reproducible

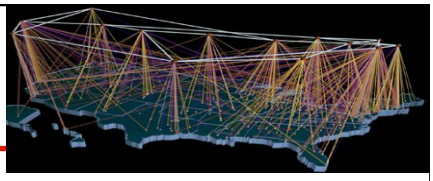


Eigenvalue problem; $Ax = \lambda x$
$n=8\times10^9$
(see: MathWorks Cleve's Corner)

The matrix is the transition probability matrix of the Markov chain; $Ax = x$

Source: Monika Henzinger, Google & Cleve Moler

Next Generation Web



- To treat CPU cycles and software like commodities.
- Enable the coordinated use of geographically distributed resources – in the absence of central control and existing trust relationships.
- Computing power is produced much like utilities such as power and water are produced for consumers.
- Users will have access to "power" on demand
- This is one of our efforts at UT.

•Why Parallel Computing

- Desire to solve bigger, more realistic applications problems.
- Fundamental limits are being approached.
- More cost effective solution

27

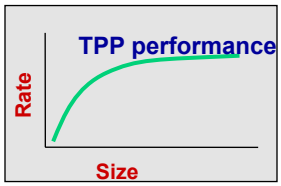
ICL UT

TOP500 SUPERCOMPUTER

H. Meuer, H. Simon, E. Strohmaier, & JD

- Listing of the 500 most powerful Computers in the World
- Yardstick: Rmax from LINPACK MPP

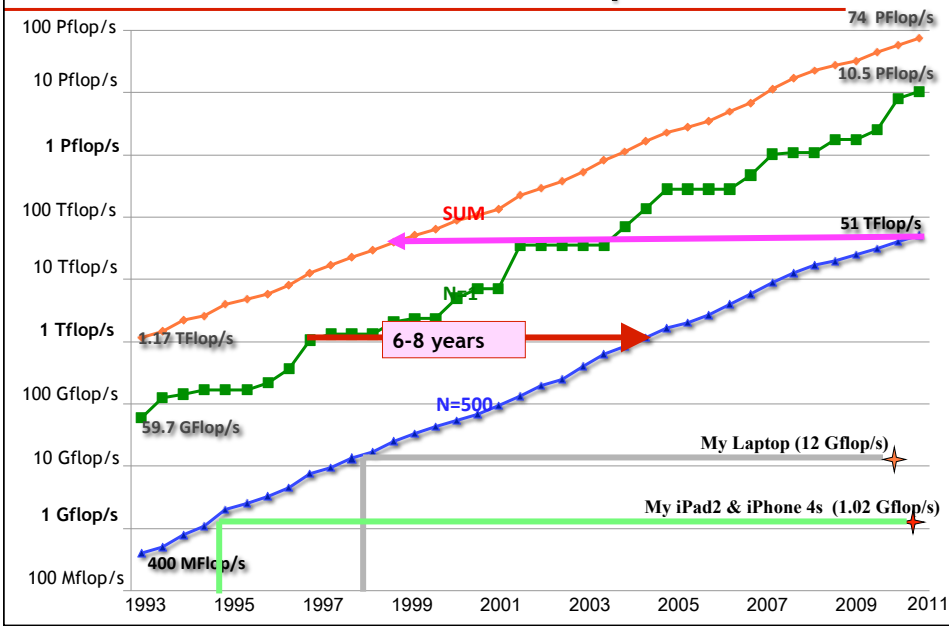
$Ax=b$, *dense problem*



- Updated twice a year
  SC'xy in the States in November
  Meeting in Germany in June
- All data available from **www.top500.org**

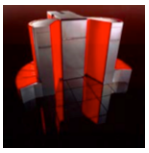
28

Performance Development



Looking at the Gordon Bell Prize

(Recognize outstanding achievement in high-performance computing applications and encourage development of parallel processing)

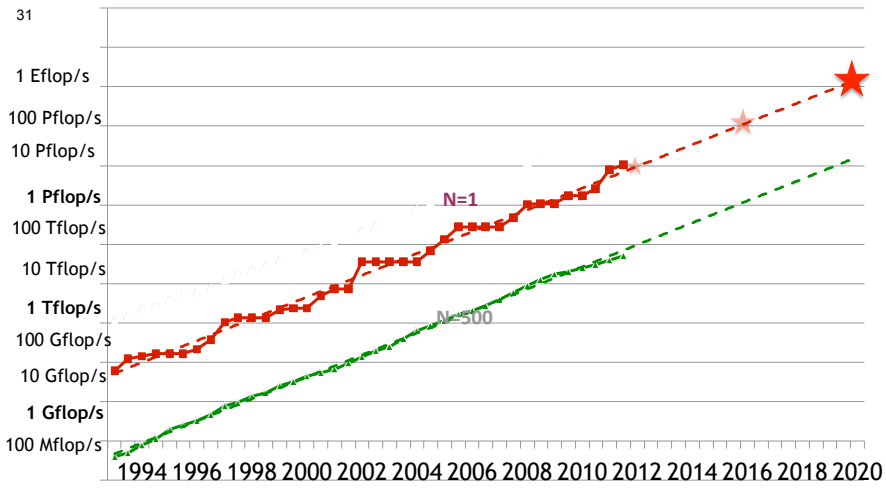


- 1 GFlop/s; 1988; Cray Y-MP; 8 Processors
  - Static finite element analysis

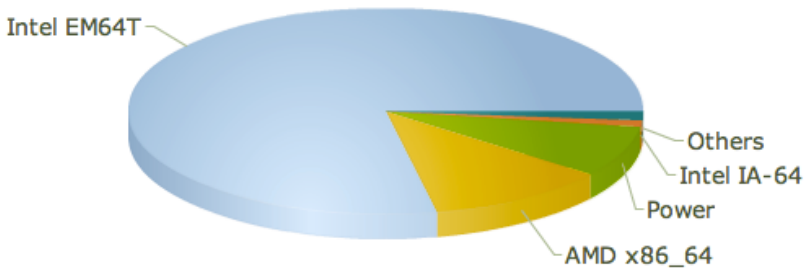
- 1 TFlop/s; 1998; Cray T3E; 1024 Processors
  - Modeling of metallic magnet atoms, using a variation of the locally self-consistent multiple scattering method.

- 1 PFlop/s; 2008; Cray XT5; 1.5×10^5 Processors
  - Superconductive materials
- 1 EFlop/s; ~2018; ?; 1×10^7 Processors (10^9 threads)

Performance Development in Top500



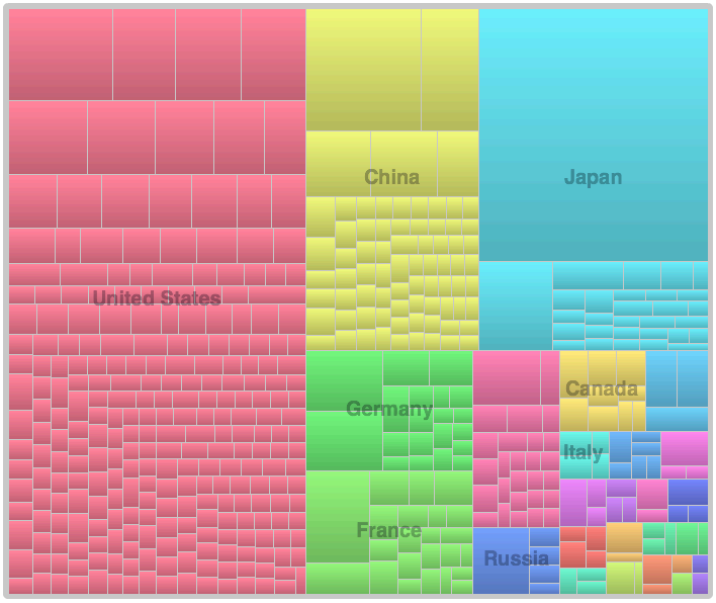
Processors Used in the Top500 Systems



Intel 77% (384)
AMD 13% (63)
IBM 9.8% (49)

