High-Performance Computing Today

Jack Dongarra
Innovative Computing Laboratory
University of Tennessee
and
Oak Ridge National Laboratory
http://www.cs.utk.edu/~dongarra/

1

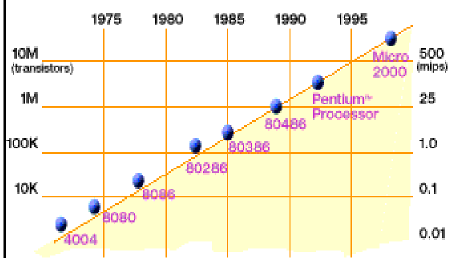
Outline

- **Look at trends in HPC**
  - **Top500 statistics**
- **Performance of Super-Scalar Processors**
  - **ATLAS**
- **Performance Monitoring**
  - **PAPI**
- **NetSolve**
  - **Example of grid middleware**

In pioneer days, they used oxen for heavy pulling, and when one ox couldn't budge a log they didn't try to grow a larger ox. We shouldn't be trying for bigger computers, but for more systems of computers.-- Grace Hopper

2

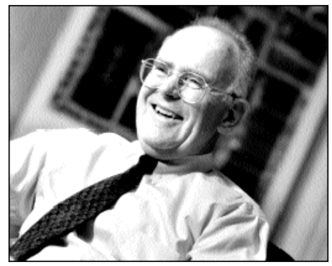
Technology Trends: Microprocessor Capacity



2X transistors/Chip Every 1.5 years

Called "Moore's Law"

Microprocessors have become smaller, denser, and more powerful.



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

3

High Performance Computers & Numerical Libraries

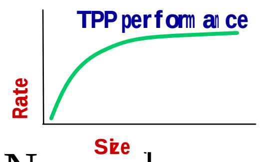
- **20 years ago**
  - **$1x10^6$ Floating Point Ops/sec (Mflop/s)**
    - » Scalar based
    - » Loop unrolling
- **10 years ago**
  - **$1x10^9$ Floating Point Ops/sec (Gflop/s)**
    - » Vector & Shared memory computing, bandwidth aware
    - » Block partitioned, latency tolerant
- **Today**
  - **$1x10^{12}$ 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

