Trends in High Performance Computing and Using Numerical Libraries on Clusters

Jack Dongarra
University of Tennessee

INNOVATIVE COMPUTING LABORATORY
COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF TENNESSEE

Outline

- Look at Clusters in the context of
  - Top500 Supercomputers (Snapshot from June 2002)
  - Top100 Clusters (Based on Theoretical Peak)
- Self Adapting Numerical Software (SANS) effort
  - Automatic Translation for Linear Algebra Software (ATLAS)
  - LAPACK for Clusters (LFC)
  - Self-Adaptive Linear Solver Architecture (SALSA)

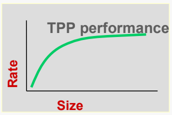
2

TOP500 SUPERCOMPUTER

H. Meuer, H. Simon, E. Strohmaier, & JD

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

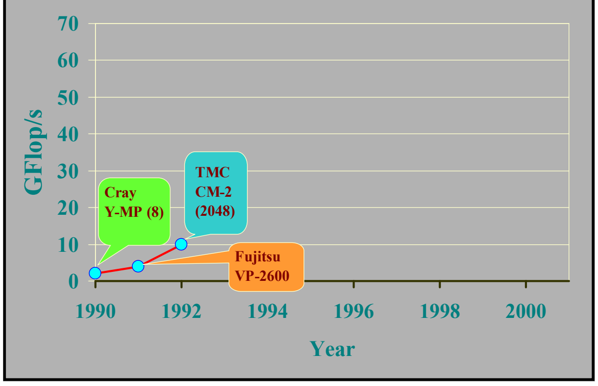
  Ax=b, dense problem



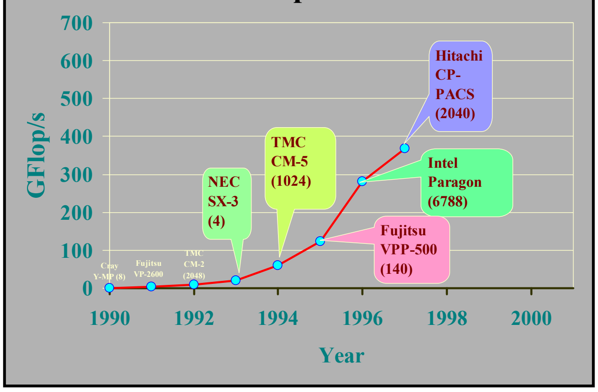
- Updated twice a year
  SC'xy in the States in November
  Meeting in Mannheim, Germany in June
- All data available from **www.top500.org**

