High Performance Computing, Computational Grid, and Numerical Libraries

Jack Dongarra
Innovative Computing Lab
University of Tennessee
http://www.cs.utk.edu/~dongarra/

Outline

- Efforts in self adapting software
- Two approaches to Grid numerical libraries, some early experiments
  - NetSolve - Grid enabled portal - software servers
  - GrADS Project - Software Technology for Problem Solving on Computational Grids

Software Technology & Performance

- Tendency to focus on hardware
- Software required to bridge an ever widening gap
- 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.

3

Software Issues:

- Predictability 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

4

Self Adapting Software

- **Software system that …**
  - Obtains information on the underlying system where they will run.
  - Adapts application to the presented data and the available resources perhaps provide automatic algorithm selection
  - During execution perform optimization and perhaps reconfigure based on newly available resources.
  - Allow the user to provide for faults and recover without additional users involvement

5

Motivation Self Adapting Numerical Software (SANS) Effort

- Optimizing software to exploit the features of a given processor has historically been an exercise in hand customization.
  - Time consuming and tedious
  - Hard to predict performance from source code
  - Growing list of kernels to tune
  - Must be redone for every architecture and compiler
    - Compiler technology often lags architecture
    - Best algorithm may depend on input, so some tuning may be needed at run-time.
    - Not all algorithms semantically or mathematically equivalent
  - Need for quick/dynamic deployment of optimized routines.

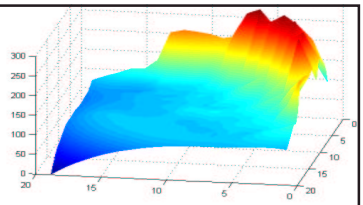
6

Self Adapting Numerical Software - SANS Effort

- Provide software technology to aid in high performance on commodity processors, clusters, and grids.
- Pre-run time (library building stage) and run time optimization.
- Integrated performance modeling and analysis
- Automatic algorithm selection – polyalgorithmic functions
- Automated installation process
- Can be expanded to areas such as communication software and selection of numerical algorithms

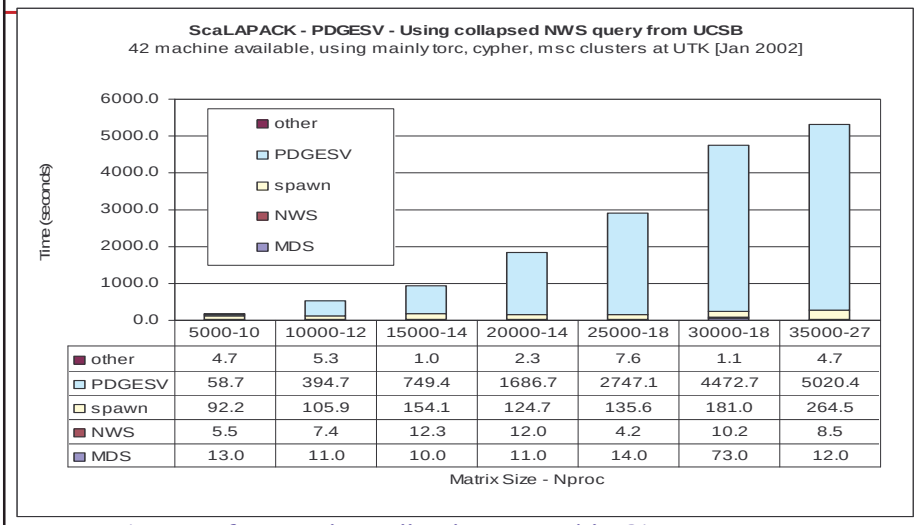
Software Generation Strategy - ATLAS BLAS

- 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 modern architectures
  - Need reasonable C compiler
- Today ATLAS in used within various ASCI and SciDAC activities and by Matlab, Mathematica, Octave, Maple, Debian, Scyld Beowulf, SuSE,…

ATLAS (DGEMM n = 500)



	5000-10	10000-12	15000-14	20000-14	25000-18	30000-18	35000-27
other	4.7	5.3	1.0	2.3	7.6	1.1	4.7
PDGESV	58.7	394.7	749.4	1686.7	2747.1	4472.7	5020.4
spawn	92.2	105.9	154.1	124.7	135.6	181.0	264.5
NWS	5.5	7.4	12.3	12.0	4.2	10.2	8.5
MDS	13.0	11.0	10.0	11.0	14.0	73.0	12.0

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

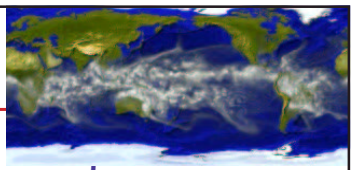
9

Some Automatic Tuning Projects

- ATLAS (www.netlib.org/atlas) (Dongarra, Whaley)
- PHIPAC (www.icsi.berkeley.edu/~bilmes/phipac) (Bilmes,Asanovic,Vuduc,Demmel)
- Sparse matrix operations, (Yelick, Im & Dongarra, Eijkhout)
- Communication topologies (Dongarra)
- FFTs and Signal Processing
  - FFTW (www.fftw.org)
    - Won 1999 Wilkinson Prize for Numerical Software
  - SPIRAL (www.ece.cmu.edu/~spiral)
    - Extensions to other transforms, DSPs
  - UHFFT
    - Extensions to higher dimension, parallelism

