CLUSTER2001

IEEE International Conference on Cluster Computing

High Performance Computing and Trends

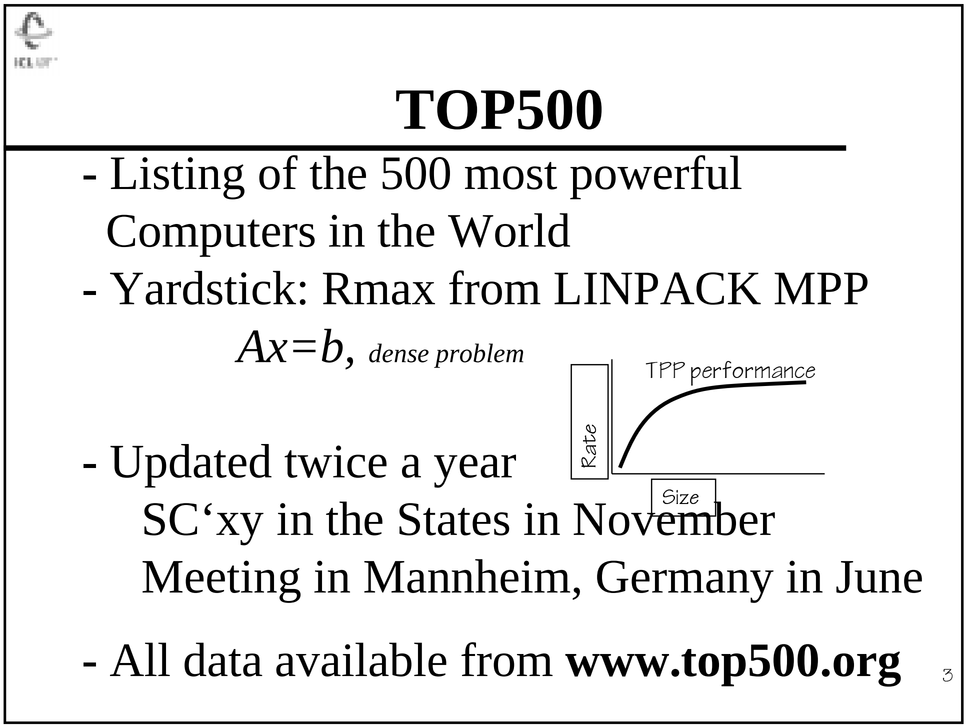
Jack Dongarra
Innovative Computing Laboratory
University of Tennessee

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

Sutton Place Hotel, Newport Beach
Oct. 8-11, 001

Outline

- **Trends**
  - **Through the "eyes" of the Top500**
- **Self Adapting Numerical Software (SANS) Effort**
  - **Software Technology to aid in high performance on clusters and commodity processors.**