3

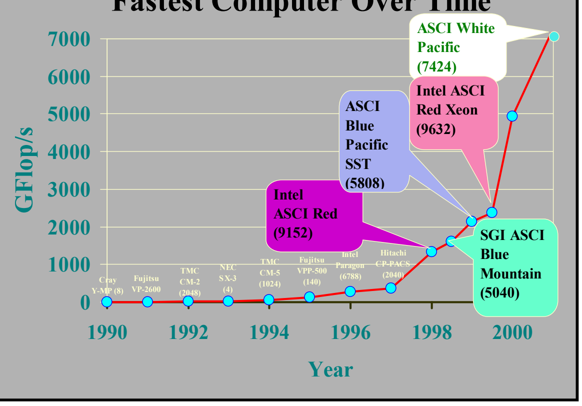
Fastest Computer Over Time



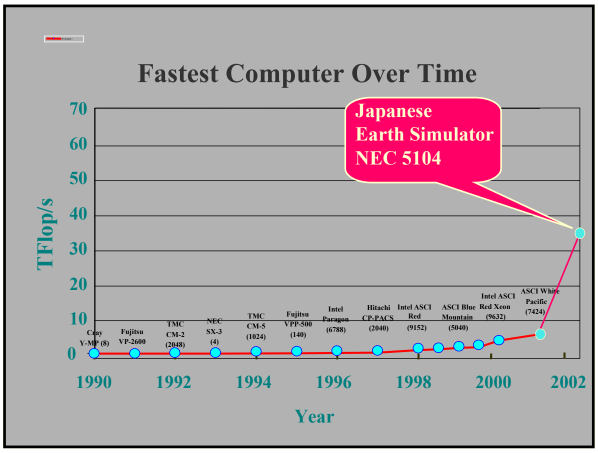
Fastest Computer Over Time



Fastest Computer Over Time



Title goes here

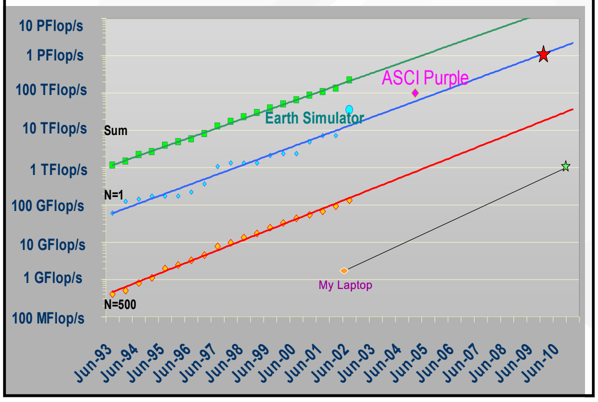


Top10 of the Top500

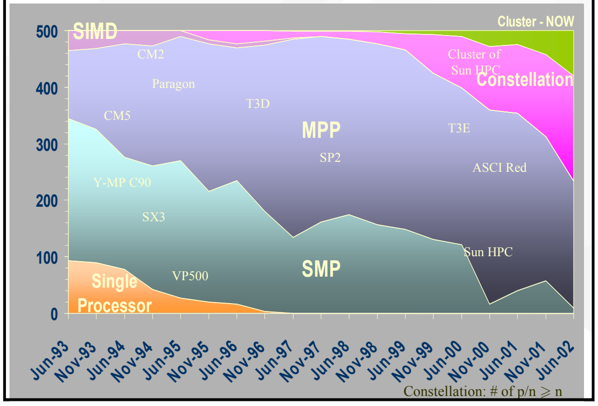
Rank	Manufacturer	Computer	R_{max} [TF/s]	Installation Site	Country	Year	Area of Installation	# Proc
1	NEC	Earth-Simulator	35.86	Earth Simulator Center	Japan	2002	Research	5120
2	IBM	ASCI White SP Power3	7.23	Lawrence Livermore National Laboratory	USA	2000	Research	8192
3	HP	AlphaServer SC ES45 1 GHz	4.46	Pittsburgh Supercomputing Center	USA	2001	Academic	3016
4	HP	AlphaServer SC ES45 1 GHz	3.98	Commissariat a l'Energie Atomique (CEA)	France	2001	Research	2560
5	IBM	SP Power3 375 MHz	3.05	NERSC/LBNL	USA	2001	Research	3328
6	HP	AlphaServer SC ES45 1 GHz	2.92	Los Alamos National Laboratory	USA	2002	Research	2048
7	Intel	ASCI Red	2.38	Sandia National Laboratory	USA	1999	Research	9632
8	IBM	pSeries 690 1.3 GHz	2.31	Oak Ridge National Laboratory	USA	2002	Research	864
9	IBM	ASCI Blue Pacific SST, IBM SP 604e	2.14	Lawrence Livermore National Laboratory	USA	1999	Research	5808
10	IBM	pSeries 690 1.3 Ghz	2.00	IBM/US Army Reseach Lab (ARL)	USA	2002	Vendor	768

