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/

1

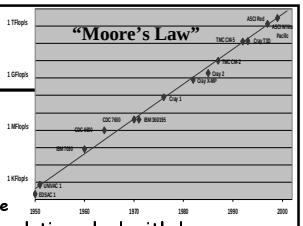
High Performance Computers

- ~ 20 years ago
  - 1×10^6 Floating Point Ops/sec (Mflop/s)
    - Scalar based
- ~ 10 years ago
  - 1×10^9 Floating Point Ops/sec (Gflop/s)
    - Vector & Shared memory computing, bandwidth aware
    - Block partitioned, latency tolerant
- ~ Today
  - 1×10^{12} Floating Point Ops/sec (Tflop/s)
    - Highly parallel, distributed processing, message passing, network based
    - data decomposition, communication/computation
- ~ 10 years away
  - 1×10^{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

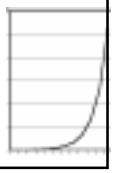
2

"Moore's Wall"

— Horst Simon, NERSC



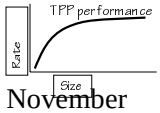
- Moore's Law predicts exponential growth
  - Performance doubling every 18 months
  - Usually plotted on semi-log scale,appears as straight line
- Human experience has a hard time deal with log scale
  - We are sitting at the bend of an exponential curve
- From our perspective Moore's Law appears as a "wall"
  - In a few years technology will again be completely different
  - Hard to predict what the future will be.



TOP500

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

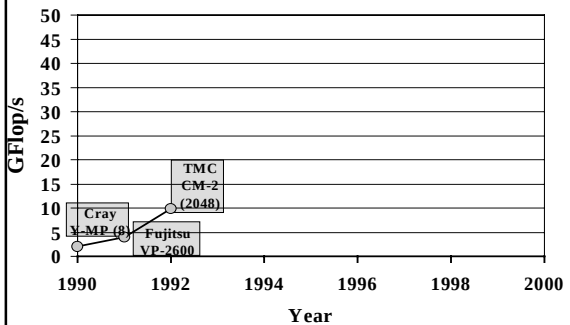
  $Ax=b$, *dense problem*



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

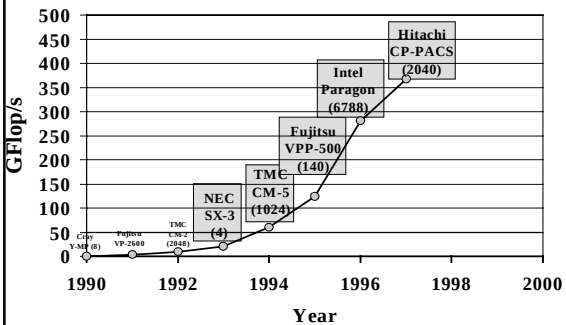
4

Fastest Computer Over Time

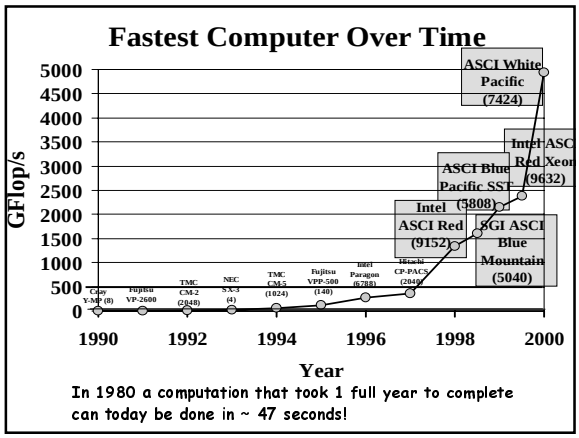


In 1980 a computation that took 1 full year to complete can now be done in ~ 10 hours!