4

TOP500

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

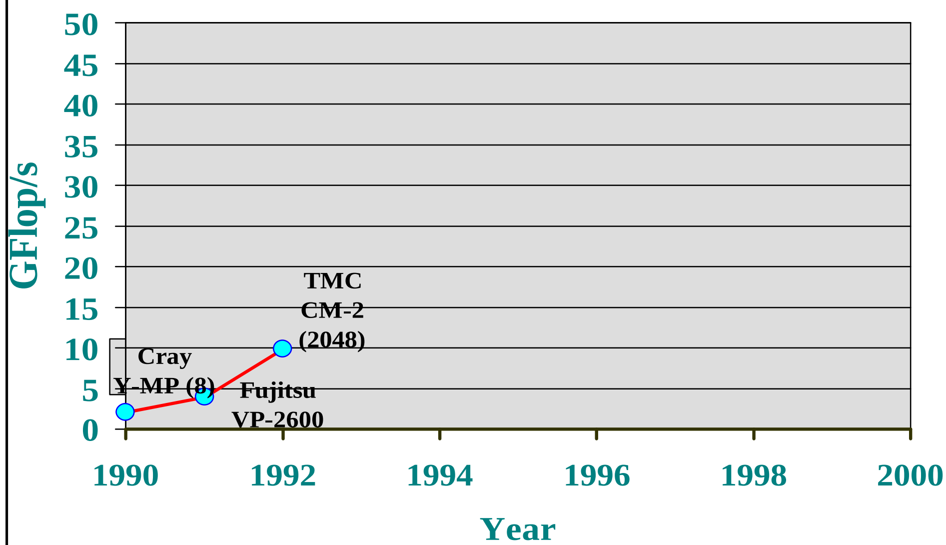
Ax=b, dense problem



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

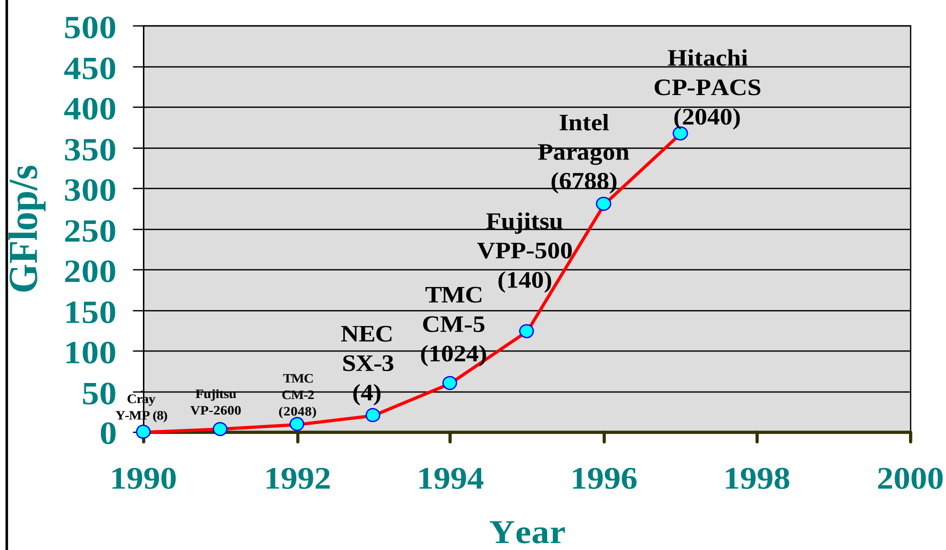
5

Fastest Computer Over Time



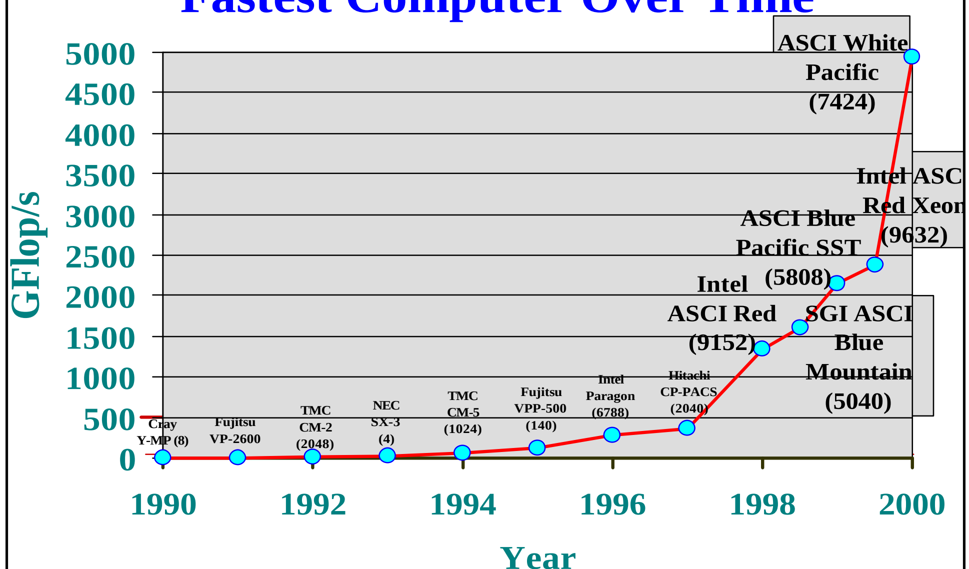
In 1980 a computation that took 1 full year to complete can now be done in 1 month!

Fastest Computer Over Time



In 1980 a computation that took 1 full year to complete can now be done in 4 days!

Fastest Computer Over Time

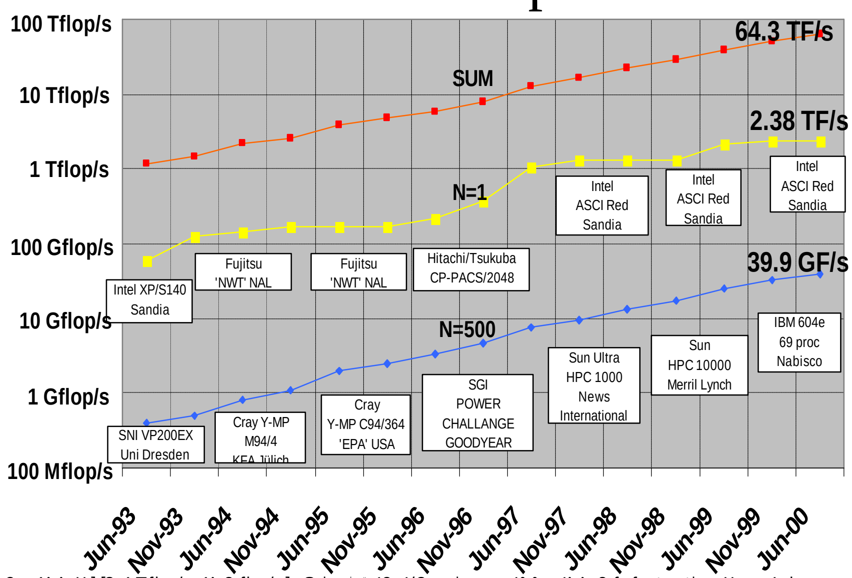


In 1980 a computation that took 1 full year to complete can today be done in 1 hour!

Top 10 Machines (June 2000)

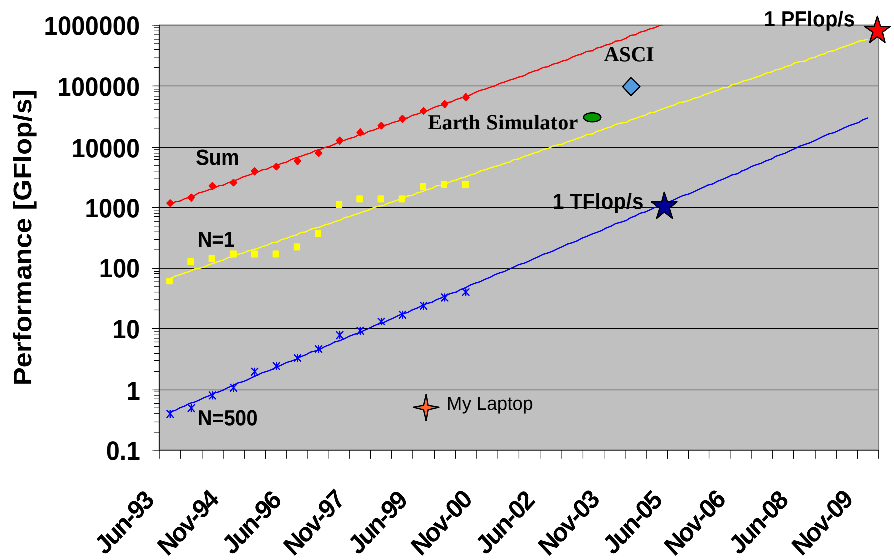
Rank	Company	Machine	Procs	Gflop/s	Place	Country	Year
1	Intel	ASCI Red	9632	2380	Sandia National Labs Albuquerque	USA	1999
2	IBM	ASCI Blue-Pacific SST, IBM SP 604e	5808	2144	Lawrence Livermore National Laboratory Livermore	USA	1999
3	SGI	ASCI Blue Mountain	6144	1608	Los Alamos National Laboratory Los 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



