An Overview of Trilinos:

Packages for Parallel Formulation and Solution of Scientific and Engineering Problems

Michael A. Heroux
Sandia National Laboratories

CScADS Workshop, July-August 2007

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

Trilinos Contributors

Chris Baker
Developer of Anasazi, RBGen

Ross Bartlett
Lead Developer of MOOCHO, Stratimikos, RTOp, Thyra
Developer of Rythmos

Paul Boggs
Developer of Thyra

Erik Boman
Lead Developer Isorropia

Todd Coffey
Lead Developer of Rythmos

Jason Cross
Developer of Jpetra

David Day
Developer of Komplex

Clark Dohrmann
Lead developer of CLAPS

Michael Gee
Developer of ML, Moertel, NOX

Bob Heaphy
Lead developer of Trilinos SQA

Mike Heroux
Trilinos Project Leader
Lead Developer of Epetra, AztecOO, Kokkos, Komplex, IFPACK, Thyra, Tpetra
Developer of Amesos, Belos, EpetraExt, Jpetra, Teuchos

Ulrich Hetmaniuk
Developer of Anasazi

Robert Hoekstra
Lead Developer of EpetraExt
Developer of Epetra, Thyra, Tpetra

Russell Hooper
Developer of NOX

Vicki Howle
Lead Developer of Meros
Developer of Belos and Thyra

Jonathan Hu
Developer of ML

Sarah Knepper
Developer of Komplex

Tammy Kolda
Lead Developer of NOX

Joe Kotulski
Lead Developer of Pliris

Rich Lehoucq
Developer of Anasazi and Belos

Kevin Long
Lead Developer of Thyra
Developer of Belos and Teuchos

Roger Pawlowski
Lead Developer of NOX

Michael Phenow
Trilinos Webmaster
Lead Developer of New_Package
Developer WebTrilinos

Eric Phipps
Lead developer Sacado
Developer of LOCA, NOX

Dennis Ridzal
Lead Developer of Aristos

Marzio Sala
Lead Developer of Didasko, Galeri, IFPACK, WebTrilinos
Developer of ML, Amesos

Andrew Salinger
Lead Developer of LOCA, Capo

Paul Sexton
Developer of Epetra and Tpetra

Bob Shuttleworth
Developer of Meros.

Chris Siefert
Developer of ML

Bill Spotz
Lead Developer of PyTrilinos
Developer of Epetra, New_Package

Ken Stanley
Lead Developer of Amesos and New_Package

Heidi Thornquist
Lead Developer of Anasazi, Belos, RBGen and Teuchos

Ray Tuminaro
Lead Developer of ML and Meros

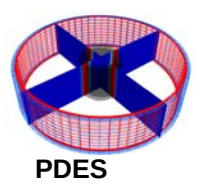
Jim Willenbring
Developer of Epetra and New_Package.
Trilinos library manager

Alan Williams
Lead Developer Isorropia
Developer of Epetra, EpetraExt, AztecOO, Tpetra

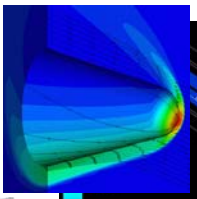
Background/Motivation

Sandia National Laboratories

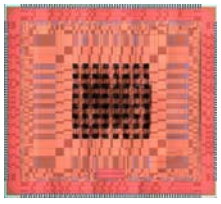
Target Problems: PDES and more…

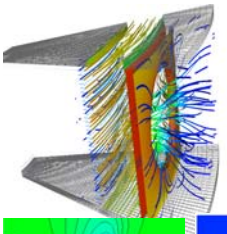


PDES

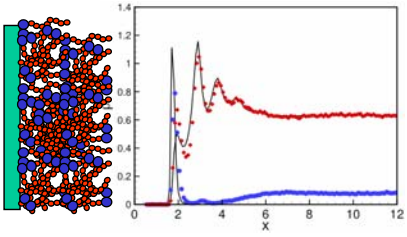


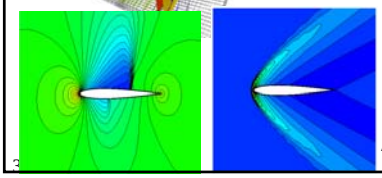
Circuits





Inhomogeneous Fluids





And More…

Target Platforms: Any and All
(Now and in the Future)

- Desktop: Development and more…
- Capability machines:
  - Redstorm (XT3), Clusters
  - BG/L.
  - Large-count multicore nodes.
- Parallel software environments:
  - MPI of course.
  - UPC, CAF, threads, vectors,…
  - Combinations of the above.
- User "skins":
  - C++/C, Python
  - Fortran.
  - Web, CCA.