TOP500

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

  Ax=b, dense problem

- Updated twice a year

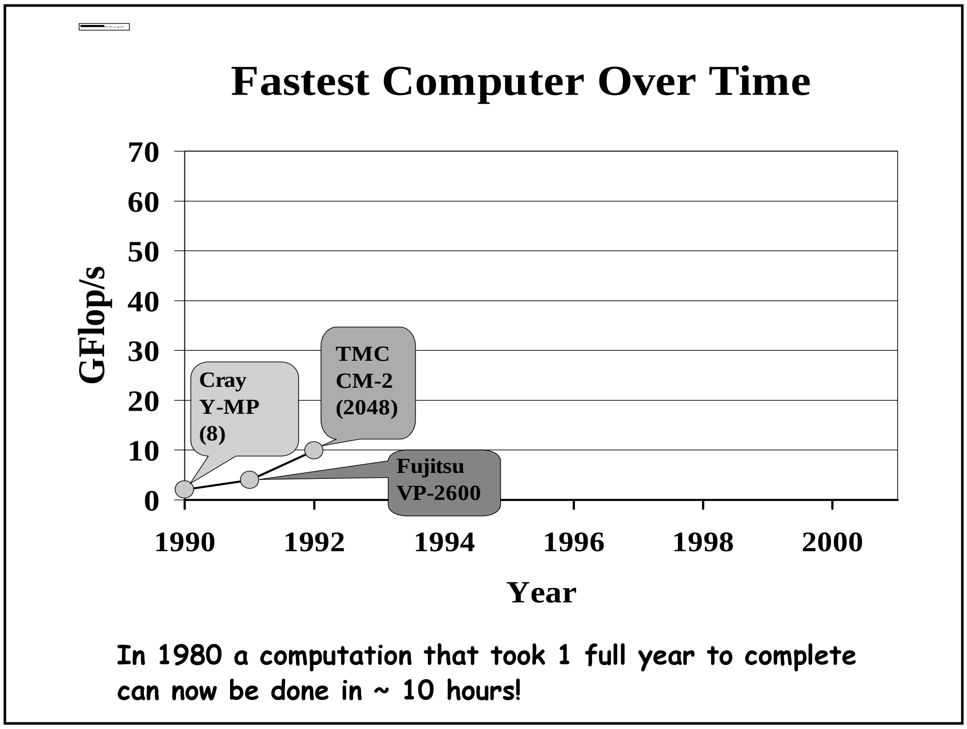
  SC‘xy in the States in November

  Meeting in Mannheim, Germany in June

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

TPP performance
Rate
Size

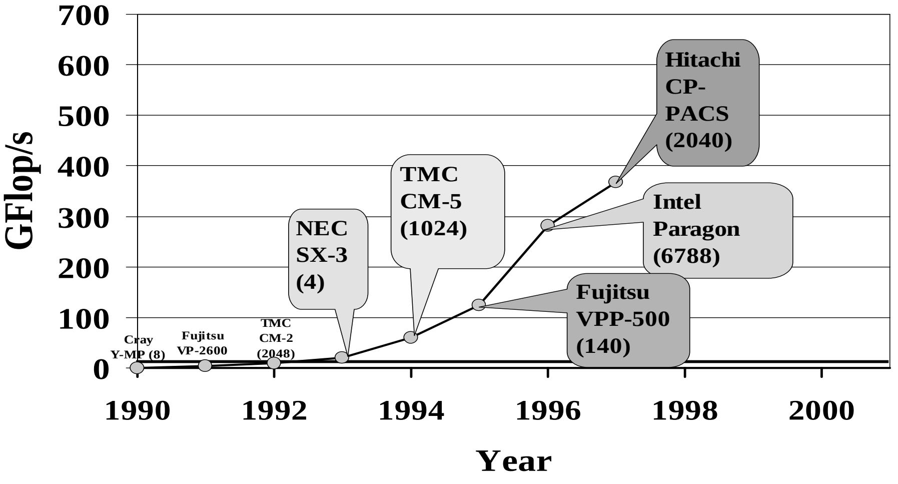
3



Fastest Computer Over Time

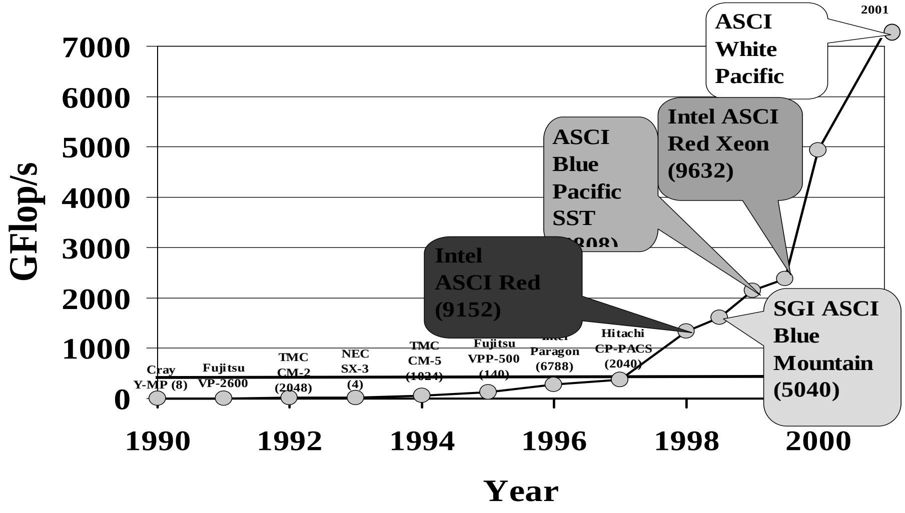
In 1980 a computation that took 1 full year to complete can now be done in ~ 10 hours!

Fastest Computer Over Time



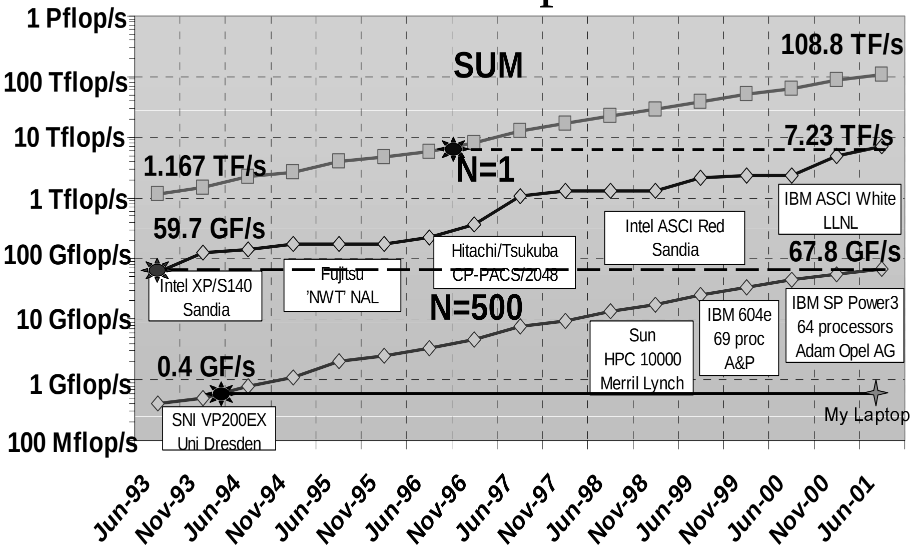
In 1980 a computation that took 1 full year to complete can now be done in ~ 16 minutes!

