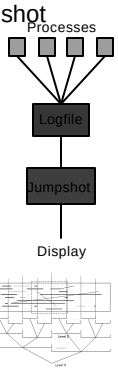
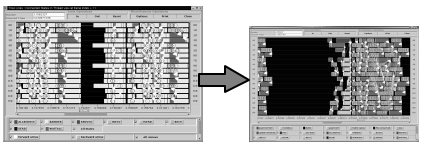
MPI Related Software

- Profiling Libraries and Tools
- Visualizing Program Behavior
- Timing
- Performance Measurement and Tuning
- High Level Libraries

Performance Visualization with Jumpshot

- For detailed analysis of parallel program behavior, timestamped events are collected into a log file during the run.
- A separate display program (Jumpshot) aids the user in conducting a post mortem analysis of program behavior.

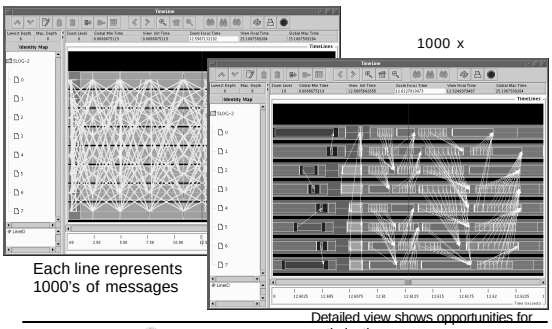




Profiling Libraries

- MPI provides mechanism to intercept calls to MPI functions
- For each MPI_ function corresponding PMPI_ version
- User can write custom version of for example MPI_Send then call PMPI_Send to send
- If user library is loaded before the standard one, users calls are executed
- Profiling libraries and tools are at
  - http://ftp.mcs.anl.gov/pub/mpi/mpe.tar

Using Jumpshot to look at FLASH at multiple Scales



Timing in MPI

- Use MPI_Wtime
  - Time in seconds since an arbitrary time in the past.
  - high-resolution, elapsed (or wall) clock.
  - MPI_WTICK gives the resolution of MPI_WTIME.

High Performance LINPACK (HPL)

- software package that solves a (random) dense linear system in double precision (64 bits) arithmetic on distributed-memory computers
- In addition to MPI, an implementation of **either** the Basic Linear Algebra Subprograms **BLAS or** the Vector Signal Image Processing Library **VSIPL** is also needed.
- Performance estimate usually overestimates that achieved in practice
- Performance on HPL depends on tuning of BLAS
  - Vendor specific BLAS
  - ATLAS

Performance Measurement

- Mpptest
  - http://www-unix.mcs.anl.gov/mpi/mpptest/
  - measures the performance of some of the basic MPI message passing routines
  - Measures performance with many participating processes (exposing contention and scalability problems)
  - can adaptively choose the message sizes in order to isolate sudden changes in performance
- SKaMPI
  - http://liinwww.ira.uka.de/~skampi/
  - suite of tests designed to measure the performance of MPI
  - Goal is to create a database to illustrate the performance of different MPI implementations on different architectures
  - Database of results
    - http://liinwww.ira.uka.de/~skampi/cgi-bin/run_list.cgi.pl

ATLAS

- **Automatically Tuned Linear Algebra Software (ATLAS)**
  - http://math-atlas.sourceforge.net/
  - ongoing research effort focusing on applying empirical techniques in order to provide portable performance
  - provides C and Fortran77 interfaces to a portably efficient BLAS implementation, as well as a few routines from LAPACK
  - Prebuilt versions for various architectures
  - Build it from source
    - check the ATLAS errata file
    - may take several hours

High-Level Programming With MPI

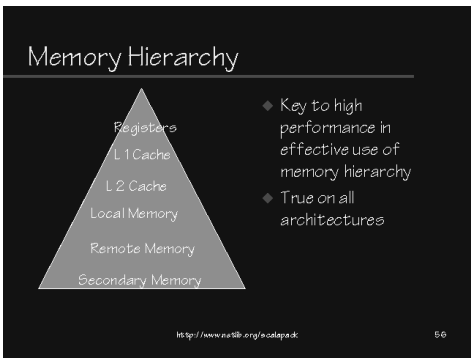
- MPI was designed from the beginning to support libraries
- Many libraries exist, both open source and commercial
- Sophisticated numerical programs can be built using libraries
  - Dense Linear algebra
  - Sparse Linear Algebra
  - Solve a PDE (e.g., PETSc)
  - Fast Fourier Transforms
  - Scalable I/O of data to a community standard file format