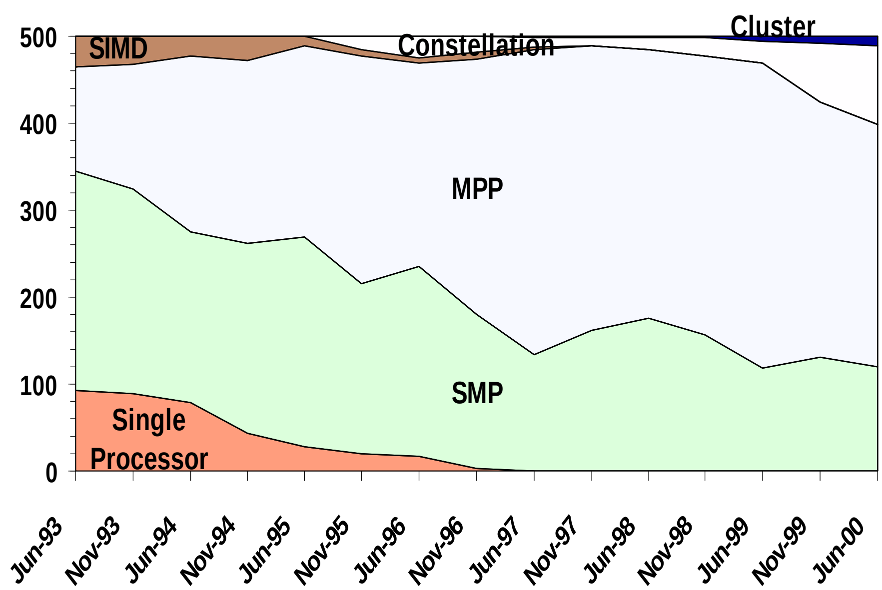
[60 G - 400 M][2.4 Tflop/s 40 Gflop/s], Schwab #19, 1/2 each year, 133 > 100 Gf, faster than Moore's law,