Motivation For Trilinos

- Sandia does LOTS of solver work.
- When I started at Sandia in May 1998:
  - Aztec was a mature package. Used in many codes.
  - FETI, PETSc, DSCPack, Spooles, ARPACK, DASPK, and many other codes were (and are) in use.
  - New projects were underway or planned in multi-level preconditioners, eigensolvers, non-linear solvers, etc…
- The challenges:
  - Little or no coordination was in place to:
    - Efficiently reuse existing solver technology.
    - Leverage new development across various projects.
    - Support solver software processes.
    - Provide consistent solver APIs for applications.
  - ASCI (now ASC) was forming software quality assurance/engineering (SQA/SQE) requirements:
    - Daunting requirements for any single solver effort to address alone.

Sandia National Laboratories

Evolving Trilinos Solution

- Trilinos[1] is an evolving framework to address these challenges:
  - Fundamental atomic unit is a *package*.
  - Includes core set of vector, graph and matrix classes (Epetra/Tpetra packages).
  - Provides a common abstract solver API (Thyra package).
  - Provides a ready-made package infrastructure (new_package package):
    - Source code management (cvs, bonsai).
    - Build tools (autotools).
    - Automated regression testing (queue directories within repository).
    - Communication tools (mailman mail lists).
  - Specifies requirements and suggested practices for package SQA.
- In general allows us to categorize efforts:
  - Efforts best done at the Trilinos level (useful to most or all packages).
  - Efforts best done at a package level (peculiar or important to a package).
  - **Allows package developers to focus only on things that are unique to their package.**

1. Trilinos loose translation: "A string of pearls"

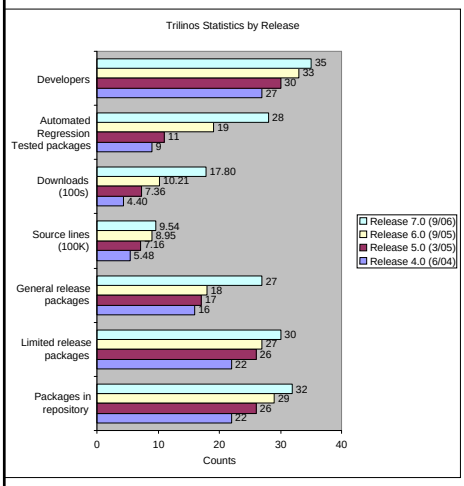
Sandia National Laboratories

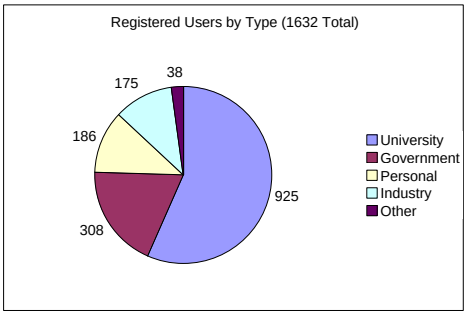
Continued Evolution

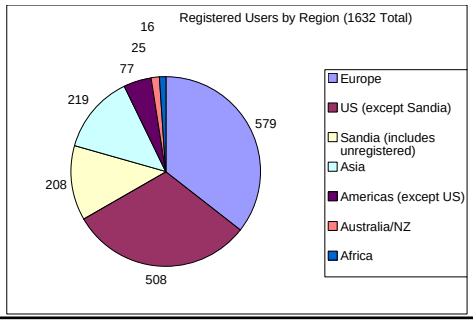
- New capabilities extend scope:
  - Sacado: AD Tools. (E. Phipps, D. Gay)
  - Entrepid: Discretizations (P. Bochev, D. Ridzal).
  - Others coming…
- Becoming more than just solvers.

