High Performance Computing and Trends: Connecting Computational Requirements with Computing Resources

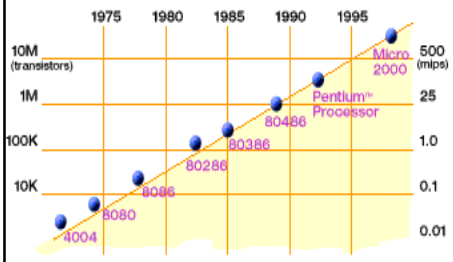
Jack Dongarra
Innovative Computing Laboratory
University of Tennessee

http://www.cs.utk.edu/~dongarra/

Outline

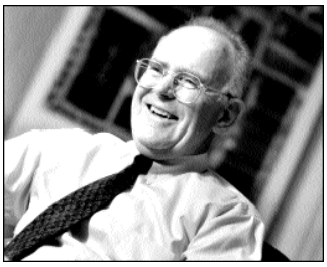
- Look at trends in HPC
  - Top500 statistics
- Grid based computing
- NetSolve
  - Example of grid middleware

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

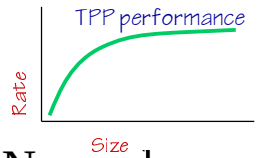
- 20 years ago
  - $1x10^6$ Floating Point Ops/sec (Mflop/s)
    - Scalar based
- 10 years ago
  - $1x10^9$ Floating Point Ops/sec (Gflop/s)
    - Vector & Shared memory computing, bandwidth aware
    - Block partitioned, latency tolerant
- Today
  - $1x10^{12}$ Floating Point Ops/sec (Tflop/s)
    -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

4

TOP500

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

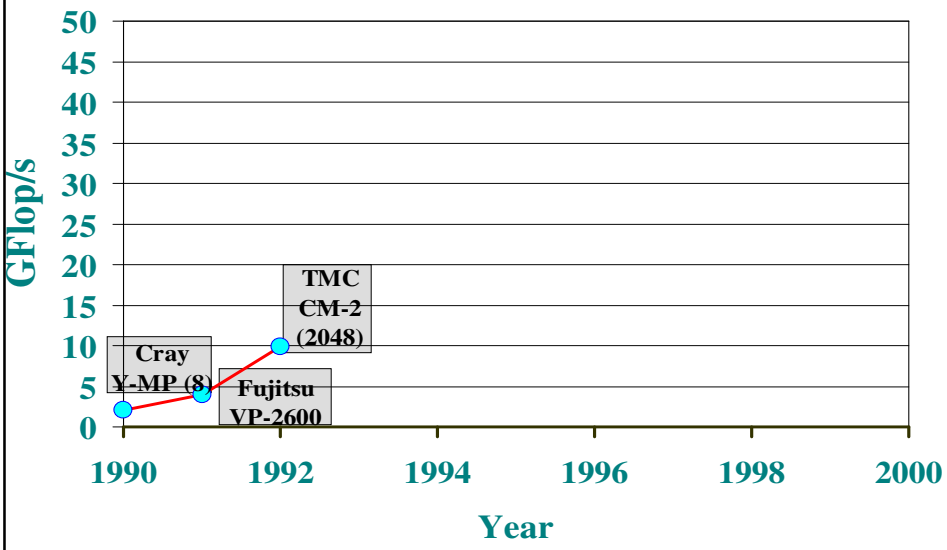
  $Ax=b$, *dense problem*



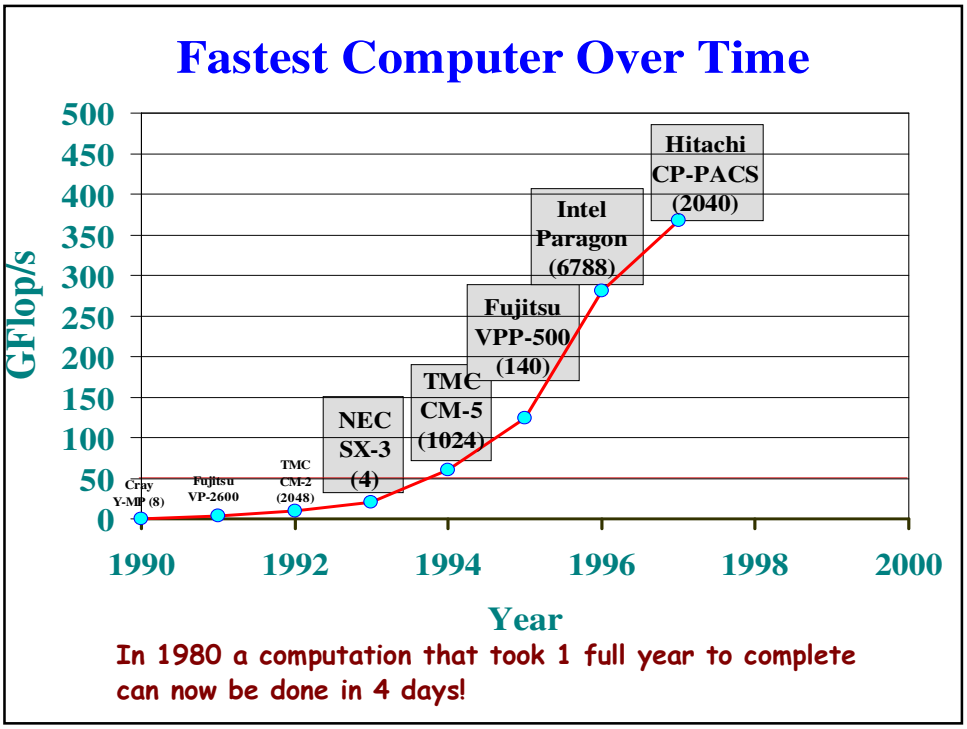
- Updated twice a year
  SC'xy in the States in November
  Meeting in Mannheim, Germany in June
- All data available from **www.top500.org**

