High Performance Computing Technologies

Jack Dongarra

University of Tennessee
Oak Ridge National Laboratory
http://www.netlib.org/utk/people/JackDongarra/

My Group in Tennessee

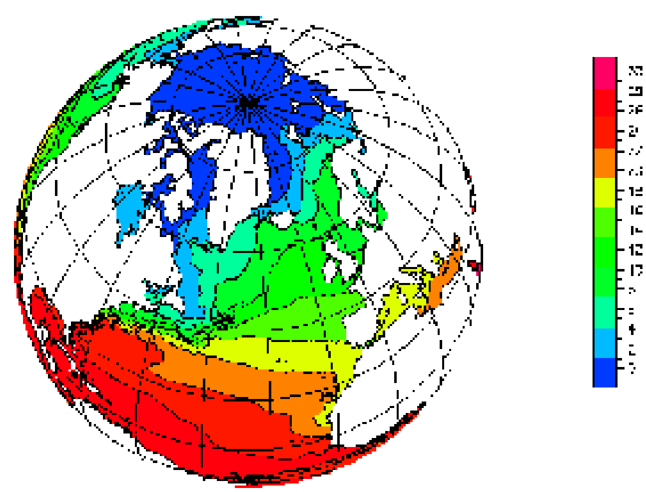
- Numerical Linear Algebra
  - Basic algorithms for HPC
  - EISPACK, LINPACK, BLAS, LAPACK, ScaLAPACK
- Heterogeneous Network Computing
  - PVM
  - MPI
- Software Repositories
  - Netlib
  - High-Performance Software Exchange
- Performance Evaluation
  - Linpack Benchmark, Top500
  - ParkBench

Computational Science

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

Why Turn to Simulation? ... Too Large

- Climate/Weather Modelling



- Data intensive problems (data-mining, oil resevoir simulation)
- Problems with large length and time scales (cosmology)

Why Parallel Computers?

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

Example: Weather Prediction (Navier-Stokes)
with 3D Grid around the Earth

$$6 \quad variables \begin{cases} temperature \\ pressure \\ humidity \\ 3 - wind \quad velocity \end{cases}$$

- 1 Kilometer Cells
- 10 slices $\rightarrow 5 \times 10^9$ cells
- each cell is 8 bytes, 2×10^{11} Bytes = 200 GBytes
- at each cell will perform 100 ops/cell
- 1 minute time step
- $\frac{100 ops/cell \times 5 \times 10^9 cells}{1 min \times 60 sec/min} = 8 GFlop/s$

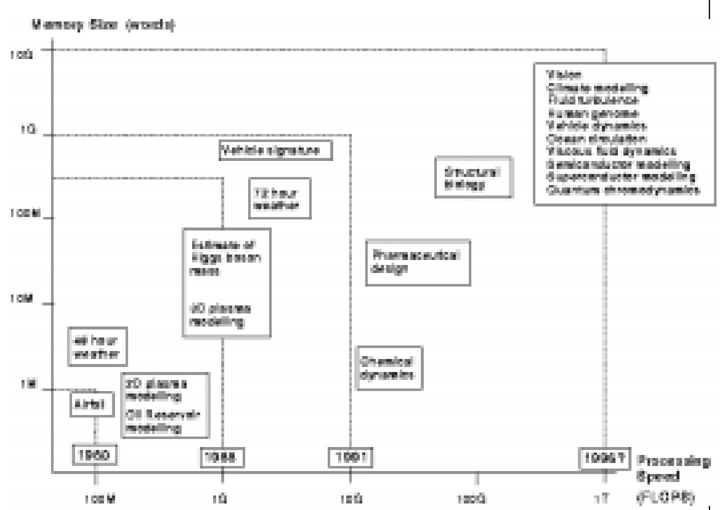
Automotive Industry

- Huge users of HPC technology: Ford (US) is 25th largest user of HPC in the world
- Main uses of simulation:
  - aerodynamics (similar to aerospace industry)
  - crash simulation
  - metal sheet forming
  - noise/vibrational optimization
  - traffic simulation
- Main gains:
  - reduced time to market of new cars;
  - increased quality;
  - reduced need to build (expensive) prototypes;
  - more efficient &; integrated manufacturing processes