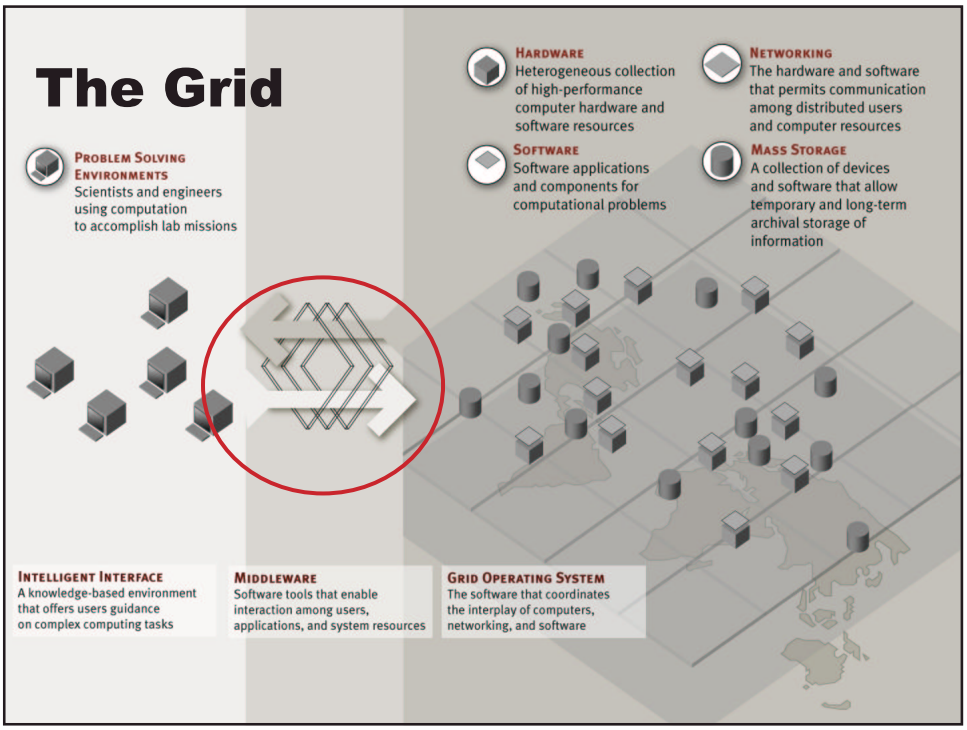
10

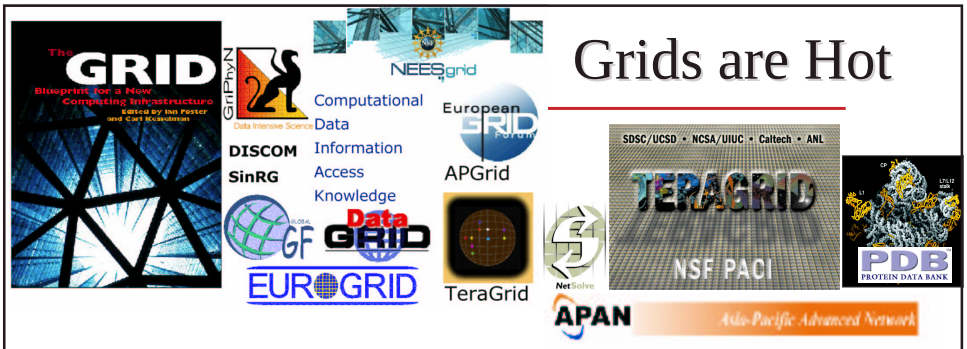
In the past: Isolation
Motivation for Grid Computing



- Today there is a complex interplay and increasing interdependence among the sciences.
- Many science and engineering problems require widely dispersed resources be operated as systems.
- What we do as collaborative infrastructure developers will have profound influence on the future of science.
- Networking, distributed computing, and parallel computation research have matured to make it possible for distributed systems to support high-performance applications, but...
  - Resources are dispersed
  - Connectivity is variable
  - Dedicated access may not be possible

Today: Collaboration[11]





Grids are Hot

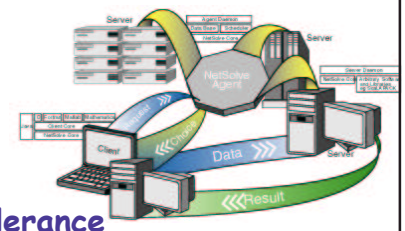
IPG NASA http://nas.nasa.gov/~wej/home/IPG
Globus http://www.globus.org/
Legion http://www.cs.virgina.edu/~grimshaw/
AppLeS http://www-cse.ucsd.edu/groups/hpcl/
NetSolve http://www.cs.utk.edu/netsolve/
NINF http://phase.etl.go.jp/ninf/
Condor http://www.cs.wisc.edu/condor/
CUMULVS http://www.epm.ornl.gov/cs/
WebFlow http://www.npac.syr.edu/users/gcf/
NGC http://www.nordicgrid.net