ScaLAPACK

- **ScaLAPACK** (or Scalable LAPACK) library includes a subset of **LAPACK** routines redesigned for distributed memory MIMD parallel computers
- http://www.netlib.org/scalapack/scalapack_home.html
- Latest in sequence of libraries LINPACK, EISPACK, LAPACK
- written in a Single-Program-Multiple-Data style using explicit message passing
- assumes matrices are laid out in a two-dimensional block cyclic decomposition
- based on block-partitioned algorithms in order to minimize the frequency of data movement between different levels of the memory hierarchy

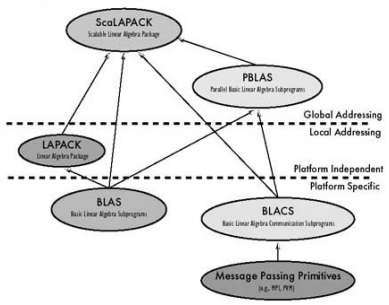
Higher Level I/O Libraries

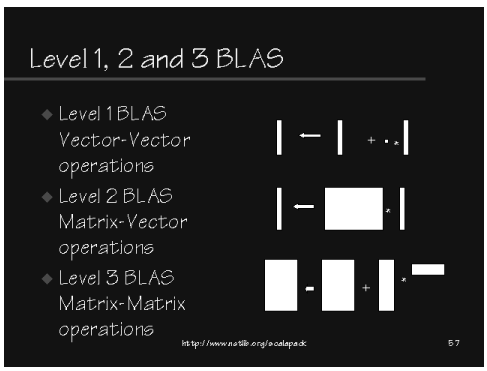
- Scientific applications work with structured data and desire mor e self-describing file formats
- netCDF and HDF5 are two popular “higher level” I/O libraries
  - Abstract away details of file layout
  - Provide standard, portable file formats
  - Include metadata describing contents
- For parallel machines, these should be built on top of MPI-IO



ScaLAPACK

- Based on
- distributed memory versions (PBLAS) of the **Level 1, 2 and 3 BLAS**,
- a set of Basic Linear Algebra Communication Subprograms (BLACS) for communication tasks that arise frequently in parallel linear algebra computations
- all interprocessor communication occurs within the **PBLAS** and the **BLACS**
- See tutorial for more details
  - http://www.netlib.org/scalapack/tutorial/





ScaLAPACK

AVAILABLE SOFTWARE:

Dense, Band, and Tridiagonal Linear Systems
- general
- symmetric positive definite

Full-Rank Linear Least Squares
Standard and Generalized Orthogonal Factorizations
Eigensolvers
- SEP: Symmetric Eigenproblem
- NEP: Nonsymmetric Eigenproblem
- GSEP: Generalized Symmetric Eigenproblem

SVD

Prototype Codes
- HPF interface to ScaLAPACK
- Matrix Sign Function for Eigenproblems
- Out-of-core solvers (LU, Cholesky, QR)
- Super LU
- PBLAS (algorithmic blocking and no alignment restrictions.)

DOCUMENTATION:

ScaLAPACK Users' Guide
http://www.netlib.org/scalapack/slug/scalapack_slug.html

Future Work
- Out-of-core Eigensolvers
- Divide and Conquer routines
- C++ and Java Interfaces

Commercial Use

ScaLAPACK has been incorporated into the following software packages:
- NAG Numerical Library
- IBM Parallel ESSL
- SGI Cray Scientific Software Library

and is being integrated into the VNI IMSL Numerical Library, as well as software libraries for Fujitsu, HP/Convex, Hitachi, and NEC.

http://www.netlib.org/scalapack/

PLAPACK

- Designed for coding linear algebra algorithms at a high level of abstraction
- http://www.cs.utexas.edu/users/plapack/
- includes Cholesky, LU, and QR factorization based solvers for symmetric positive definite, general, and overdetermined systems of equations, respectively
- More OO in style
- raising the level of abstraction sacrifices some perfromance but more sophisticated algorithms can be implemented, which allows high levels of performance to be regained

Aztec

- A massively parallel iterative solver for solving sparse linear systems
- grew out of a specific application: modeling reacting flows (**MPSalsa**)
- easy-to-use and efficient
- global distributed matrix allows a user to specify pieces (different rows for different processors) of his application matrix exactly as he would in the serial setting
- Issues such as local numbering, ghost variables, and messages are instead computed by an automated transformation function.