8

Performance Extrapolation



Architectures

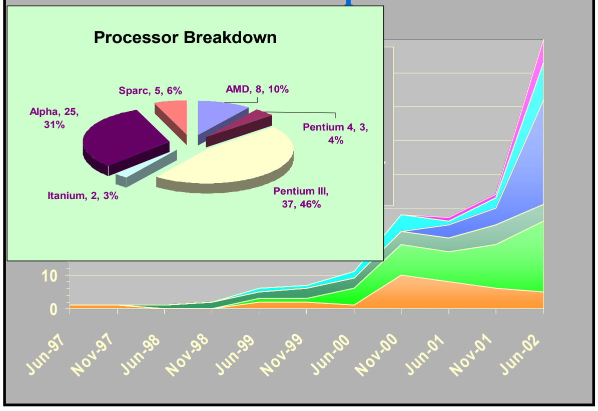


80 Clusters on the Top500

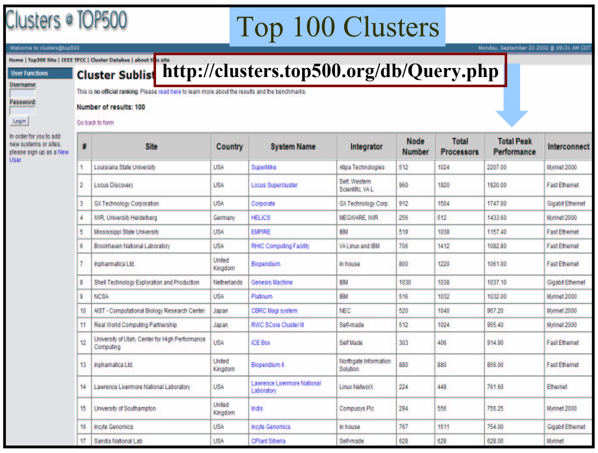
- A total of 42 Intel based and 8 AMD based PC clusters are in the TOP500.
  - 31 of these Intel based cluster are IBM Netfinity systems delivered by IBM.
- A substantial part of these are installed at industrial customers especially in the oil-industry.
  - Including 5 Sun and 5 Alpha based clusters and 21 HP AlphaServer.
- 14 of these clusters are labeled as 'Self-Made'.