Fastest Computer Over Time



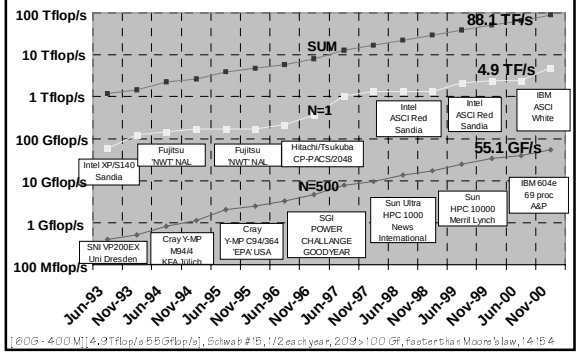
In 1980 a computation that took 1 full year to complete can now be done in ~ 16 minutes!



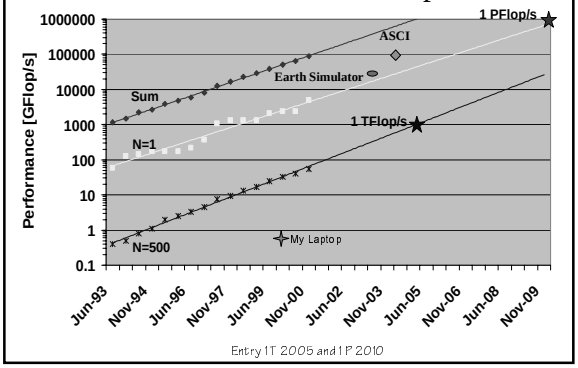
Top 10 Machines (Nov 2000)

Rank	Company	Machine	Procs	Gflop/s	Place	Country	Year
1	IBM	ASCI White	8192	4938	Livermore National Laboratory Livermore		2000
2	Intel	ASCI Red	9632	2380	Sandia National Labs Albuquerque	USA	1999
3	IBM	ASCI Blue-Pacific SST, IBM SP 604e	5808	2144	Lawrence Livermore National Laboratory Livermore	USA	1999
4	SGI	ASCI Blue Mountain	6144	1608	Los Alamos National Laboratory Los Alamos	USA	1998
5	IBM	SP Power3 375 MHz	1336	1417	Naval Oceanographic Office (NAVOCEANO)	USA	2000
6	IBM	SP Power3 375 MHz	1104	1179	National Center for Environmental Protection	USA	2000
7	Hitachi	SR8000-F1/112	112	1035	Leibniz Rechenzentrum Muenchen	Germany	2000
8	IBM	SP Power3 375 MHz, 8 way	1152	929	UCSD/San Diego Supercomputer Center	USA	2000
9	Hitachi	SR8000-F1/100	100	917	High Energy Accelerator Research Organization /KEK Tsukuba	Japan	2000
10	Cray Inc.	T3E1200	1084	892	Government	USA	1998

Performance Development



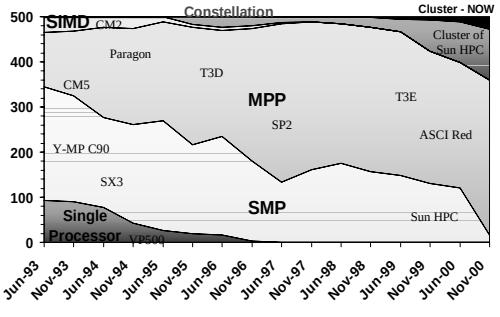
Performance Development



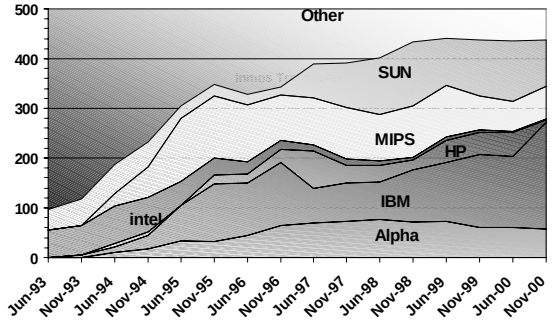
Petaflop Computers Within the Next Decade

- Five basis design points:
  - Conventional technologies
    - 4.8 GHz processor, 8000 nodes, each w/16 processors
  - Processing-in-memory (PIM) designs
    - Reduce memory access bottleneck
  - Superconducting processor technologies
    - Digital superconductor technology, Rapid Single-Flux-Quantum (RSFQ) logic & hybrid technology multi-threaded (HTMT)
  - Special-purpose hardware designs
    - Specific applications e.g. GRAPE Project in Japan for gravitational force computations
  - Schemes utilizing the aggregate computing power of processors distributed on the web
    - SETI@home