Sandia National Laboratories

Trilinos Statistics







Stats: Trilinos Download Page 06/14/2007.

External Visibility

- Awards: R&D 100, HPC SW Challenge (04).
- www.cfd-online.com:

Trilinos

A project led by Sandia to develop an object-oriented software framework for scientific computations. This is an active project which includes several state-of-the-art solvers and lots of other nice things a software engineer writing CFD codes would find useful. Everything is freely available for download once you have registered. Very good!

- Industry Collaborations: Boeing, Goodyear, ExxonMobil.
- Linux distros: Debian, Mandriva.
- Star-P Interface.
- Over 4200 downloads since March 2005.
- Occasional unsolicited external endorsements such as the following two-person exchange on mathforum.org:

> The consensus seems to be that OO has little, if anything, to offer
> (except bloat) to numerical computing.
I would completely disagree. A good example of using OO in numerics is
Trilinos: http://software.sandia.gov/trilinos/

Trilinos Presentation Forums

- ACTS “Hands-on” Tutorial:
  - Aug 21-23, 2007.
  - At Lawrence Berkeley Lab, Berkeley, CA, USA.
- Next Trilinos User Group Meeting:
  - Nov 6-8, 2007.
  - At Sandia National Laboratories, Albuquerque, NM, USA.

Sandia National Laboratories

TOPS-2 Participation

- Trilinos new member of TOPS.
- Activities start July 2008.
- Looking for application partners.
- Preparation activities:
  - Interoperability with PETSc:
    - PETSc-Trilinos Data objects interoperable.
    - Interoperability at other levels in time.
  - Fortran 9X User Interface:
    - Support construction of matrices/vectors via OO Fortran interface.
    - Support selection of solvers and parameters via Fortran.
  - Attend meetings like this one.

Sandia National Laboratories

Trilinos Package Concepts

Package: The Atomic Unit

Sandia National Laboratories

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, ML, IFPACK, Meros.
- Special package collections:
  - Petra (Epetra, Tpetra, Jpetra): Concrete Data Objects
  - Thyra: Abstract Conceptual Interfaces
  - Teuchos: Common Tools.
  - New_package: Jumpstart prototype.

Sandia National Laboratories

Objective	Package(s)
Linear algebra objects	Epetra, Jpetra, Tpetra
Krylov solvers	AztecOO, **Belos**, Komplex
ILU-type preconditioners	AztecOO, IFPACK
Multilevel preconditioners	ML, CLAPS
Eigenvalue problems	Anasazi
Block preconditioners	Meros
Direct sparse linear solvers	Amesos
Direct dense solvers	Epetra, Teuchos, Pliris
Abstract interfaces	Thyra
Nonlinear system solvers	NOX, LOCA
Time Integrators/DAEs	Rythmos
C++ utilities, (some) I/O	Teuchos, EpetraExt, Kokkos
Trilinos Tutorial	Didasko
"Skins"	PyTrilinos, WebTrilinos, Star-P, Stratimikos
Optimization	MOOCHO**, Aristos**
Archetype package	NewPackage
Others	Galeri, Isorropia, Moertel, RTOp, **Sacado**

Trilinos Package Summary

Basic Linear Algebra classes

Block Krylov Methods: Linear: CG, GMRES Scalable: OBPCG, block Davidson, Preconditioners, ...

3rd Party Direct Solver Package.

Abstract Interfaces.