32

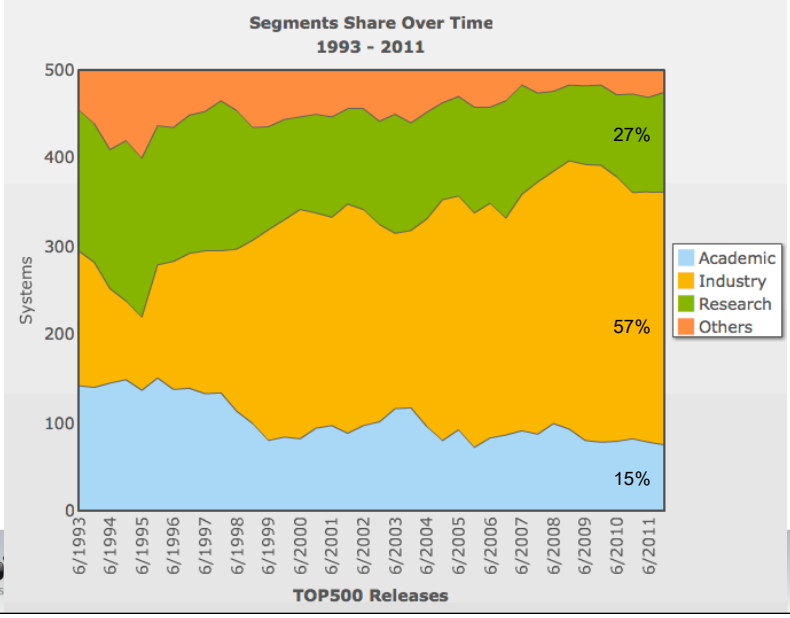
Countries Share



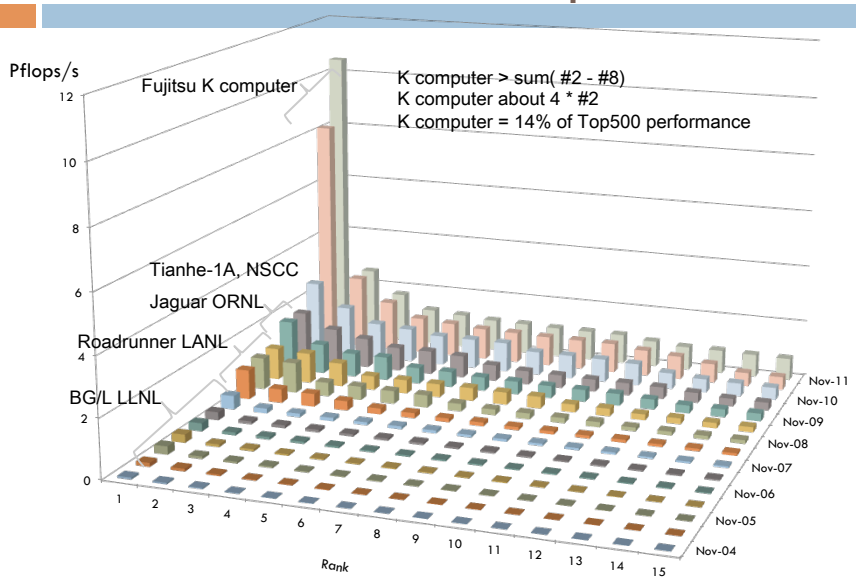
Absolute Counts

US:	263
China:	75
Japan:	30
UK:	27
France:	23
Germany:	20

Customer Segments



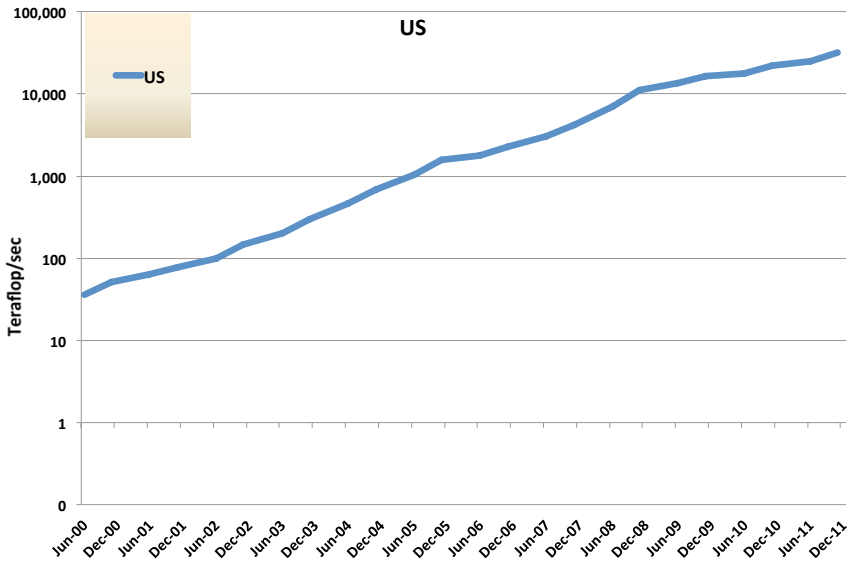
Performance of the Top15 Over Time



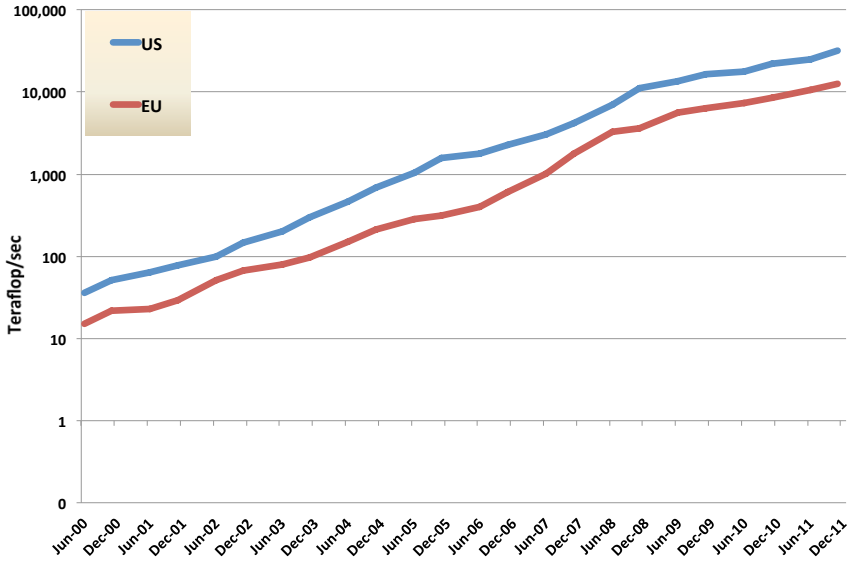
Pflop/s Club (17 systems; Peak)

Name	Peak Pflop/s	Country 6 5 2 2 1 1	
K computer	11.3	Japan	Fujitsu: Sparc/Self
Tianhe-1A	4.70	China	NUDT: Hybrid Intel/Nvidia/Self
Nebulae	2.98	China	Dawning: Hybrid Intel/Nvidia/IB
Jaguar	2.33	US	Cray: AMD/Self
TSUBAME 2.0	2.29	Japan	HP: Hybrid Intel/Nvidia/IB
Roadrunner	1.38	US	IBM: Hybrid AMD/Cell/IB
Lomonosov	1.37	Russia	T-Platform: Hybrid Intel/Nvidia/IB
Cielo	1.37	US	Cray: AMD/Self
Tianhe-1A Hunan Solution	1.34	China	NUDT: Hybrid Intel/Nvidia/Self
Pleiades	1.32	US	SGI: Intel/IB
Hopper	1.29	US	Cray: AMD/Self
Tera-100	1.25	France	Bull: Intel/IB
Kraken XT5	1.17	US	Cray: AMD/Self
Sunway Blue Light	1.07	China	Sunway: Shenwei/IB
HERMIT	1.04	Germany	Cray: AMD/Self
Mole-8.5	1.01	China	CAS: Hybrid Intel/Nvidia/IB
JUGENE	1.00	Germany	IBM: BG-P/Self

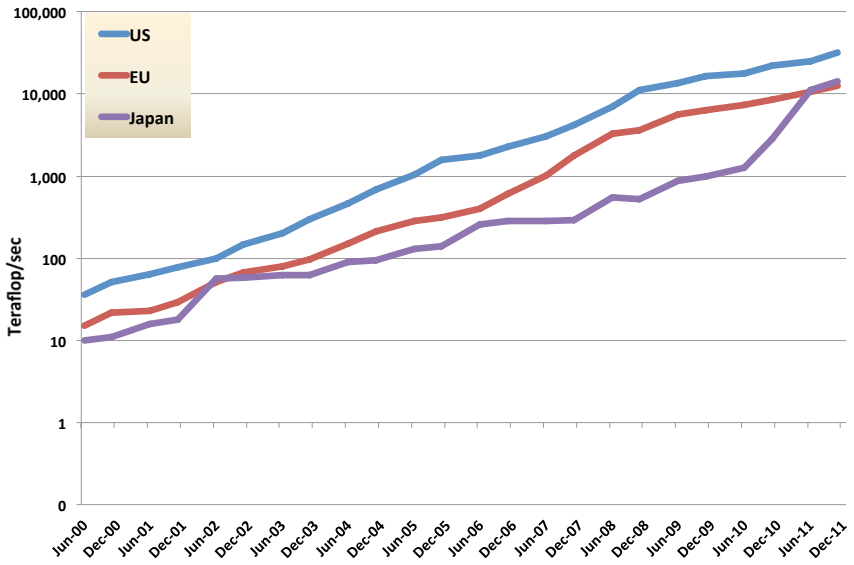
Performance of Countries



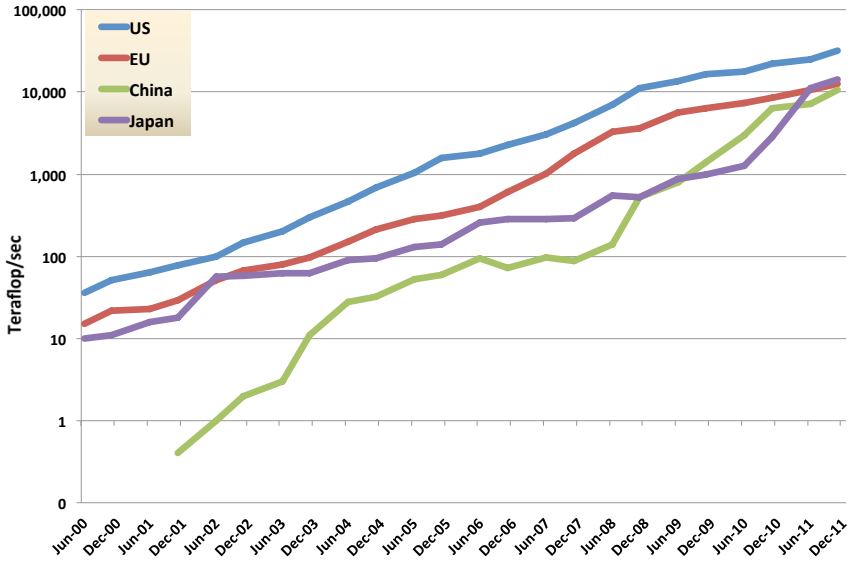
Performance of Countries



Performance of Countries



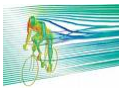
Performance of Countries



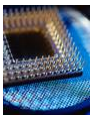
Industrial Use of Supercomputers

- Of the 500 Fastest Supercomputer
  - Worldwide, Industrial Use is > 60%

- Aerospace
- Automotive
- Biology
- CFD
- Database
- Defense
- Digital Content Creation
- Digital Media
- Electronics
- Energy
- Environment
- Finance
- Gaming
- Geophysics
- Image Proc./Rendering
- Information Processing Service
- Information Service
- Life Science
- Media
- Medicine
- Pharmaceutics
- Research
- Retail
- Semiconductor
- Telecomm
- Weather and Climate Research
- Weather Forecasting



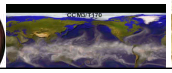












Power is an Industry Wide Problem

Google facilities

- leveraging hydroelectric power
- old aluminum plants

The New York Times **"Hiding in Plain Sight, Google Seeks More Power", by John Markoff, June 14, 2006**

Google Plant in The Dalles, Oregon, from NYT, June 14, 2006

Microsoft and Yahoo are building big data centers upstream in Wenatchee and Quincy, Wash.
– To keep up with Google, which means they need cheap electricity and readily accessible data networking

Microsoft Quincy, Wash. 470,000 Sq Ft, 47MW!