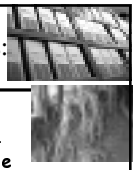
Top 500 Architectures



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

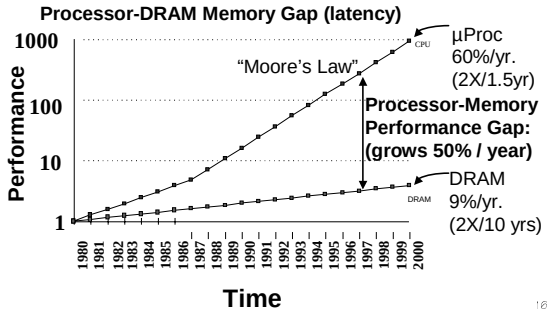
Enabled by PC hardware, networks and operating system achieving capabilities of scientific workstations at a fraction of the cost and availability of industry standard message passing libraries. However, much more of a contact sport.

14

- Peak performance
- Interconnection
- http://clusters.top500.org
- Benchmark results to follow in the coming months

15

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



16

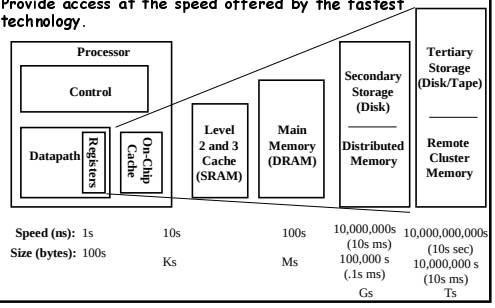
Optimizing Computation and Memory Use

- Computational optimizations
  - Theoretical peak: (# fpus)*(flops/cycle) * Mhz
    - PIII: (1 fpu)*(1 flop/cycle)*(850 Mhz) = 850 MFLOP/s
    - Athlon: (2 fpu)*(1flop/cycle)*(600 Mhz) = 1200 MFLOP/s
    - Power3: (2 fpu)*(2 flops/cycle)*(375 Mhz) = 1500 MFLOP/s
- Operations like:
  - $\alpha = x^T y$: 2 operands (16 Bytes) needed for 2 flops; at 850 Mflop/s will requires 1700 MW/s bandwidth
  - $y = \alpha x + y$: 3 operands (24 Bytes) needed for 2 flops; at 850 Mflop/s will requires 2550 MW/s bandwidth
- Memory optimization
  - Theoretical peak: (bus width) * (bus speed)
    - PIII: (32 bits)*(133 Mhz) = 532 MB/s = 66.5 MW/s
    - Athlon: (64 bits)*(133 Mhz) = 1064 MB/s = 133 MW/s
    - Power3: (128 bits)*(100 Mhz) = 1600 MB/s = 200 MW/s

17

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.

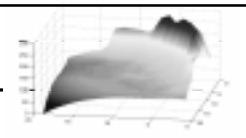


Self-Adapting Numerical Software (SANS)

- Today's processors can achieve high-performance, but this requires extensive machine-specific hand tuning.
- Operations like the BLAS require many man-hours / platform
  - Software lags far behind hardware introduction
  - Only done if financial incentive is there
- Hardware, compilers, 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.
- Need for quick/dynamic deployment of optimized routines.
- ATLAS - Automatic Tuned Linear Algebra Software

19

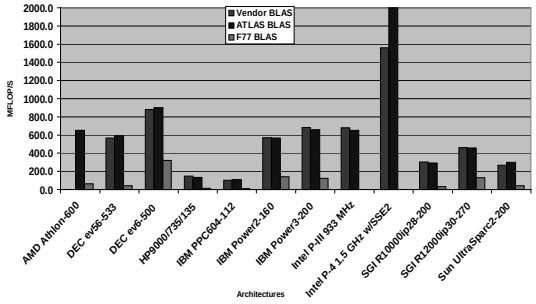
Software Generation Strategy



- Level 1 cache multiply optimizes for:
  - TLB access
  - L1 cache reuse
  - FP unit usage
  - Memory fetch
  - Register reuse
  - Loop overhead minimization
- Takes about 30 minutes 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
- Today ATLAS in use by Matlab, Mathematica, Octave, Maple, Debian, Scyld Beowulf, SuSE, …

