Lecture 2:
Overview of High-Performance Computing

Jack Dongarra
University of Tennessee

Most Important Slide

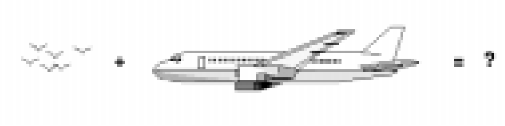
- **Netlib - software repository**
  - Go to http://www.netlib.org/
- **Register for the na-digest**
  - Go to http://www.netlib.org/na-net/
  - Register to receive the na-digest
    - http://www.netlib.org/na-net/join_mail_forw.html

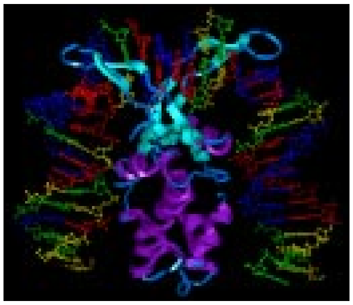
Computational Science

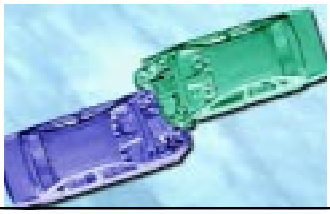
- HPC offers a new way to do science:
  - Experiment - Theory - Computation
- Computation used to approximate physical systems - Advantages include:
  - Playing with simulation parameters to study emergent trends
  - Possible replay of a particular simulation event
  - Study systems where no exact theories exist

Why Turn to Simulation?

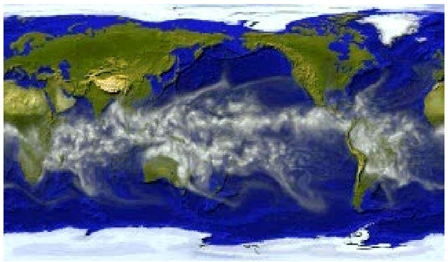
- When the problem is too . . .
  - Complex
  - Large / small
  - Expensive
  - Dangerous
- to do any other way.



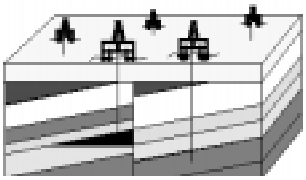


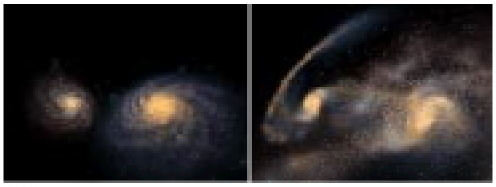


Why Turn to Simulation?



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





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

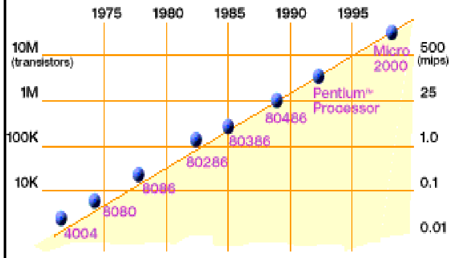
•Why Parallel Computing

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

Automotive Industry

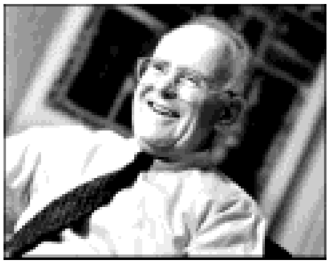
- Huge users of HPC technology;
  - Ford is 25th largest user of HPC in the world
- Main uses of simulation:
  - Aerodynamics (similar to aerospace)
  - Crash simulation
  - Metal sheet formation
  - Noise/vibration optimization
  - Traffic simulation
- Main benefits:
  - Reduced time to market of new cars
  - Increased quality
  - Reduced need to build prototypes
  - more efficient & integrated manufacturing processes

Technology Trends: Microprocessor Capacity



2X transistors/Chip Every 1.5 years

Called "Moore's Law"

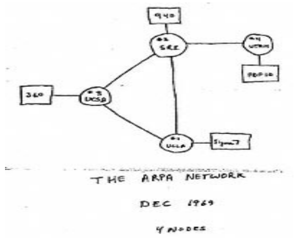


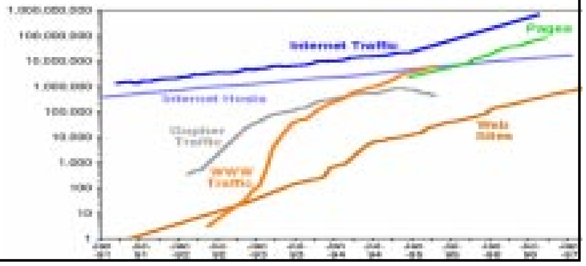
Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months.

Microprocessors have become smaller, denser, and more powerful.
Not just processors, bandwidth, storage, etc

Internet – 4th Revolution in Telecommunications

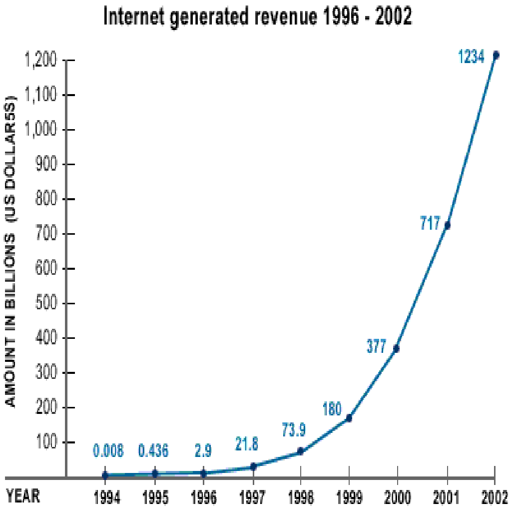
- Telephone, Radio, Television
- Growth in Internet outstrips the others
- Exponential growth since 1985
- Traffic doubles every 100 days





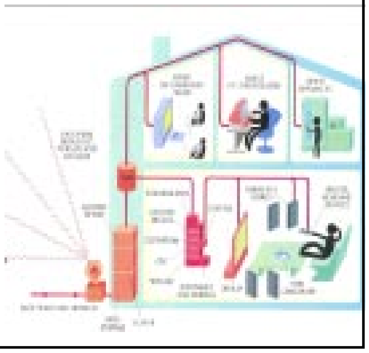
The Web Phenomenon

- 90 – 93 Web invented
- U of Illinois Mosaic released March 94, ~ 0.1% traffic
- September 93 ~ 1% traffic w/200 sites
- June 94 ~ 10% of traffic w/2,000 sites
- Today 60% of traffic w/2,000,000 sites
- Every organization, company school

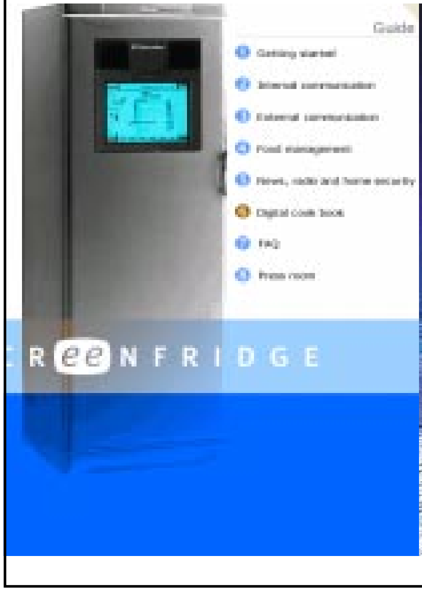


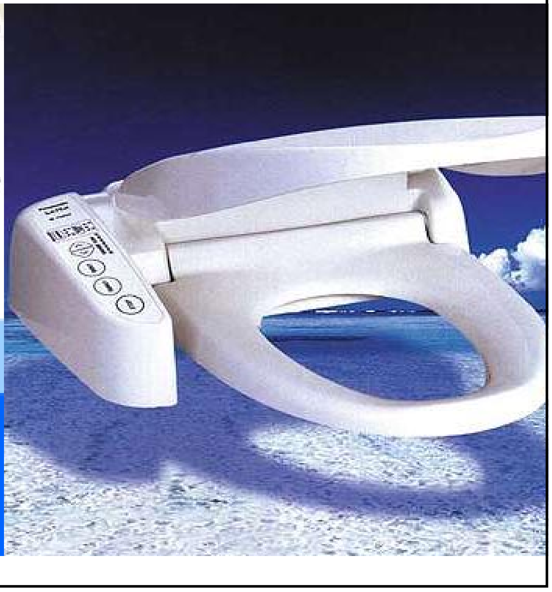
Peer to Peer Computing

- Peer-to-peer is a style of networking in which a group of computers communicate directly with each other.
- Wireless communication
- Home computer in the utility room, next to the water heater and furnace.
- Web tablets
- Imbedded computers in things all tied together.
  - Books, furniture, milk cartons, etc
- Smart Appliances
  - Refrigerator, scale, etc