Fastest Computer Over Time



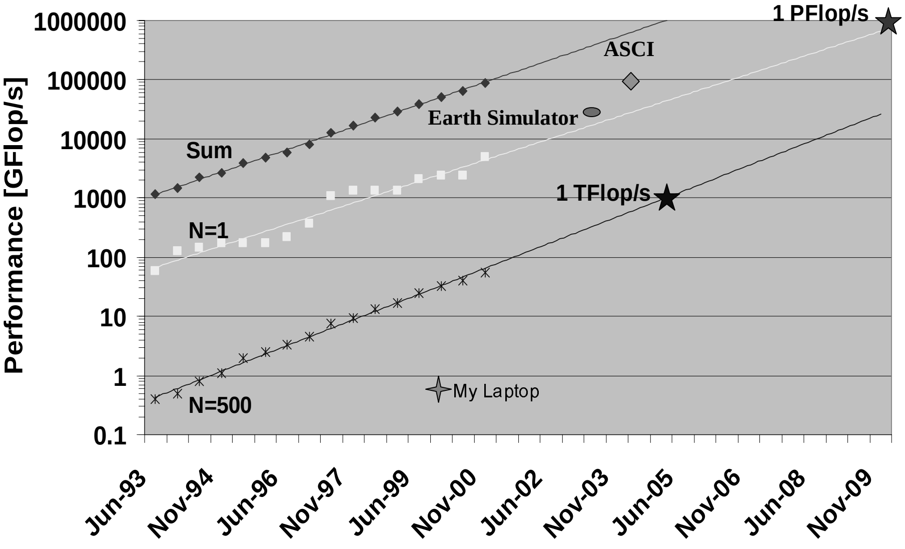
In 1980 a computation that took 1 full year to complete can today be done in ~ 27 seconds!

Performance Development



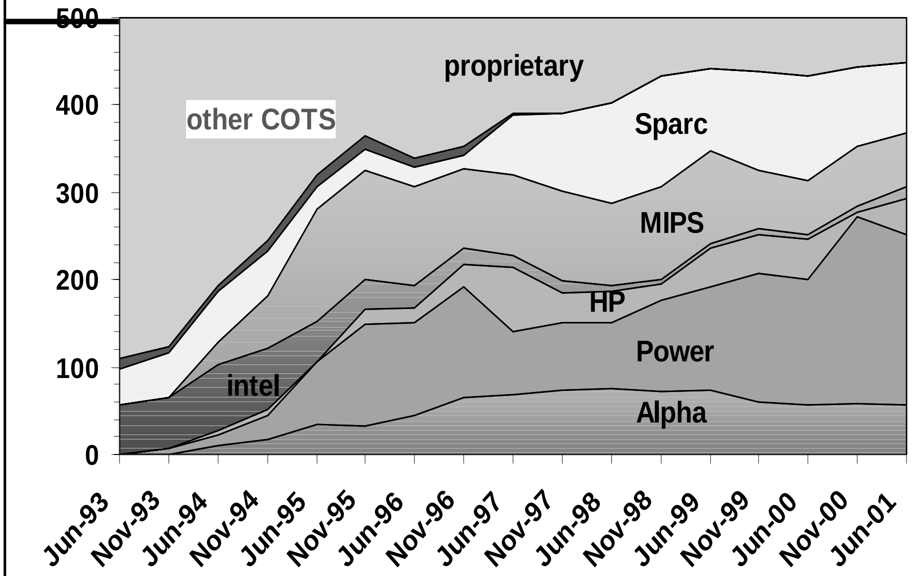
[60G - 400 M][7.2 Tflop/s 68Gflop/s], Moore's law, US Government = 27.7 TF, 51 machines