Performance Development



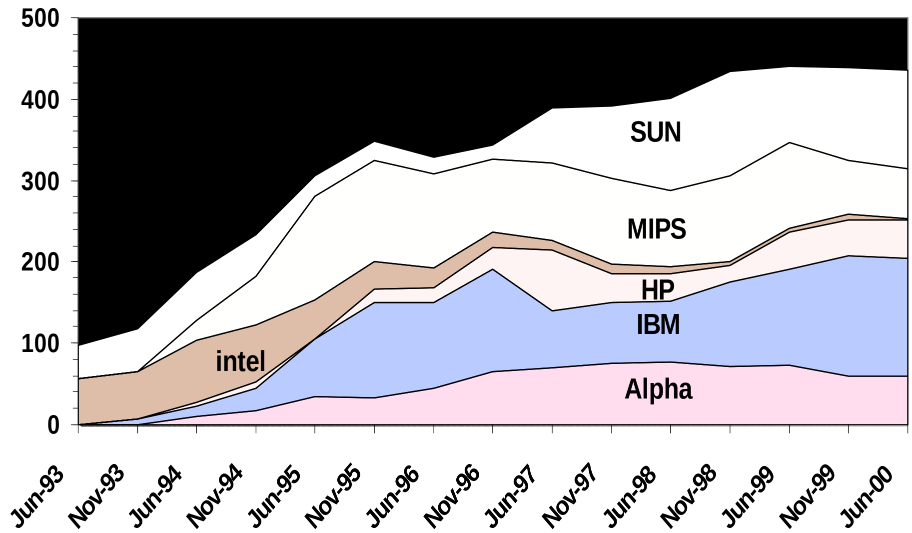
Entry 1 T 2005 and 1 P 2010

Architectures

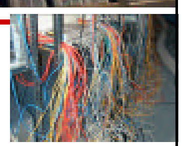


91 const, 14 clus, 275 mpp, 120 smp

Chip Technology



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**

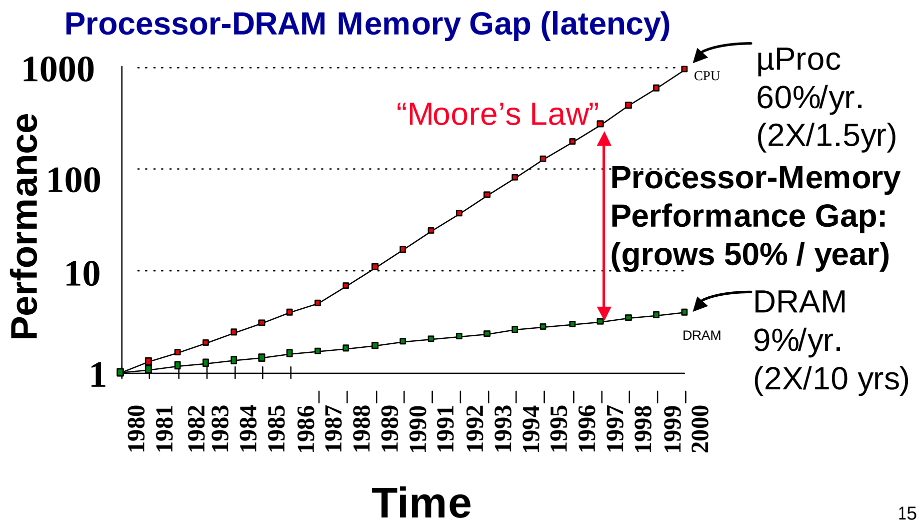
Advantages:

- **Best price-performance**
- **Low entry-level cost**
- **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.

14

Where Does the Performance Go? or Why Should I Cares About the Memory Hierarchy?



15

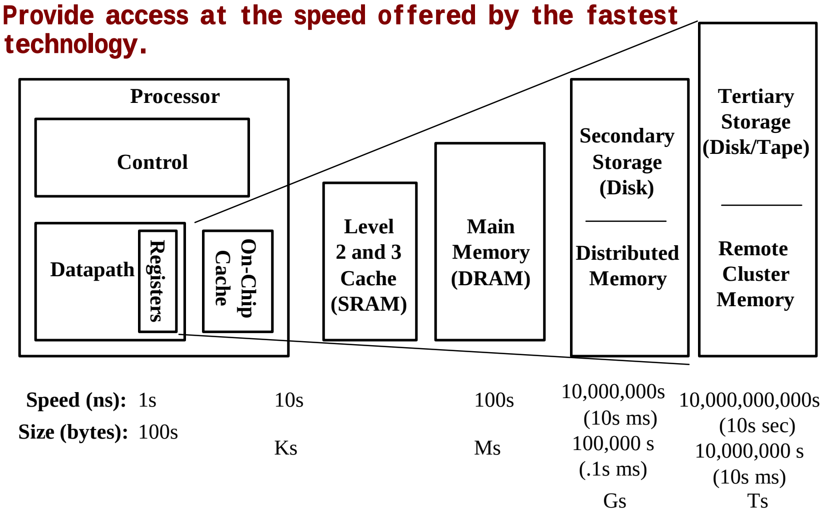
Optimizing Computation and Memory Use

- **Computational optimizations**
  - **Theoretical peak: (# fpus)*(flops/cycle) * Mhz**
    - » PIII: (1 fpu)*(1 flop/cycle)*(650 Mhz) = 650 MFLOP/s
    - » Athlon: (2 fpu)*(1flop/cycle)*(600 Mhz) = 1200 MFLOP/s
    - » Power3: (2 fpu)*(2 flops/cycle)*(375 Mhz) = 1500 MFLOP/s
- **Memory optimization**
  - **Theoretical peak: (bus width) * (bus speed)**
    - » PIII : (32 bits)*(133 Mhz) = 532 MB/s = 66.5 MW/s
    - » Athlon: (64 bits)*(200 Mhz) = 1600 MB/s = 200 MW/s
    - » Power3: (128 bits)*(100 Mhz) = 1600 MB/s = 200 MW/s
- **Memory about an order of magnitude slower**

16

Memory Hierarchy

- By taking advantage of the principle of locality:
  - Present the user with as much memory as is available in the cheapest technology.
  - Provide access at the speed offered by the fastest technology.



How To Get Performance From Commodity Processors?

- Today's processors can achieve high-performance, but this requires extensive machine-specific hand tuning.
- Hardware and software have a large design space w/many parameters
  - Blocking sizes, loop nesting permutations, loop unrolling depths, software pipelining strategies, register allocations, and instruction schedules.
  - Complicated interactions with the increasingly sophisticated micro-architectures of new microprocessors.
- Until recently, no tuned BLAS for Pentium for Linux.
- Need for quick/dynamic deployment of optimized routines.
- ATLAS - Automatic Tuned Linear Algebra Software
  - PhiPac from Berkeley
  - FFTW from MIT (http://www.fftw.org)

18

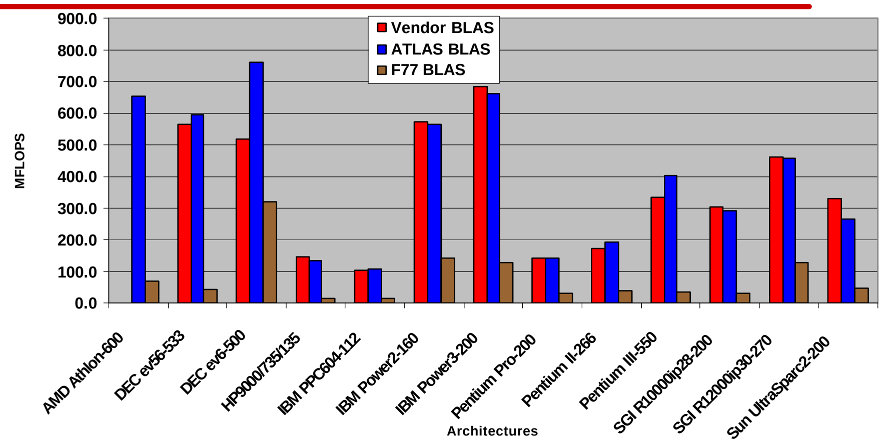
ATLAS

- **An adaptive software architecture**
  - **High-performance**
  - **Portability**
  - **Elegance**