13

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

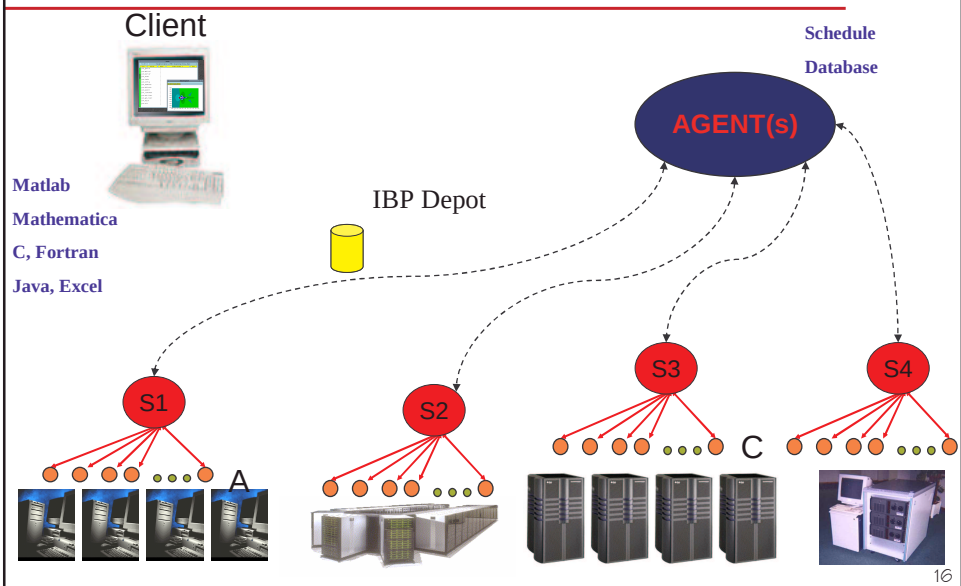
14

NetSolve Network Enabled Server

- NetSolve is an example of a Grid based hardware/software/data server.
- Based on a Remote Procedure Call model but with …
  - resource discovery, dynamic problem solving capabilities, load balancing, fault tolerance asynchronicity, security, …
- Easy-of-use paramount
- Its about providing transparent access to resources.

15

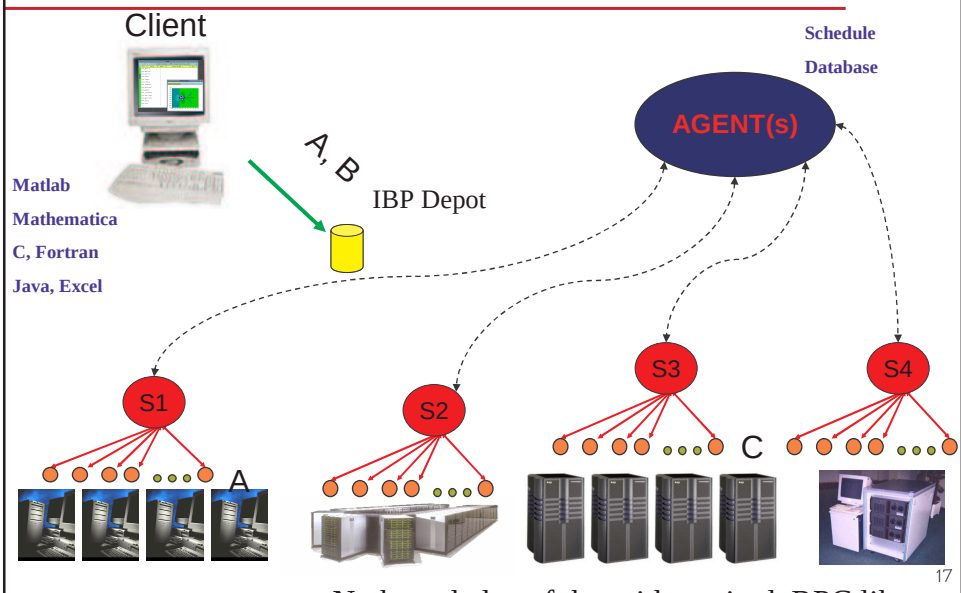
NetSolve: The Big Picture



16

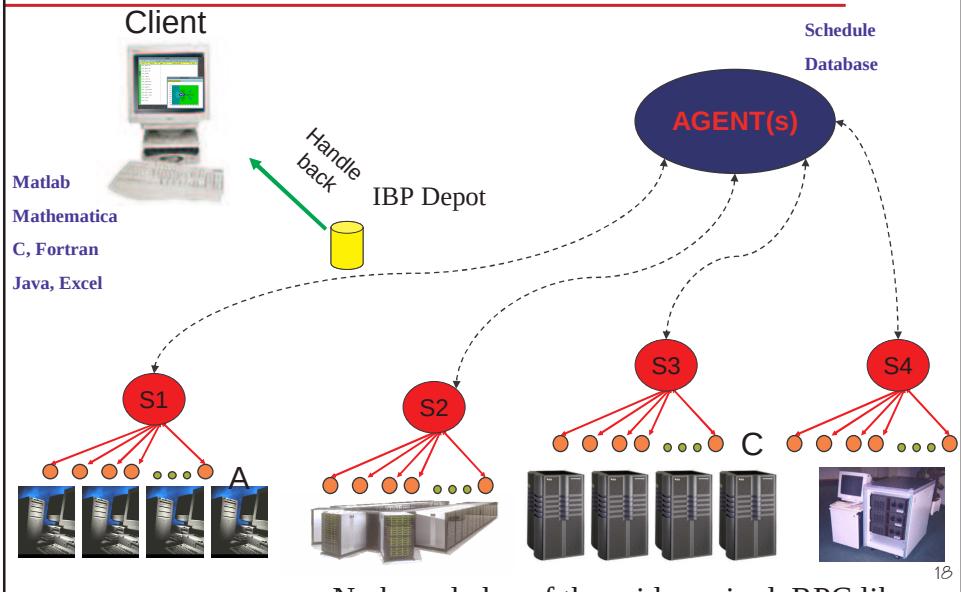
No knowledge of the grid required, RPC like.