Performance Development



Entry 1 T 2005 and 1 P 2010

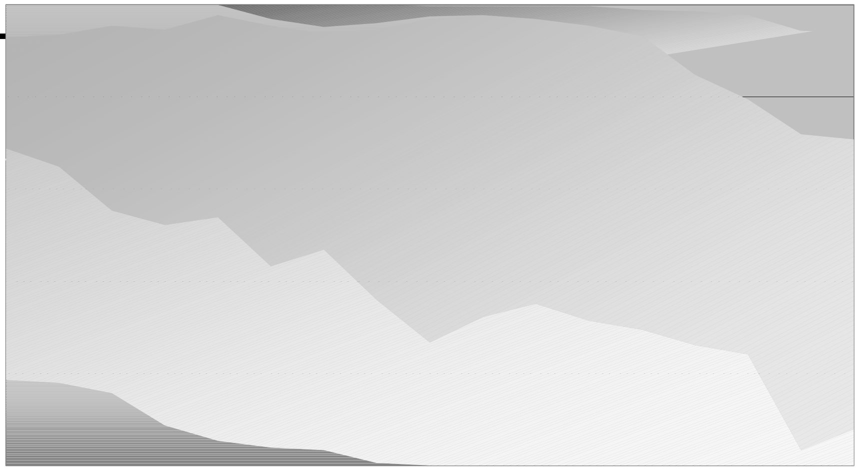
Chip Technology



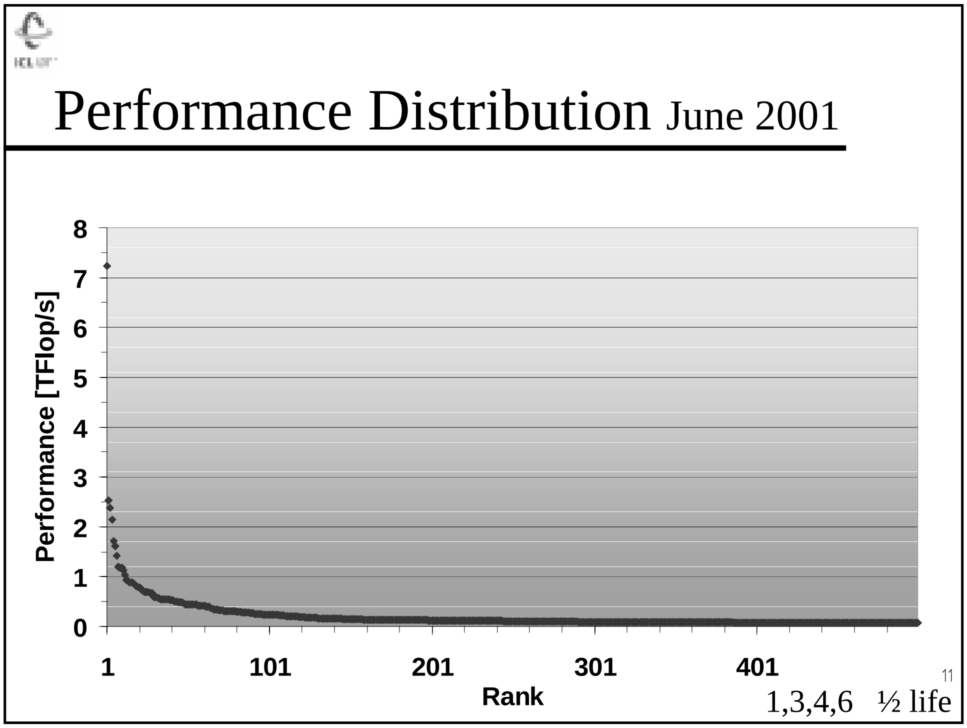
9

16 companies 8 now

Architectures

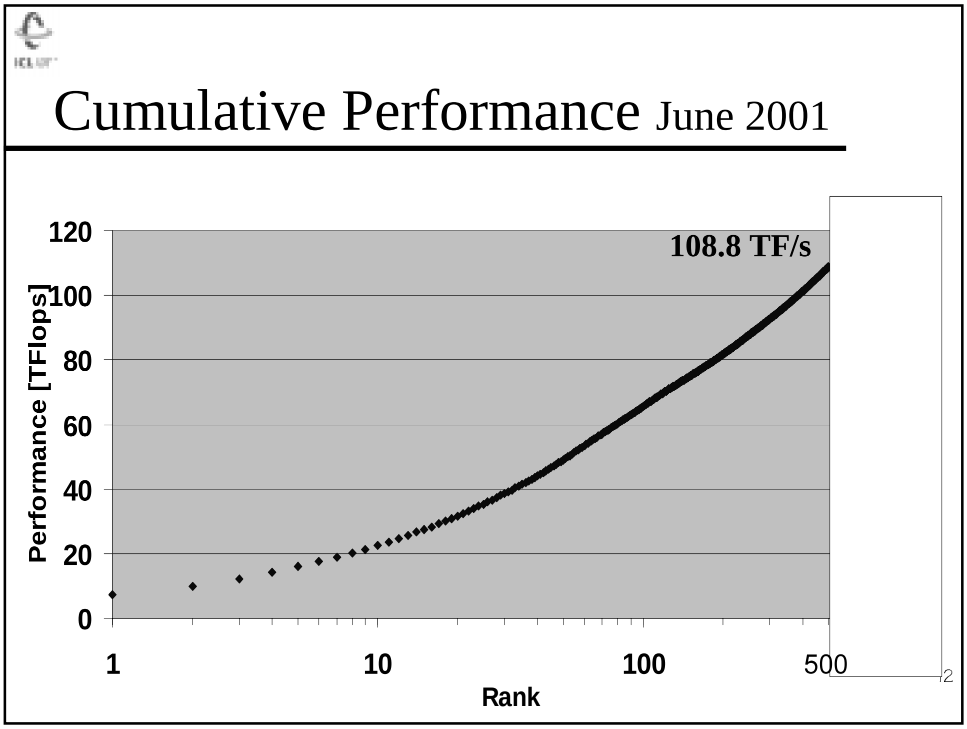


10

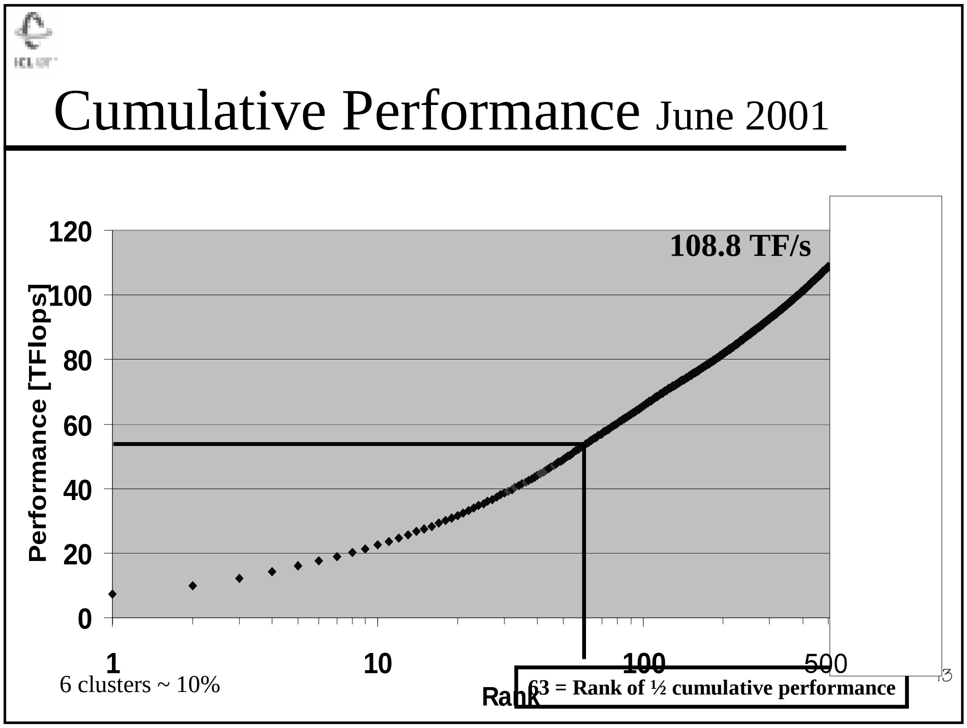


Performance Distribution June 2001

1,3,4,6 ½ life



Cumulative Performance June 2001

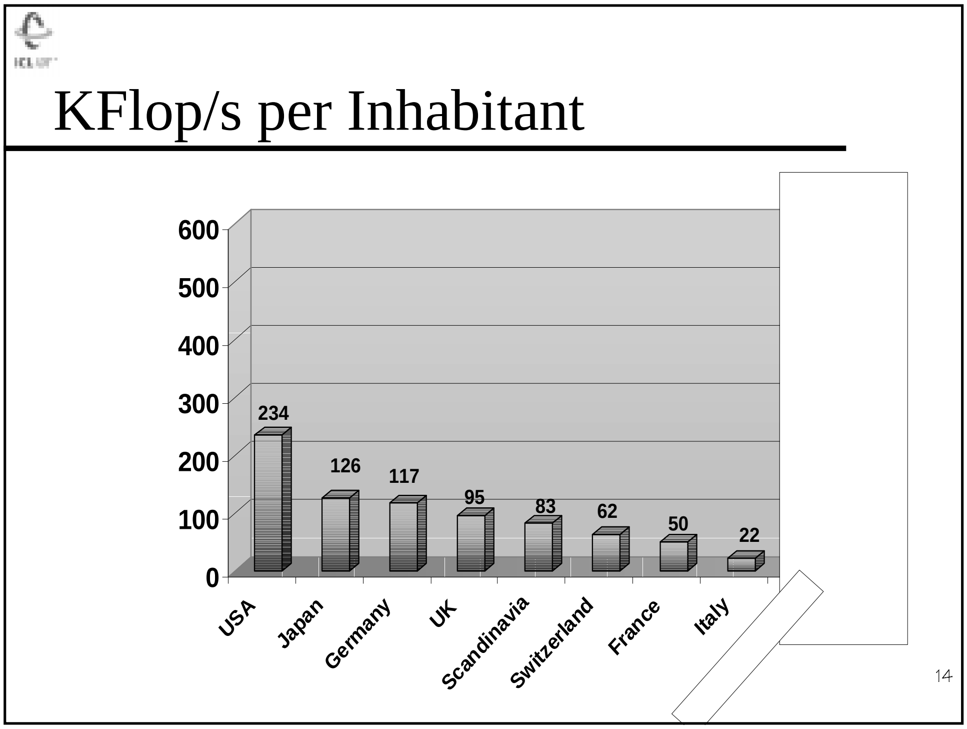


Cumulative Performance June 2001

6 clusters ~ 10%

63 = Rank of ½ cumulative performance

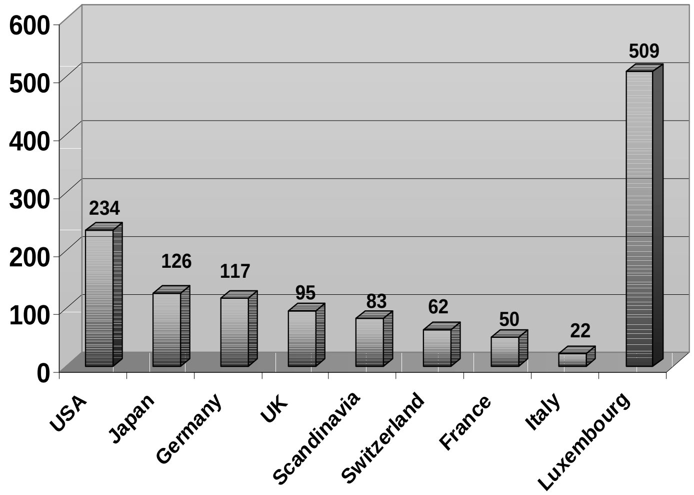
13



KFlop/s per Inhabitant

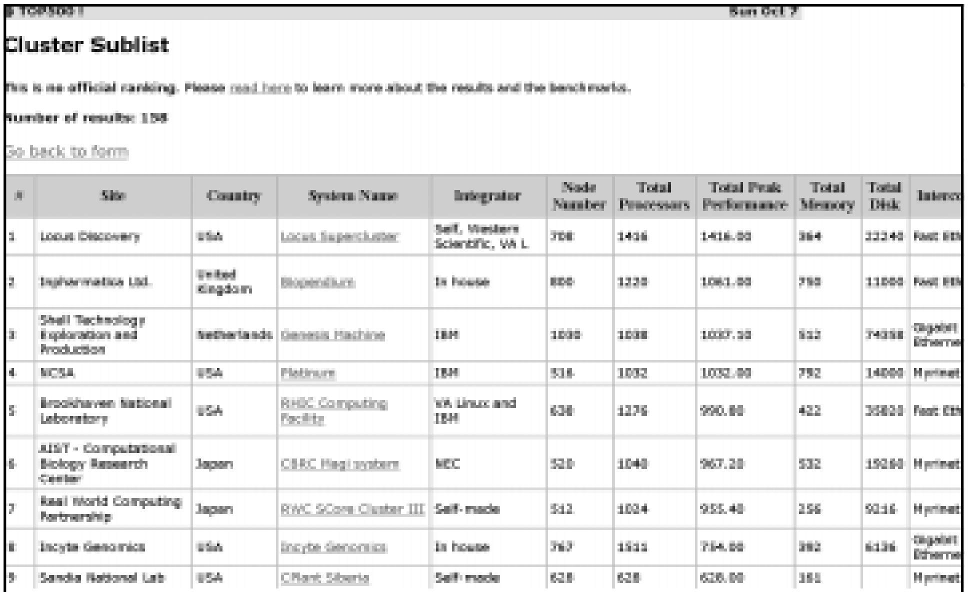
14

KFlop/s per Inhabitant



What’s new with the TOP500?

- Benchmark Implementation HPL:
  High Performance Linpack
  - **http://icl.cs.utk.edu/hpl/**
  - **Needs only**
    - **MPI**
    - **BLAS or VSIPL**
  - **Highly scalable and efficient**
- **Top Clusters**