Grand Challenge Science

- US Office of Science and Technology Policy
- Some Definitions A Grand Challenge is a fundamental problem in science or engineering, with potentially broad economic, political and/or scientific impact, that could be advanced by applying High Performance Computing resources
- The Grand Challenges of High Performance Computing are those projects which are almost too difficult to investigate using current supercomputers!

GC Computing Requirements

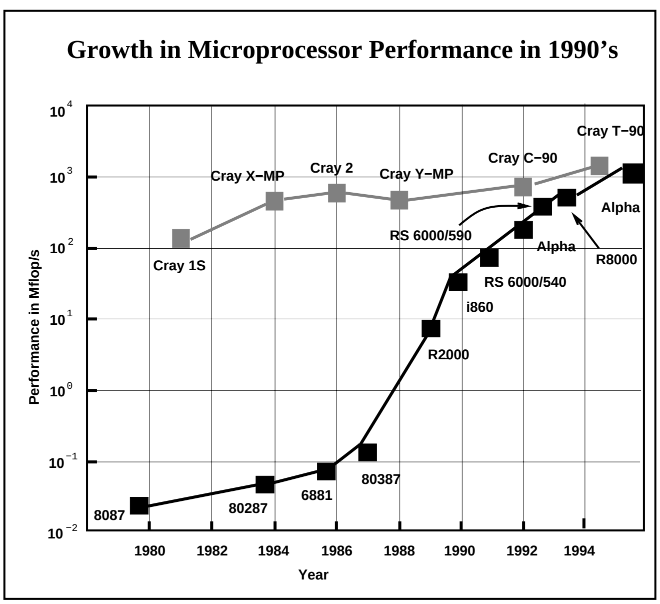


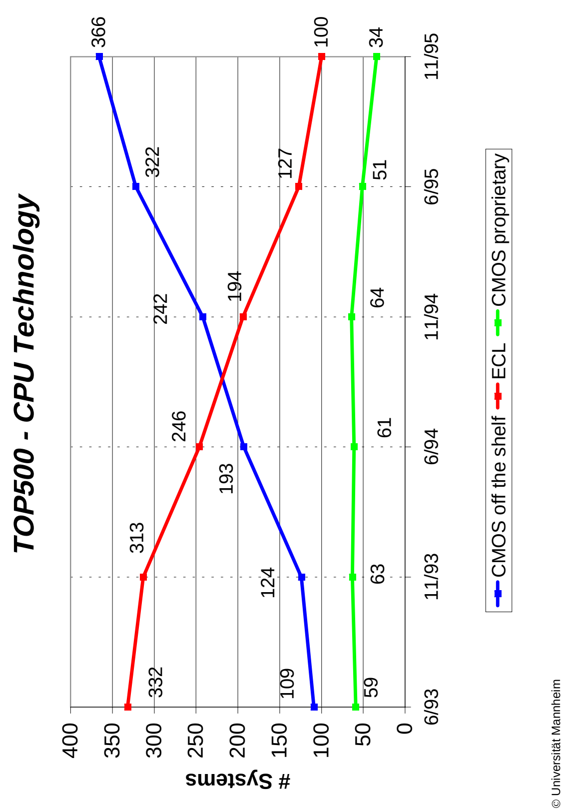
GC Summary

- Computational science is a (relatively) new method of investigating the world
- Current generation of high performance computers are making an impact in many areas of science
- New Grand Challenges appearing – e.g., global modeling, computational geography
- Users still want more power!
- ... and all this applies to HPC in business
- Maybe the problems in computational science are not so different from those in business ...?