Spare Linear Systems

- SuperLU
  - http://crd.lbl.gov/~xiaoye/SuperLU/
  - direct solution of large, sparse, nonsymmetric systems
  - SuperLU for sequential machines
  - SuperLU_MT for shared memory parallel machines
  - SuperLU_DIST for distributed memory
  - perform an LU decomposition with partial pivoting and triangular system solves through forward and back substitution
  - Distributed memory version uses static pivoting instead to avoid large numbers of small messages

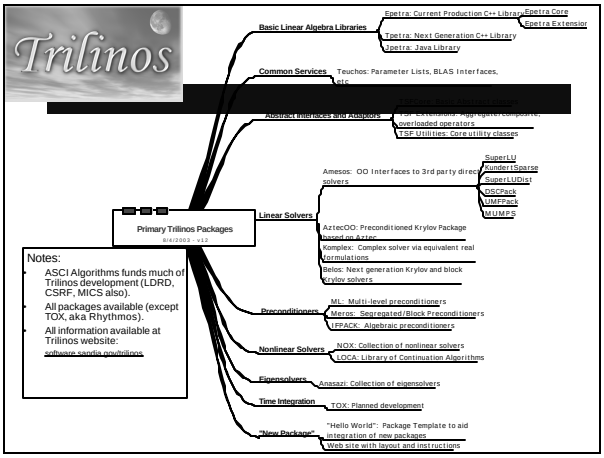
Trilinos

- an effort to develop parallel solver algorithms and libraries within an object-oriented software framework for the solution of large-scale, complex multi-physics engineering and scientific applications
- unique design feature of Trilinos is its focus on packages
- Aztec now part of Trilinos

Trilinos Packages

- Trilinos is a collection of *Packages*.
- Each package is:
  - Focused on important, state-of-the-art algorithms in its problem regime.
  - Developed by a small team of domain experts.
  - Self-contained: No explicit dependencies on any other software packages (with some special exceptions).
  - Configurable/buildable/documented on its own.
- Sample packages: NOX, AztecOO, IFPACK, Meros.
- Special package collections: Petra, TSF, Teuchos.

Package	Description	Release			
		3.1 (Current)	3.1 (Limited)	4 (Current)	4 (Limited)
Amesos	3rd Party Direct Solver Suite		X	X	X
Anasazi	Eigensolver package				X
AztecOO	Linear Iterative Methods	X	X	X	X
Belos	Block Linear Solvers				X
Epetra	Basic Linear Algebra	X	X	X	X
EpetraExt	Extensions to Epetra		X	X	X
Ifpack	Algebraic Preconditioners	X	X	X	X
Jpetra	Java Petra Implementation				X
Kokkos	Sparse Kernels			X	X
Komplex	Complex Linear Methods	X	X	X	X
LOCA	Bifurcation Analysis Tools	X	X	X	X
Meros	Segregated Preconditioners		X		X
ML	Multi-level Preconditioners	X	X	X	X
NewPackage	Working Package Prototype	X	X	X	X
NOX	Nonlinear solvers	X	X	X	X
Pliris	Dense direct Solvers			X	X
Teuchos	CommonUtilities			X	X
TSFCore	Abstract Solver API			X	X
TSFExt	Extensions to TSFCore			X	X
Tpetra	Templated Petra				X
Totals		**8**	**11**	**15**	**20**



Three Special Trilinos Package Collections

- **Petra**: Package of concrete linear algebra classes: Operators, matrices, vectors, graphs, etc.
  - Provides working, parallel code for basic linear algebra computations.
- **TSF**: Packages of abstract solver classes: Solvers, preconditioners, matrices, vectors, etc.
  - Provides an application programmer interface (API) to any other package that implements TSF interfaces.
  - Inspired by HCL.
- **Teuchos (pronounced Tef-hos)**: Package of basic tools:
  - Common Parameter list, smart pointer, error handler, timer.
  - Interface to BLAS, LAPACK, MPI, XML, ...
  - Common traits mechanism.
  - Goal: Portable tools that enhance interoperability between packages.