Full "Vertical" Solver Coverage

Trilinos Packages

· Optimization Problems: · Unconstrained: · Constrained:	 Find $u \in \Re^n$ that minimizes $f(u)$ Find $y \in \Re^m$ and $u \in \Re^n$ that minimizes $f(y,u)$ s.t. $c(y,u)=0$	MOOCHO
· Transient Problems: · DAEs/ODEs:	Solve $f(\dot{x}(t),x(t),t)=0$ $t \in [0,T],\ x(0)=x_0,\ \dot{x}(0)=x_0'$ for $x(t) \in \Re^n,\ t \in [0,T]$	Rythmos
· Nonlinear Problems: · Nonlinear equations: · Stability analysis:	Given nonlinear op $c(x,u) \in \Re^{n+m} \rightarrow \Re^n$ Solve $c(x)=0$ for $x \in \Re^n$ For $c(x,u)=0$ find space $u \in U \ni \frac{\partial c}{\partial x}$ singular	NOX LOCA
· Implicit Linear Problems: · Linear equations: · Eigen problems:	Given linear ops (matrices) $A, B \in \Re^{n \times n}$ Solve $Ax=b$ for $x \in \Re^n$ Solve $Av=\lambda Bv$ for (all) $v \in \Re^n,\ \lambda \in \Re$	AztecOO, Belos, Ifpack,ML,etc. Anasazi
· Explicit Linear Problems: · Matrix/graph equations: · Vector problems:	 Compute $y=Ax;\ A=A(G);\ A \in \Re^{m \times n}, G \in \square^{m \times n}$ Compute $y=\alpha x+\beta w;\ \alpha =< x,y >;\ x,y \in \Re^n$	Epetra, Tpetra

New in Trilinos 8.0

- Sacado (E. Phipps, D. Gay) AD tools via templated types:
  - Write residual evaluation using template scalar type.
  - Get Jacobian "for free."
- Belos (H. Thornquist, M. Parks) Block Krylov Linear solvers:
  - Block CG.
  - Block GMRES:
    - Modular implementation: e.g., orthogonalization interface, several adapters.
    - All BLAS 3 kernels.
  - Pseudo-Block GMRES:
    - Synch'ed independent GMRES's.
    - BLAS 3 SpMM, SpSM.
  - Single RHS GCRO-DR: Recycling GMRES:
    - Useful for sequences of systems: $A^i x^i = b^i, \quad i = 1, \ldots$
- Aristos (D. Ridzal) Nonlinear continuous optimization package :
  - Based on full-space SQP methods.
  - Specifically designed for large-scale constrained optimization problems with inexact linearized constraint equations solves.