  - **List by peak performance today**
  - **Information given on processors, interconnect, etc**
  - **http://clusters.top500.org**



§ TOP500 ! Sun Oct 7

Cluster Sublist

This is no official ranking. Please read here to learn more about the results and the benchmarks.

Number of results: 158

Go back to form

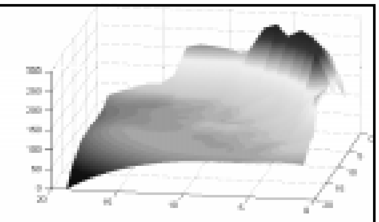
#	Site	Country	System Name	Integrator	Node Number	Total Processors	Total Peak Performance	Total Memory	Total Disk	Interc
1	Locus Discovery	USA	Locus Supercluster	Self, Western Scientific, VA L	708	1416	1416.00	364	22240	Fast Eth
2	Inpharmatica Ltd.	United Kingdom	Biopendium	In house	800	1220	1061.00	750	11000	Fast Eth
3	Shell Technology Exploration and Production	Netherlands	Genesis Machine	IBM	1030	1038	1037.10	512	74358	Gigabit Etherne
4	NCSA	USA	Platinum	IBM	516	1032	1032.00	792	14000	Myrinet
5	Brookhaven National Laboratory	USA	RHIC Computing Facility	VA Linux and IBM	638	1276	990.80	422	35820	Fast Eth
6	AIST - Computational Biology Research Center	Japan	CBRC Magi system	NEC	520	1040	967.20	532	15260	Myrinet
7	Real World Computing Partnership	Japan	RWC SCore Cluster III	Self-made	512	1024	955.40	256	9216	Myrinet
8	Incyte Genomics	USA	Incyte Genomics	In house	767	1511	734.00	382	6136	Gigabit Etherne
9	Sandia National Lab	USA	CPlant Siberia	Self-made	628	628	628.00	161		Myrinet

- ♦ Today the ranking is by theoretical peak performance.
- ♦ Interconnect, processors, memory, OS, application area, …
- ♦ http://clusters.top500.org
- ♦ Benchmark results to follow in the coming months

Self-Adapting Numerical Software (SANS) Effort

- ♦ The complexities of modern processors or clusters makes it difficult to analytically predict or model by hand the performance.
- ♦ Operations as simple as 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.