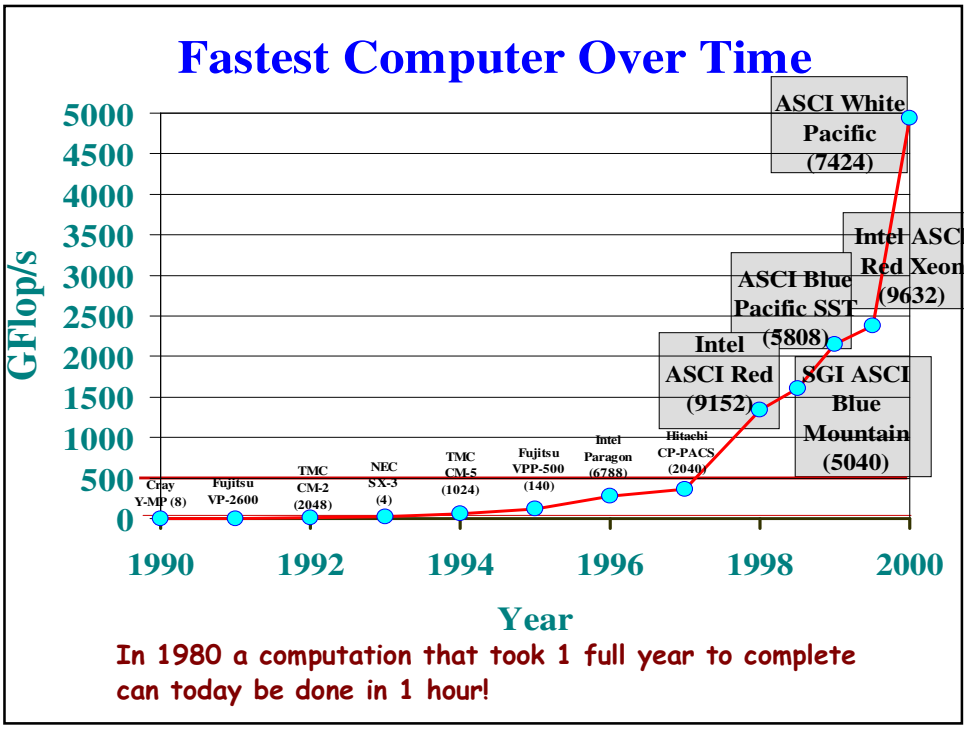
5

Fastest Computer Over Time



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

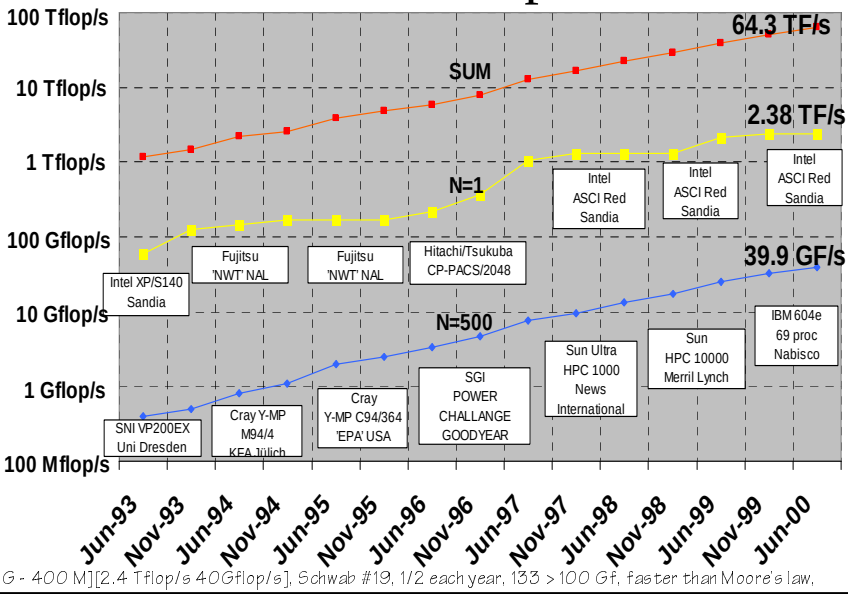

Fastest Computer Over Time
GFlop/s
500
450
400
350
300
250
200
150
100
50
0
1990
1992
1994
1996
1998
2000
Year
Cray Y-MP (8)
Fujitsu VP-2600
TMC CM-2 (2048)
NEC SX-3 (4)
TMC CM-5 (1024)
Fujitsu VPP-500 (140)
Intel Paragon (6788)
Hitachi CP-PACS (2040)
In 1980 a computation that took 1 full year to complete can now be done in 4 days!