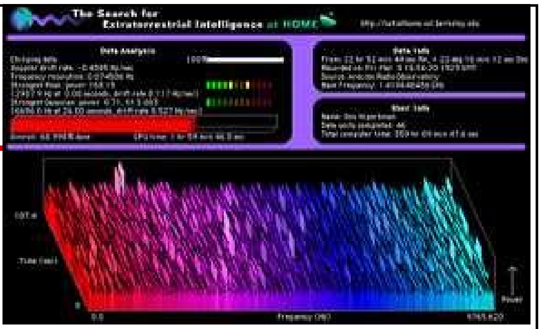


Internet On Everything



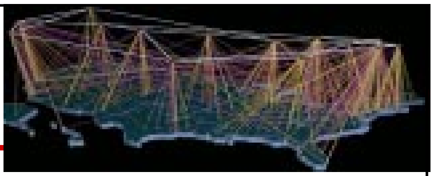


SETI@home



- Use thousands of Internet-connected PCs to help in the search for extraterrestrial intelligence.
- Uses data collected with the Arecibo Radio Telescope, in Puerto Rico
- When their computer is idle or being wasted this software will download a 300 kilobyte chunk of data for analysis.
- The results of this analysis are sent back to the SETI team, combined with the crunched data from the many thousands of other SETI@home participants.

Next Generation Web

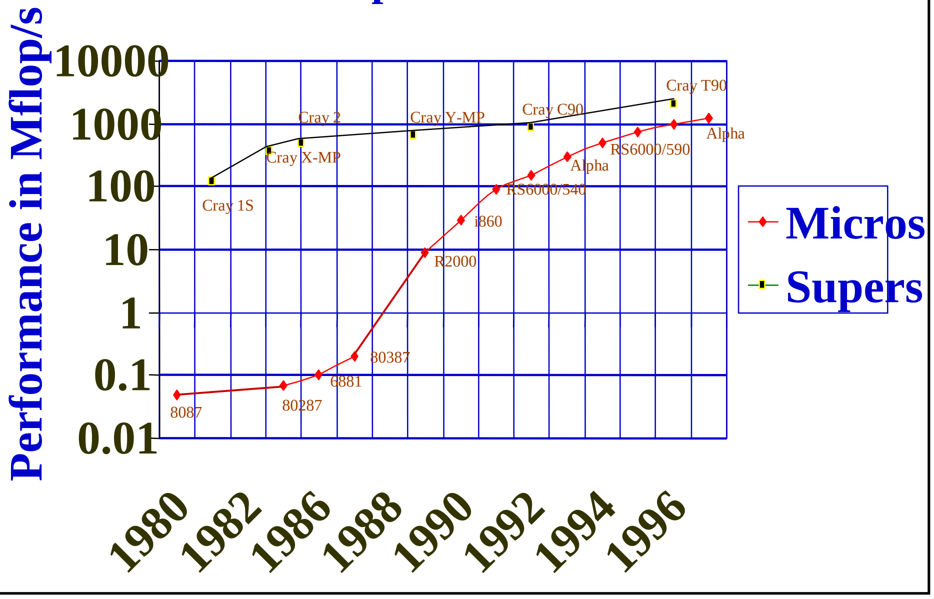


- To treat CPU cycles and software like commodities.
- Enable the coordinated use of geographically distributed resources – in the absence of central control and existing trust relationships.
- Computing power is produced much like utilities such as power and water are produced for consumers.
- Users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

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.

Growth of Microprocessor Performance



Other Examples: Sony PlayStation2

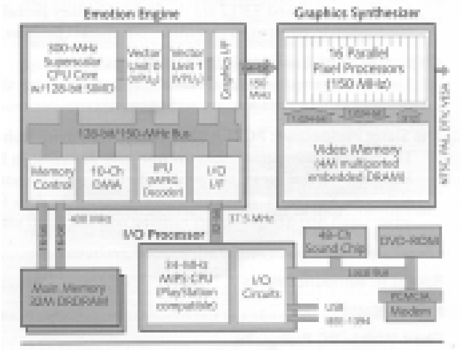


Figure 1. PlayStation 2000 employs an unprecedented level of parallelism to achieve workstation-class 3D performance.

Figure 2. PlayStation 2000 screenshot. (Source: Namco)

- Emotion Engine: 6.2 Gflop/s, 75 million polygons per second (Microprocessor Report, 13:5)
  - Superscalar MIPS core + vector coprocessor + graphics/DRAM
  - Claim: "*Toy Story*" realism brought to games
  - About $250

Sony PlayStation2 Export Limits?

svtech/news/breaking/merc/docs/060772.htm

Japan limits Playstation2 export, fears military use

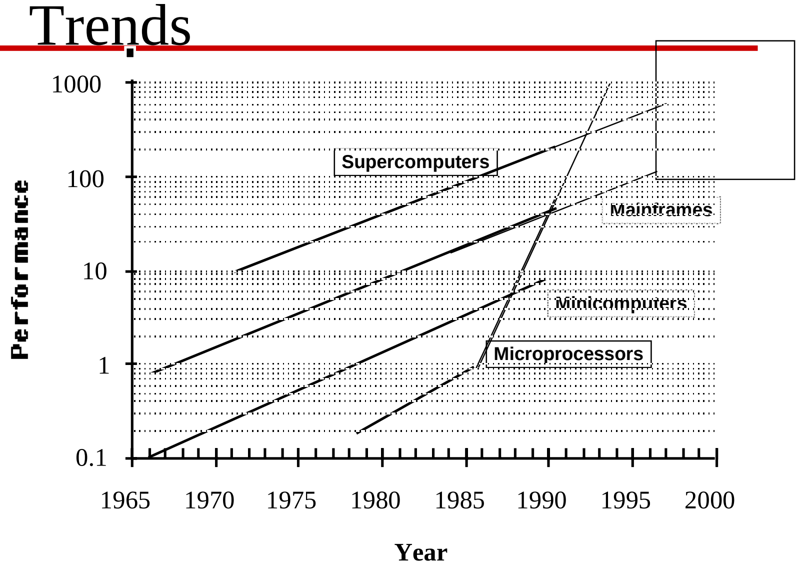
TOKYO, April 16 (Reuters) - Japan has slapped export controls on Sony Corp's new, hugely popular Playstation2 video game because the machine is so sophisticated it could be used for military purposes, media said on Sunday.

The hit home game machine, which includes a digital video disc (DVD) player and will eventually offer Internet access, is Sony's most profitable product. The company said it had shipped 1.4 million in the month after the game's March 4 launch.