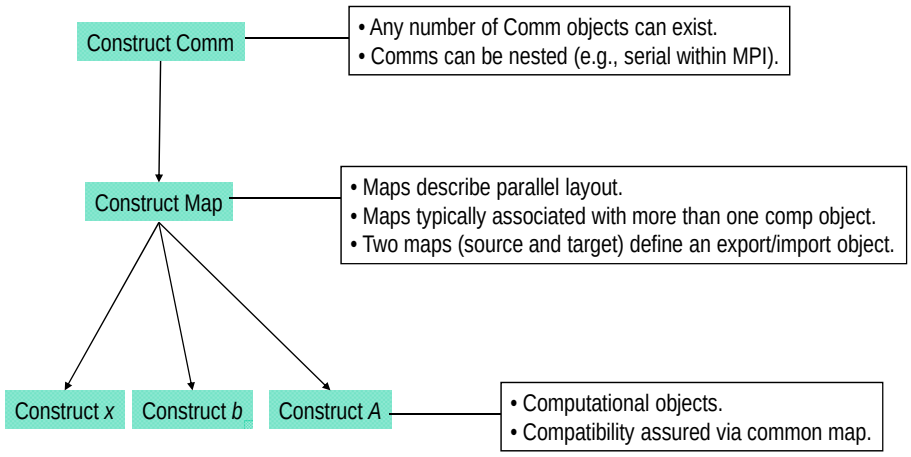
Managing Parallel Data

The Petra Object Model

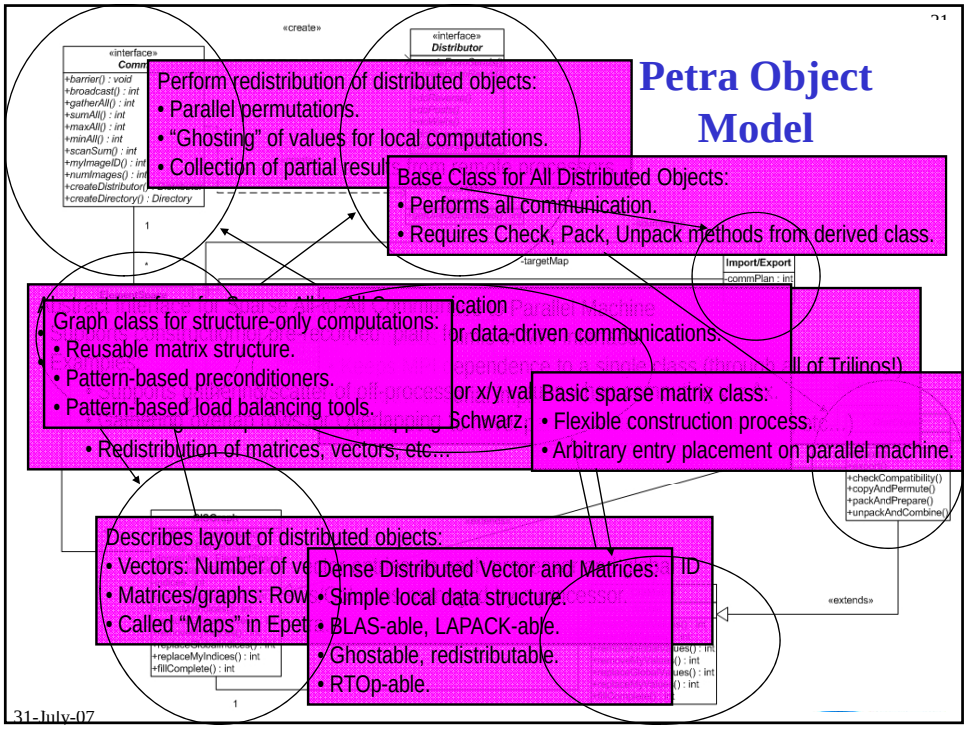
A Simple Epetra/AztecOO Program

```
// Header files omitted…
int main(int argc, char *argv[]) {
  MPI_Init(&argc,&argv); // Initialize MPI, MpiComm
  Epetra_MpiComm Comm( MPI_COMM_WORLD );

// ***** Map puts same number of equations on each pe *****

  int NumMyElements = 1000 ;
  Epetra_Map Map(-1, NumMyElements, 0, Comm);
  int NumGlobalElements = Map.NumGlobalElements();

// ***** Create an Epetra_Matrix  tridiag(-1,2,-1) *****

  Epetra_CrsMatrix A(Copy, Map, 3);
  double negOne = -1.0; double posTwo = 2.0;

  for (int i=0; i<NumMyElements; i++) {
    int GlobalRow = A.GRID(i);
    int RowLess1 = GlobalRow - 1;
    int RowPlus1 = GlobalRow + 1;
    if (RowLess1!=-1)
      A.InsertGlobalValues(GlobalRow, 1, &negOne, &RowLess1);
    if (RowPlus1!=NumGlobalElements)
      A.InsertGlobalValues(GlobalRow, 1, &negOne, &RowPlus1);
    A.InsertGlobalValues(GlobalRow, 1, &posTwo, &GlobalRow);
  }
A.FillComplete(); // Transform from GIDs to LIDs

  // ***** Create x and b vectors *****
  Epetra_Vector x(Map);
  Epetra_Vector b(Map);
  b.Random(); // Fill RHS with random #s

// ***** Create Linear Problem *****
  Epetra_LinearProblem problem(&A, &x, &b);

  // ***** Create/define AztecOO instance, solve *****
  AztecOO solver(problem);
  solver.SetAztecOption(AZ_precond, AZ_Jacobi);
  solver.Iterate(1000, 1.0E-8);

// ***** Report results, finish **********************
  cout << "Solver performed " << solver.NumIters()
       << " iterations." << endl
       << "Norm of true residual = "
       << solver.TrueResidual()
       << endl;

  MPI_Finalize() ;
  return 0;
}
```


Typical Flow of Epetra Object Construction



Sandia National Laboratories



22

Petra Implementations

- Three version under development:
- **Epetra** (Essential Petra):
  - Current production version.
  - Restricted to real, double precision arithmetic.
  - Uses stable core subset of C++ (circa 2000).
  - Interfaces accessible to C and Fortran users.
- **Tpetra** (Templated Petra):
  - Next generation C++ version.
  - Templated scalar and ordinal fields.
  - Uses namespaces, and STL: Improved usability/efficiency.
- **Jpetra** (Java Petra):
  - Pure Java. Portable to any JVM.
  - Interfaces to Java versions of MPI, LAPACK and BLAS via interfaces.