20

ATLAS (DGEMM n = 500)



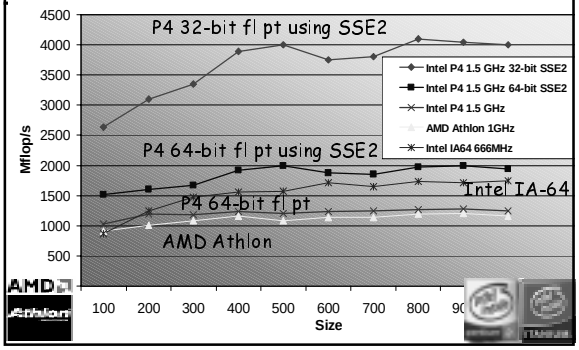
- ATLAS is faster than all other portable BLAS implementations and it is comparable with machine-specific libraries provided by the vendor.

Related Tuning Projects

- PHiPAC
  - Portable High Performance ANSI C www.icsi.berkeley.edu/~bilmes/phipac initial automatic GEMM generation project
- FFTW Fastest Fourier Transform in the West
  - www.fftw.org
- UHFFT
  - tuning parallel FFT algorithms
  - rodin.cs.uh.edu/~mirkovic/fft/parfft.htm
- SPIRAL
  - Signal Processing Algorithms Implementation Research for Adaptable Libraries maps DSP algorithms to architectures
- Sparsity
  - Sparse-matrix-vector and Sparse-matrix-matrix multiplication www.cs.berkeley.edu/~ejim/publication/ tunes code to sparsity structure of matrix more later in this tutorial

22

ATLAS Matrix Multiply (64 & 32 bit floating point results)



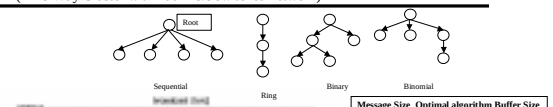
Machine-Assisted Application Development and Adaptation

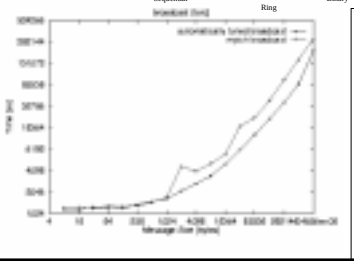
- **Communication libraries**
  - Optimize for the specifics of one's configuration.
- **Algorithm layout and implementation**
  - Look at the different ways to express implementation

24

Work in Progress: ATLAS-like Approach Applied to Broadcast

(PII 8 Way Cluster with 100 Mb/s switched network)

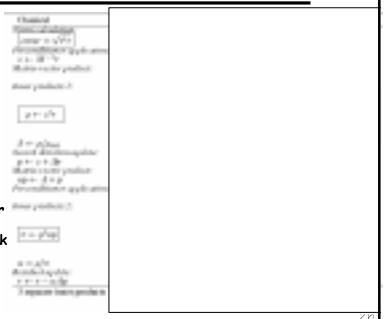




Message Size (bytes)	Optimal algorithm	Buffer Size (bytes)
8	binomial	8
16	binomial	16
32	binary	32
64	binomial	64
128	binomial	128
256	binomial	256
512	binomial	512
1K	sequential	1K
2K	binary	2K
4K	binary	2K
8K	binary	2K
16K	binary	4K
32K	binary	4K
64K	ring	4K
128K	ring	4K
256K	ring	4K
512K	ring	4K
1M	binary	4K

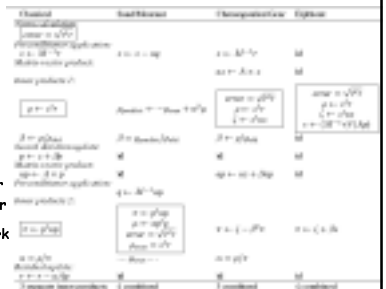
CG Variants by Dynamic Selection at Run Time

- Variants combine inner products to reduce communication bottleneck at the expense of more scalar ops.
- Same number of iterations, no advantage on a sequential processor
- With a large number of processor and a high-latency network may be advantages.
- Improvements can range from 15% to 50% depending on size.