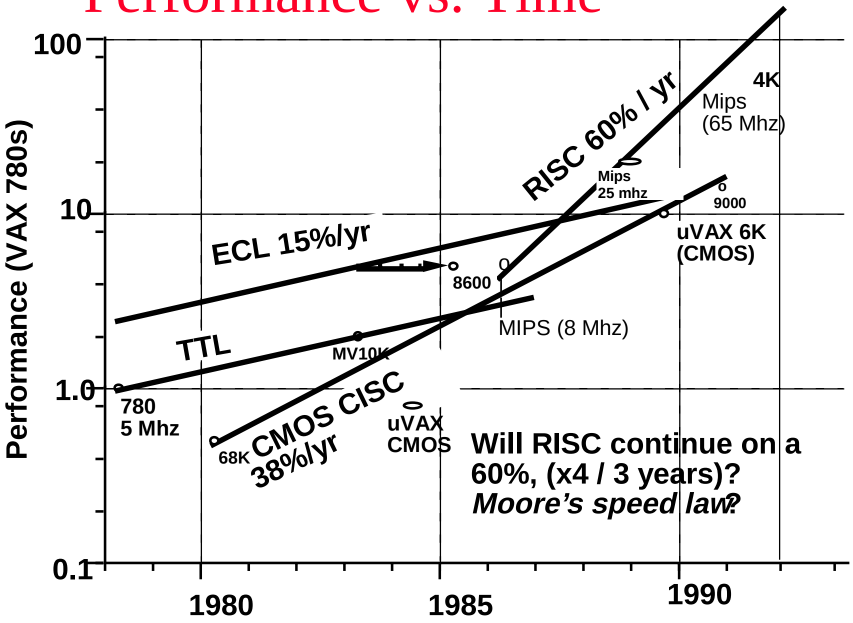
The console and its eight-megabyte memory card have been designated as ``general-purpose products related to conventional weapons'' because they contain components that could be used for military devices such as missile guidance systems, Kyodo news agency

---

Processor Performance Trends

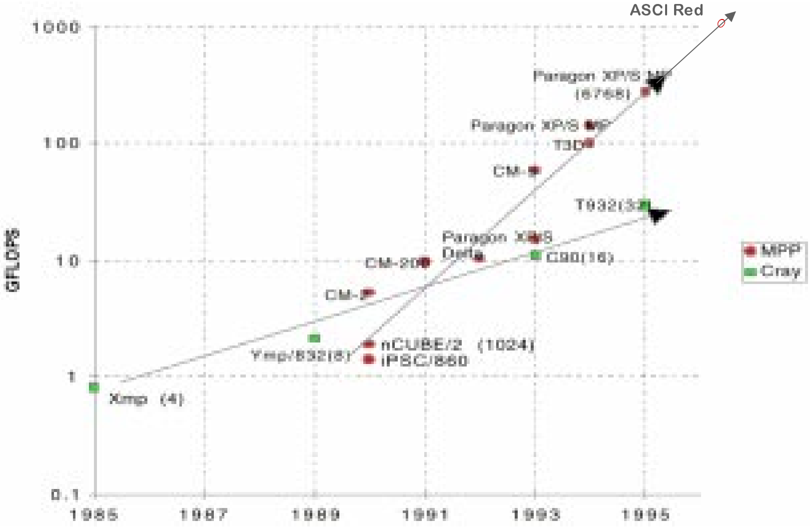


Performance vs. Time



Will RISC continue on a 60%, (x4 / 3 years)?
Moore's speed law?

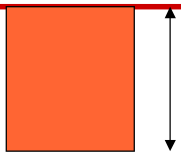
Trends in Computer Performance



Where Has This Performance Improvement Come From?

- Technology?
- Organization?
- Instruction Set Architecture?
- Software?
- Some combination of all of the above?

How fast can a serial computer be?



1 Tflop 1 TB
sequential
machine

r = .3 mm

- Consider the 1 Tflop sequential machine
  - data must travel some distance, r, to get from memory to CPU
  - to get 1 data element per cycle, this means 10^{12} times per second at the speed of light, $c = 3 \times 10^8$ m/s
  - so $r < c/10^{12} = .3$ mm
- Now put 1 TB of storage in a .3 mm^2 area
  - each word occupies about 3 $Angstroms^2$, the size of a small atom

Microprocessor Transistors & Paralle

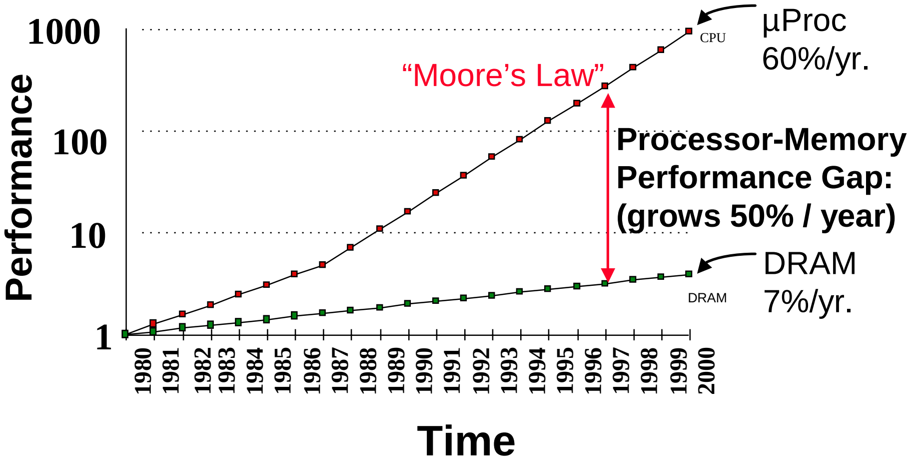
Thread-Level
Parallelism?

Instruction-Level
Parallelism

Bit-Level
Parallelism

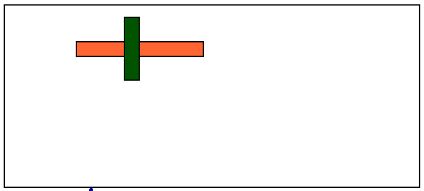
Name

Processor-DRAM Gap (latency)



1st Principles

- What happens when the feature size shrinks by a factor of x ?



- Clock rate goes up by x
  - actually less than x, because of power consumption
- Transistors per unit area goes up by x^2
- Die size also tends to increase
  - typically another factor of ~x
- Raw computing power of the chip goes up by ~ x^4 !
  - of which x^3 is devoted either to parallelism or locality

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.

"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

Finding Enough Parallelism

- Suppose only part of an application seems parallel
- Amdahl's law
  - let s be the fraction of work done sequentially, s (1-s) is fraction parallelizable
  - P = number of processors
- Even if the parallel part speeds up perfectly may be limited by the sequential part