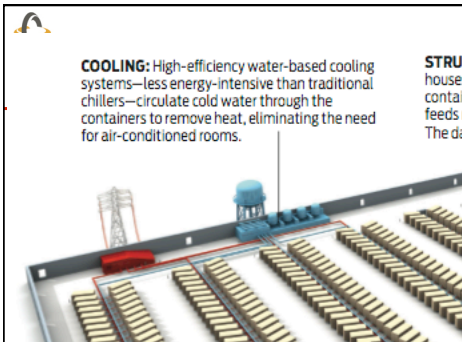
42

Commercial Data Centers

Facebook
300,000 sq ft
1.5 cents per kW hour
Prineville OR

Microsoft 700,000 sq ft in Chicago

Apple 500,000 sq ft in Rural NC 4 cents kW/h



November 2011: The TOP10

Rank	Site	Computer	Country	Cores	Rmax [Pflops]	% of Peak
1	RIKEN Advanced Inst for Comp Sci	K computer Fujitsu SPARC64 VIIIfx + custom	Japan	705,024	10.5	93
2	Nat. SuperComputer Center in Tianjin	Tianhe-1A, NUDT Intel + Nvidia GPU + custom	China	186,368	2.57	55
3	DOE / OS Oak Ridge Nat Lab	Jaguar, Cray AMD + custom	USA	224,162	1.76	75
4	Nat. Supercomputer Center in Shenzhen	Nebulea, Dawning Intel + Nvidia GPU + IB	China	120,640	1.27	43
5	GSIC Center, Tokyo Institute of Technology	Tusbame 2.0, HP Intel + Nvidia GPU + IB	Japan	73,278	1.19	52
6	DOE / NNSA LANL & SNL	Cielo, Cray AMD + custom	USA	142,272	1.11	81
7	NASA Ames Research Center/NAS	Plelades SGI Altix ICE 8200EX/8400EX + IB	USA	111,104	1.09	83
8	DOE / OS Lawrence Berkeley Nat Lab	Hopper, Cray AMD + custom	USA	153,408	1.054	82
9	Commissariat a l'Energie Atomique (CEA)	Tera-10, Bull Intel + IB	France	138,368	1.050	84
10	DOE / NNSA Los Alamos Nat Lab	Roadrunner, IBM AMD + Cell GPU + IB	USA	122,400	1.04	76

November 2011: The TOP10

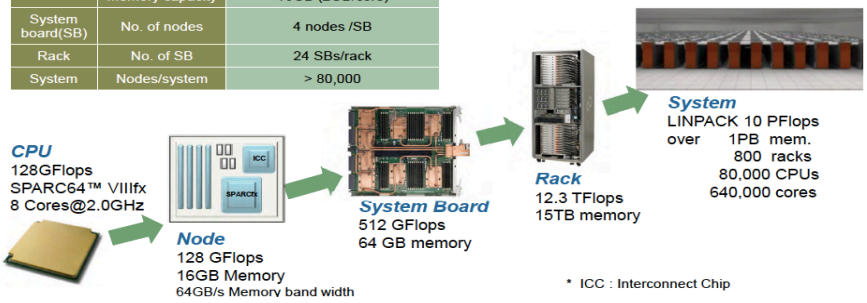
Rank	Site	Computer	Country	Cores	Rmax [Pflops]	% of Peak	Power [MW]	MFlops /Watt
1	RIKEN Advanced Inst for Comp Sci	K computer Fujitsu SPARC64 VIIIfx + custom	Japan	705,024	10.5	93	12.7	830
2	Nat. SuperComputer Center in Tianjin	Tianhe-1A, NUDT Intel + Nvidia GPU + custom	China	186,368	2.57	55	4.04	636
3	DOE / OS Oak Ridge Nat Lab	Jaguar, Cray AMD + custom	USA	224,162	1.76	75	7.0	251
4	Nat. Supercomputer Center in Shenzhen	Nebulea, Dawning Intel + Nvidia GPU + IB	China	120,640	1.27	43	2.58	493
5	GSIC Center, Tokyo Institute of Technology	Tusbame 2.0, HP Intel + Nvidia GPU + IB	Japan	73,278	1.19	52	1.40	865
6	DOE / NNSA LANL & SNL	Cielo, Cray AMD + custom	USA	142,272	1.11	81	3.98	279
7	NASA Ames Research Center/NAS	Plelades SGI Altix ICE 8200EX/8400EX + IB	USA	111,104	1.09	83	4.10	265
8	DOE / OS Lawrence Berkeley Nat Lab	Hopper, Cray AMD + custom	USA	153,408	1.054	82	2.91	362
9	Commissariat a l'Energie Atomique (CEA)	Tera-10, Bull Intel + IB	France	138,368	1.050	84	4.59	229
10	DOE / NNSA Los Alamos Nat Lab	Roadrunner, IBM AMD + Cell GPU + IB	USA	122,400	1.04	76	2.35	446
500	IT Service	IBM Cluster, Intel + GigE	USA	7,236	.051	53		

Japanese K Computer

K computer Specifications

CPU (SPARC64 VIIIfx)	Cores/Node	8 cores (@2GHz)
	Performance	128GFlops
	Architecture	SPARC V9 + HPC extension
	Cache	L1(I/D) Cache : 32KB/32KB L2 Cache : 6MB
	Power	58W (typ. 30 C)
	Mem. bandwidth	64GB/s.
Node	Configuration	1 CPU / Node
	Memory capacity	16GB (2GB/core)
System board(SB)	No. of nodes	4 nodes /SB
Rack	No. of SB	24 SBs/rack
System	Nodes/system	> 80,000

Inter-connect	Topology	6D Mesh/Torus
	Performance	5GB/s. for each link
	No. of link	10 links/ node
	Additional feature	H/W barrier, reduction
	Architecture	Routing chip structure (no outside switch box)
Cooling	CPU, ICC*	Direct water cooling
	Other parts	Air cooling



New Linpack run with 705,024 cores at 10.51 Pflop/s (88,128 CPUs), 12.7 MW; 29.5 hours
Fujitsu to have a 100 Pflop/s system in 2014

47

China

- First Chinese Supercomputer to use a Chinese Processor
  - Sunway BlueLight MPP
  - ShenWei SW1600 processor, 16 core, 65 nm, fabbed in China
  - 125 Gflop/s peak
  - #14 with 139,364 cores, .796 Pflop/s & 1.07 Pflop/s Peak
  - Power Efficiency 741 Mflops/W

- Coming soon, Loongson (Godson) processor
  - 8-core, 65nm Loongson 3B processor runs at 1.05 GHz, with a peak performance of 128 Gflop/s

The New York Times

China Has Homemade Supercomputer Gain

By JOHN MARKOFF
Published: October 28, 2011

China has made its first supercomputer based on Chinese microprocessor chips, an advance that surprised high-performance computing specialists in the United States.

48

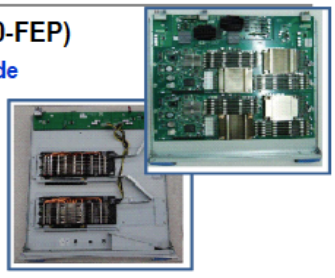
10+ Pflop/s Systems Planned in the States

- DOE Funded, Titan at ORNL, Based on Cray design w/AMD & Nvidia accelerators,
  - 20 Pflop/s, 2012
- DOE Funded, Sequoia at Lawrence Livermore Nat. Lab, Based on IBM's BG/Q,
  - 20 Pflop/s, 2012
- DOE Funded, BG/Q at Argonne National Lab, Based on IBM's BG/Q,
  - 10 Pflop/s, 2012
- NSF Funded, Blue Waters at University of Illinois UC, Based Cray XE6/XK6 hybrid,
  - 11.5 Pflop/s, 2012
- NSF Funded, U of Texas, Austin, Based on Dell/Intel MIC,
  - 10 Pflop/s, 2013

Tianhe-1A

Main configuration of TH-1A system

- 7,168 compute nodes (YH-X5670-FEP)
  - 2 six-core CPU and 1 GPU per node
  - CPU: Xeon X5670 (Westmere)
    - Processor speed - 2.93GHz
  - GPU: nVIDIA M2050
    - Connected with CPU by PCI-E
  - 32GB memory per node
  - 2U height



7168(nodes) × 2(CPU) ×2.93(GHz)× 6(Cores) ×4 =1.008PFlops

\+

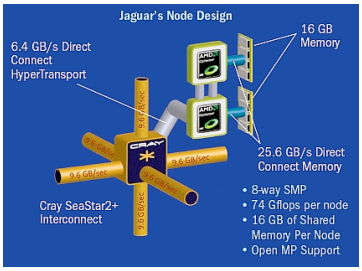
7168(nodes) × 1(GPU) ×1.15(GHz)*448(CUDA Cores) =3.692PFlops

Total: 4,701,061 GFlops

#3 ORNL's Jaguar System - Cray XT5

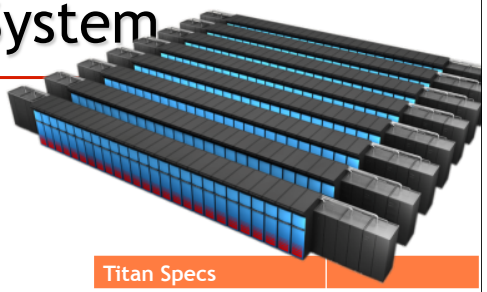
2.3 Pflop/s system with more than 224K processor cores using AMD's 6 Core chip.