11

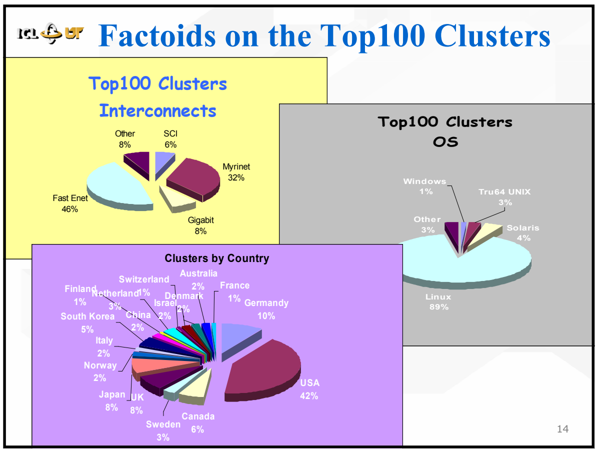
Cluster on the Top500

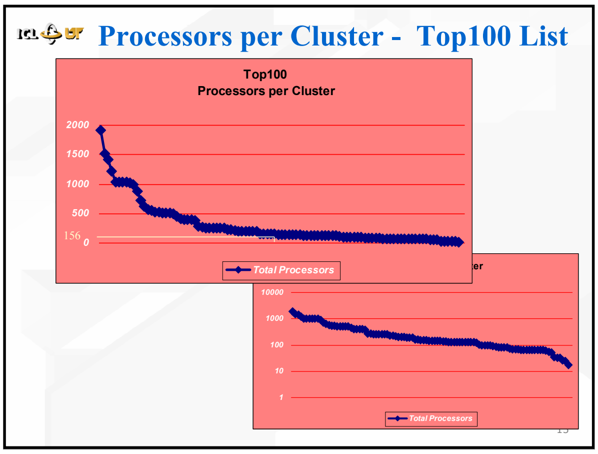


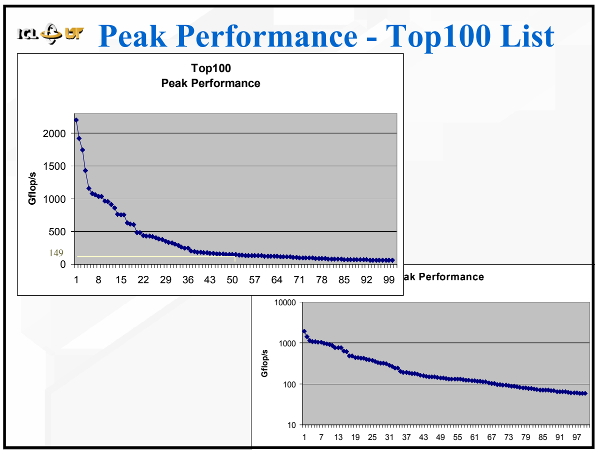
Title goes here



#	Site	Country	System Name	Integrator	Node Number	Total Processors	Total Peak Performance	Interconnect
1	Louisiana State University	USA	SuperMike	Atipa Technologies	512	1024	2207.00	Myrinet 2000
2	Locus Discovery	USA	Locus Supercluster	Self, Western Scientific, VA L	960	1920	1920.00	Fast Ethernet
3	GX Technology Corporation	USA	Corporate	GX Technology Corp.	912	1504	1747.00	Gigabit Ethernet
4	IWR, University Heidelberg	Germany	HELICS	MEGWARE, IWR	256	512	1433.60	Myrinet 2000
5	Mississippi State University	USA	EMPIRE	IBM	519	1038	1157.40	Fast Ethernet
6	Brookhaven National Laboratory	USA	RHIC Computing Facility	VA Linux and IBM	706	1412	1082.80	Fast Ethernet
7	Inpharmatica Ltd.	United Kingdom	Biopendium	In house	800	1220	1061.00	Fast Ethernet
8	Shell Technology Exploration and Production	Netherlands	Genesis Machine	IBM	1030	1038	1037.10	Gigabit Ethernet
9	NCSA	USA	Platinum	IBM	516	1032	1032.00	Myrinet 2000
10	AIST - Computational Biology Research Center	Japan	CBRC Magi system	NEC	520	1040	967.20	Myrinet 2000
11	Real World Computing Partnership	Japan	RWC SCore Cluster III	Self-made	512	1024	955.40	Myrinet 2000
12	University of Utah, Center for High Performance Computing	USA	ICE Box	Self Made	303	406	914.90	Fast Ethernet
13	Inphamatica Ltd.	United Kingdom	Biopendium II	Northgate Information Solution	880	880	856.00	Fast Ethernet
14	Lawrence Livermore National Laboratory	USA	Lawrence Livermore National Laboratory	Linux NetworX	224	448	761.60	Ethernet
15	University of Southampton	United Kingdom	Iridis	Compusys Plc	294	556	756.25	Myrinet 2000
16	Incyte Genomics	USA	Incyte Genomics	In house	767	1511	754.00	Gigabit Ethernet
17	Sandia National Lab	USA	CPlant Siberia	Self-made	628	628	628.00	Myrinet







Self-Adapting Numerical Software (SANS) Effort

- The complexities of modern processors and clusters makes it difficult to achieve high performance
- Hardware, compilers, and software have a large design space w/many parameters
- Kernels of Computation Routines
  - Focus on where the most time is spent
- Need for quick/dynamic deployment of optimized routines.
- Algorithm layout and implementation
  - Look at the different ways to express implementation

17

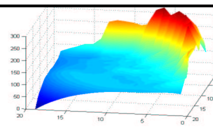
What is Self Adapting Performance Tuning of Software?