$$\text{Speedup}(P) = \text{Time}(1)/\text{Time}(P)$$

$$<= 1/(s + (1-s)/P)$$

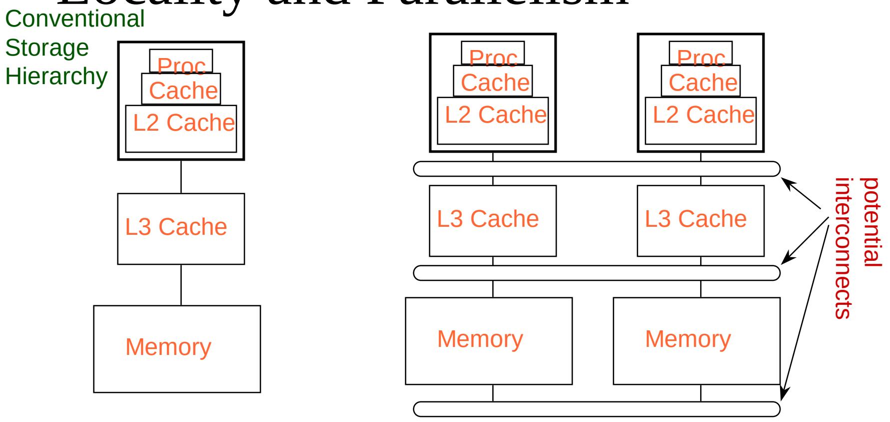
$$<= 1/s$$

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

---

Locality and Parallelism

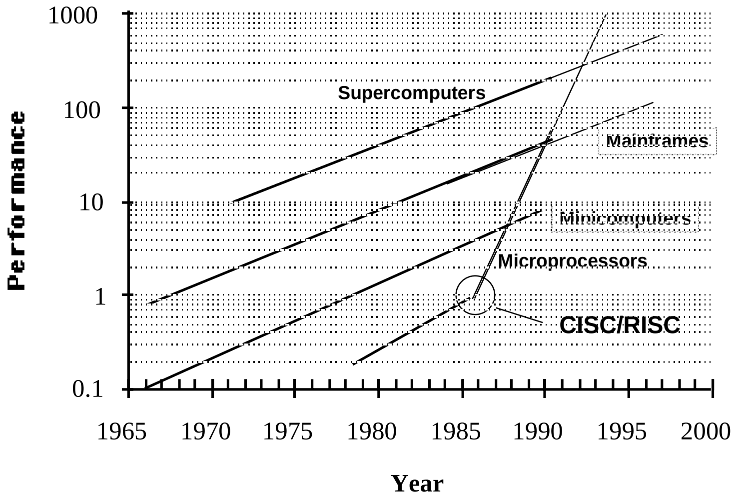


- Large memories are slow, fast memories are small
- Storage hierarchies are large and fast <u>on average</u>
- Parallel processors, collectively, have large, fast $
  - the slow accesses to "remote" data we call "communication"
- Algorithm should do most work on local data

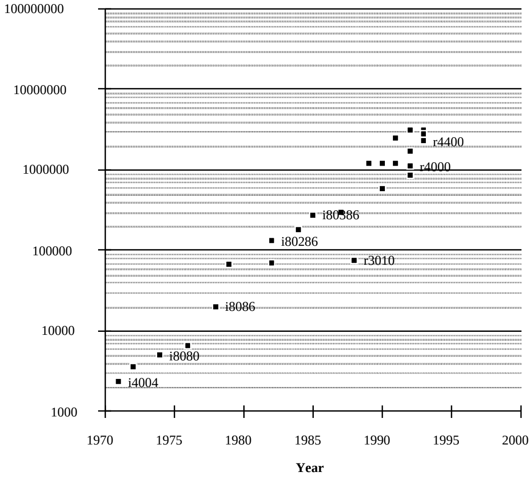
Load Imbalance

- 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

Performance Trends Revisited (Architectural Innovation)



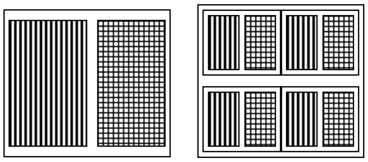
Performance Trends Revisited (Microprocessor Organization)



- Bit Level Parallelism
- Pipelining
- Caches
- Instruction Level Parallelism
- Out-of-order Xeq
- Speculation
- . . .

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*

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

Performance Numbers on RISC Processors

- Using Linpack Benchmark

Machine	MHz	Linpack n=100 Mflop/s	Ax=b n=1000 Mflop/s	Peak Mflop/s
DEC Alpha	612	287(23%)	764 (59%)	1224
IBM RS/397	160	315 (49%)	532 (83%)	640
HP PA (Expl V)	240	203 (21%)	731 (76%)	960
SGI Origin 2K	195	114 (29%)	344 (88%)	390
SUN Ultra II	336	154(23%)	461 (69%)	672
Intel Pent. P6	200	63 (31%)	97 (49%)	200
Cray T90	454	705 (39%)	1603 (89%)	1800
Cray C90	238	387 (41%)	902 (95%)	952
Cray Y-MP	166	161 (48%)	324 (97%)	333

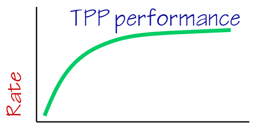
High Performance Computers

- ~ 20 years ago
  - 1x10^6 Floating Point Ops/sec (Mflop/s)
    - Scalar based
- ~ 10 years ago
  - 1x10^9 Floating Point Ops/sec (Gflop/s)
    - Vector & Shared memory computing, bandwidth aware
    - Block partitioned, latency tolerant
- ~ Today
  - 1x10^{12} Floating Point Ops/sec (Tflop/s)
    - Highly parallel, distributed processing, message passing, network based
    - data decomposition, communication/computation
- ~ 10 years away
  - 1x10^{15} Floating Point Ops/sec (Pflop/s)
    - Many more levels MH, combination/grids&HPC
    - More adaptive, LT and bandwidth aware, fault tolerant, extended precision, attention to SMP nodes

TOP500

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

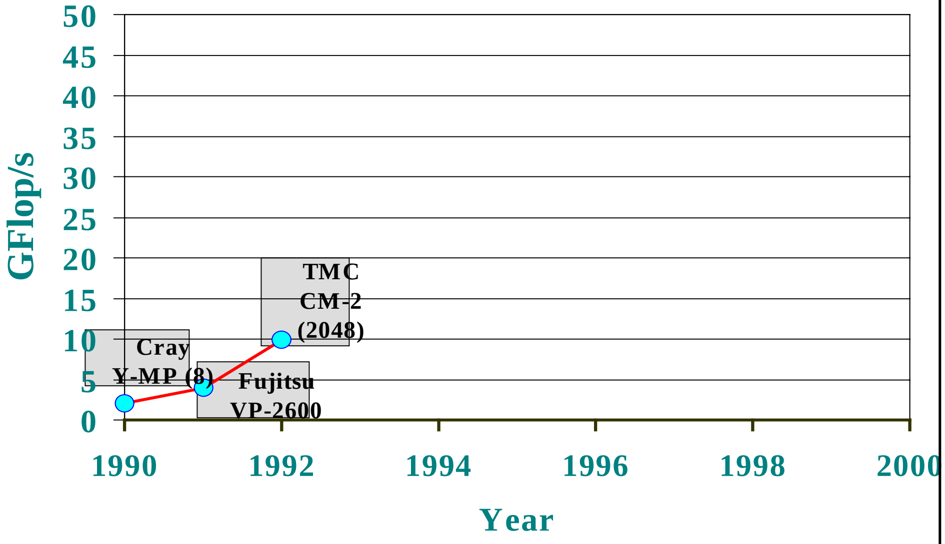
Ax=b, dense problem



- Updated twice a year
  SC‘xy in the States in November
  Meeting in Mannheim, Germany in June

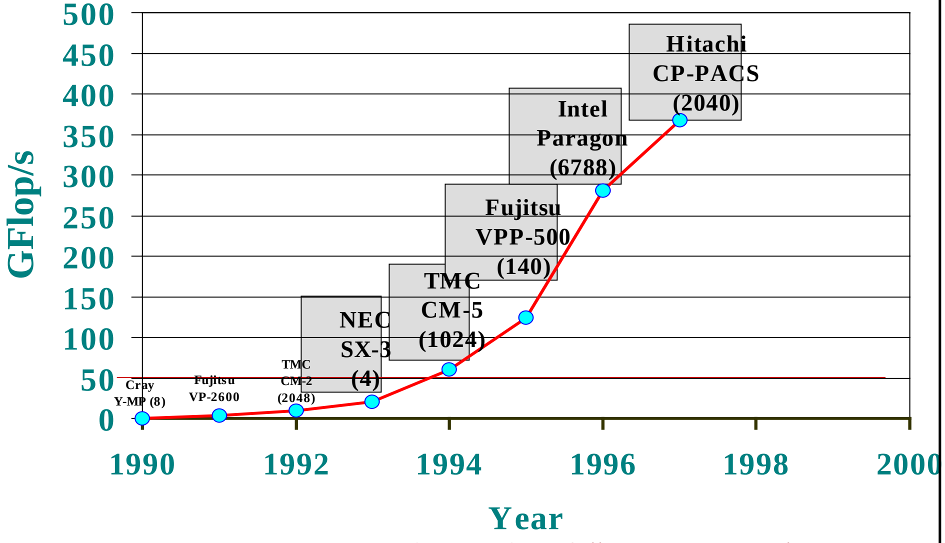
All data available from **www.top500.org**