Peak performance	**2.3 PF**
System memory	300 TB
Disk space	10 PB
Disk bandwidth	240+ GB/s
Interconnect bandwidth	374 TB/s



U.S. DEPARTMENT OF ENERGY | Office of Science

ORNL's "Titan" System



- Upgrade of existing Jaguar Cray XT5
- Cray Linux Environment operating system
- Gemini interconnect
  - 3-D Torus
  - Globally addressable memory
  - Advanced synchronization features
  - AMD Opteron 6200 processor (Interlagos)
- New accelerated node design using NVIDIA multi-core accelerators
  - 2011: 960 NVIDIA M2090 "Fermi" GPUs
  - 2012: 10-20 PF NVIDIA "Kepler" GPUs
- 10-20 PFlops peak performance
  - Performance based on available funds
- 600 TB DDR3 memory (2x that of Jaguar)

Titan Specs	
Compute Nodes	18,688
Login & I/O Nodes	512
Memory per node	32 GB + 6 GB
NVIDIA "Fermi" (2011)	665 GFlops
# of Fermi chips	960
NVIDIA "Kepler" (2012)	>1 TFlops
Opteron	2.2 GHz
Opteron performance	141 GFlops
Total Opteron Flops	2.6 PFlops
Disk Bandwidth	~ 1 TB/s

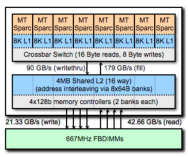
NSF Supercomputing Centers

University of Illinois - Blue Waters will be the powerhouse of the National Science Foundation's strategy to support supercomputers for scientists nationwide

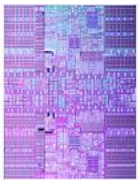
T1	**Blue Waters**	**NCSA/Illinois**	**10 Pflop/s peak; 1 Pflop/s *sustained* per second in 2012**
T2	**Kraken**	**NICS/U of Tennessee**	**1 Pflop/s peak per second**
T2	**Ranger**	**TACC/U of Texas**	**504 Tflop/s peak per second**
T3	**Campuses across the U.S.**	**Several sites**	**50-100 Tflop/s peak per second**

Today's Multicores

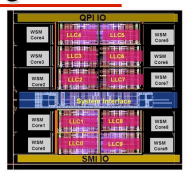
99% of Top500 Systems Are Based on Multicore



Sun Niagra2 (8 cores)

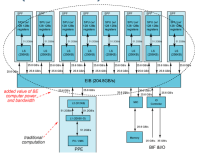


IBM Power 7 (8 cores)

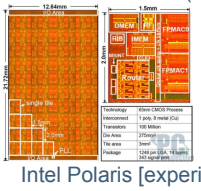


Intel Westmere (10 cores)

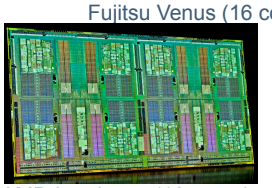
Fujitsu Venus (16 cores)



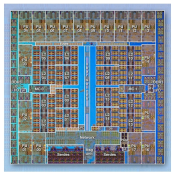
IBM Cell (9 cores)



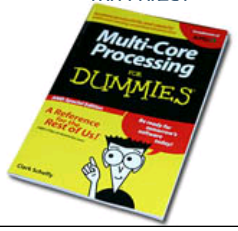
Intel Polaris [experimental]



AMD Interlagos (16 cores)

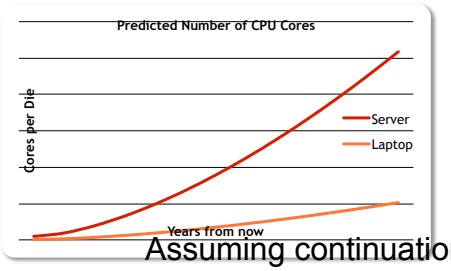


IBM BG/Q (18 cores)

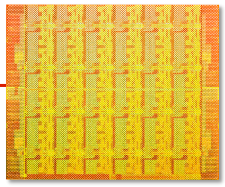


General Purpose CPU Concurrency Trends

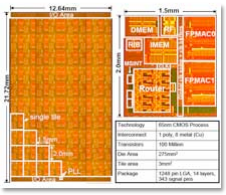
- Today
  - Typical server node chip ~ 8 cores
  - 1k node cluster ➔ 8,000 cores
  - Laptop ~ 2 cores (low power)
- By 2020
  - Typical server node chip ~400 cores
  - 1k node cluster ➔ 400,000 cores
  - Laptop ~ 100 cores (low power)



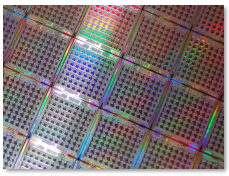
Assuming continuation of 58% historical density improvement per year



Intel SCC 48 cores



Intel 80 cores (teraflop



Tilera 100 GP cores

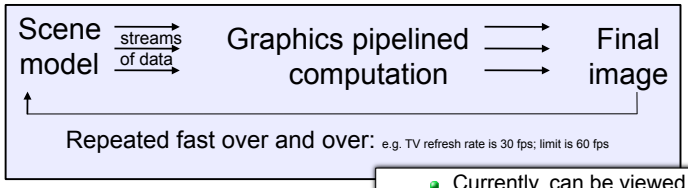
Future Computer Systems

- Most likely be a hybrid design
- Think standard multicore chips and accelerator (GPUs)
- Today accelerators are attached
- Next generation more integrated
- Intel's "Knights Corner" and "Knights Ferry" to come.
  - 48 x86 cores
- AMD's Fusion in 2011 - 2013
  - Multicore with embedded graphics ATI
- Nvidia's Project Denver plans to develop an integrated chip using ARM architecture in 2013.

56

Evolution of GPUs

GPUs: excelling in graphics rendering



- Currently, can be viewed as **multithreaded multicore vector units**

This type of computation:

- Requires **enormous computational power**
- Allows for **high parallelism**
- Needs **high bandwidth *vs* low latency**

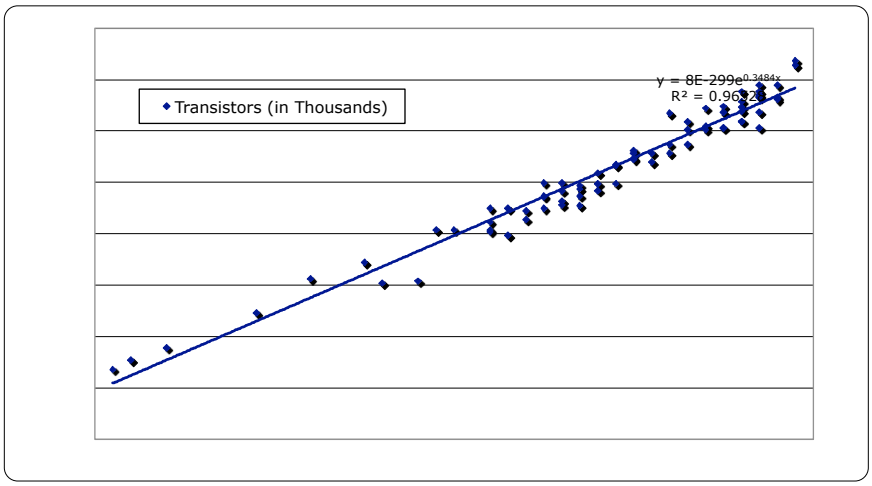
(as low latencies can be compensated with deep graphics pipeline)

Obviously, this pattern of computation is common with many other applications

Challenges of using GPUs

- **High levels of parallelism**
  Many GPU cores, serial kernel execution
  [e.g. 240 in the Nvidia Tesla; up to 512 in *Fermi* - to have concurrent kernel execution]
- **Hybrid/heterogeneous architectures**
  Match algorithmic requirements to architectural strengths
  [e.g. small, non-parallelizable tasks to run on CPU, large and parallelizable on GPU]
- **Compute *vs* communication gap**
  Exponentially growing gap; persistent challenge
  [Processor speed improves 59%, memory bandwidth 23%, latency 5.5%]
  [on all levels, e.g. a GPU Tesla C1070 (4 x C1060) has compute power o O(1,000)
  Gflop/s but GPUs communicate through the CPU using O(1) GB/s connection]