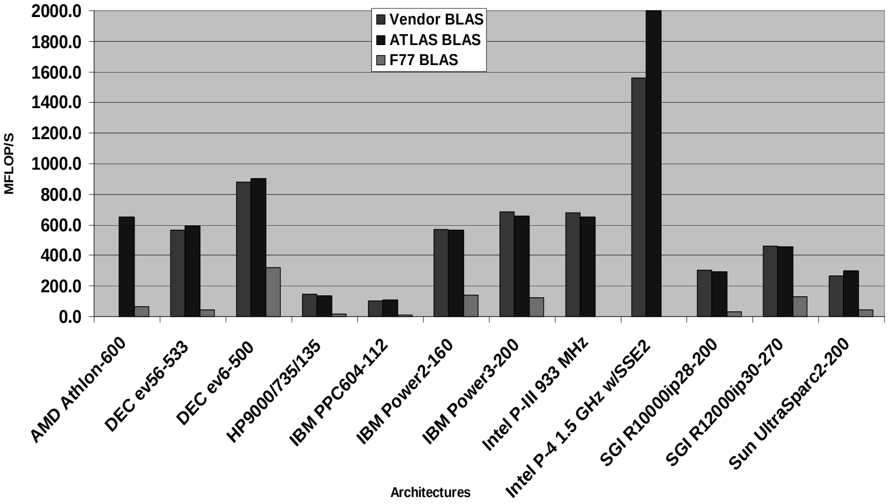
Software Generation Strategy - ATLAS BLAS



- ♦ Parameter study of the hw
- ♦ Generate multiple versions of code, w/difference values of key performance parameters
- ♦ Run and measure the performance for various versions
- ♦ Pick best and generate library
- ♦ Level 1 cache multiply optimizes for:
  - ➢ TLB access
  - ➢ L1 cache reuse
  - ➢ FP unit usage
  - ➢ Memory fetch
  - ➢ Register reuse
  - ➢ Loop overhead minimization
- ♦ Takes ~ 20 minutes to run.
- ♦ "New" model of high performance programming where critical code is machine generated using parameter optimization.
- ♦ Designed for RISC arch
  - ➢ Super Scalar
  - ➢ Need reasonable C compiler
- ♦ Today ATLAS in use by Matlab, Mathematica, Octave, Maple, Debian, Scyld Beowulf, SuSE, …

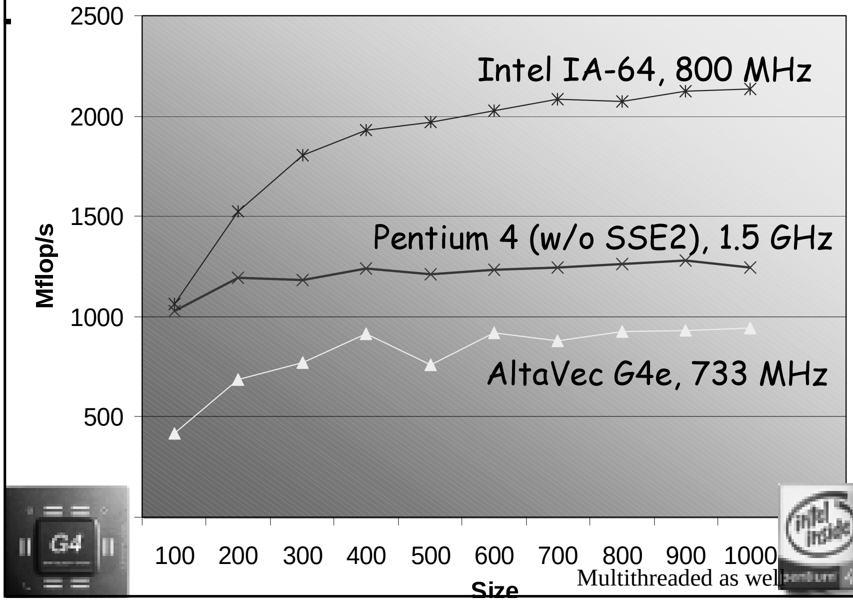
19

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.

ATLAS Matrix Multiply (64-bit floating point results)



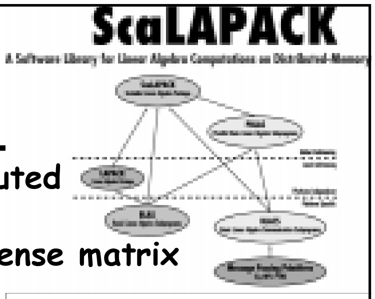
Multithreaded as wel

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
  - University of Tennessee

22

ScaLAPACK


ScaLAPACK
A Software Library for Linear Algebra Computations on Distributed-Memory

- ScaLAPACK is a portable distributed memory numerical library
- Complete numerical library for dense matrix computations
- Designed for distributed parallel computing (MPP & Clusters) using MPI
- One of the first math software packages to do this
- Numerical software that will work on a heterogeneous platform
- In use today by IBM, HP-Convex, Fujitsu, NEC, Sun, SGI, Cray, NAG, IMSL, …
  - Tailor performance & provide support