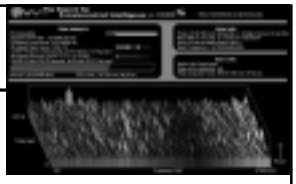
CG Variants by Dynamic Selection at Run Time

- Variants combine inner products to reduce communication bottleneck at the expense of more scalar ops.
- Same number of iterations, no advantage on a sequential processor
- With a large number of processor and a high-latency network may be advantages.
- Improvements can range from 15% to 50% depending on size.



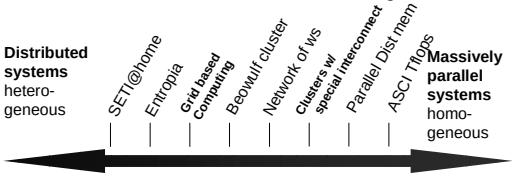
SETI@home

- 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 thousands of other participants.



- **Largest distributed computation project in existence**
  - ~ 400,000 machines
  - Averaging 26 Tflop/s
- Today many companies trying this for profit.

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

The Grid

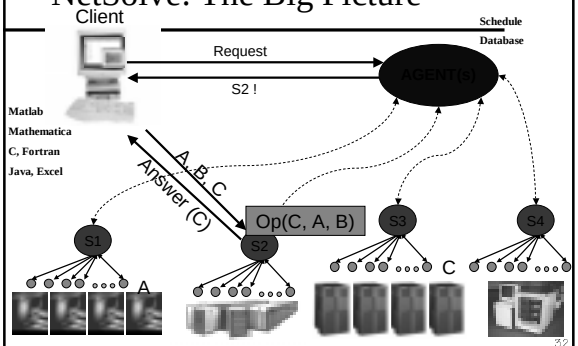
- To treat CPU cycles and software like commodities.
- Napster on steroids.
- Enable the coordinated use of geographically distributed resources – in the absence of central control and existing trust relationships.
- Computing power is produced much like utilities such as power and water are produced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

NetSolve
Network Enabled Server

- NetSolve is an example of a grid based hardware/software server.
- Easy-of-use paramount
- Based on a RPC model but with …
  - resource discovery, dynamic problem solving capabilities, load balancing, fault tolerance asynchronicity, security, …
- Other examples are NEOS from Argonne and NINF Japan.
- Use a resource, not tie together geographically distributed resources for a single application.

31

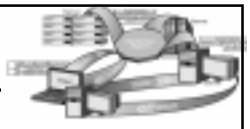
NetSolve: The Big Picture



32

No knowledge of the grid required, RPC like.

Basic Usage Scenarios



- Grid based numerical library routines
  - User doesn't have to have software library on their machine, LAPACK, SuperLU, ScaLAPACK, PETSc, AZTEC, ARPACK
- 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
  - In use today, up to 200 servers in 9 countries

33

Futures for Numerical Algorithms and Software

- Numerical software will be adaptive, exploratory, and intelligent
- Determinism in numerical computing will be gone.
  - After all, its not reasonable to ask for exactness in numerical computations.
  - Auditability of the computation, reproducibility at a cost
- Importance of floating point arithmetic will be undiminished.
  - 16, 32, 64, 128 bits and beyond.
- Reproducibility, fault tolerance, and auditability
- Adaptivity is a key so applications can function appropriately

34

Contributors to These Ideas

- Top500
  - Erich Strohmaier, LBL
  - Hans Meuer, Mannheim U
- Linear Algebra
  - Victor Eijkhout, UTK
  - Piotr Luszczek, UTK
  - 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/

Many opportunities within the group at Tennessee

35