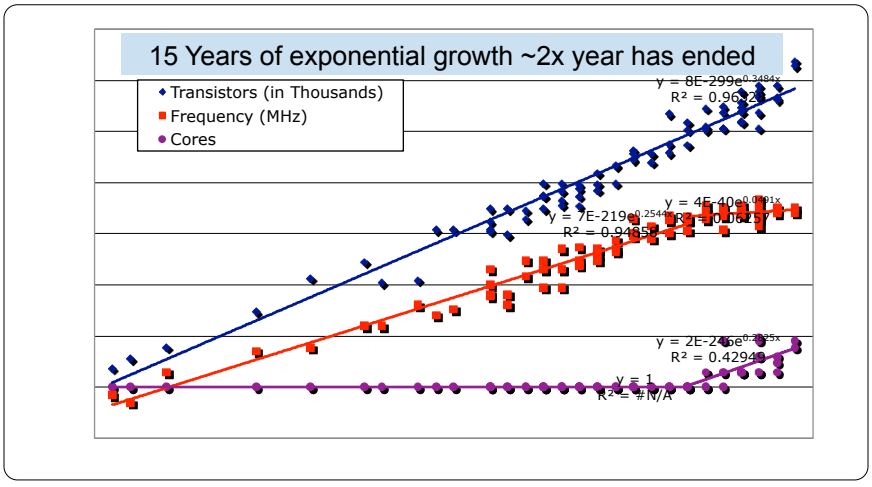
Moore's Law is Alive and Well



Data from Kunle Olukotun, Lance Hammond, Herb Sutter, Burton Smith, Chris Batten, and Krste Asanoviç

Slide from Kathy Yelick

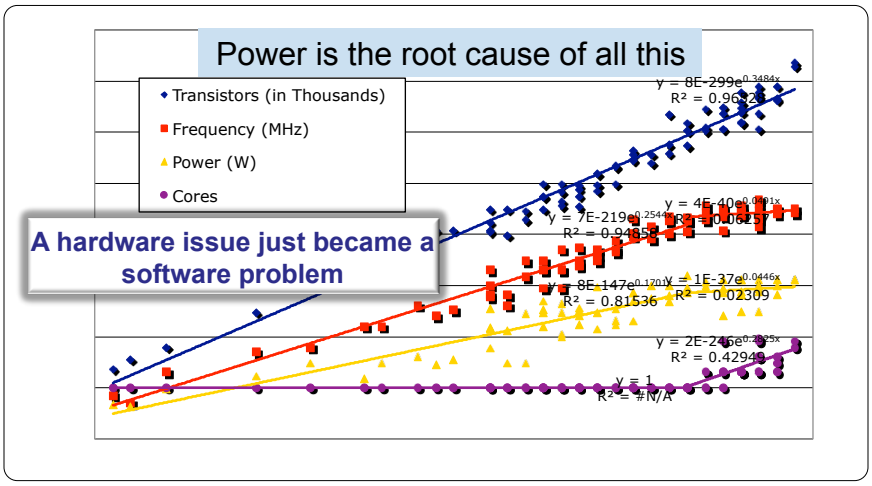
But Clock Frequency Scaling Replaced by Scaling Cores / Chip



Data from Kunle Olukotun, Lance Hammond, Herb Sutter, Burton Smith, Chris Batten, and Krste Asanoviç

Slide from Kathy Yelick

Performance Has Also Slowed, Along with Power



Data from Kunle Olukotun, Lance Hammond, Herb Sutter, Burton Smith, Chris Batten, and Krste Asanoviç

Slide from Kathy Yelick

Power Cost of Frequency

- Power ∝ Voltage^2 x Frequency (V^{2}F)
- Frequency ∝ Voltage
- Power ∝ Frequency^3

	Cores	V	Freq	Perf	Power	PE (Bops/watt)
Superscalar	1	1	1	1	1	1
"New" Superscalar	1X	1.5X	1.5X	1.5X	3.3X	0.45X

2

Power Cost of Frequency

- Power ∝ Voltage^2 x Frequency (V^{2}F)
- Frequency ∝ Voltage
- Power ∝ Frequency^3

	Cores	V	Freq	Perf	Power	PE (Bops/watt)
Superscalar	1	1	1	1	1	1
"New" Superscalar	1X	1.5X	1.5X	1.5X	3.3X	0.45X
Multicore	2X	0.75X	0.75X	1.5X	0.8X	1.88X

(Bigger # is better)

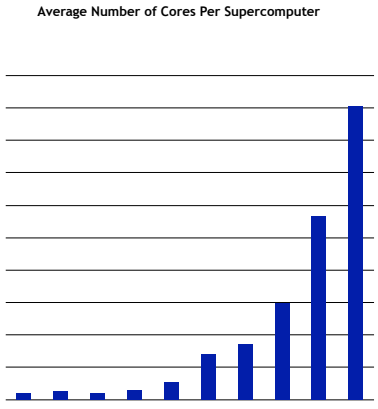
50% more performance with 20% less power

Preferable to use multiple slower devices, than one superfast device

63

Moore's Law Reinterpreted

- Number of cores per chip doubles every 2 year, while clock speed decreases (not increases).
  - Need to deal with systems with millions of concurrent threads
    - Future generation will have billions of threads!
  - Need to be able to easily replace inter-chip parallelism with intro-chip parallelism
- Number of threads of execution doubles every 2 year

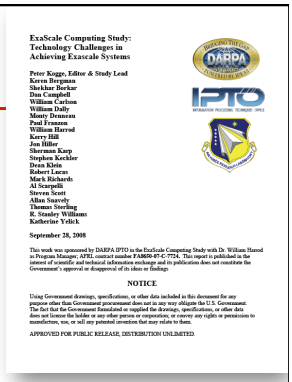


Major Changes to Software

- **Must rethink the design of our software**
  - **Another disruptive technology**
    - Similar to what happened with cluster computing and message passing
  - **Rethink and rewrite the applications, algorithms, and software**

65

Exascale Computing


ExaScale Computing Study:
Technology Challenges in
Achieving Exascale Systems

Peter Kogge, Editor & Study Lead
Keren Bergman
Shekhar Borkar
Dan Campbell
William Carlson
William Dally
Monty Denneau
Paul Franzon
William Harrod
Kerry Hill
Jon Hiller
Sherman Karp
Stephen Keckler
Dean Klein
Robert Lucas
Mark Richards
Al Scarpelli
Steven Scott
Allan Snavely
Thomas Sterling
R. Stanley Williams
Katherine Yelick

September 28, 2008

This work was sponsored by DARPA IPTO in the ExaScale Computing Study with Dr. William Harrod as Program Manager; AFRL contract number FA8650-07-C-7724. This report is published in the interest of scientific and technical information exchange and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

NOTICE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government furnished or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED.

- Exascale systems are likely feasible by 2017±2
- 10-100 Million processing elements (cores or mini-cores) with chips perhaps as dense as 1,000 cores per socket, clock rates will grow more slowly
- 3D packaging likely
- Large-scale optics based interconnects
- 10-100 PB of aggregate memory
- Hardware and software based fault management
- Heterogeneous cores
- Performance per watt — stretch goal 100 GF/watt of sustained performance $\Longrightarrow$ >> 10 - 100 MW Exascale system
- Power, area and capital costs will be significantly higher than for today's fastest systems

66

Google: exascale computing study

Factors that Necessitate Redesign of Our Software

- Steepness of the ascent from terascale to petascale to exascale
- Extreme parallelism and hybrid design
  - Preparing for million/billion way parallelism
- Tightening memory/bandwidth bottleneck
  - Limits on power/clock speed implication on multicore
  - Reducing communication will become much more intense
  - Memory per core changes, byte-to-flop ratio will change
- Necessary Fault Tolerance
  - MTTF will drop
  - Checkpoint/restart has limitations

Software infrastructure does not exist today

www.exascale.org

Conclusions

- For the last decade or more, the research investment strategy has been overwhelmingly biased in favor of hardware.
- This strategy needs to be rebalanced - barriers to progress are increasingly on the software side.
- Moreover, the return on investment is more favorable to software.
  - Hardware has a half-life measured in years, while software has a half-life measured in decades.
- High Performance Ecosystem out of balance
  - Hardware, OS, Compilers, Software, Algorithms, Applications
    - No Moore's Law for software, algorithms and applications

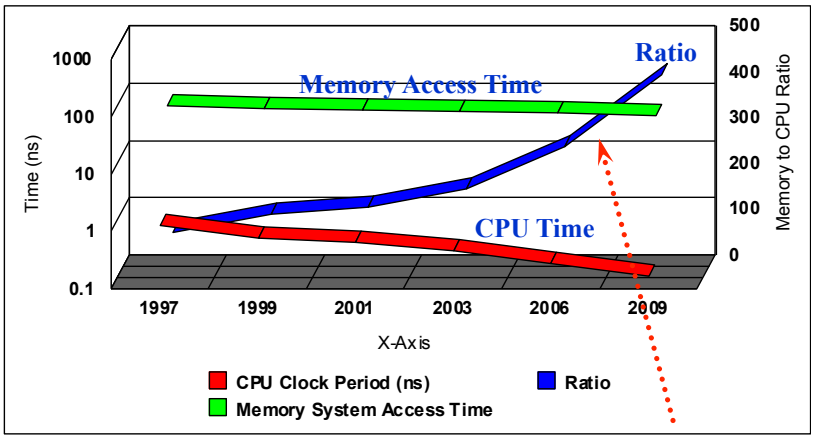
69

Why Fast Machines Run Slow

- Latency
  - Waiting for access to memory or other parts of the system
- Overhead
  - Extra work that has to be done to manage program concurrency and parallel resources the real work you want to perform
- Starvation
  - Not enough work to do due to insufficient parallelism or poor load balancing among distributed resources
- Contention
  - Delays due to fighting over what task gets to use a shared resource next. Network bandwidth is a major constraint.