NetSolve: The Big Picture



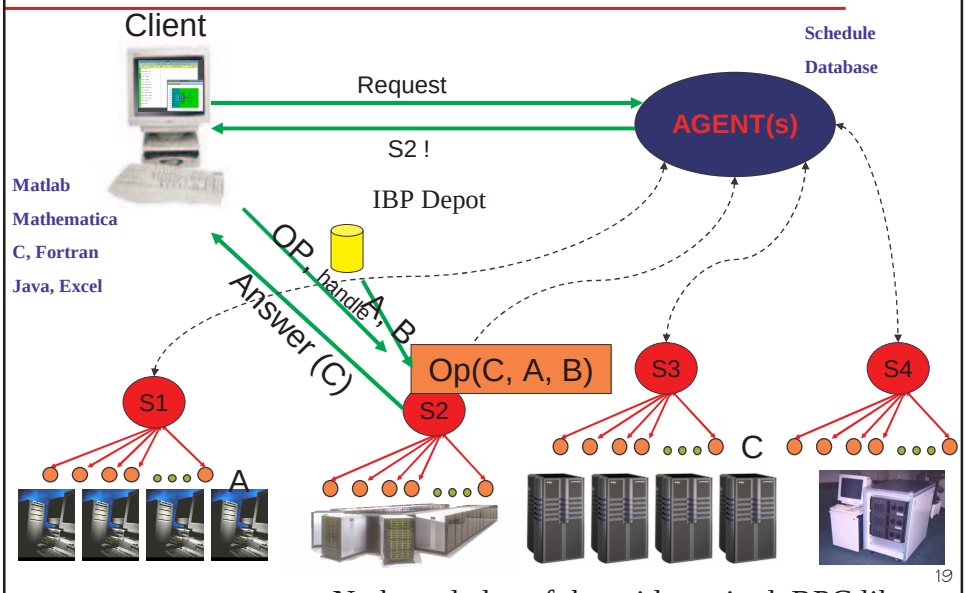
No knowledge of the grid required, RPC like.

NetSolve: The Big Picture



No knowledge of the grid required, RPC like.

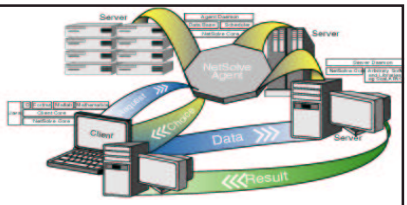
NetSolve: The Big Picture



19

No knowledge of the grid required, RPC like.

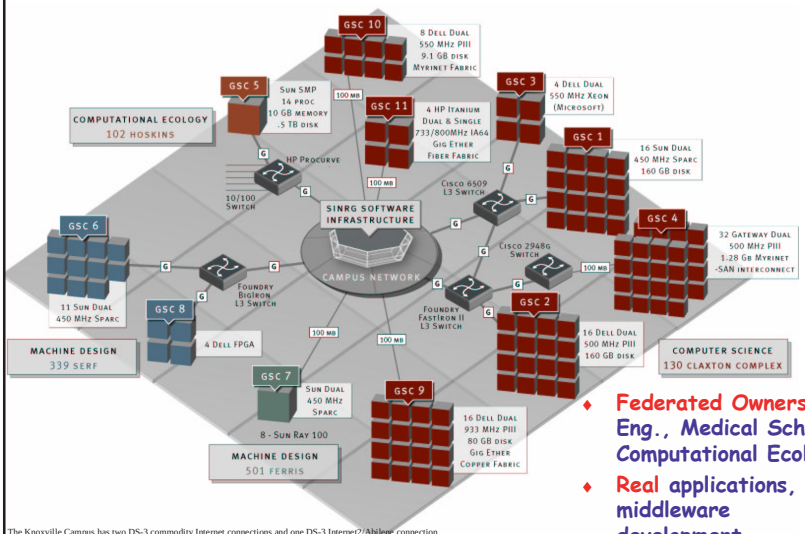
Basic Usage Scenarios



- 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
- Can plug into Globus, Condor, NINF, …

20

University of Tennessee Deployment: Scalable Intracampus Research Grid: SInRG



The Knoxville Campus has two DS-3 commodity Internet connections and one DS-3 Internet2/Abilene connection. An OC-3 ATM link routes IP traffic between the Knoxville campus, National Transportation Research Center, and Oak Ridge National Laboratory. UT participates in several national networking initiatives including Internet2 (I2), Abilene, the federal Next Generation Internet (NGI) initiative, Southern Universities Research Association (SURA) Regional Information Infrastructure (RII), and Southern Crossroads (SoX).

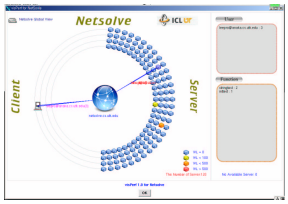
The UT campus consists of a meshed ATM OC-12 being migrated over to switched Gigabit by early 2002.