Fastest Computer Over Time

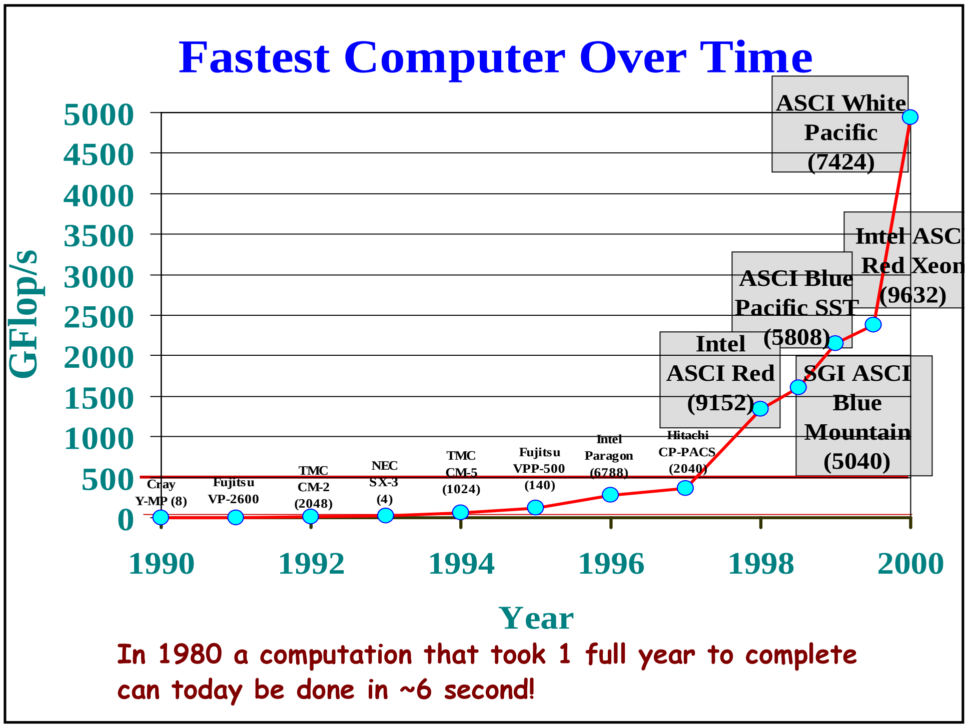


In 1980 a computation that took 1 full year to complete can now be done in ~ 50 minutes!

Fastest Computer Over Time

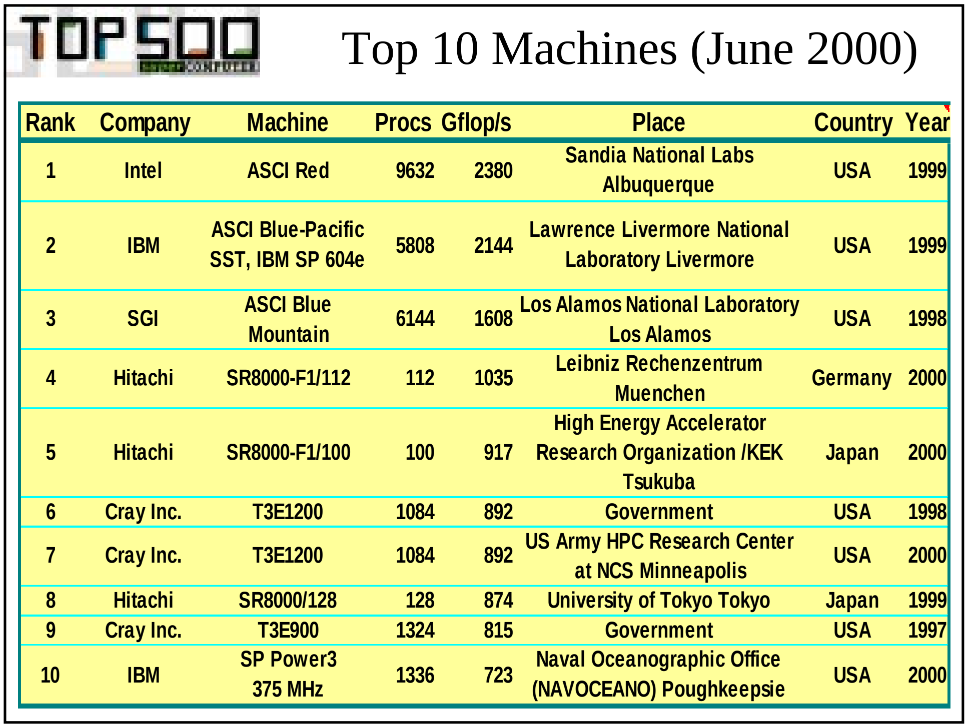


In 1980 a computation that took 1 full year to complete can now be done in ~ 1.3 minutes!



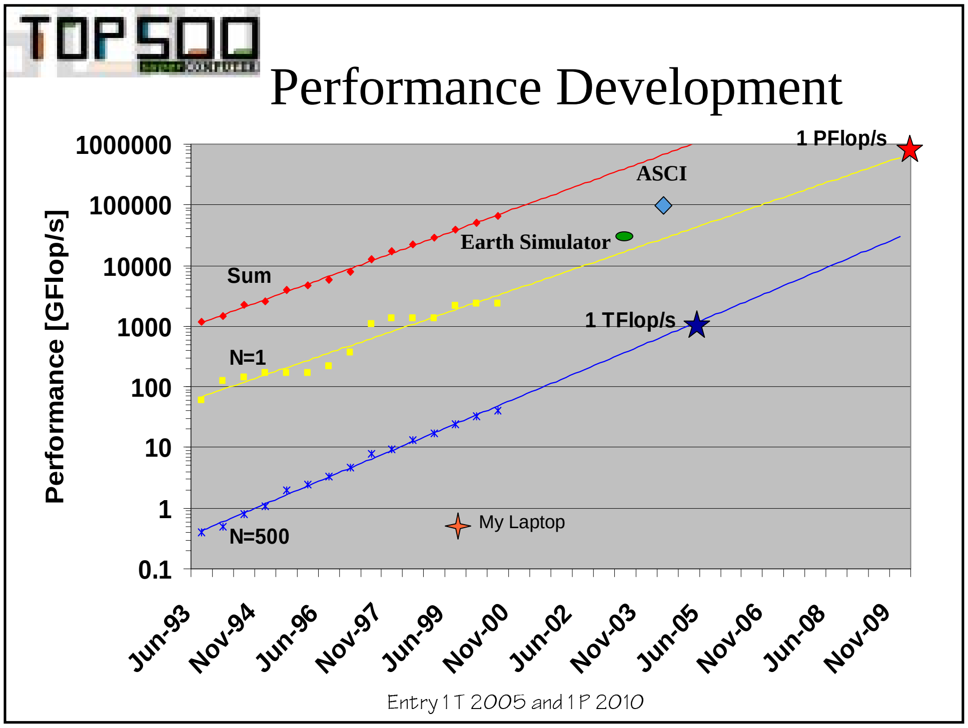
Fastest Computer Over Time

In 1980 a computation that took 1 full year to complete can today be done in ~6 second!