How ScaLAPACK Works

- To use ScaLAPACK a user must:
  - Download the package and auxiliary packages (like PBLAS, BLAS, BLACS, & MPI) to the machines.
  - If heterogeneous collection of machines, make sure proper versions available.
  - Write a SPMD program which
    - Sets up the logical 2-D process grid
    - Places the data on the logical process grid
    - Calls the library routine in a SPMD fashion
    - Collects the solution after the library routine finishes
  - The user must allocate the processors and decide the number of processes the application will run on
  - The user must start the application
    - "mpirun –np *N* user_app"
      - Note: the number of processors is fixed by the user before the run
  - Upon completion, return the processors to the pool of resources

hetero

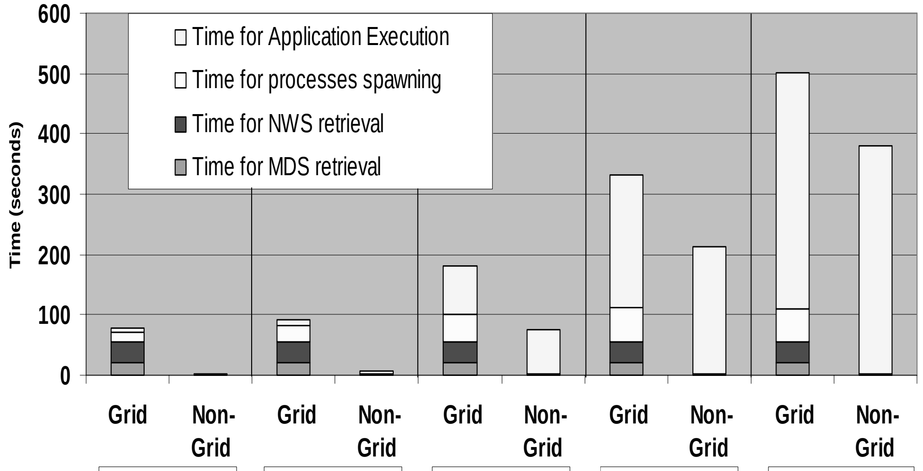
Self Adapting Numerical Library

- Want to relieve the user of some of the tasks
- Make decisions on which machines to use based on the user's problem and the state of the system
  - Optimize for the best time to solution dynamically
    - Optimization problem involves: resources, problem size, software, …
  - Distribute the data on the processors and collections of results
  - Start the SPMD library routine on all the platforms
  - Check to see if the computation is proceeding as planned
    - If not perhaps migrate application

2 efforts

25

Grid ScaLAPACK vs Non-Grid ScaLAPACK, Dedicated Torc machines



N=600, NB=40, 2 torc procs. Ratio: 46.12	N=1500, NB=40, 4 torc procs. Ratio: 15.03	N=5000, NB=40, 6 torc procs. Ratio: 2.25	N=8000, NB=40, 8 torc procs. Ratio: 1.52	N=10,000, NB=40, 8 torc procs. Ratio: 1.29

26

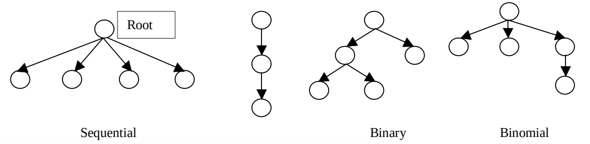
Machine-Assisted Application Development and Adaptation

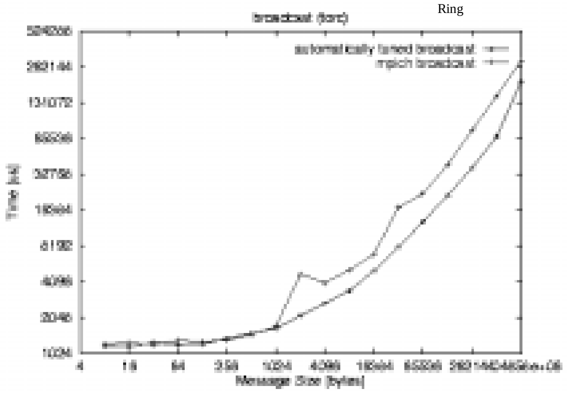
- ♦ **Communication libraries**
  - ➢**Optimize for the specifics of one's configuration.**
- ♦ **Algorithm layout and implementation**
  - ➢**Look at the different ways to express implementation**

27

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

8

Futures for Numerical Algorithms and Software on Clusters and Grids

- 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 effectively use the resources.

Contributors

Contributors

- Top500
  - Erich Strohmaier, NERSC
  - Hans Meuer, Mannheim U
  - http://www.top500.org
- SANS-Effort
  - Victor Eijkhout, UTK
  - Antoine Petitet, Sun France
  - Kenny Roche, UTK
  - Sathish Vadhiyar, UTK
  - Clint Whaley, UTK

For additional information see…

www.netlib.org/top500/

www.netlib.org/atlas/

icl.cs.utk.edu/grads/

www.cs.utk.edu/~dongarra/

Many opportunities within the group at Tennessee