- Federated Ownership: CS, Chem Eng., Medical School, Computational Ecology, El. Eng.
- Real applications, middleware development, logistical networking

21

NetSolve- Things Not Touched On

- Security
  - Using Kerberos V5 for authentication.
- Separate Server Characteristics
  - Implementing Hardware and Software servers
- Hierarchy of Argents
  - More scalable configuration
- Monitor NetSolve Network
  - Track and monitor usage
- Network status
  - Network Weather Service
- Internet Backplane Protocol
  - Middleware for managing and using remote storage.
- Fault Tolerance
- Local / Global Configurations
- Dynamic Nature of Servers
- Automated Adaptive Algorithm Selection
  - Dynamic determine the nest algorithm based on system status and nature of user problem
- NetSolve evolving into GridRPC
  - Being worked on under GGF with joint with NINF

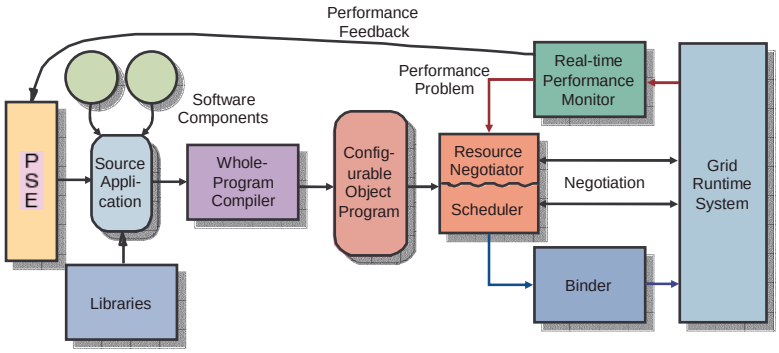


NETWORK WEATHER SERVICE

22

NSF/NGS
GrADS - GrADSoft Architecture

- **Goal: reliable performance on dynamically changing resources**

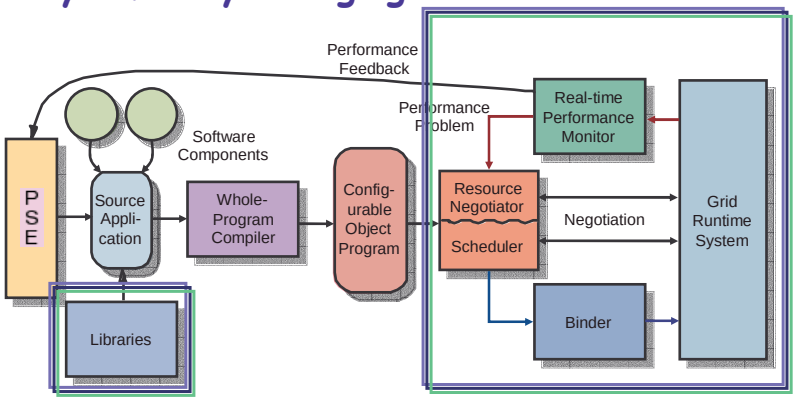


PIs: Ken Kennedy, Fran Berman, Andrew Chein, Keith Cooper, JD, Ian Foster, Lennart Johnsson, Dan Reed, Carl Kesselman, John Mellor-Crummey, & Rich Wolski

23

NSF/NGS
GrADS - GrADSoft Architecture

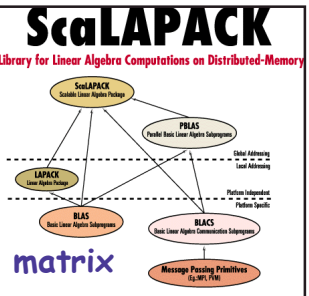
- **Goal: reliable performance on dynamically changing resources**



PIs: Ken Kennedy, Fran Berman, Andrew Chein, Keith Cooper, JD, Ian Foster, Lennart Johnsson, Dan Reed, Carl Kesselman, John Mellor-Crummey, & Rich Wolski

24

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

25

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

26

ScaLAPACK Grid Enabled

- Implement a version of a ScaLAPACK library routine that runs on the Grid.
  - Make use of resources at the user's disposal
  - Provide the best time to solution
  - Proceed without the user's involvement
- Make as few changes as possible to the numerical software.
- Assumption is that the user is already "Grid enabled" and runs a program that contacts the execution environment to determine where the execution should take place.

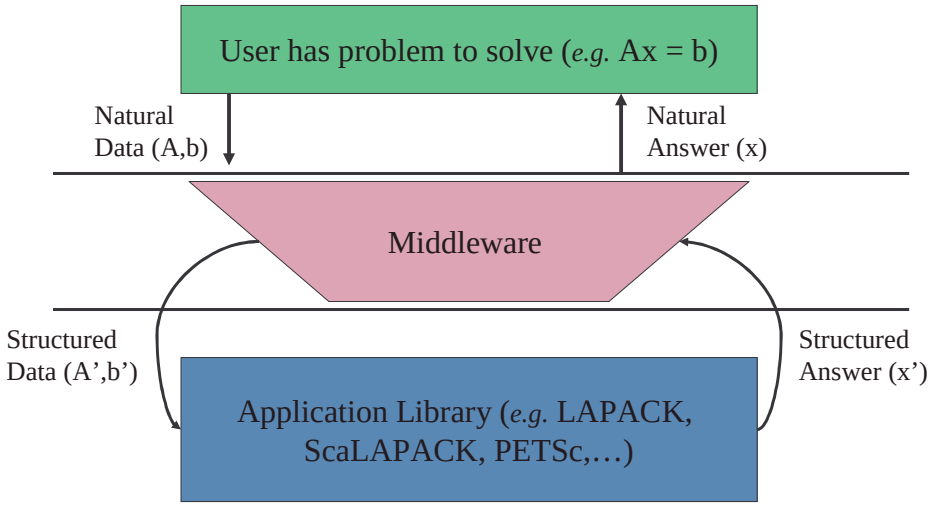
27

GrADS 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 machines that can 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