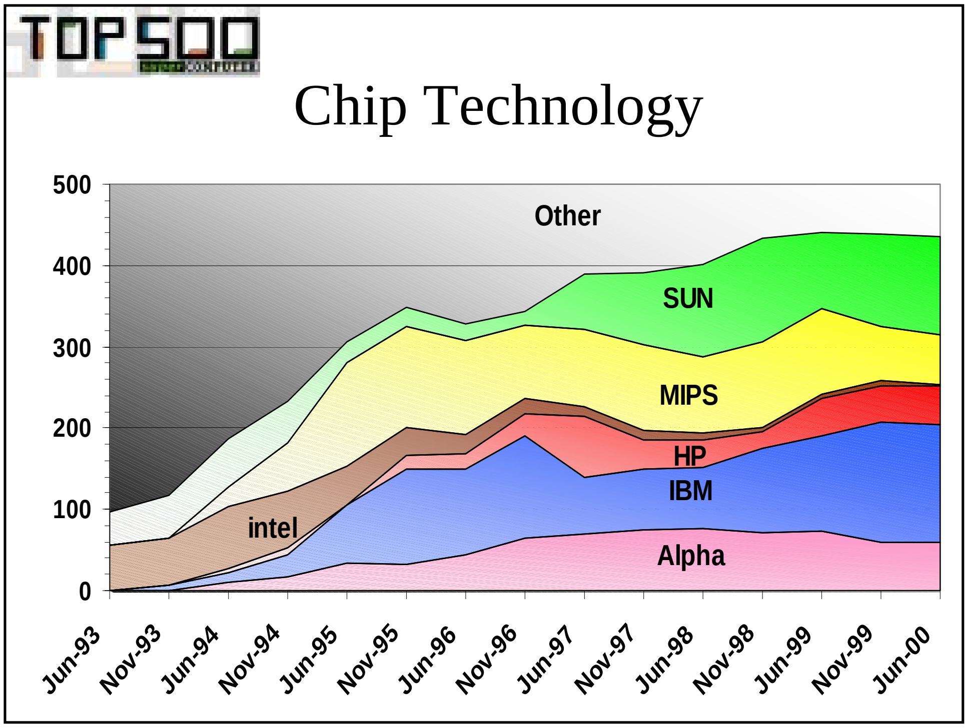
Top 10 Machines (June 2000)

Rank	Company	Machine	Procs	Gflop/s	Place	Country	Year
1	Intel	ASCI Red	9632	2380	Sandia National Labs Albuquerque	USA	1999
2	IBM	ASCI Blue-Pacific SST, IBM SP 604e	5808	2144	Lawrence Livermore National Laboratory Livermore	USA	1999
3	SGI	ASCI Blue Mountain	6144	1608	Los Alamos National Laboratory Los Alamos	USA	1998
4	Hitachi	SR8000-F1/112	112	1035	Leibniz Rechenzentrum Muenchen	Germany	2000
5	Hitachi	SR8000-F1/100	100	917	High Energy Accelerator Research Organization /KEK Tsukuba	Japan	2000
6	Cray Inc.	T3E1200	1084	892	Government	USA	1998
7	Cray Inc.	T3E1200	1084	892	US Army HPC Research Center at NCS Minneapolis	USA	2000
8	Hitachi	SR8000/128	128	874	University of Tokyo Tokyo	Japan	1999
9	Cray Inc.	T3E900	1324	815	Government	USA	1997
10	IBM	SP Power3 375 MHz	1336	723	Naval Oceanographic Office (NAVOCEANO) Poughkeepsie	USA	2000



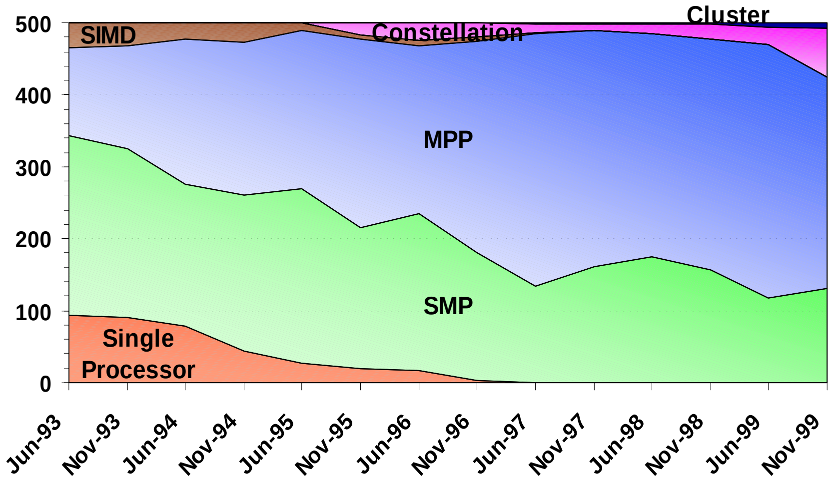
Performance Development

Entry 1 T 2005 and 1 P 2010



Chip Technology

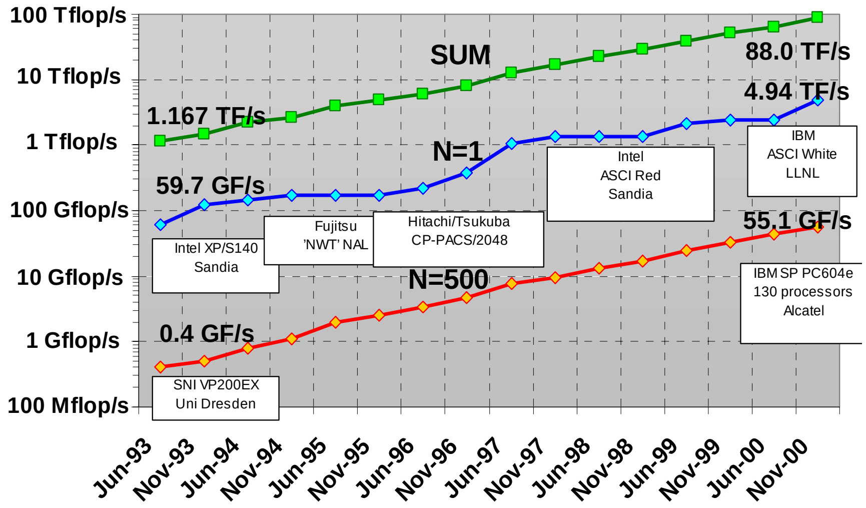
Architectures



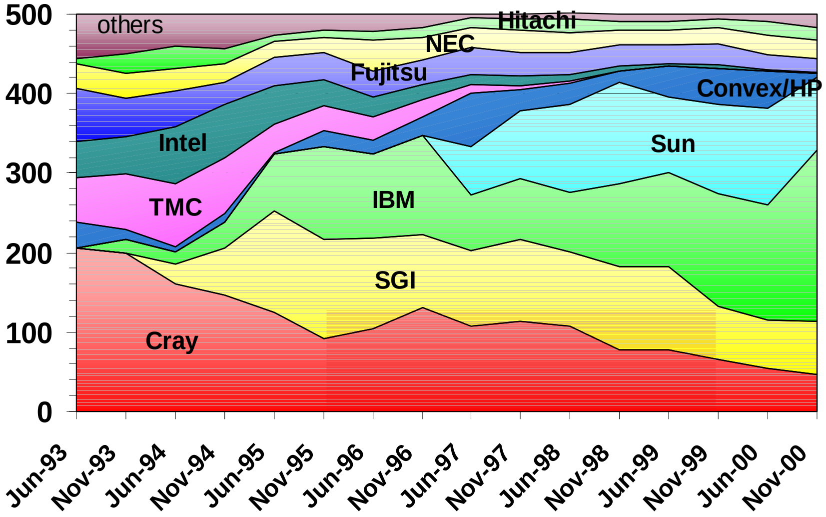
The Maturation of Highly Parallel Technology

- Affordable parallel systems now out-perform the best conventional supercomputers.
- Performance per dollar is particularly favorable.
- The field is thinning to a few very capable systems.
- Reliability is greatly improved.
- Third-party scientific and engineering applications are appearing.
- Business applications are appearing.
- Commercial customers, not just research labs, are acquiring systems.