High-Performance Computing Today

- In the past decade, the world has experienced one of the most exciting periods in computer development
- Computer performance improvements have been dramatic - a trend that promises to continue for the next several years.
- One reason for the improved performance is the rapid advance in microprocessor technology.
- Microprocessors have become smaller, denser, and more powerful.
- If cars had made equal progress, you could buy a car for a few dollars, drive it across the country in a few minutes, and "park" the car in your pocket!
- 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.





Scalable Multiprocessors

What is Required?

- Must scale the local memory bandwidth linearly.
- Must scale the global interprocessor communication bandwidth.
- Scaling memory bandwidth cost-effectively requires separate, distributed memories.
- Cost-effectiveness also requires best price-performance in individual processors.

What we get

- Compelling Price/Performance
- Tremendous scalability
- Tolerable entry price
- Tackle intractable problems

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.**

Cray v Cray

- Cray Research Inc. v Cray Computer Company
- CRI: Founded by Seymour Cray in 1972, the father of the supercomputer
- Business based on vector supercomputers & later MPP
  - Cray1 (‘76), XMP(‘82), YMP(‘87), C90(‘92), J90(‘93), T90 (‘95), ...
  - Cray1 (‘76), Cray2(‘85), Cray3(?)
  - T3D (‘94), T3E (‘96), ...
- Seymour Cray left to form CCC in 1989 to develop exotic processor technology (Cray 3)
- 1994 CCC went bust
- 1995 CRI returned to profit + huge order backlog

Silicon Graphics Inc. (SGI)

- The new kids on the block ...
- Founded in 1981 as a Stanford University spin-out
- Sales originally based on graphics workstations
  - Graphics done in hardware
  - exception to the rule of custom built chips being less cost effective than general-purpose processors running software
- All machines use mass produced processors from MIPS Computer Systems (now an SGI subsidiary)
- Aggressively marketed

SGI Today

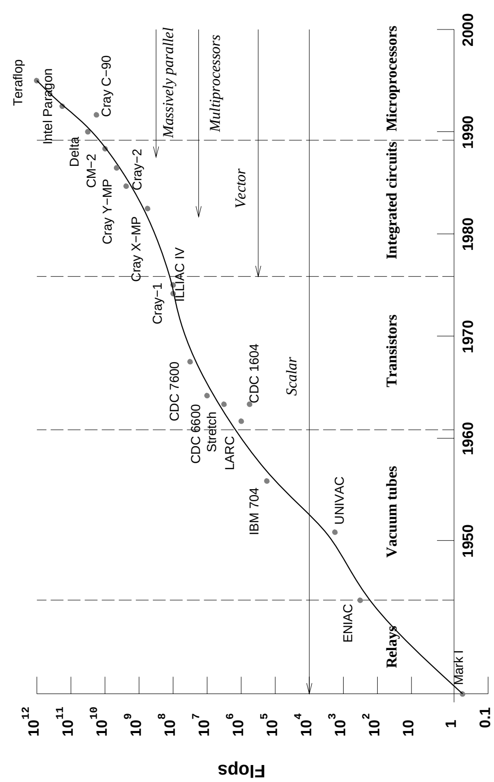
- New markets: move away from graphics workstations to general purpose HPC: introduction of parallelism
- Current: POWER CHALLENGE
- Aim:
  sell affordable / accessible / entry-level / scalable HPC
- Market position: 23% of machines in "Top 500" list
- Interesting asides:
  - MIPS announce deal to supply processors for the next generation of Nintendo machines: HPC feeding into the mainstream
  - Feb. 26, 1996: SGI buy 75% of CRI stock: low end HPC having strong influence on high end HPC