70

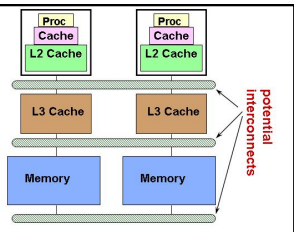
Latency in a Single System



THE *WALL*

71

Memory hierarchy



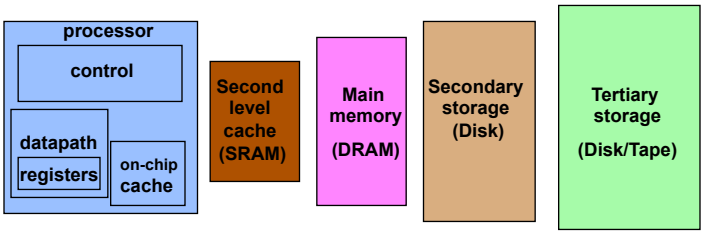
- **Typical latencies for today's technology**

Hierarchy	Processor clocks
Register	1
L1 cache	2-3
L2 cache	6-12
L3 cache	14-40
Near memory	100-300
Far memory	300-900
Remote memory	$O(10^3)$
Message-passing	$O(10^3)$-$O(10^4)$

72

Memory Hierarchy

- Most programs have a high degree of locality in their accesses
  - spatial locality: accessing things nearby previous accesses
  - temporal locality: reusing an item that was previously accessed
- Memory hierarchy tries to exploit locality

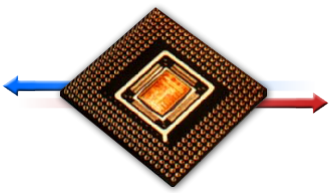


Speed	1ns	10ns	100ns	10ms	10sec
Size	B	KB	MB	GB	TB

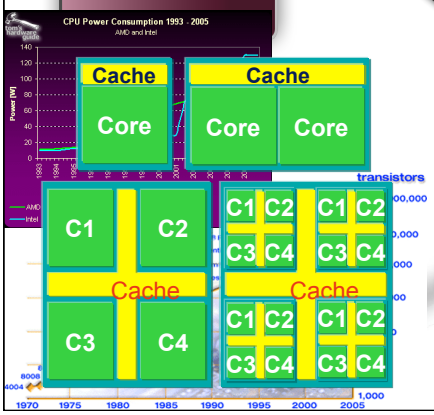
Increasing CPU Performance: A Delicate Balancing Act

Increasing the number of gates into a tight knot and decreasing the cycle time of the processor

Lower Voltage



Increase Clock Rate & Transistor Density



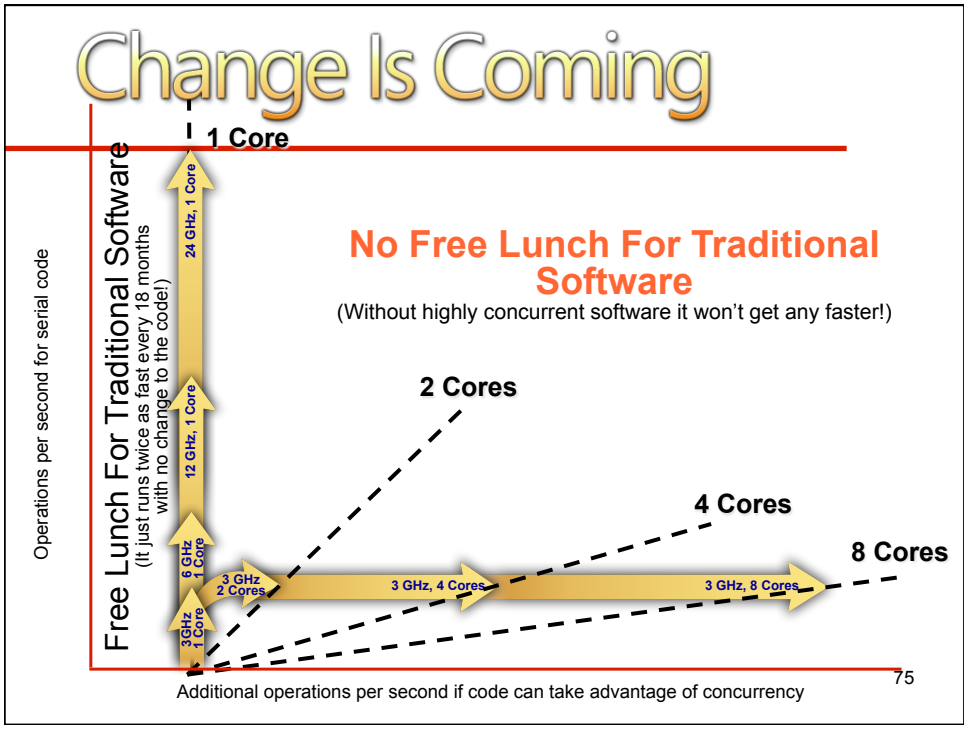
We have seen increasing number of gates on a chip and increasing clock speed.

Heat becoming an unmanageable problem, Intel Processors > 100 Watts

We will not see the dramatic increases in clock speeds in the

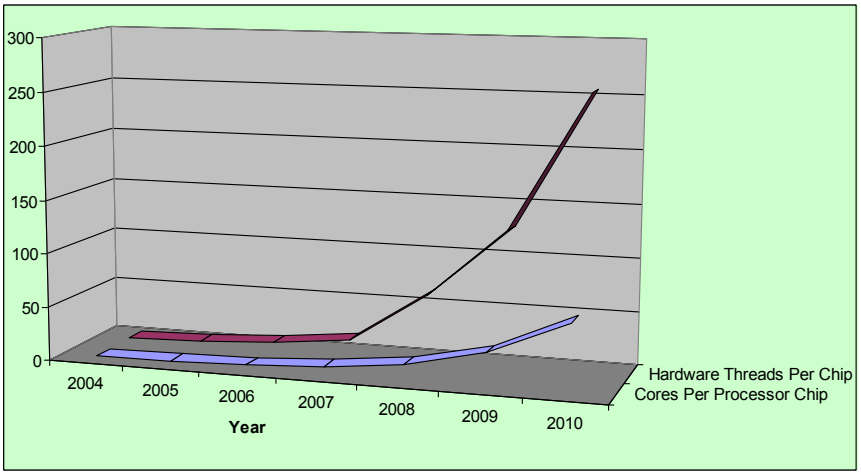
However, the number of gates on a chip will continue to in

74



CPU Desktop Trends 2004-2010

- Relative processing power will continue to double every 18 months
- 256 logical processors per chip in late 2010



Percentage of peak

- A rule of thumb that often applies
  - A contemporary RISC processor, for a spectrum of applications, delivers (i.e., sustains) 10% of peak performance
- There are exceptions to this rule, in both directions
- Why such low efficiency?
- There are two primary reasons behind the disappointing percentage of peak
  - IPC (in)efficiency
  - Memory (in)efficiency

77

IPC

- Today the theoretical IPC (instructions per cycle) is 4 in most contemporary RISC processors (6 in Itanium)
- Detailed analysis for a spectrum of applications indicates that the average IPC is 1.2–1.4
- We are leaving ~75% of the possible performance on the table…

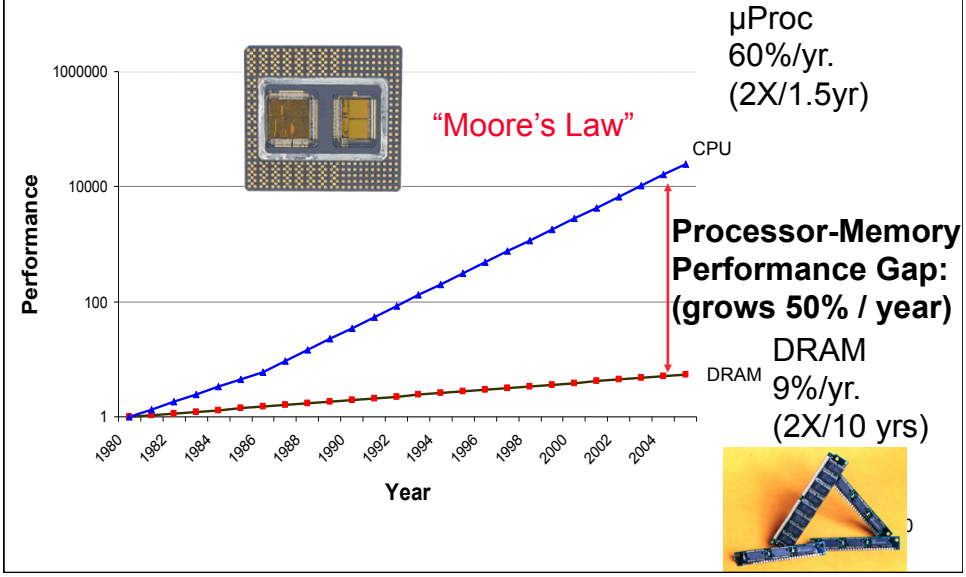
78

Memory bandwidth

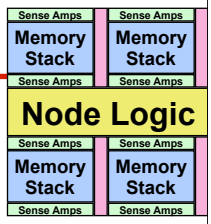
- To provide bandwidth to the processor the bus either needs to be faster or wider
- Busses are limited to perhaps 400-800 MHz
- Links are faster
  - Single-ended 0.5–1 GT/s
  - Differential: 2.5–5.0 (future) GT/s
  - Increased link frequencies increase error rates requiring coding and redundancy thus increasing power and die size and not helping bandwidth
- Making things wider requires pin-out (Si real estate) and power
  - Both power and pin-out are serious issue

79

Processor-DRAM Memory Gap



Processor in Memory (PIM)



- PIM merges logic with memory
  - Wide ALUs next to the row buffer
  - Optimized for memory throughput, not ALU utilization
- PIM has the potential of riding Moore's law while
  - greatly increasing effective memory bandwidth,
  - providing many more concurrent execution threads,
  - reducing latency,
  - reducing power, and
  - increasing overall system efficiency
- It may also simplify programming and system design

81

Principles of Parallel Computing

- Parallelism and Amdahl's Law
- Granularity
- Locality
- Load balance
- Coordination and synchronization
- Performance modeling

➡ All of these things makes parallel programming even harder than sequential programming.