Performance Development



Manufacturer



Top500 Conclusions

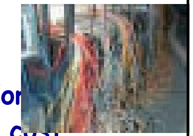
- **Microprocessor based supercomputers have brought a major change in accessibility and affordability.**
- **MPPs continue to account of more than half of all installed high-performance computers worldwide.**

High-Performance Computing Directions: Beowulf-class PC Clusters

Definition:

- COTS PC Nodes
  - Pentium, Alpha, PowerPC, SMP
- COTS LAN/SAN Interconnect
  - Ethernet, Myrinet, Giganet, ATM
- Open Source Unix
  - Linux, BSD
- Message Passing Computing
  - MPI, PVM
  - HPF

Advantages:



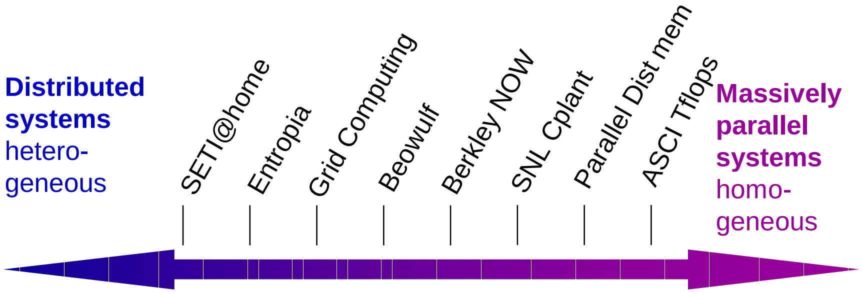
- Best price-perfor
- Low entry-level cos
- Just-in-place configuration
- Vendor invulnerable
- Scalable
- Rapid technology tracking

Enabled by PC hardware, networks and operating system achieving capabilities of scientific workstations at a fraction of the cost and availability of industry standard message passing libraries.

Clusters on the Top500

Rank	Manufacturer	Computer	Rmax	Installation Site	Country	Year	Area of Installation	# Proc	Rpeak
50	Sun	HPC 4500 Cluster	272.1	Sun Burlington	USA	1998	Vendor	720	483.84
62	Self-made	CPlant Cluster	232.6	Sandia National Laboratories Albuquerque	USA	1999	Research	580	580
187	Sun	HPC 10000 400 MHz Cluster	68.77	KT Freetel Seoul	Korea	1999	Industry Telecomm	110	88
207	Self-made	NT Supercluster	62.59	NCSA Urbana-Champaign	USA	1999	Research	256	140.8
237	Compaq	AlphaServer Cluster	61.3	Institute of Physical and Chemical Res. (RIKEN) Wako	Japan	1999	Research	140	140
271	SGI	ORIGIN 2000 250 MHz - Eth-Cluster	54.68	The Sabre Group Ft Worth	USA	1999	Industry Transportation	448	224
272	SGI	ORIGIN 2000 250 MHz - Eth-Cluster	54.68	America On Line (AOL)	USA	1999	Industry WWW	320	160
273	SGI	ORIGIN 2000 250 MHz - Eth-Cluster	54.68	Industrial Light & Magic	USA	1999	Industry Image Proc./Rendering	224	112
364	Self-made	Avalon Cluster	48.6	Los Alamos National Laboratory/CNLS Los Alamos	USA	1998	Academic	140	140.4
366	SGI	ORIGIN 2000 300 MHz - Eth-Cluster	48.33	Industrial Light & Magic	USA	1999	Industry Image Proc./Rendering	128	76.8
367	SGI	ORIGIN 2000 300 MHz - Eth-Cluster	48.33	Sikorsky Stratford	USA	2000	Industry Aerospace	128	76.8
368	SGI	ORIGIN 2000 300 MHz - Eth-Cluster	48.33	Toshiba Tokyo	Japan	2000	Industry Electronics	128	76.8
381	Self-made	AC3 Velocity/MS NT4	47.38	Cornell Theory Center Ithaca	USA	1999	Academic	256	126
394	Sun	HPC 10000 400 MHz Cluster	45.46	Boeing IDS Group Orange County	USA	1999	Industry Aerospace	72	57.6

Distributed and Parallel Systems



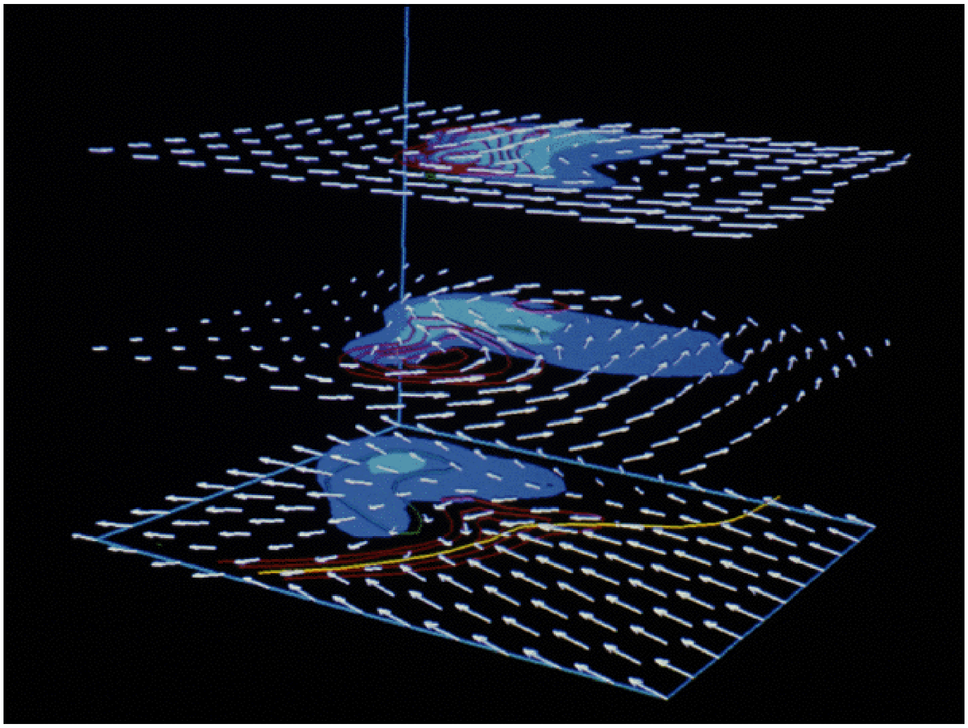
- Gather (unused) resources
- Steal cycles
- System SW manages resources
- System SW adds value
- 10% - 20% overhead is OK
- Resources drive applications
- Time to completion is not critical
- Time-shared

- 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

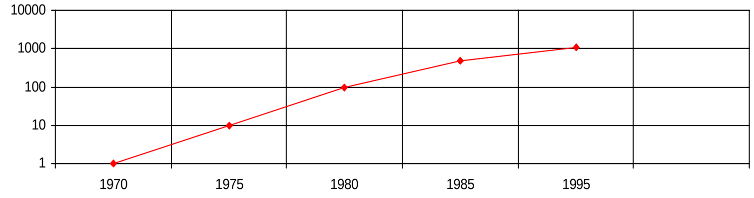
Virtual Environments

```
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
```

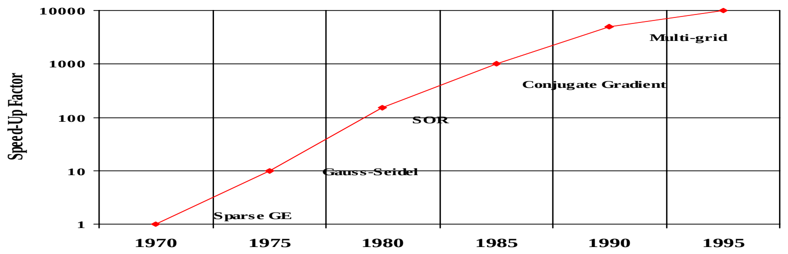
Do they make any sense?



Performance Improvements for Scientific Computing Problems



Derived from Computational Methods



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.

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.