- **Self Adapting during library installation**
  - Taylor to the specifics of the machine
  - Example is Automatically Tuned Linear Algebra Sw (ATLAS)
- **Self Adapting to the available resources**
  - Adapt to things like the processor type, number of processors, size of problem
  - Example is LAPACK For Clusters (LFC)
- **Self Adapting to the user's problem**
  - Adapt to the user data by providing automatic algorithm selection
  - Example is Self-Adaptive Linear Solver Architecture (SALSA)
- **Self Adapting in the presence of failures**
  - Allow the user to provide for faults and recover without additional users involvement
  - Example is FT-MPI based algorithm

18

Title goes here

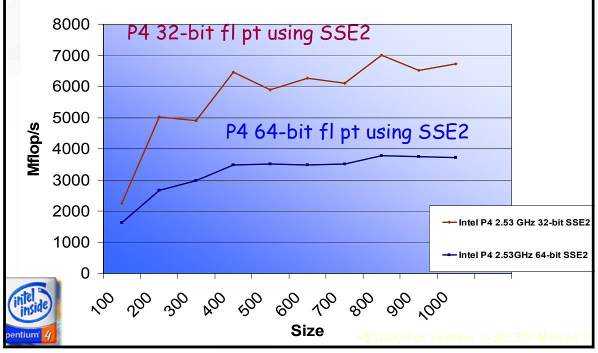
Software Generation Strategy - ATLAS BLAS



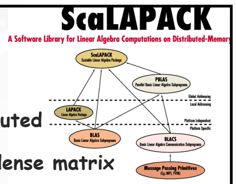
- Parameter study of the hw
- Generate multiple versions of code, w/difference values of key performance parameters
- Run and measure the performance for various versions
- Pick best and generate library
- Level 1 cache multiply optimizes for:
  - TLB access
  - L1 cache reuse
  - FP unit usage
  - Memory fetch
  - Register reuse
  - Loop overhead minimization
- Takes ~ 20 minutes to run, generates Level 1,2, & 3 BLAS
- "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 used within various ASCI and SciDAC activities and by Matlab, Mathematica, Octave, Maple, Debian, Scyld Beowulf, SuSE,...

19

ATLAS Matrix Multiply Intel Pentium 4 at 2.53GHz – using SSE2



ScaLAPACK



- 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
- Funding from DOE, NSF, and DARPA
- In use today by IBM, HP-Convex, Fujitsu, NEC, Sun, SGI, Cray, NAG, IMSL, …
  - Tailor performance & provide support

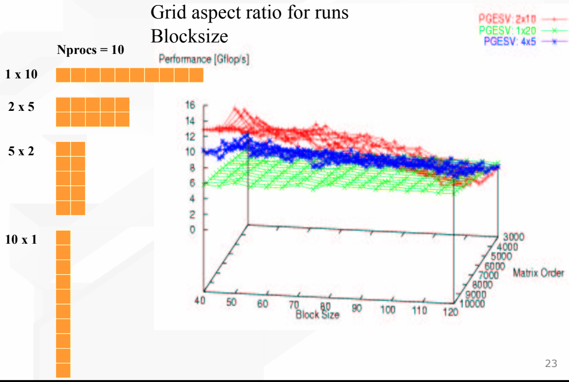
21

To Use ScaLAPACK a User Must:

- Download the package and auxiliary packages (like PBLAS, BLAS, BLACS, & MPI) to the machines.
- Write a SPMD program which
  - Sets up the logical 2-D process grid
  - Places the data on the logical process grid
  - Calls the numerical 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, if problem size changes dynamically …
- Upon completion, return the processors to the pool of resources

22

Needs an expert to do the tuning:
Number of processors
Grid aspect ratio for runs
Blocksize