82

"Automatic" Parallelism in Modern Machines

- Bit level parallelism
  - within floating point operations, etc.
- Instruction level parallelism (ILP)
  - multiple instructions execute per clock cycle
- Memory system parallelism
  - overlap of memory operations with computation
- OS parallelism
  - multiple jobs run in parallel on commodity SMPs

Limits to all of these -- for very high performance, need user to identify, schedule and coordinate parallel tasks

83

Finding Enough Parallelism

- Suppose only part of an application seems parallel
- Amdahl's law
  - let f_s be the fraction of work done sequentially, $(1-f_s)$ is fraction parallelizable
  - N = number of processors

- Even if the parallel part speeds up perfectly may be limited by the sequential part

84

Amdahl's Law

Amdahl's Law places a strict limit on the speedup that can be realized by using multiple processors. Two equivalent expressions for Amdahl's Law are given below:

$t_N = (f_p/N + f_s)t_1$ Effect of multiple processors on run time

$S = 1/(f_s + f_p/N)$ Effect of multiple processors on speedup

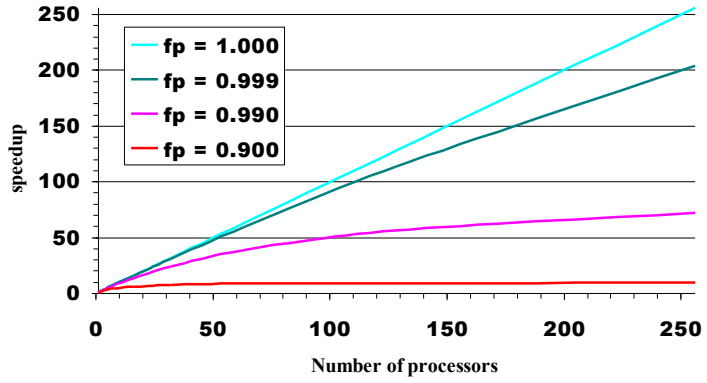
Where:

f_s = serial fraction of code
f_p = parallel fraction of code = 1 - f_s
N = number of processors

85

Illustration of Amdahl's Law

It takes only a small fraction of serial content in a code to degrade the parallel performance. It is essential to determine the scaling behavior of your code before doing production runs using large numbers of processors



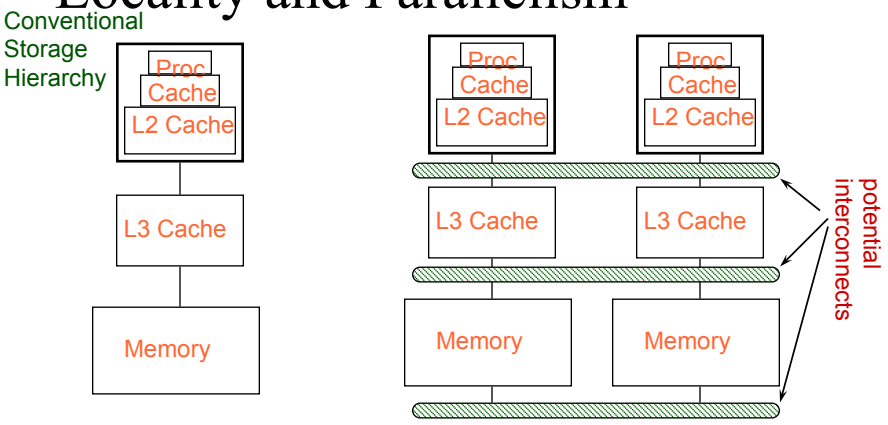
86

Overhead of Parallelism

- Given enough parallel work, this is the biggest barrier to getting desired speedup
- Parallelism overheads include:
  - cost of starting a thread or process
  - cost of communicating shared data
  - cost of synchronizing
  - extra (redundant) computation
- Each of these can be in the range of milliseconds (=millions of flops) on some systems
- Tradeoff: Algorithm needs sufficiently large units of work to run fast in parallel (I.e. large granularity), but not so large that there is not enough parallel work

87

Locality and Parallelism



- Large memories are slow, fast memories are small
- Storage hierarchies are large and fast on average
- Parallel processors, collectively, have large, fast $
  - the slow accesses to "remote" data we call "communication"
- Algorithm should do most work on local data

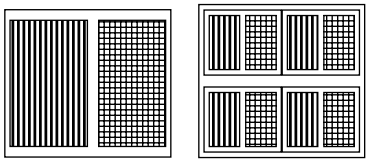
Load Imbalance

- Load imbalance is the time that some processors in the system are idle due to
  - insufficient parallelism (during that phase)
  - unequal size tasks
- Examples of the latter
  - adapting to "interesting parts of a domain"
  - tree-structured computations
  - fundamentally unstructured problems
- Algorithm needs to balance load

89

What is Ahead?

- Greater instruction level parallelism?
- Bigger caches?
- Multiple processors per chip?
- Complete systems on a chip? (Portable Systems)



- High performance LAN, Interface, and Interconnect

90

Directions

- Move toward shared memory
  - SMPs and Distributed Shared Memory
  - Shared address space w/deep memory hierarchy
- Clustering of shared memory machines for scalability
- Efficiency of message passing and data parallel programming
  - Helped by standards efforts such as MPI and HPF