- **ATLAS is faster than all other portable BLAS implementations and it is comparable with machine-specific libraries provided by the vendor.**

19

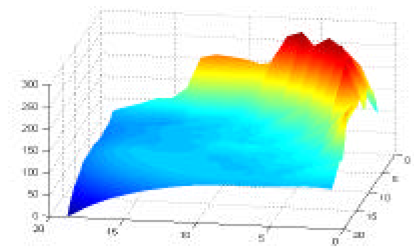
ATLAS Across Various Architectures (DGEMM n=500)



- **ATLAS is faster than all other portable BLAS implementations and it is comparable with machine-specific libraries provided by the vendor.**

20

Code Generation Strategy



- On-chip multiply optimizes for:
  - TLB access
  - L1 cache reuse
  - FP unit usage
  - Memory fetch
  - Register reuse
  - Loop overhead minimization
- Takes a 30 minutes to a hour to run.
- New model of high performance programming where critical code is machine generated using parameter optimization.
- Code is iteratively generated & timed until optimal case is found. We try:
  - Differing NBs
  - Breaking false dependencies
  - M, N and K loop unrolling
- Designed for RISC arch
  - Super Scalar
  - Need reasonable C compiler

21

Plans for ATLAS

- **Software Release, available today:**
  - Level 1, 2, and 3 BLAS implementations
  - See: www.netlib.org/atlas/
- **Next Version:**
  - Multi-treading
  - Java generator
- **Futures:**
  - Optimize message passing system
  - Runtime adaptation
    - Sparsity analysis
    - Iterative code improvement
  - Specialization for user applications
  - Adaptive libraries

22

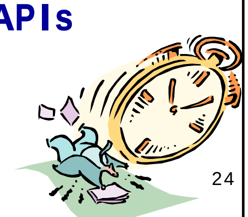
Tools for Performance Evaluation

- Timing and performance evaluation has been an art
  - Resolution of the clock
  - Issues about cache effects
  - Different systems
- Situation about to change
  - Today's processors have internal counters

23

Performance Counters

- Almost all high performance processors include hardware performance counters.
- Some are easy to access, others not available to users.
- On most platforms the APIs, if they exist, are not appropriate for a common user, functional or well documented.
- Existing performance counter APIs
  - Cray T3E
  - SGI MIPS R10000
  - IBM Power series
  - DEC Alpha pfm pseudo-device interface
  - Windows 95, NT and Linux



24

Performance Data That May Be Available

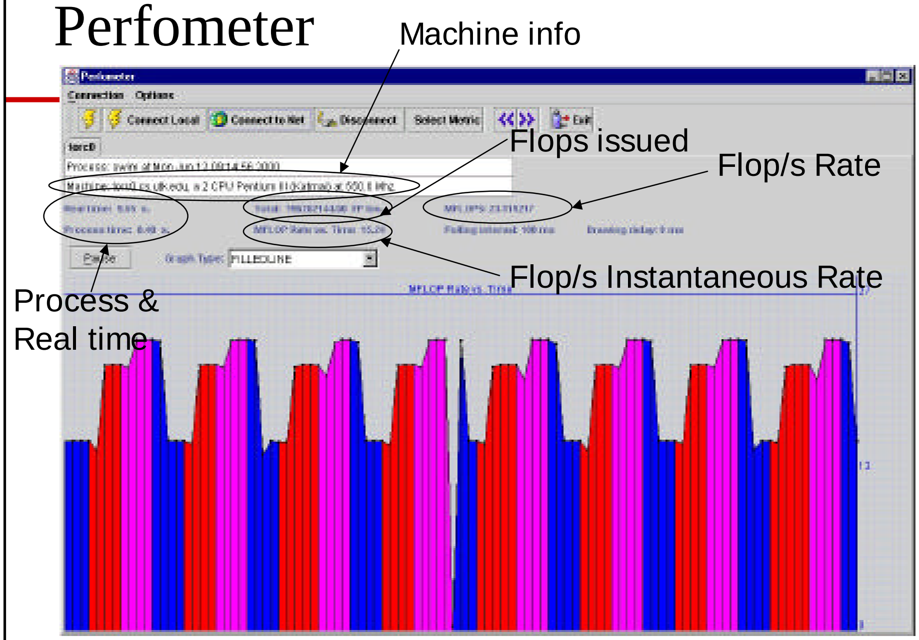
- Cycle count
- Floating point instruction count
- Integer instruction count
- Instruction count
- Load/store count
- Branch taken / not taken count
- Branch mispredictions
- Pipeline stalls due to memory subsystem
- Pipeline stalls due to resource conflicts
- I/D cache misses for different levels
- Cache invalidations
- TLB misses
- TLB invalidations

25

PAPI's Graphical Tools Perfometer Usage

- Application is instrumented with PAPI
  - call perfometer()
- Will be layered over the best existing vendor-specific APIs for these platforms
- Sections of code that are of interest are designated with specific colors
  - Using a call to set_perfometer(color)
- Application is started, at the call to performeter a task is spawned to collect and send the information to a Java applet containing the graphical view.