Fastest Computer Over Time
GFlop/s
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1990
1992
1994
1996
1998
2000
Year
Cray Y-MP (8)
Fujitsu VP-2600
TMC CM-2 (2048)
NEC SX-3 (4)
TMC CM-5 (1024)
Fujitsu VPP-500 (140)
Intel Paragon (6788)
Hitachi CP-PACS (2040)
Intel ASCI Red (9152)
ASCI Blue Pacific SST (5808)
Intel ASCI Red Xeon (9632)
SGI ASCI Blue Mountain (5040)
ASCI White Pacific (7424)
In 1980 a computation that took 1 full year to complete can today be done in 1 hour!

Top 10 Machines (June 2000)

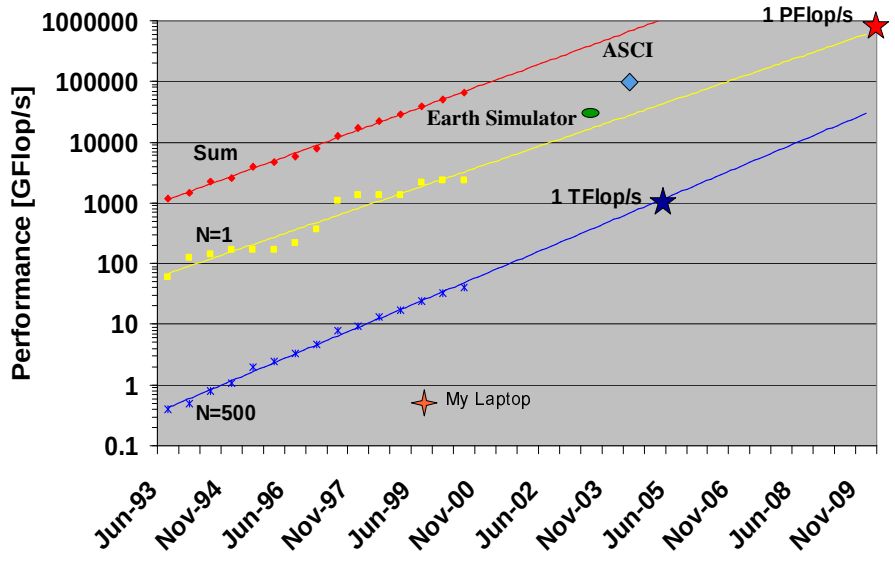
Rank	Company	Machine	Procs	Gflop/s	Place	Country	Year
1	Intel	ASCI Red	9632	2380	Sandia National Labs Albuquerque	USA	1999
2	IBM	ASCI Blue-Pacific SST, IBM SP 604e	5808	2144	Lawrence Livermore National Laboratory Livermore	USA	1999
3	SGI	ASCI Blue Mountain	6144	1608	Los Alamos National Laboratory Los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



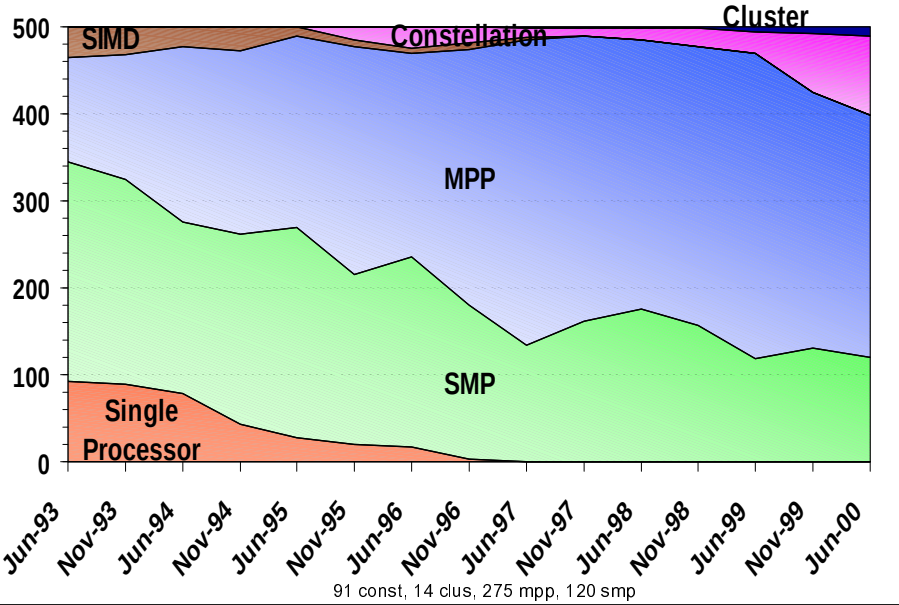
[60G - 400 M][2.4 Tflop/s 40Gflop/s], Schwab #19, 1/2 each year, 133 > 100 Gf, faster than Moore's law.