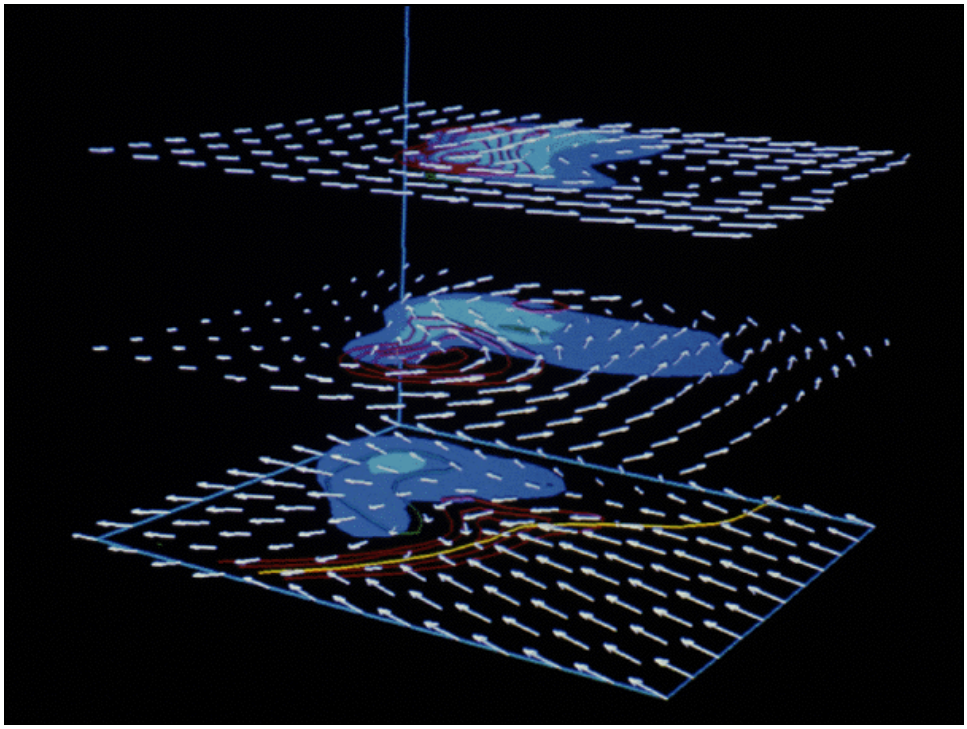
91

Virtual Environments

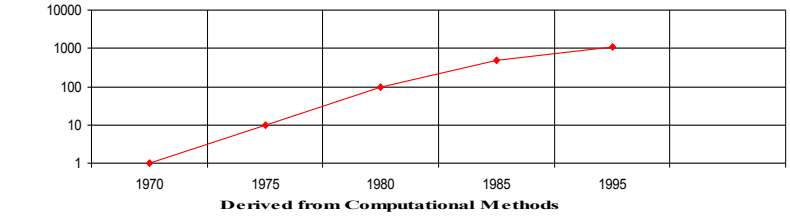
0.32E-08 0.00E+00 0.00E+00 0.00E+00 0.38E-06 0.13E-05 0.22E-05 0.33E-05 0.59E-05 0.11E-04
0.18E-04 0.23E-04 0.23E-04 0.21E-04 0.67E-04 0.38E-03 0.90E-03 0.18E-02 0.30E-02 0.43E-02
0.50E-02 0.51E-02 0.49E-02 0.44E-02 0.39E-02 0.35E-02 0.31E-02 0.28E-02 0.27E-02 0.26E-02
0.26E-02 0.27E-02 0.28E-02 0.30E-02 0.33E-02 0.36E-02 0.38E-02 0.39E-02 0.39E-02 0.38E-02
0.34E-02 0.30E-02 0.27E-02 0.24E-02 0.21E-02 0.18E-02 0.16E-02 0.14E-02 0.11E-02 0.96E-03
0.79E-03 0.63E-03 0.48E-03 0.35E-03 0.24E-03 0.15E-03 0.80E-04 0.34E-04 0.89E-05 0.16E-05
0.18E-06 0.34E-08 0.00E+00 0.00E+00 0.00E+00 0.00E+00 0.00E+00 0.00E+00 0.00E+00 0.00E+00
0.00E+00 0.00E+00 0.00E+00 0.00E+00 0.24E-08 0.00E+00 0.00E+00 0.00E+00 0.29E-06 0.11E-05
0.19E-05 0.30E-05 0.53E-05 0.96E-05 0.15E-04 0.20E-04 0.20E-04 0.18E-04 0.27E-04 0.23E-03
0.65E-03 0.14E-02 0.27E-02 0.40E-02 0.49E-02 0.51E-02 0.49E-02 0.45E-02 0.40E-02 0.35E-02
0.31E-02 0.28E-02 0.27E-02 0.26E-02 0.26E-02 0.27E-02 0.28E-02 0.30E-02 0.33E-02 0.36E-02
0.38E-02 0.39E-02 0.39E-02 0.37E-02 0.34E-02 0.30E-02 0.27E-02 0.24E-02 0.21E-02 0.18E-02
0.16E-02 0.14E-02 0.12E-02 0.98E-03 0.81E-03 0.65E-03 0.51E-03 0.38E-03 0.27E-03 0.17E-03
0.99E-04 0.47E-04 0.16E-04 0.36E-05 0.62E-06 0.41E-07 0.75E-10 0.00E+00 0.00E+00 0.00E+00
0.00E+00 0.00E+00 0.00E+00 0.00E+00 0.00E+00 0.00E+00 0.00E+00 0.00E+00 0.15E-08 0.00E+00
0.00E+00 0.00E+00 0.19E-06 0.84E-06 0.16E-05 0.27E-05 0.47E-05 0.82E-05 0.13E-04 0.17E-04
0.17E-04 0.15E-04 0.16E-04 0.10E-03 0.41E-03 0.11E-02 0.23E-02 0.37E-02 0.48E-02 0.51E-02
0.49E-02 0.45E-02 0.40E-02 0.35E-02 0.31E-02 0.28E-02 0.27E-02 0.26E-02 0.26E-02 0.27E-02
0.28E-02 0.31E-02 0.33E-02 0.36E-02 0.38E-02 0.39E-02 0.38E-02 0.36E-02 0.33E-02 0.29E-02

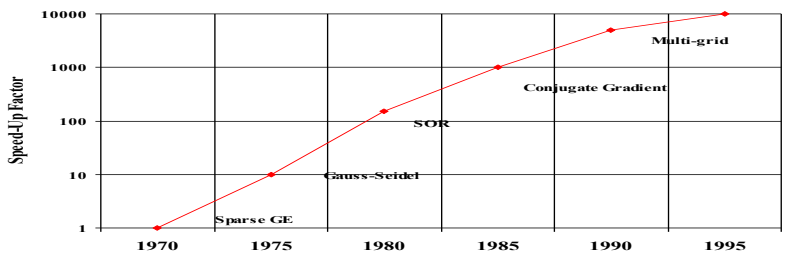
Do they make any sense?

92



Performance Improvements for Scientific Computing Problems





94

Different Architectures

- Parallel computing: single systems with many processors working on same problem
- Distributed computing: many systems loosely coupled by a scheduler to work on related problems
- Grid Computing: many systems tightly coupled by software, perhaps geographically distributed, to work together on single problems or on related problems

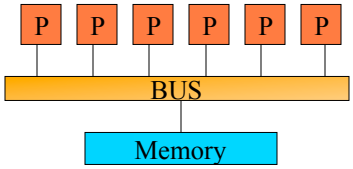
95

Types of Parallel Computers

- The simplest and most useful way to classify modern parallel computers is by their memory model:
  - shared memory
  - distributed memory

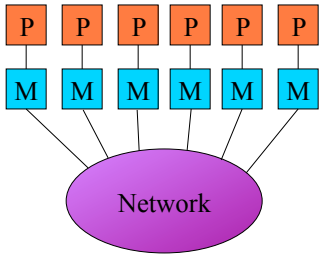
96

Shared vs. Distributed Memory



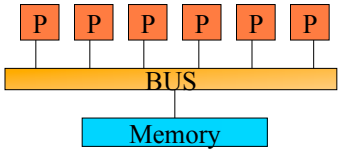
Shared memory - single address space. All processors have access to a pool of shared memory. (Ex: SGI Origin, Sun E10000)

Distributed memory - each processor has it's own local memory. Must do message passing to exchange data between processors. (Ex: CRAY T3E, IBM SP, clusters)



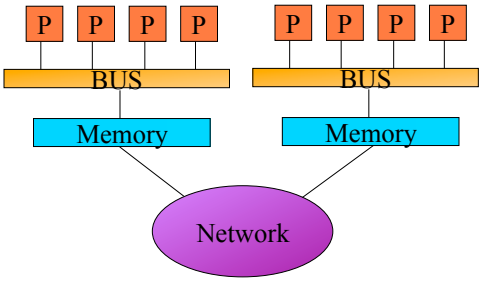
97

Shared Memory: UMA vs. NUMA



Uniform memory access (UMA): Each processor has uniform access to memory. Also known as symmetric multiprocessors (Sun E10000)

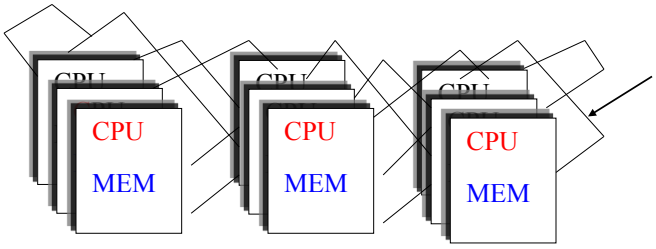
Non-uniform memory access (NUMA): Time for memory access depends on location of data. Local access is faster than non-local access. Easier to scale than SMPs (SGI Origin)



98

Distributed Memory: MPPs vs. Clusters

- Processors-memory nodes are connected by some type of interconnect network
  - Massively Parallel Processor (MPP): tightly integrated, single system image.
  - Cluster: individual computers connected by s/w



99

Processors, Memory, & Networks

- Both shared and distributed memory systems have:
  1. processors: now generally commodity RISC processors
  2. memory: now generally commodity DRAM
  3. network/interconnect: between the processors and memory (bus, crossbar, fat tree, torus, hypercube, etc.)
- We will now begin to describe these pieces in detail, starting with definitions of terms.

100

Interconnect-Related Terms

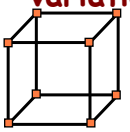
- Latency: How long does it take to start sending a "message"? Measured in microseconds.
  - (Also in processors: How long does it take to output results of some operations, such as floating point add, divide etc., which are pipelined?)
- Bandwidth: What data rate can be sustained once the message is started? Measured in Mbytes/sec.

101

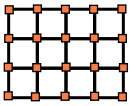
Interconnect-Related Terms

Topology: the manner in which the nodes are connected.

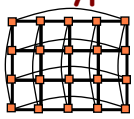
- Best choice would be a fully connected network (every processor to every other). Unfeasible for cost and scaling reasons.
- Instead, processors are arranged in some variation of a grid, torus, or hypercube.



3-d hypercube



2-d mesh



2-d torus

102

Highly Parallel Supercomputing: Where Are We?

- **Performance:**
  - Sustained performance has dramatically increased during the last year.
  - On most applications, sustained performance per dollar now exceeds that of conventional supercomputers. But...
  - Conventional systems are still faster on some applications.
- **Languages and compilers:**
  - Standardized, portable, high-level languages such as HPF, PVM and MPI are available. But ...
  - Initial HPF releases are not very efficient.
  - Message passing programming is tedious and hard to debug.
  - Programming difficulty remains a major obstacle to usage by mainstream scientist.

103

Highly Parallel Supercomputing: Where Are We?

- **Operating systems:**
  - Robustness and reliability are improving.
  - New system management tools improve system utilization. But...
  - Reliability still not as good as conventional systems.
- **I/O subsystems:**
  - New RAID disks, HiPPI interfaces, etc. provide substantially improved I/O performance. But...
  - I/O remains a bottleneck on some systems.

104

The Importance of Standards - Software

- Writing programs for MPP is hard ...
- But ... one-off efforts if written in a standard language
- Past lack of parallel programming standards ...
  - ... has restricted uptake of technology (to "enthusiasts")
  - ... reduced portability (over a range of current architectures and between future generations)
- Now standards exist: (PVM, MPI & HPF), which ...
  - ... allows users & manufacturers to protect software investment
  - ... encourage growth of a "third party" parallel software industry & parallel versions of widely used codes

105

The Importance of Standards - Hardware

- Processors
  - commodity RISC processors
- Interconnects
  - high bandwidth, low latency communications protocol
  - no de-facto standard yet (ATM, Fibre Channel, HPPI, FDDI)
- Growing demand for total solution:
  - robust hardware + usable software
- HPC systems containing all the programming tools / environments / languages / libraries / applications packages found on desktops

106

The Future of HPC

- The expense of being different is being replaced by the economics of being the same
- HPC needs to lose its "special purpose" tag
- Still has to bring about the promise of scalable general purpose computing ...
- ... but it is dangerous to ignore this technology
- Final success when MPP technology is embedded in desktop computing
- Yesterday's HPC is today's mainframe is tomorrow's workstation

107