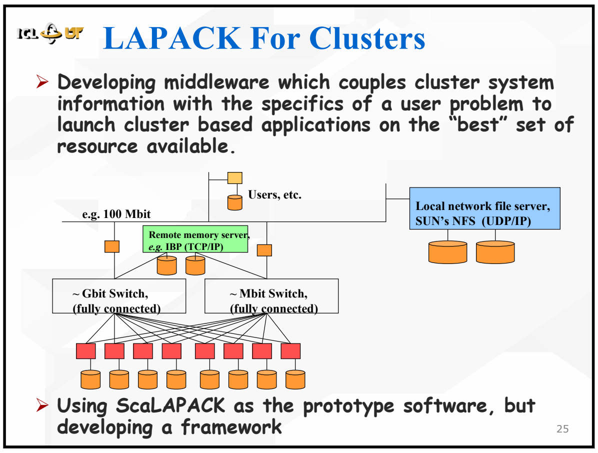
23

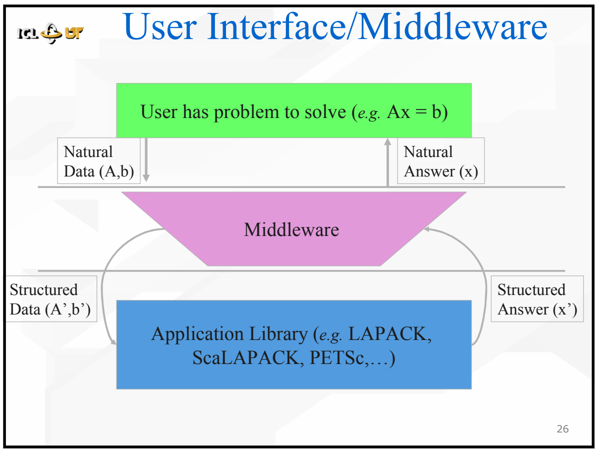
Cluster Numerical Library

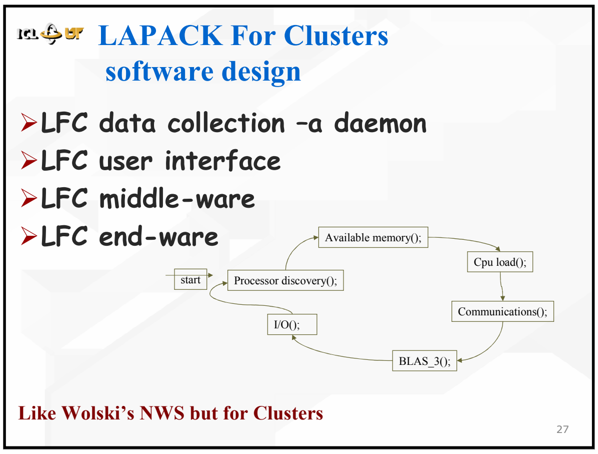
- Want to relieve the user of some of the tasks
- Make decisions on which machines to use based on the user's problem and the state of the system
  - Determinate set of procs that should be used
  - Optimize for the best time to solution
  - Distribute the data on the processors and collections of results
  - Start the SPMD library routine on all the platforms
  - Check to see if the computation is proceeding as planned
    - If not perhaps migrate application