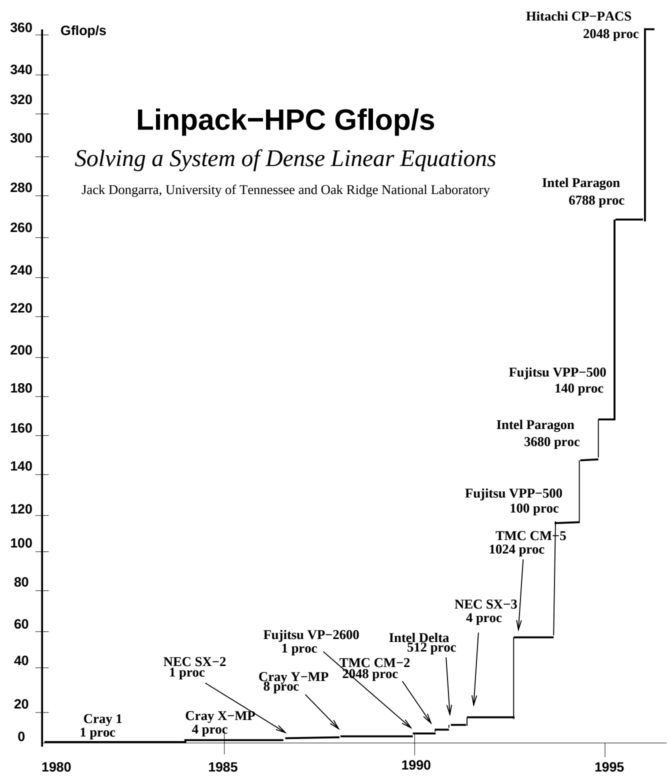
The Giants

- No longer just biding their time
- IBM: released SP2 in 1994 (based on workstation chips);
  - Market position: 21% of machines in "Top 500" list
- DEC: Memory Channel architecture released (1994) from networking and workstation processor experience
  - Market position: 3% of machines in "Top 500" list
- Intel: early experiences with hypercube machines (1982-90) 1995: won contract for US Government "Teraflops machine"
  - Market position: 5% of machines in "Top 500" list
- HP Convex: HP bought Convex in 1994, to bring together workstation knowledge & HPC
  - Market position: 4% of machines in "Top 500" list
- ... but how many of them are making a profit in MPP systems?
- Others: Fujitsu (7%), NEC (8%), Hitachi (3%), Tera, Meiko (2%)

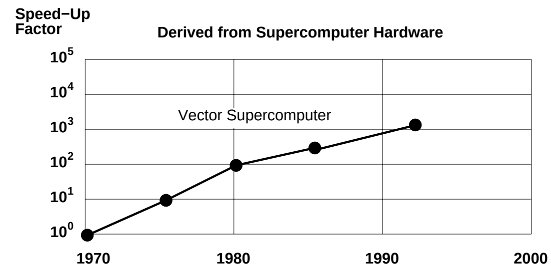
Scientific Computing: 1986 vs. 1996

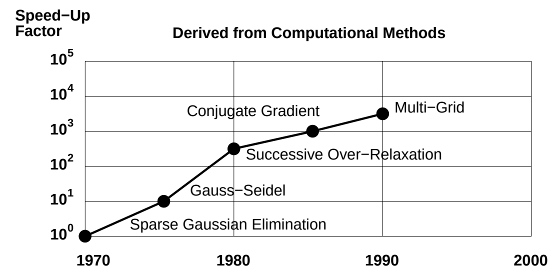
- **1986:**
  1. **Minisupercomputers (1 - 20 Mflop/s): Alliant, Convex, DEC.**
  2. **Parallel vector processors (PVP) (20 - 2000 Mflop/s): CRI, CDC, IBM.**
- **1996:**
  1. **PCs (200 Mflop/s): Intel Pentium Pro**
  2. **RISC workstations (10 - 1000 Mflop/s): DEC, HP, IBM, SGI, Sun.**
  3. **RISC based symmetric multiprocessors (SMP) (0.5 - 15 Gflop/s): HP-Convex, DEC, and SGI-CRI.**
  4. **Parallel vector processors (1 - 250 Gflop/s): SGI-CRI, Fujitsu, and NEC.**
  5. **Highly parallel processors (1 - 250 Gflop/s): HP-Convex, SGI-CRI, Fujitsu, IBM, NEC, Hitachi**