Performance Development



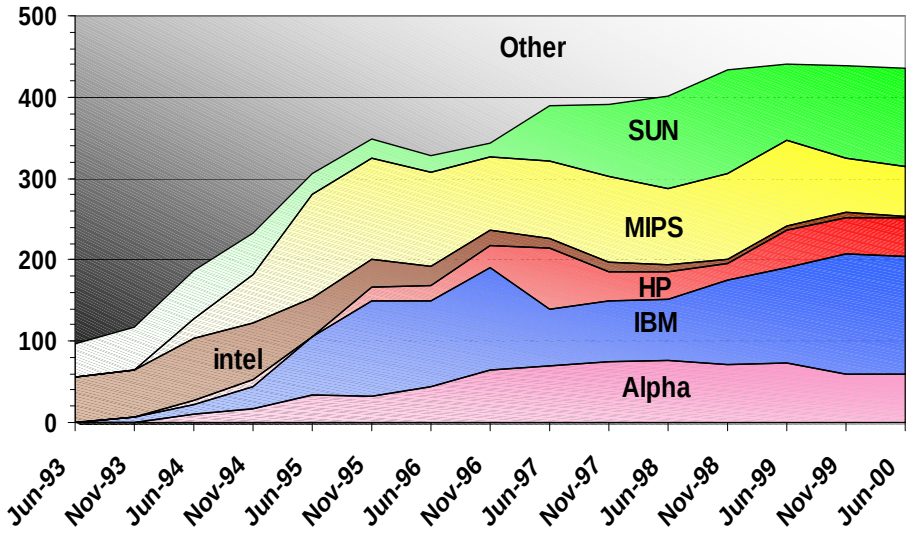
Entry 1 T 2005 and 1 P 2010

Architectures

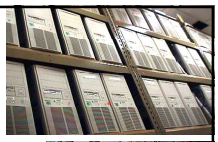


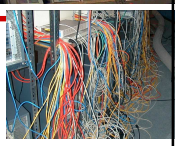
91 const, 14 clus, 275 mpp, 120 smp

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

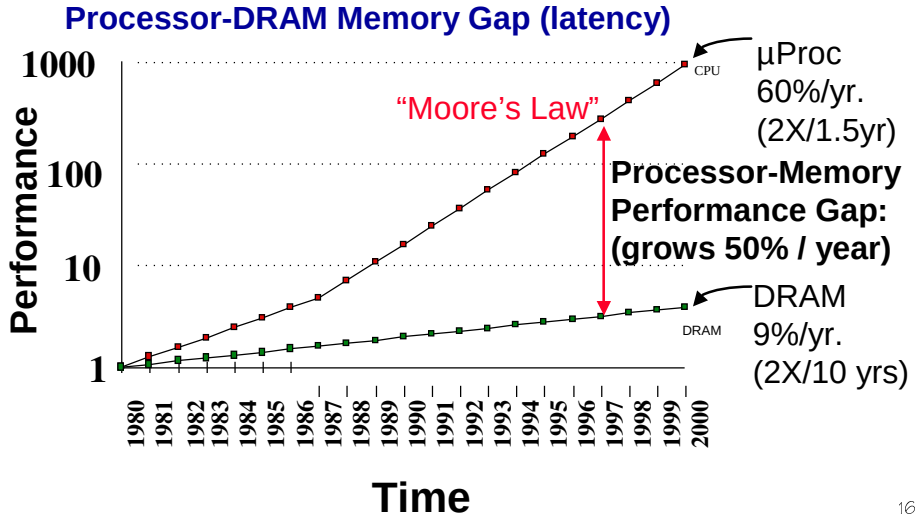
Enabled by PC hardware, networks and operating system achieving capabilities of scientific workstations at a fraction of the cost and availability of industry standard message passing libraries.

14

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	Alphleet 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	11.2
364	Self-made	Avalon Cluster	**48.6**	Los Alamos National Laboratory /CNLS Los Alamos	USA	1998	Academic	140	149.4
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

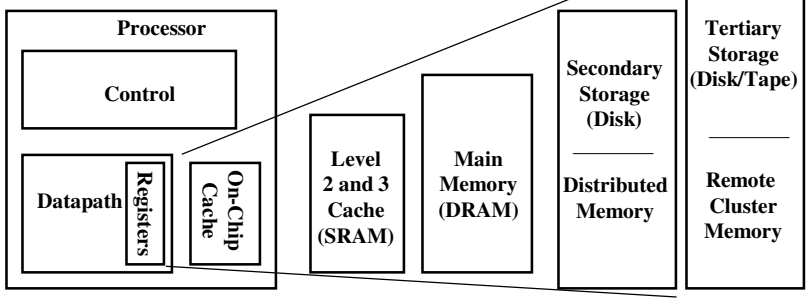
Where Does the Performance Go? or Why Should I Cares About the Memory Hierarchy?