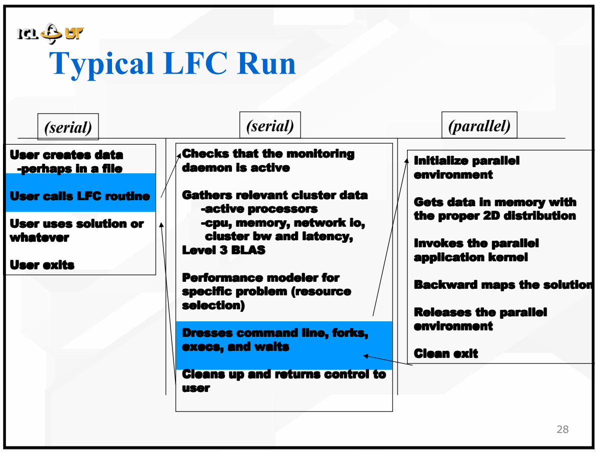
24

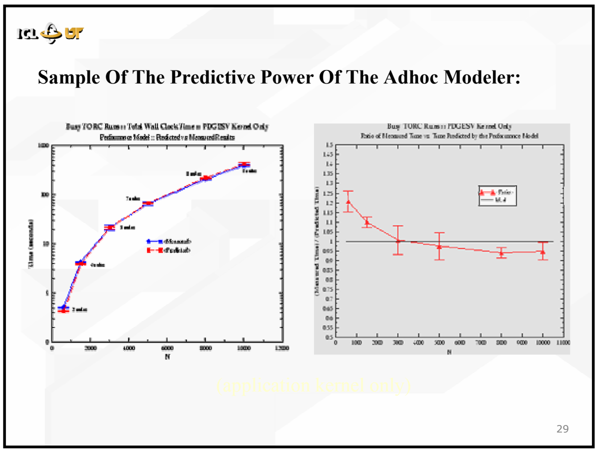
Title goes here

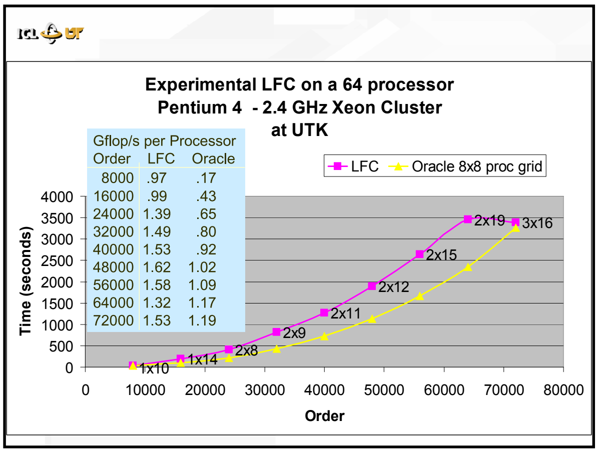












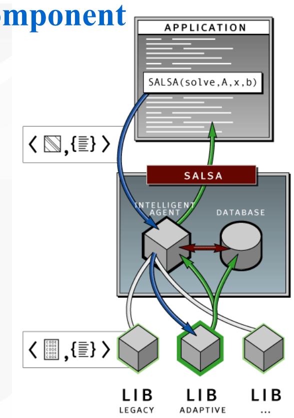
Run-Time Adaptivity for Linear Systems

- Many possible methods: Nature of data is prime consideration in choice
- Dense systems: fairly cut and dry, only adapt to infrastructure
- Sparse systems: a mess. Direct and iterative methods, multigrid, different preconditioners. No one algorithm best for sparse system.

31

Intelligent Component

- System to mediate between user application and multiple possible libraries
- Self-Adaptivity and Learning Behavior
  - Heuristics are tuned based on data
    - System gradually gets smarter (database)
  - The system can educate the user
- User Interaction
  - User can guide the system by providing further information
  - System teaches user about properties of the data



Future SANS Effort

- Intelligent Component
  - Automates method selection based on data, algorithm, and system attributes
- System component
  - Provides intelligent management of and access to clusters and computational grids
- History database
  - Records relevant info generated by the IC and maintains past performance data
- Fault Tolerant Aspect
  - Transparently detect and recover from failure
    - FT-MPI
    - Algorithmic Fault Tolerance

33

Collaborators

- TOP500
  - H. Mauer, Mannheim U
  - H. Simon, NERSC
  - E. Strohmaier, NERSC
- SANS-Effort
  - Jeffrey Chen, UTK
  - Jun Ding, UTK
  - Tom Eidson, ICASE
  - Victor Eijkhout, UTK
  - Piotr Luszczek, UTK
  - Kenny Roche, UTK
  - Sathish Vadhiyar, UTK
- HPL and ATLAS
  - Antoine Petitet, Sun
  - Clint Whaley, FSU
- Availability
  - Top500
    - http://www.top500.org/
    - http://clusters.top500.org/
  - ATLAS
    - http://icl.cs.utk.edu/atlas/
  - LFC
    - 5 drivers from ScaLAPACK coming soon
  - Algorithm Fault Tolerance
    www.cs.utk.edu/~plank/plank/papers/ADCKP.html

34