Perfometer



Call Perfometer('red')

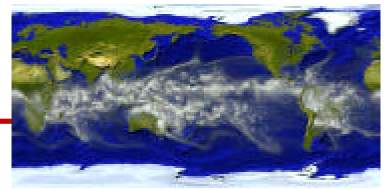
27

Go To Demo

PAPI

28

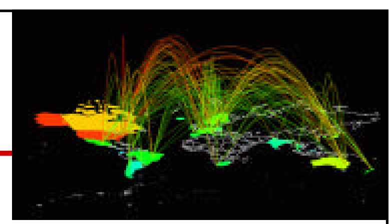
Trends in Computational Science and Engineering



- Multi-scale, Multi-physics, Multi-dimensional simulations of realistic complexity
- Growing use of dynamic adaptive algorithms
- Strong interplay between observation and simulation (e.g., cosmology, weather)
- Impact of the WWW
  - accelerated pace of research due to electronic publishing
  - proliferation of digital archives
  - emergence of workbenches and portals

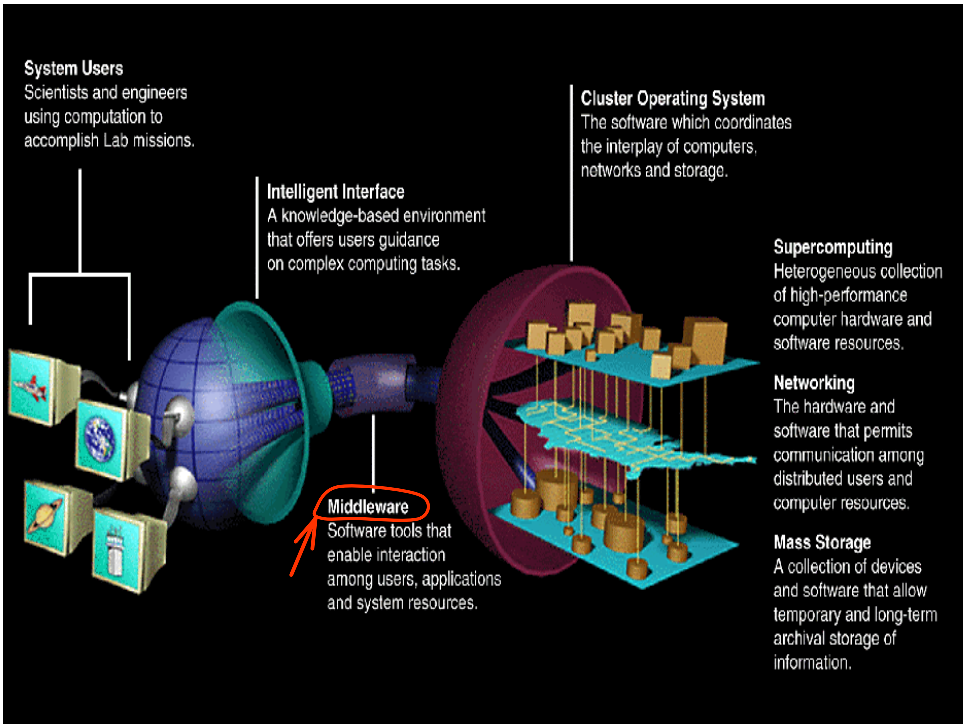
29

Grid Computing

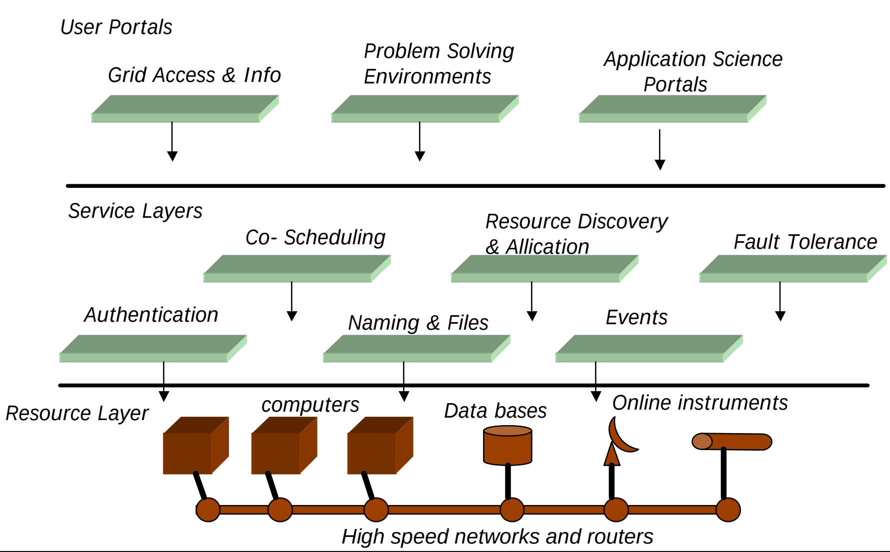


- To treat CPU cycles and software like commodities, an application should be:
  - Ubiquitous -- able to interface to the system at any point and leverage whatever is available
  - Resource Aware -- capable of managing heterogenity
  - Adaptive -- able to tailor its behavior dynamically so that it gets maximum performance benefit from the services and resources at hand