Dependence vs. Interoperability

- Although most Trilinos packages have no explicit dependence, each package must interact with *some* other packages:
  - NOX needs operator, vector and solver objects.
  - AztecOO needs preconditioner, matrix, operator and vector objects.
  - Interoperability is enabled at configure time. For example, NOX:
    --enable-nox-lapack compile NOX lapack interface libraries
    --enable-nox-epetra compile NOX epetra interface libraries
    --enable-nox-petsc compile NOX petsc interface libraries
- Trilinos is a vehicle for:
  - Establishing interoperability of Trilinos components…
  - Without compromising individual package autonomy.
- Trilinos offers five basic interoperability mechanisms.

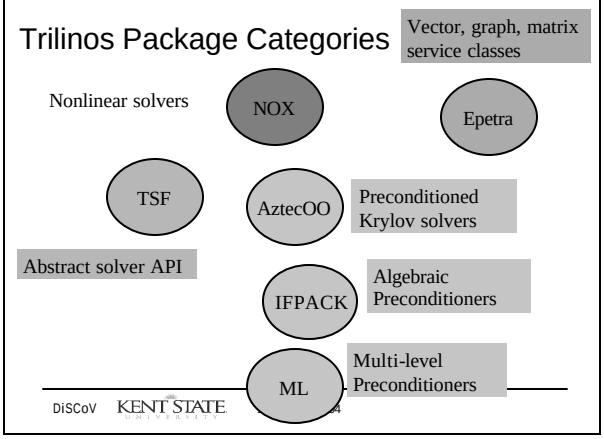
Interoperability Example: AztecOO

- AztecOO: Preconditioned Krylov Solver Package.
- Primary Developer: Mike Heroux.
- Minimal *explicit, essential* dependence on other Trilinos packages.
  - Uses abstract interfaces to matrix/operator objects.
  - Has independent configure/build process (but can be invoked at T rilinos level).
  - Sole dependence is on Epetra (but easy to work around).
- *Interoperable* with other Trilinos packages:
  - Accepts user data as Epetra matrices/vectors.
  - Can use Epetra for internal matrices/vectors.
  - Can be used via TSF abstract interfaces.
  - Can be built via Trilinos configure/build process.
  - Can provide solver services for NOX.
  - Can use IFPACK, ML or AztecOO objects as preconditioners.

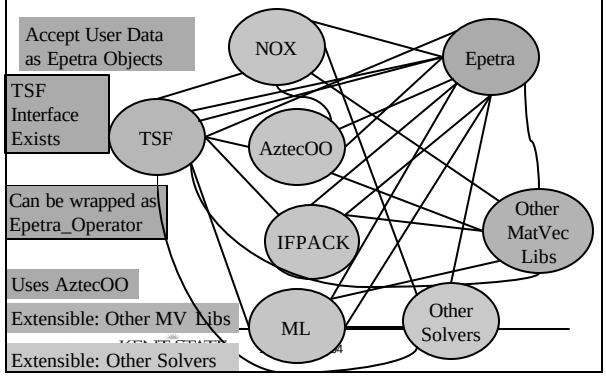
Trilinos Interoperability Mechanisms

- M1: *Package* accepts user data as Epetra or TSF objects.
  =>Applications using Epetra/TSF can use *package*.
- M2: *Package* can be used via TSF abstract solver classes.
  => Applications or other packages using TSF can use *package*.
- M3: *Package* can use Epetra for private data.
  => *Package* can then use other packages that understand Epetra.
- M4: *Package* accesses solver services via TSF interfaces.
  => *Package* can then use other packages that implement TSF interfaces.
- M5: *Package* builds under Trilinos `configure` scripts.
  => *Package* can be built as part of a suite of packages.
  => Cross-package dependencies can be handled automatically.