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.



Speed (ns):	1s	10s	100s	10,000,000s (10s ms)	10,000,000,000s (10s sec)
Size (bytes):	100s	Ks	Ms	100,000 s (.1s ms)	10,000,000 s (10s ms)
				Gs	Ts

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**

18

Software Technology & Performance

- Tendency to focus on computer hardware
- Software required to bridge an ever widening gap
  - hardware capabilities and user needs
- Gaps between usable and deliverable performance is very steep
  - Performance only if the data and controls are setup just right
    - » Otherwise, dramatic performance degradations, very unstable situation
    - » Will become more unstable
- Challenge of Libraries, PSEs and Tools is formidable with Tflop/s level, even greater with Pflops, some might say insurmountable.

19

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)

20

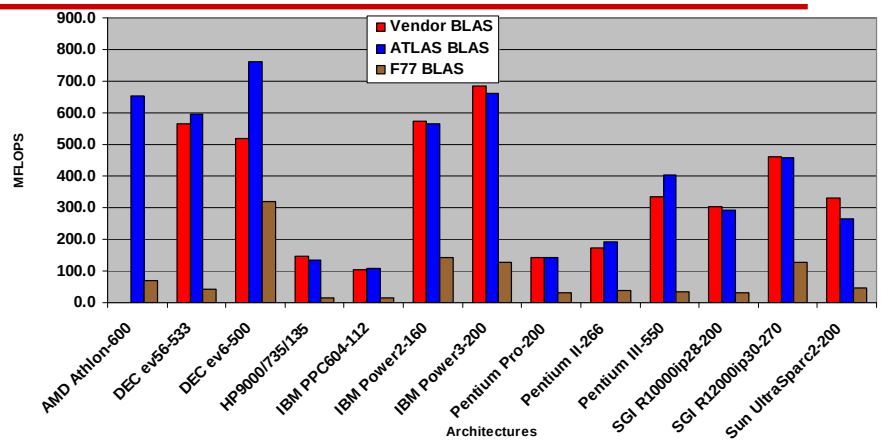
ATLAS

- 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.

21

ATLAS Across Various Architectures (DGEMM n=500)

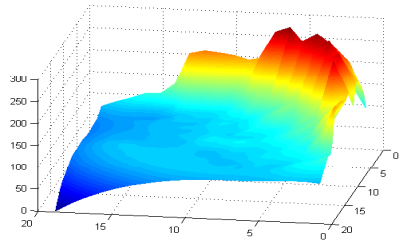


- ATLAS is faster than all other portable BLAS implementations and it is comparable with machine-specific libraries provided by the vendor.

22

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

23

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

24

Changing Algorithmic Approach

- Compute by the fast, possibly unreliable, method - "living dangerously"
  - Check a posteriori whether a problem occurred
  - Fix any problems by the reliable, perhaps much slower method
- Predictability and robustness of accuracy and performance
  - May require higher precision (128 bit floating point)
  - Perhaps interval arithmetic may play a role.
    - » However problems with composibility
- Reproducibility, fault tolerance, and auditability
- Adaptivity is a key so applications can function appropriately

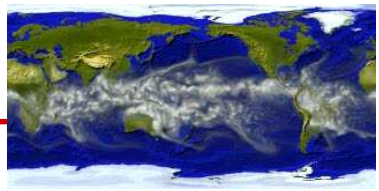
25

Futures for Numerical Algorithms and Software

- Numerical computing will be adaptive, iterative, exploratory, and intelligent.
- Determinism in numerical computing will be gone.
  - After all, its not reasonable to ask for exactness in numerical computations.
  - Audibility of the computation, reproducibility at a cost
- Importance of floating point arithmetic will be undiminished.
  - 16, 32, 64, 128 bits and beyond.
- New methods, multipole methods and their descendants will be ubiquitous.
- Standards are critical, need to evolve

26

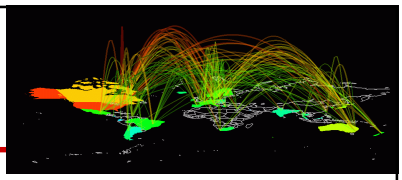
Trends in Computational Science and Engineering