30



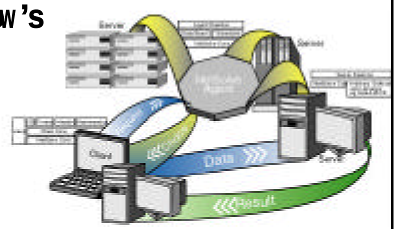
The Grid Architecture Picture



32

Motivation for NetSolve

Design an easy-to-use tool to provide efficient and uniform access to a variety of scientific packages on UNIX and Window's platforms

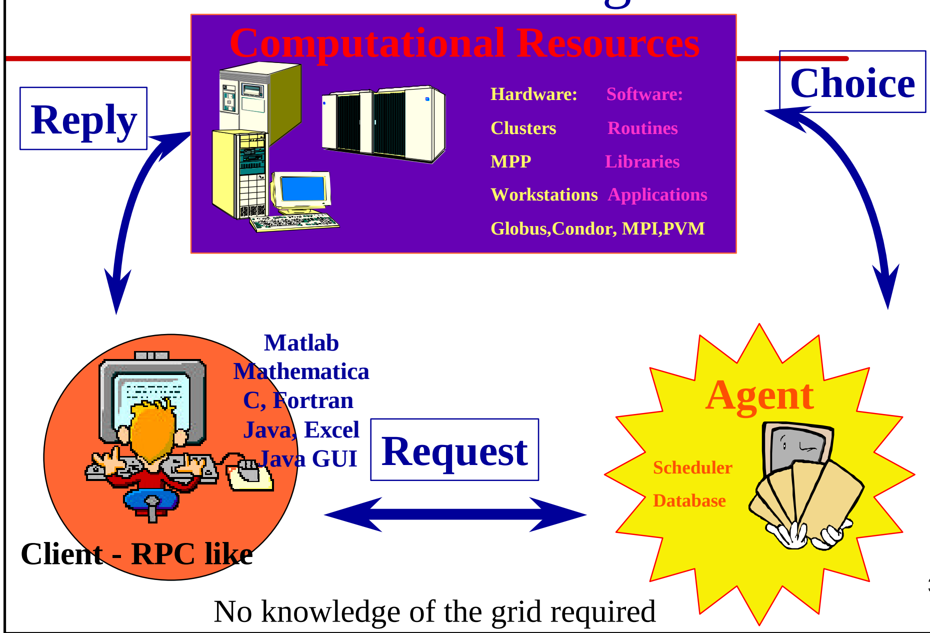


Basics

- Client-Server Design
- Non-hierarchical system
- Load Balancing and Fault Tolerance
- Heterogeneous Environment Supported
- Multiple and simple client interfaces
- Built on standard components

33

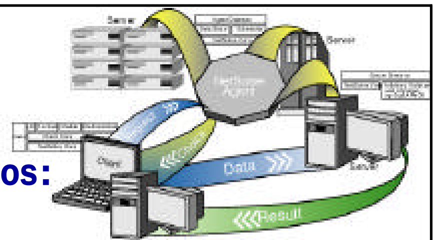
NetSolve - The Big Picture



No knowledge of the grid required

34

NetSolve



- Three deployment scenarios:
  - Client, servers and agents anywhere on Internet (3(10)-150(80-ws/mpp)-Mcell)
  - Client, servers and agents on an Intranet
  - Client, server and agent on the same machine
- "Blue Collar" Grid Based Computing
  - User can set things up, no "su" required
  - Does not require deep knowledge of network programming
- Smart Libraries
  - "Rent" access to routines
  - Decouple interface

35

NetSolve Usage Scenarios

- Grid based library routines
  - Users doesn't have to have library routines on their machine
- Task farming applications
  - "Pleasantly parallel" execution
  - eg Parameter studies
- Remote application execution
  - Complete packages with user specifying input parameters

36

NetSolve - MATLAB Interface

Synchronous Call

```
>> define sparse matrix A
>> define rhs
>> [x, its] = netsolve('itmeth', 'petsc', A, rhs, 1.e-6, 50);
…
>> [x, its] = petsc(A, rhs);   % for PETSc
>> [x, its] = aztec(A, rhs);   % for AZTEC
>> [x] = superlu(A, rhs);      % for SuperLU
>> [x] = ma28(A, rhs);         % for MA28
```

Asynchronous Calls also available

NetSolve - FORTRAN Interface

Easy to ‘switch’ to NetSolve

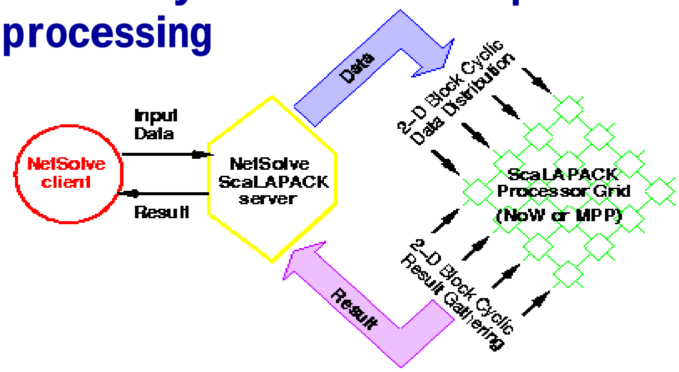
```
parameter( MAX = 100)
double precision A(MAX,MAX), B(MAX)
integer IPIV(MAX), N, INFO, LWORK
integer NSINFO

call DGESV(N,1,A,MAX,IPIV,B,MAX,INFO)


call NETSL(‘DGESV()’,NSINFO,
           N,1,A,MAX,IPIV,B,MAX,INFO)
```