Sandia National Laboratories

31-July-07

Epetra Performance Optimization Guide
SAND2005-1668

- Topics:
  - 3rd Party Libraries: BLAS and LAPACK
  - Epetra_MultiVector Data Layout
  - Epetra_CrsGraph Construction
  - Epetra_CrsMatrix Construction
  - Selecting the Right Sparse Matrix Class
  - Parallel Data Redistribution
  - General Practices
- Tiered Approach to practices.

Sandia National Laboratories

Practice Categories

Each practice falls into one of three categories:

VSR — Very Strongly Recommended - Practices necessary for Epetra to perform well.

SR — Strongly Recommended - Practices that are definitely a good thing or that have proved to be valuable.

R — Recommended - Practices that are probably a good idea.

Sandia National Laboratories

Data Model Wrap-Up

- Our target apps and computer platforms require flexible data model.
- Petra Object Model (POM) supports:
  - Arbitrary placement of vector, graph, matrix entries on parallel machine.
  - Arbitrary redistribution, ghosting and collection of distributed data.
  - Coarse-grain abstraction for key objects.
- This flexibility is needed by LALs:
  - Algebraic and multi-level preconditioners.
  - Concrete distributed matrix kernels.
  - Direct methods.
- Also needed for future architectures: Multicore, GPUs, CELL, etc.
- Also needed for future parallel languages & libraries: UPC, CAF, etc.
- Final Notes:
  - POM is complex: BUT Non-LALs (ANAs) do not rely on it.
  - Parallel Data Redistribution is a huge topic I must skip for this talk (ask about it at break if interested).

Templates and Generic Programming

Investment in Generic Programming

- 2nd Generation Trilinos packages are templated on:
  - OrdinalType (think **int**).
  - ScalarType (think **double**).
- Examples:
  Teuchos::SerialDenseMatrix<int, double> A;
  Teuchos::SerialDenseMatrix<short, float> B;
- Scalar/Ordinal Traits mechanism completes support for genericity.
- The following packages support templates:

  Teuchos (Basic Tools)
  Thyra (Abstract Interfaces)
  Tpetra (including MPI support)
  Belos (Krylov and Block Krylov Linear),
  IFPACK (algebraic preconditioners, next version),
  Anasazi (Eigensolvers),
  Sacado (AD),
  MOOCHO(Optimization)

Sandia National Laboratories

Potential Benefits of Templated Types

Templated scalar (and ordinal) types have great potential:

- Generic: Algorithms expressed over abstract field.
- High Performance: Partial template specialization + Fortran.
- Facilitate variety of algorithmic studies.
- Allow study of asymptotic behavior of discretizations.
- Facilitate debugging: Reduces FP error as source error.
- UQ, QMU studies.
- Use your imagination…
- All new Trilinos packages are templated.
- Dark side: Compilation times can soar (but don't have to).

Sandia National Laboratories

Other Interface Packages: “Skins”

PyTrilinos, WebTrilinos, ForTrilinos

Sandia National Laboratories

PyTrilinos

- PyTrilinos provides Python access to Trilinos packages.
- Uses SWIG to generate bindings.
- Epetra, AztecOO, Galeri, IFPACK, ML, NOX, LOCA, Teuchos, Thyra, Anasazi, Amesos and NewPackage are support.
- Possible to:
  - Define RowMatrix implementation in Python.
  - Use from Trilinos C++ code.
- Performance for large grain is equivalent to C++.
- Several times hit for very fine grain code.