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

The Importance of Standards - Hardware

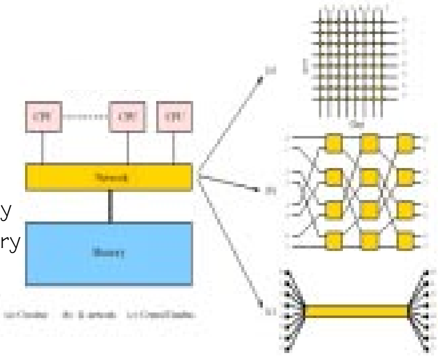
- Processors
  - commodity RISC processors
- Interconnects
  - high bandwidth, low latency communications protocol
  - no de-facto standard yet (ATM, Fibre Channel, HPPI, FDDI)
- Growing demand for total solution:
  - robust hardware + usable software
- HPC systems containing all the programming tools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

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

Different Architectures

- **SIMD**
  - **Vector Computers**
- **MIMD**
  - **Shared Memory**
    - » NEC SX Vector Computer
    - » Cray J90/T90 Vector Computer
  - **Distributed Shared Memory**
    - » SGI Origin Distributed Shared Memory
    - » HP-Convex Distributed Shared Memory
    - » SUN Enterprise Shared Memory
  - **Distributed Memory**
    - » Cray T3E Distributed Memory
    - » IBM SP Distributed Memory
    - » Fujitsu VPP Vector Distributed Memory
    - » Cluster of Processors (COWS, NOWS, Beowulf)

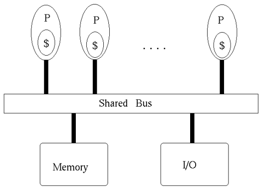


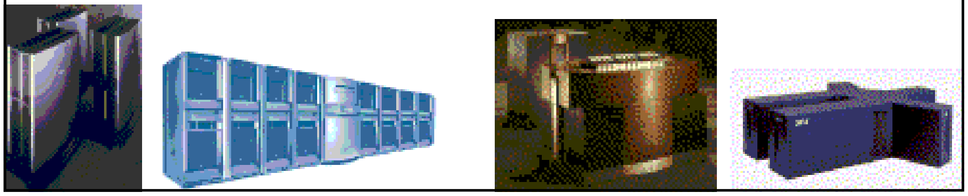
SISD

- **RISC - Superscalar**

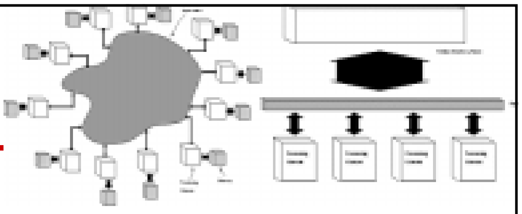
Architectures

- **MIMD Shared memory**
  - SGI Origin
  - HP-Convex Exempler
  - Cray T90/J90
  - NEC SX-4
  - Sun Enterprise

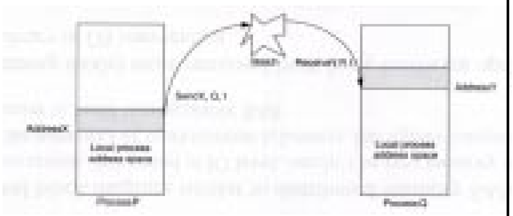


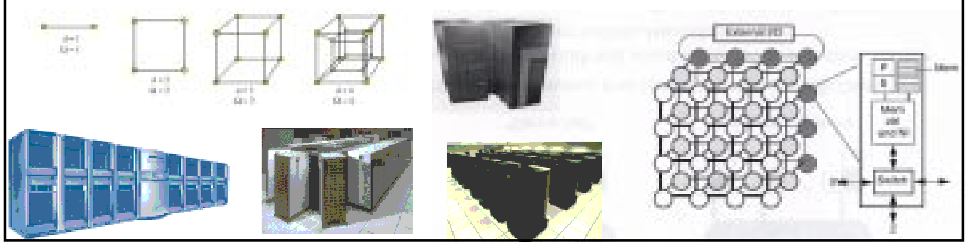


Distributed Memory

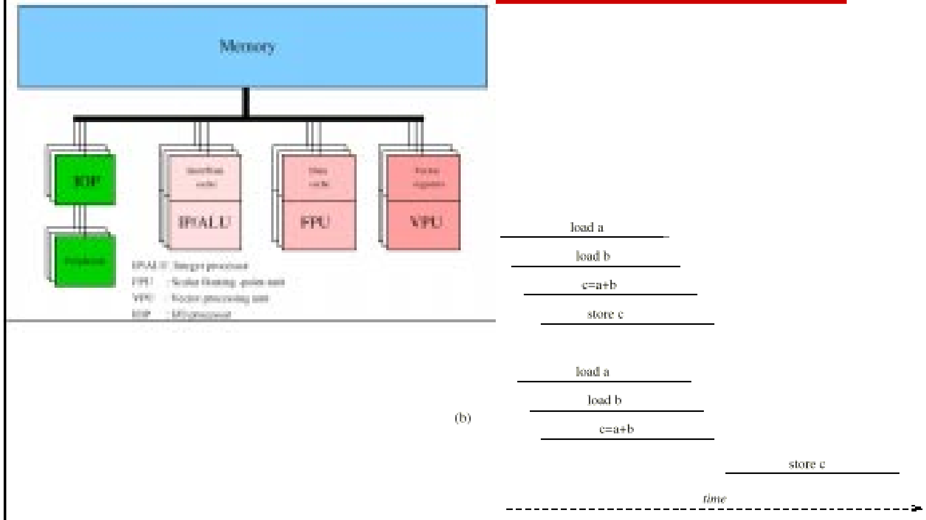


- IBM SP
- CRAY/SGI T3E
- SGI Origin 2000
- HP/Convex Exempler
- Sun Enterprise
- (Intel)



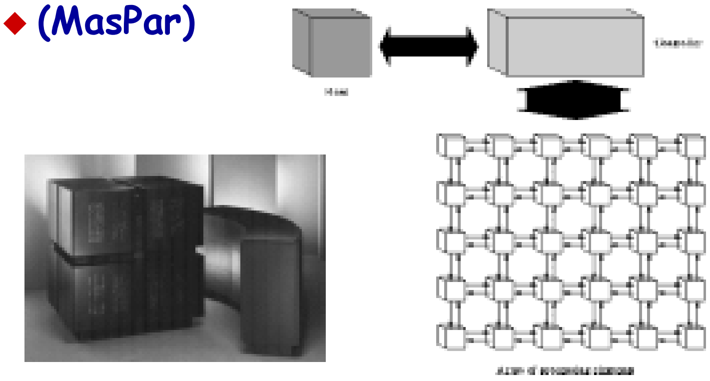


Vector Architecture (SIMD)



SIMD

- ◆ **(Thinking Machine CM-2)**
- ◆ **(MasPar)**



SGI/Cray Vector Computers

System parameters:

Model	Cray J90(se)	Cray T90
Clock cycle	10 ns (100 MHz)	2.2 ns (455 MHz)
Theor. peak performance	Per processor	
	200 Mflop/s	1.8 Gflop/s
No. of processors	4-32	1-32
Maximal	6.4 Gflop/s	58 Gflop/s
Main memory	<= 4 GB	<= 8 GB
Memory bandwidth		
Single proc. bandwidth	1.6 GB/s	24 GB/s

Achieving TeraFlops

- **In 1991, 1 Gflop/s**
- **1000 fold increase**
  - **Architecture**
    - exploiting parallelism
  - **Processor, communication, memory**
    - Moore's Law
  - **Algorithm improvements**
    - block-partitioned algorithms

Future: Petaflops (10^{15}fl pt ops/s)

Today $\approx \sqrt{10^{15}}$ flops for our workstations

- A Pflop for 1 second ≈ a typical workstation computing for 1 year.
- From an algorithmic standpoint
  - concurrency
  - data locality
  - latency & sync
  - floating point accuracy
  - dynamic redistribution of workload
  - new language and constructs
  - role of numerical libraries
  - algorithm adaptation to hardware failure

A Petaflops Computer System

- 1 Pflop/s sustained computing
- Between 10,000 and 1,000,000 processors
- Between 10 TB and 1PB main memory
- Commensurate I/O bandwidth, mass store, etc.
- If built today, cost $40 B and consume 1 TWatt.
- May be feasible and “affordable” by the year 2010