Hiding the Parallel Processing

? **User maybe unaware of parallel processing**

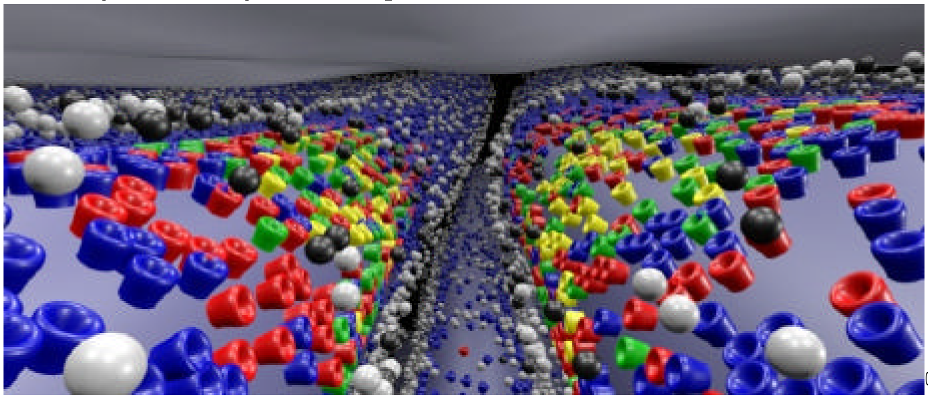


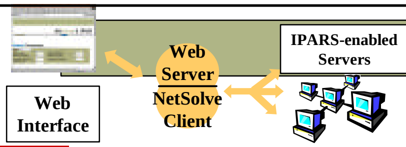
? **NetSolve takes care of the starting the message passing system, data distribution, and returning the results.**

39

MCell: 3-D Monte-Carlo Simulation of Neuro-Transmitter Release in Between Cells

•Developed at: Salk Institute (T. Bartol), Cornell U. (J. Stiles)
•Study how neurotransmitters diffuse and activate receptors in synapses
•blue unbounded, red singly bounded, green doubly bounded closed, yellow doubly bounded open





NetSolve & IPARS

- Integrated Parallel Accurate Reservoir Simulator.
  - Mary Wheeler's group, UT-Austin
- Reservoir and Environmental Simulation.
  - models black oil, waterflood, compositions
  - 3D transient flow of multiple phase
- Integrates Existing Simulators.
- Framework simplified development
  - Provides solvers, handling for wells, table lookup.
  - Provides pre/postprocessor, visualization.
- Full IPARS access without Installation.
- IPARS Interfaces Now Available:
  - C, FORTRAN, Matlab, Mathematica, and Web.

41

NetSolve Applications and Interactions

- Tool integration
  - Globus - Middleware infrastructure (ANL/SSI)
  - Condor - Workstation farm (U Wisconsin)
  - NWS - Network Weather Service (U Tennessee)
  - SCIRun - Computational steering (U Utah)
  - Ninf - NetSolve-like system, (ETL,Tsukuba)
- Library usage
  - LAPACK/ScaLAPACK - Parallel dense linear solvers
  - SuperLU/MA28 - Parallel sparse direct linear solvers(UCB/RAL)
  - PETSc/Aztec - Parallel iterative solvers (ANL/SNL)
  - Other areas as well (not just linear algebra)
- Applications
  - MCell - Microcellular physiology (UCSD/Salk)
  - IPARS - Reservoir Simulator (UTexas, Austin)
  - Virtual Human - Pulmonary System Model (ORNL)
  - RSICC - Radiation Safety sw/simulation (ORNL)
  - LUCAS - Land usage modeling (U Tennessee)
  - ImageVision - Computer Graphics and Vision (Graz U)

42

Conclusion

- Exciting time to be in scientific computing
- Network computing is here
- The Grid offers tremendous opportunities for collaboration
- Important to develop algorithms and software that will work effectively in this environment

43

Contributors to These Ideas

- Top500
  - Erich Strohmaier, UTK
  - Hans Meuer, Mannheim U
- ATLAS
  - Antoine Petitet, UTK
  - Clint Whaley, UTK
- PAPI
  - Shirley Browne, UTK
  - Nathan Garner, UTK
  - Kevin London, UTK
  - Phil Mucci, UTK
- NetSolve
  - Dorian Arnold, UTK
  - Susan Blackford, UTK
  - Henri Casanova, UCSD
  - Michelle Miller, UTK
  - Sathish Vadhiyar, UTK

For additional information see…

www.netlib.org/top500/
www.netlib.org/atlas/
icl.cs.utk.edu/projects/papi/
www.netlib.org/netsolve/
www.cs.utk.edu/~dongarra/

Many opportunities within group

44

45