Performance Improvement for Scientific Computing Problems





Department of Energy's Accelerated Strategic Computing Initiative

- 5-year, $1B program designed to deliver tera-scale computing capability.
- "Stockpile Stewardship" - safe and reliable maintenance of the nation's nuclear arsenal in the absence of nuclear testing.
- Advanced computations, specifically 3-D modeling and simulation capability, are viewed as the backbone of "stockpile stewardship"
- 5 generations of HPC will be delivered over the lifetime of the program.
- First machine is a single massively parallel 1.8 teraflop computer. Intel Paragon based on 9000 200 Mflop/s Pentium processors to Sandia Labs by the end of 1996.
- Second machine is a $93M system from IBM consists of clusters of shared-memory processors. 3 Tflop/s system is scheduled for demonstration in December 1998 to LLNL.
- Third machine is a NUMA system from SGI-CRI. Schedule for LANL.
- Remaining two machines will deliver capability in the 30- and 100-Tflop/s range.

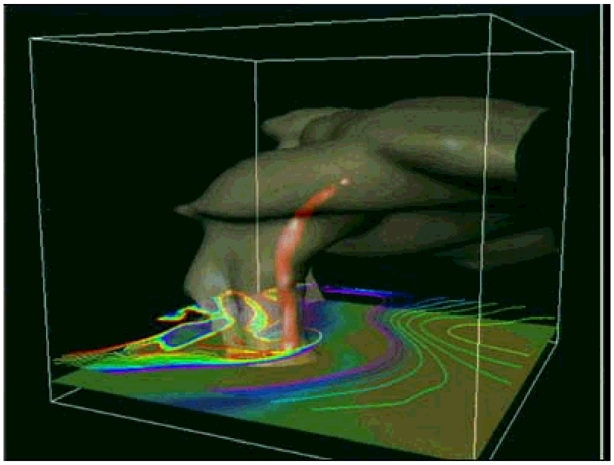
Virtual Environments

- When the number crunchers finish crunching, the user is facedwith the mammoth task of making sense of the data. As visualization and computation become ever more closely coupled, new environments for scientific discovery emerge: 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?



Alternative Supercomputing Resources

- Vast numbers of under utilized workstations available to use.
- Huge numbers of unused processor cycles and resources that could be put to good use in a wide variety of applications areas.
- Reluctance to buy Supercomputer due to their cost and short life span.
- Distributed computer resources "fit" better into today's funding model.

MIMD, multicomputer: networked workstations

Enabling software technology: PVM (Parallel Virtual Machine) available from netlib@ornl.gov

Enabling software technology: MPI (Message Passing Interface) available from netlib@ornl.gov

very active research area; about 150 software products;
catalog available NHSE

Enabling hardware technology: high bandwidth interconnect is not here yet;

Ethernet: msec latencies and 100's of Kbyte/sec bandwidth insufficient

Other technology is on the verge of becoming available:
HIPPI products, Fibre Channel, ATM.

THE METACOMPUTER: ONE FROM MANY

- Birth of a Concept
- The term "metacomputing" was coined around 1987 by NCSA Director, Larry Smarr. But the genesis of metacomputing took place years earlier.
- Goals for the research community was to provide a "Seamless Web" linking the user interface on the workstation and supercomputers.