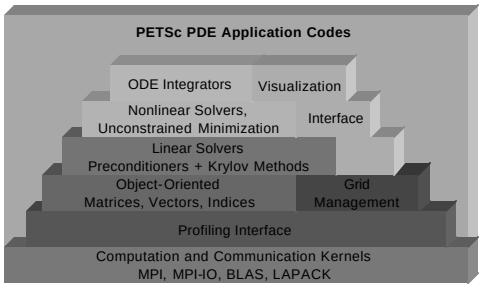
Trilinos Package Categories



Trilinos Package Interoperability



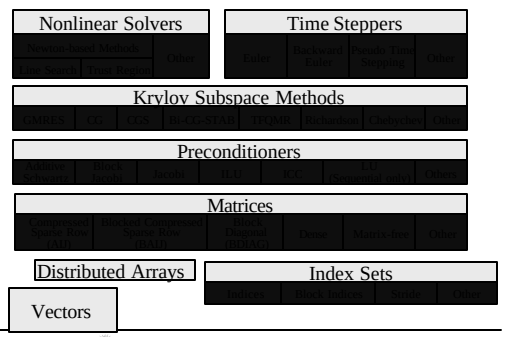
Structure of PETSc



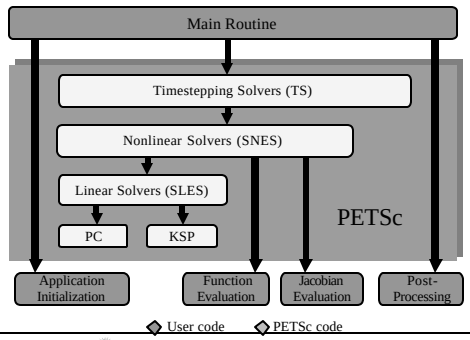
The PETSc Library

- PETSc provides routines for the parallel solution of systems of equations that arise from the discretization of PDEs
  - Linear systems
  - Nonlinear systems
  - Time evolution
- PETSc also provides routines for
  - Sparse matrix assembly
  - Distributed arrays
  - General scatter/gather (e.g., for unstructured grids)

PETSc Numerical Components



Flow of Control for PDE Solution



Fast Fourier Transform

- FFTW - **Fastest Fourier Transform in the West**
- **MPI** parallel transforms are only available in 2.1.5
- Received the 1999 J. H. Wilkinson Prize for Numerical Software
- Features
  - Speed. (Supports SSE/SSE2/3dNow!/Altivec, new in version 3.0.)
  - Both one-dimensional and **multi-dimensional** transforms.
  - **Arbitrary-size** transforms. (Sizes with small prime factors are best, but FFTW uses O(N log N) algorithms even for prime sizes.)
  - Fast transforms of **purely real** input or output data.
  - Parallel transforms: parallelized code for platforms with Cilk or for SMP machines with some flavor of threads (e.g. POSIX). An MPI version for distributed-memory transforms is also available, currently only as part of FFTW 2.1.5.
  - **Portable** to any platform with a C compiler.

Eigenvalue Problems

- ScaLAPACK and PLAPACK
- ARPACK
  - designed to compute a few eigenvalues and corresponding eigenvectors of a general n by n matrix A.
  - most appropriate for large sparse or structured matrices A where structured means that a matrix-vector product w <- Av requires order n rather than the usual order n2 floating point operations
  - based upon an algorithmic variant of the Arnoldi process called the Implicitly Restarted Arnoldi Method (IRAM)
  - Reverse Communication Interface
    - No need for user to pass the matrix to library
    - Can work with any user defined data structure or with matrices that are operatively defined

Load balancing

- Read about Graph Partitioning Algorithms
- Parmetis
  - MPI-based parallel library that implements a variety of algorithms for partitioning unstructured graphs, meshes, and for computing fill-reducing orderings of sparse matrices.
  - http://www-users.cs.umn.edu/~karypis/metis/parmetis/
- Chaco
- Zoltan

Applications

- Gaussian
  - predicts the energies, molecular structures, and vibrational frequencies of molecular systems, along with numerous molecular properties derived from these basic computation types
- Fluent
  - Computational fluid dynamics
- MSC/Nastran
  - CAE/structural finite element code
- LS-DYNA
  - general purpose nonlinear finite element program
- NAMD
  - recipient of a 2002 Gordon Bell Award, is a parallel molecular dynamics code designed for high-performance simulation of large biomolecular systems
- NWChem
- provides many methods to compute the properties of molecular and periodic systems using standard quantum mechanical descriptions of the electronic wavefunction or density

Some Research Areas

- MPI-2 RMA interface
  - Can we get high performance?
- Fault Tolerance and MPI
  - Are intercommunicators enough?
- MPI on 64K processors
  - Umm…how do we make this work :)?
  - Reinterpreting the MPI "process"
- MPI as system software infrastructure
  - With dynamic processes and fault tolerance, can we build services on MPI?

Getting MPI for your cluster

- MPI standard
  - http://www.mcs.anl.gov/mpi/
- MPICH
  - http://www.mcs.anl.gov/mpi/mpich
  - Either MPICH-1 or
  - MPICH-2
- LAM
  - http://www.lam-mpi.org
- MPICH-GM
  - http://www.myricom.com
- MPICH-G2
  - http://www.niu.edu/mpi
- Many other versions see book