- 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

27

Grid Computing



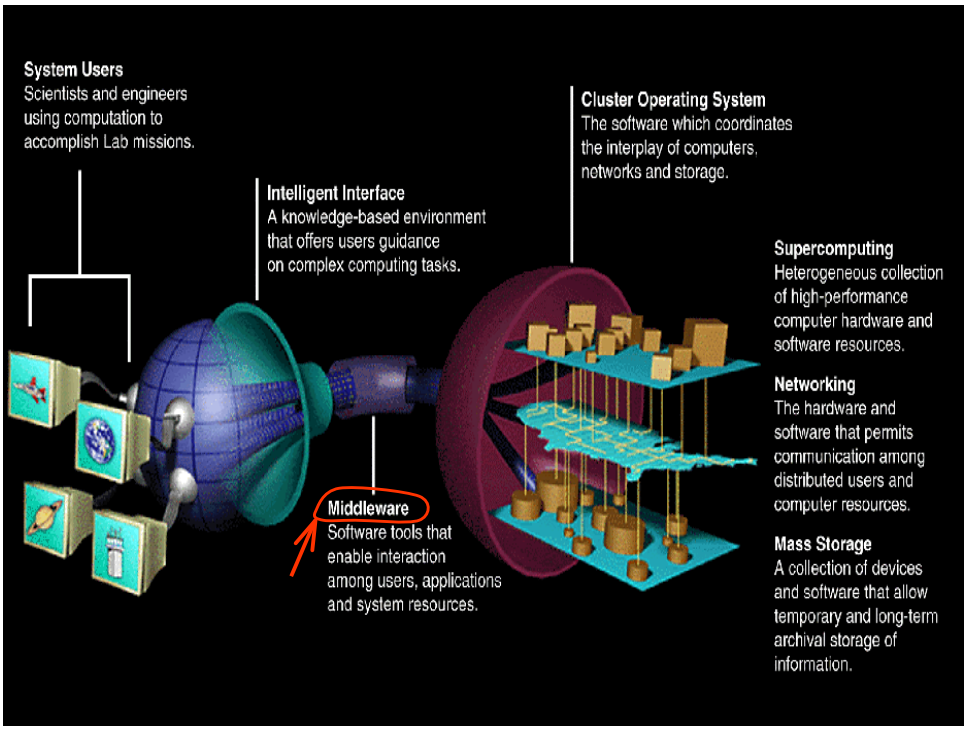
- To treat CPU cycles and software like commodities, an application should be:
  - Ubiquitous -- able to interface to the system at any point and leverage whatever is available
  - Resource Aware -- capable of managing heterogenity
  - Adaptive -- able to tailor its behavior dynamically so that it gets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

28

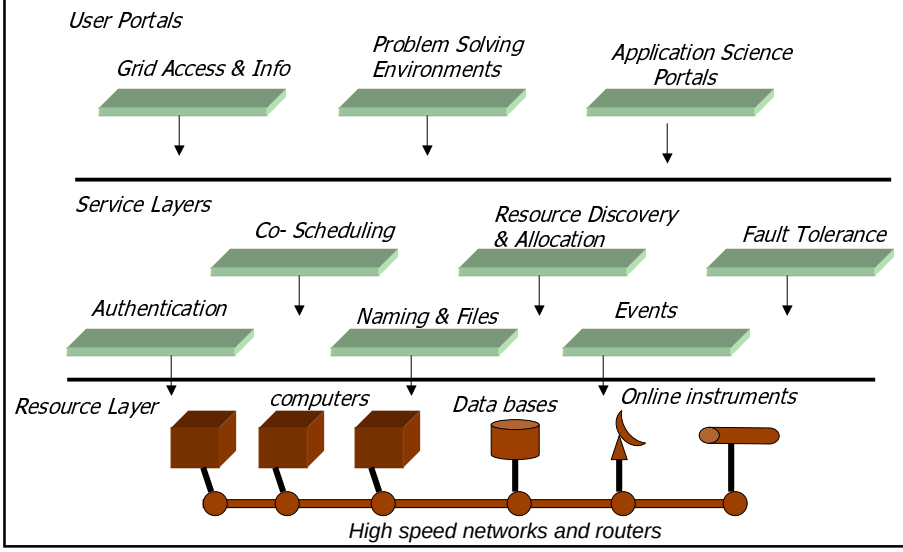
What is a Grid Anyway?

- A "Grid" is a set of resources
  - computers, networks, databases, instruments and people
  - all tied together by a common set of ubiquitous distributed services.
- The service layers are designed to provide the *seamless fabric* that allows us to build the problem solving environments (PSE) that the applications scientists want.