MetaComputer Summary

- Many parts and functions of a metacomputer are being tested on a small scale today.
- Much research remains to create a balanced system of computational power and mass storage connected by high-speed networks.
- The ultimate goal is to have a Scalable Distributed Operating System

Java

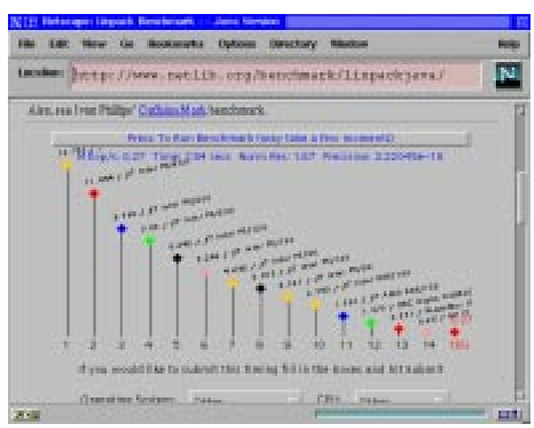
- Java likely to be a dominant language.
  - C++ like language
  - Taking the web/world by storm
  - No pointers or memory deallocation
  - Portability achieved via abstract machine
- Java is a convenient user interface builder which allows one to develop quickly customized interfaces.
- Internet is slow and getting slower, many activities focus on intranets.

Open Universal WebWindows
A Revolution in the Software Industry

- In future one will not write software for either
  - Windows95/NT, UNIX, Digital VMS, etc
- Rather one will write software for WebWindows defined as the operating environment for the World Wide Web
- WebWindows builds on top of Web Servers and Web Client open interfaces as in
  - CGI interface for servers
  - Java or equivalent applet technology for clients
- Applications written for WebWindows will be portable to all computers running Web Servers or Clients which hide hardware and native OS specifics.

Java Linpack Benchmark

- Should Java be taken seriously for numerical computations?
- 3 months ago the fastest Java performance was 1 Mflop/s on a 600 Mflop/s processor.
- Top performer today is 13.7 Mflop/s for a P6 using Netscape 3.0 JIT
- URL http://www.netlib.org/benchmark/linpackjava/



Metacomputing in the Future The Future Trends...

- Long term is hard to predict- See changes over the last 5 Years!!
- Can see trends, however...

Metacomputing in the Future

Hardware Trends (5-10 Years) Computers

- Millions (100-300) of ”settop” boxes
- One in every US household
- More worldwide
- Ranging from Supercomputing to Personal Digital Assistants.

Metacomputing in the Future Hardware Trends (5-10 Years) Networks

- Networks (1-20 MByte/s) fulfill needs of ”home” entertainment industry.
- Technologies ranging from high band-width fibre to Electromagnetic types such as Microwave.

Metacomputing in the Future Hardware Trends (5-10 Years) Software

- Very hard to predict in a relatively short term- JAVA has been a product for about a year!!
- Ubiquitous and pervasive (WWW/JAVA-like).
- Can forget about underlying h/w and OS.
- Metacomputing “plug-ins”
- Micro-kernel-like JAVA based servers with add-on services that can support Metacomputing (load balancing, migration, checkpointing, etc...)

Highly Parallel Supercomputing: Where Are We?

1. 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.
2. 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?

1. Operating systems:
   - Robustness and reliability are improving.
   - New system management tools improve system utilization.

     But
   - Reliability still not as good as conventional systems.
2. I/O subsystems:
   - New RAID disks, HiPPI interfaces, etc. provide substantially improved I/O performance.

     But
   - I/O remains a bottleneck on some systems.

Current Situation...

- An ongoing thread of research in scientific computing is the efficient solution of large problems.
- Various mechanisms have been developed to perform computations across diverse platforms. The most common mechanism involves software libraries.
- Some software libraries are highly optimized for only certain platforms and do not provide a convenient interface to other computer systems.
- Other libraries demand considerable programming effort from the user, who may not have the time to learn the required programming techniques.
- While a limited number of tools have been developed to alleviate these difficulties, such tools themselves are usually available only on a limited number of computer systems.

The Importance of Standards (I)

Software

- Writing programs for MPP is hard ...
- But ... one-off effort 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 (II)

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