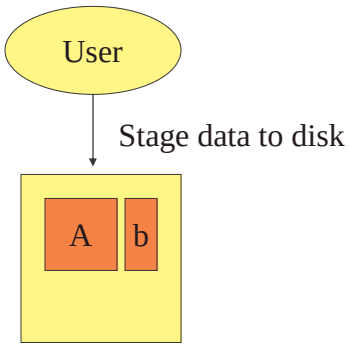
28

Big Picture…



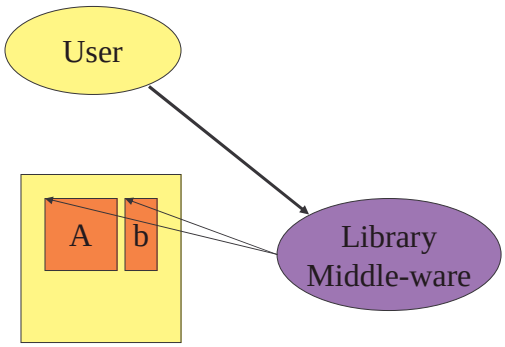
29

Numerical Libraries for Clusters



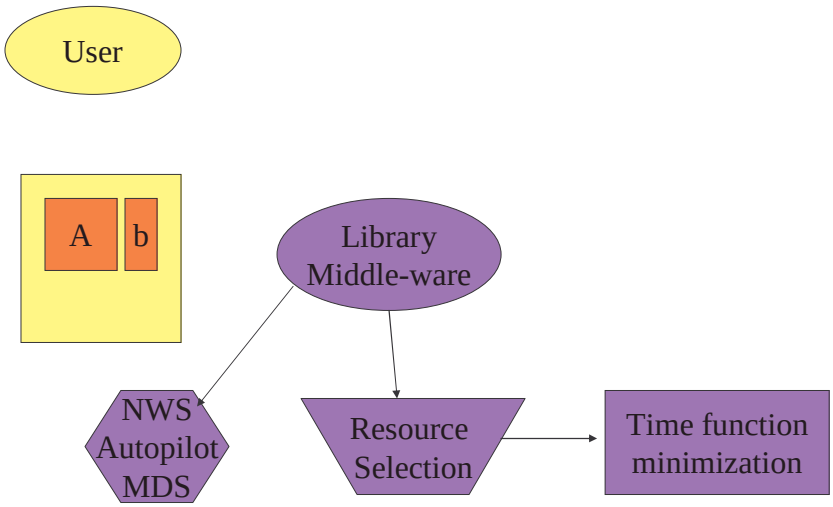
30

Numerical Libraries for Clusters



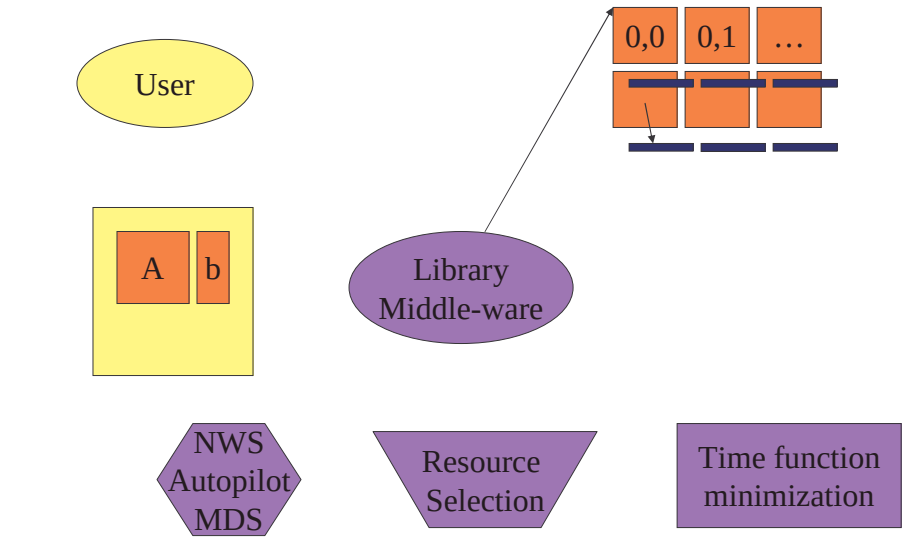
31

Numerical Libraries for Clusters



32

Numerical Libraries for Clusters



33

Can use Grid infrastructure, i.e.Globus/NWS, but doesn't have to.