29



The Grid Architecture Picture



31

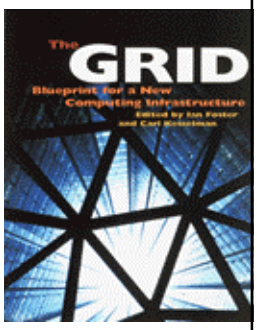
Many Software Issues to Face …

- Predictability and robustness of accuracy and performance.
- Run-time resource management and dynamic algorithm selection.
- Support for a multiplicity of programming environments and plugability.
- Reproducibility, fault tolerant, and auditability of the computations.
- New algorithmic techniques for latency tolerant and miserly bandwidth applications.
- Support for long running computations
- Major challenge adaptability of application and software
  - Resource utilization, failures, new resources

32

Many Active Research Projects

- Akenti
- AppLeS *
- Arcade
- Condor
- CUMULVUS *
- Ecce
- Electronic Notebook *
- EveryWare *
- Globus
- Habanero
- Harness *
- IceT
- IPG NAS-NASA
- JINI
- LoCI *
- Legion
- NCSA Workbench Project
- NEOS
- NetSolve *
- NINF
- Ninja
- NWS *
- PAWS
- POEMS
- PVM *
- Teraweb
- WebFlow



http://www.computingportals.org

* ORNL and/or UTK projects

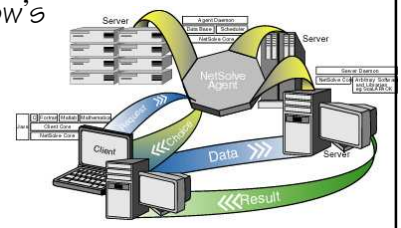
The Changing Face of Numerical Software and Libraries

- Have been involved with open source software
  - Linpack, BLAS, LAPACK, ScaLAPACK, ATLAS
  - PVM, MPI
- Looking for mechanisms that will allow community to use software in an easy to use fashion.
- Allow for portability / heterogeneous / robust
- Grid aware

34

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

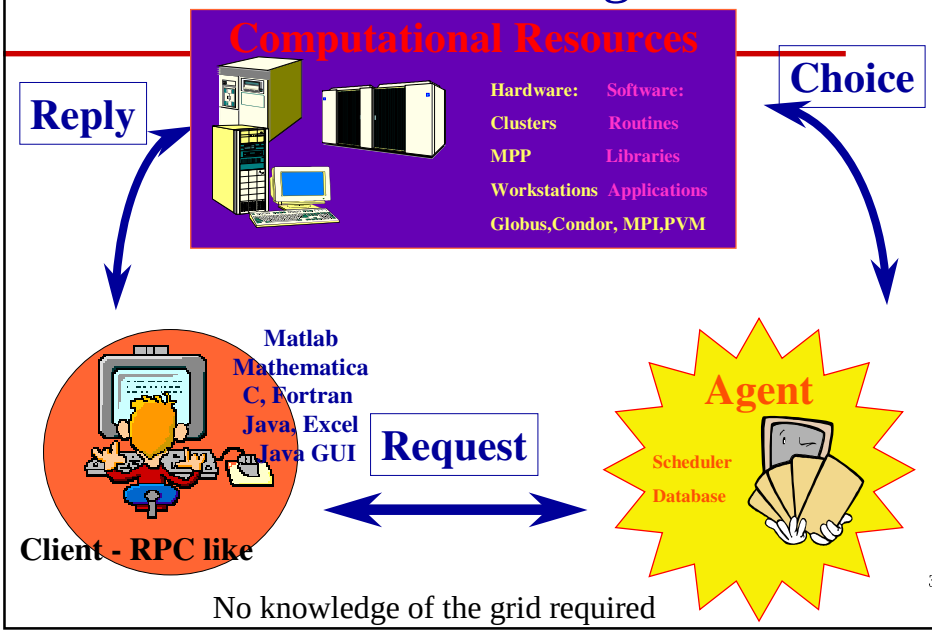


Basics

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

35

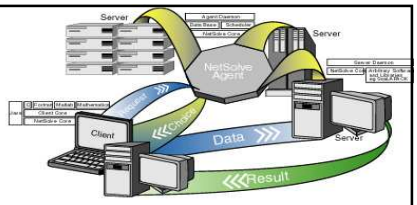
NetSolve - The Big Picture



No knowledge of the grid required

36

NetSolve Basic Usage Scenarios



- Grid based numerical library routines
  - Users doesn't have to have software library on their machine
- Task farming applications
  - "Pleasantly parallel" execution
  - eg Parameter studies
- Remote application execution
  - Complete applications with user specifying input parameters and receiving output
- "Blue Collar" Grid Based Computing
  - Does not require deep knowledge of network programming
  - Level of expressiveness right for many users
  - User can set things up, no "su" required
- Smart Libraries
  - Decision on which algorithm and software based on input data
  - Decouple interface (common component)

37

NetSolve - MATLAB Interface

Synchronous Call

```
>> define sparse matrix A
>> define rhs
>> [x, its] = netsolve( 'itmeth', 'petsc', A, rhs );
...
>> [x, its] = netsolve( 'itmeth', 'aztec', A, rhs );
>> [x, its] = netsolve( 'solve', 'superlu', A, rhs );
>> [x, its] = netsolve( 'solve', 'ma28', A, rhs );
```