Experimental Hardware / Software Grid

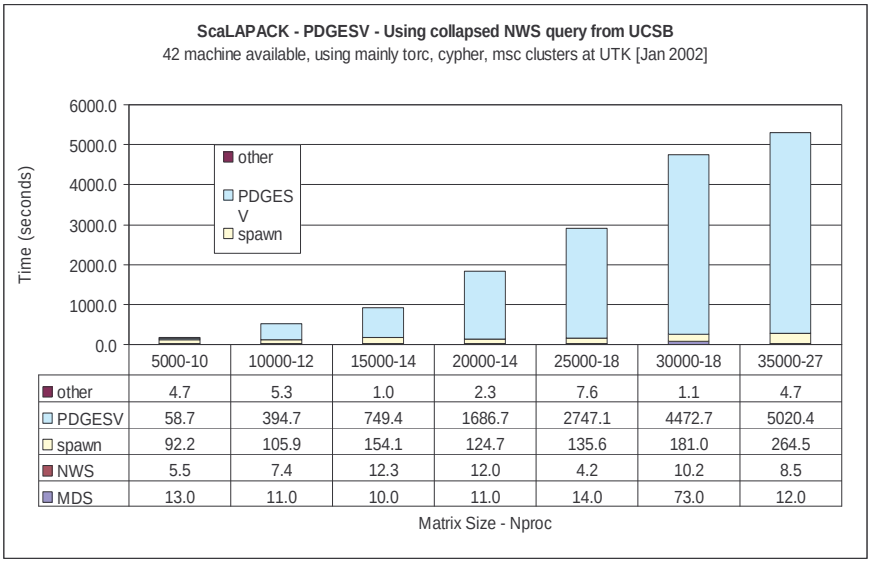
MacroGrid Testbed	TORC	CYPHER	OPUS
Type	Cluster 8 Dual Pentium III	Cluster 16 Dual Pentium III	Cluster 8 Pentium II
OS	Red Hat Linux 2.2.15 SMP	Debian Linux 2.2.17 SMP	Red Hat Linux 2.2.16
Memory	512 MB	512 MB	128 or 256 MB
CPU speed	550 MHz	500 MHz	265 – 448 MHz
Network	Fast Ethernet (100 Mbit/s) (3Com 3C905B) and switch (BayStack 350T) with 16 ports	Gigabit Ethernet (SK-9843) and switch (Foundry FastIron II) with 24 ports	Myrinet (LANai 4.3) with 16 ports each

- Globus version 1.1.3
- Autopilot version 2.3
- NWS version 2.0.pre2
- MPICH-G version 1.1.2
- ScaLAPACK version 1.6
- ATLAS/BLAS version 3.0.2
- BLACS version 1.1
- PAPI version 1.1.5
- GrADS' "Crafted code"

Independent components being put together and interacting

34

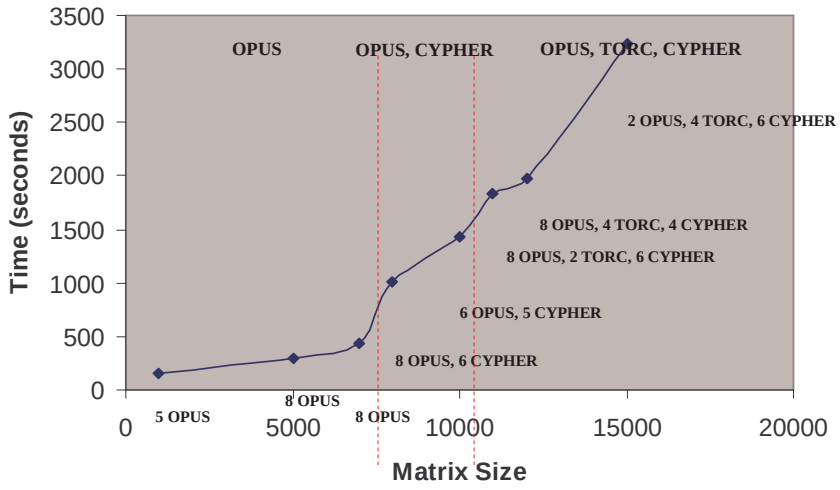
PDGESV Experiments: Time Breakdown



	5000-10	10000-12	15000-14	20000-14	25000-18	30000-18	35000-27
other	4.7	5.3	1.0	2.3	7.6	1.1	4.7
PDGESV	58.7	394.7	749.4	1686.7	2747.1	4472.7	5020.4
spawn	92.2	105.9	154.1	124.7	135.6	181.0	264.5
NWS	5.5	7.4	12.3	12.0	4.2	10.2	8.5
MDS	13.0	11.0	10.0	11.0	14.0	73.0	12.0

35

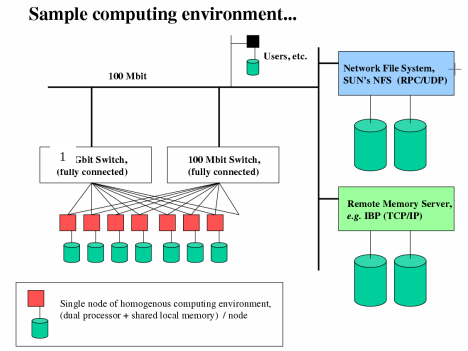
ScaLAPACK across 3 Clusters



36

LAPACK For Clusters

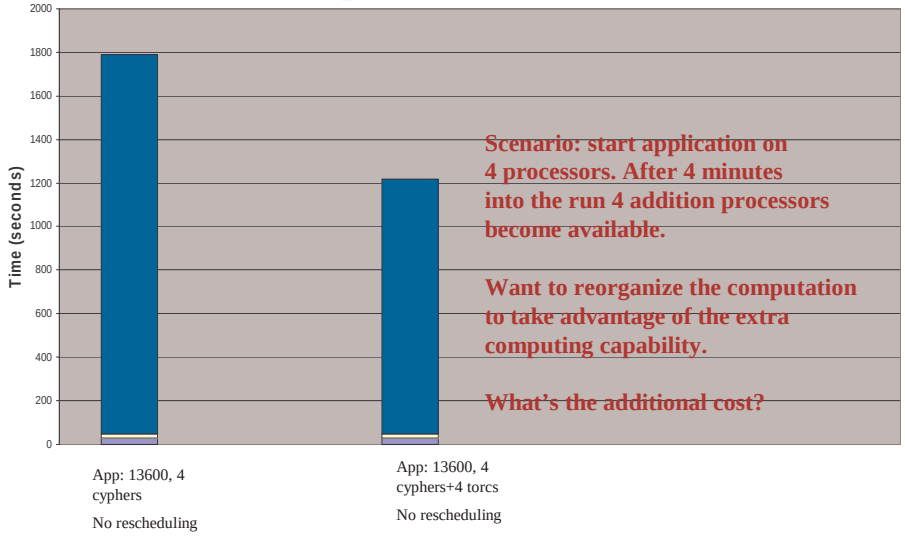
- Developing middleware which couples cluster system information with the specifics of a user problem to launch cluster based applications on the "best" set of resource available.



- Using ScaLAPACK as the prototype software

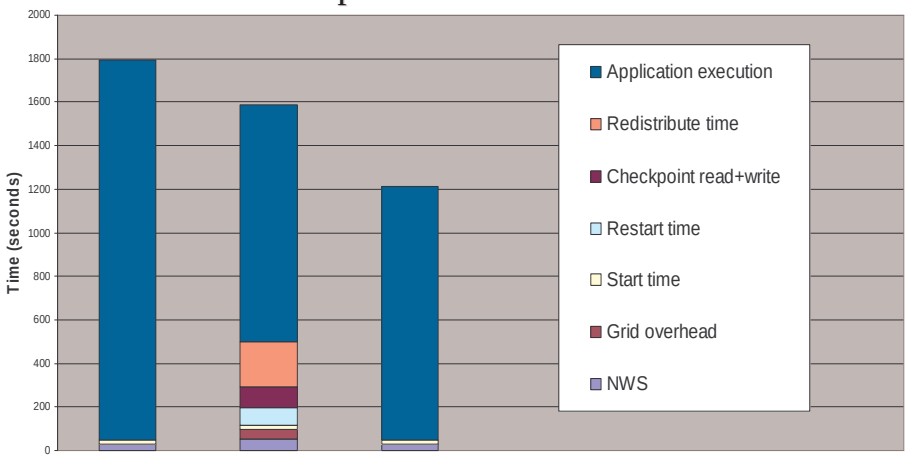
37

Rescheduling/Redistribution Experimental Results



38

Rescheduling/Redistribution Experimental Results



App: 13600, 4 cyphers

No rescheduling

App1: 13600, 4 torcs

App2: 13600, 4 torcs+4 cyphers, introduced 4 mins. into the run

With rescheduling

App: 13600, 4 cyphers+4 torcs

No rescheduling

About 12% better performance even with redistribution and restart.

39

Fault Tolerance in the Message Passing

- Critical for many Grid and Cluster applications
- MPI wasn't designed to be fault tolerant
- Number of projects
  - FT-MPI at University of Tennessee

40

Algorithmic Fault Tolerance

- Important that this is built into the algorithms.
- Not good enough to have it in the message passing.
- Working on numerical library design for ScaLAPACK and PETSc that will be fault tolerant.

41

Research Directions

- Parameterizable libraries
- Fault tolerant algorithms
- Annotated libraries
- Hierarchical algorithm libraries
- "Grid" (network) enabled strategies

A new division of labor between compiler writers, library writers, and algorithm developers and application developers will emerge.

42

Futures for Numerical Algorithms and Software on Clusters and Grids

- Retargetable Libraries - 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.

43

Conclusion

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

44

Collaborators

Many opportunities within the group at Tennessee

- ATLAS
  - Antoine Petitet, Sun
  - Clint Whaley, FSU
- SANS
  - Victor Eijkhout, UTK
- GrADS
  - Sathish Vadhiyar, UTK
  - Asim YarKhan, UTK
  - Ken Kennedy, Fran Berman, Andrew Chein, Ian Foster, Carl Kesselman, Lennart Johnsson, Dan Reed, Rich Wolski
- LFC
  - Jeffrey Chen, UTK
  - Piotr Luszczek, UTK
  - Kenny Roche, UTK
- FT-MPI
  - Graham Fagg, UTK/HLRS
  - Tone Bukovsky, UTK
  - Jeremy Miller, UTK

- Software Availability
  - ATLAS
    - icl.cs.utk.edu/atlas/
  - NetSolve
    - icl.cs.utk.edu/netsolve/
  - LFC
    - 5 drivers from ScaLAPACK around the end of summer
    - Next look at iterative solvers
  - FT-MPI
    - Available end of summer.