Asynchronous Calls also available

NetSolve - FORTRAN Interface

Easy to 'switch' to NetSolve

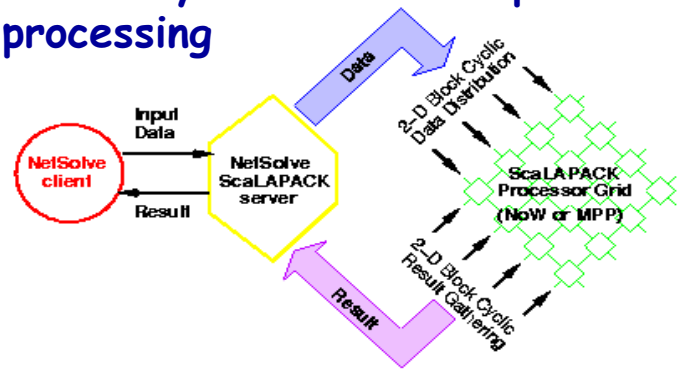
```
parameter( MAX = 100)
double precision A(MAX,MAX), B(MAX)
integer IPIV(MAX), N, INFO, LWORK
integer NSINFO

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

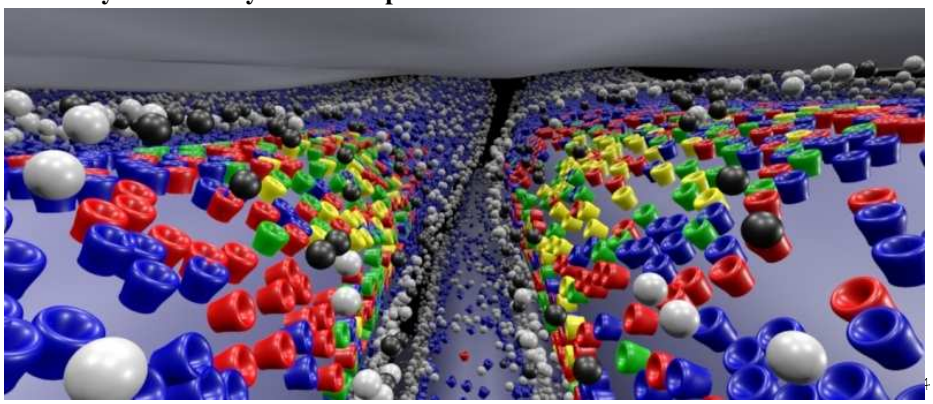


- NetSolve takes care of the starting the message passing system, data distribution, and returning the results.

40

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

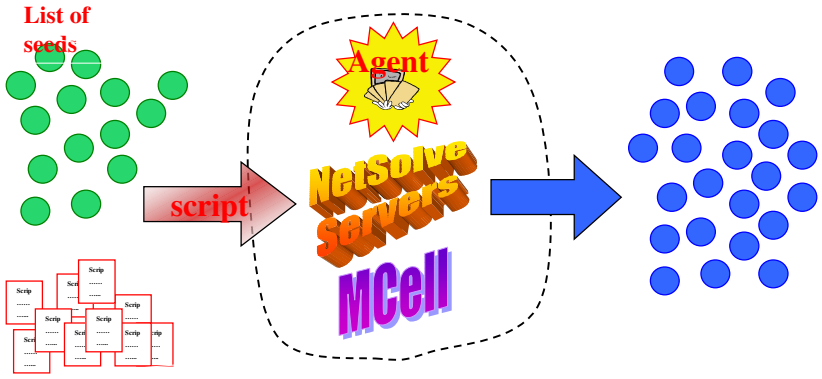
- Developed at: Salk Institute (T. Bartol), CMU (J. Stiles)
- Study how neurotransmitters diffuse and activate receptors in synapses
- blue unbounded, red singly bounded, green doubly bounded closed, yellow doubly bounded open



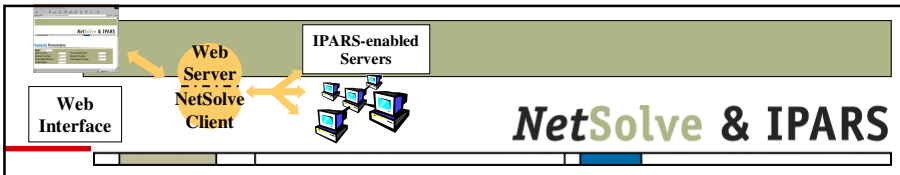
41

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

- Developed at: Salk Institute, CMU
- In the past, manually run on available workstations
- Transparent Parallelism, Load balancing, Fault-tolerance
- Fits the farming semantic and need for NetSolve



42



- 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.

43

Conclusion

- Exciting time to be in scientific computing
- Network computing is here
- Computational grids enable qualitatively new applications but introduce hard technical issues
- Important to develop algorithms and software that will work effectively in this environment
- Large-scale deployment, application underway

44

Contributors to These Ideas

- Top500
  - Erich Strohmaier, UTK
  - Hans Meuer, Mannheim U
- ATLAS
  - Antoine Petitet, UTK
  - Clint Whaley, 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/

www.netlib.org/netsolve/

www.cs.utk.edu/~dongarra/

45

Opportunities in the group for visiting and permanent positions