Sandia National Laboratories

C++ vs. Python: Equivalent Code

```
#include "mpi.h"
#include "Epetra_MpiComm.h"
#include "Epetra_Vector.h"
#include "Epetra_Time.h"
#include "Epetra_RowMatrix.h"
#include "Epetra_CrsMatrix.h"
#include "Epetra_Time.h"
#include "Epetra_LinearProblem.h"
#include "Trilinos_Util_CrsMatrixGallery.h"
using namespace Trilinos_Util;
int main(int argc, char *argv[])
{
MPI_Init(&argc, &argv);
Epetra_MpiComm Comm(MPI_COMM_WORLD);
int nx = 1000;
int ny = 1000 * Comm.NumProc();
CrsMatrixGallery Gallery("laplace_2d", Comm);
Gallery.Set("ny", ny);
Gallery.Set("nx", nx);
Gallery.Set("problem_size",nx*ny);
Gallery.Set("map_type", "linear");
Epetra_LinearProblem* Problem =
        Gallery.GetLinearProblem();
assert (Problem != 0);
// retrieve pointers to solution (lhs), right-hand side (rhs)
// and matrix itself (A)
Epetra_MultiVector* lhs = Problem->GetLHS();
Epetra_MultiVector* rhs = Problem->GetRHS();
Epetra_RowMatrix* A = Problem->GetMatrix();
Epetra_Time Time(Comm);
for (int i = 0 ; i < 10 ; ++i)
A->Multiply(false, *lhs, *rhs);
cout << Time.ElapsedTime() << endl;
MPI_Finalize();
return(EXIT_SUCCESS);
} // end of main()
```

```
#! /usr/bin/env python
from PyTrilinos import Epetra, Triutils
Epetra.Init()
Comm = Epetra.PyComm()
nx = 1000
ny = 1000 * Comm.NumProc()
Gallery = Triutils.CrsMatrixGallery("laplace_2d",
      Comm)
Gallery.Set("nx", nx)
Gallery.Set("ny", ny)
Gallery.Set("problem_size", nx * ny)
Gallery.Set("map_type", "linear")
Matrix = Gallery.GetMatrix()
LHS = Gallery.GetStartingSolution()
RHS = Gallery.GetRHS()
Time = Epetra.Time(Comm)
for i in xrange(10):
Matrix.Multiply(False, LHS, RHS)
print Time.ElapsedTime()
Epetra.Finalize()
```

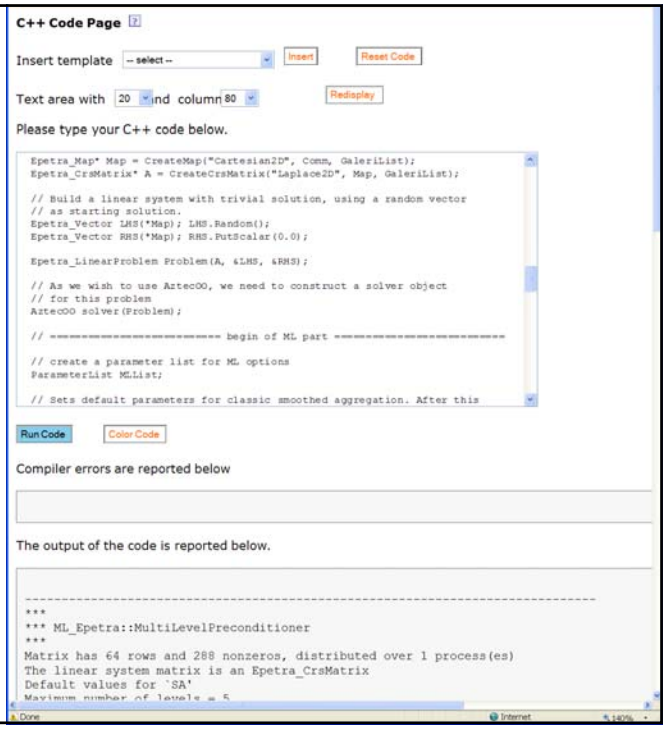
Sandia National Laboratories

Web Interface

- Yet another Trilinos Package
- Supports:
  - Use of Trilinos via simple Web Editor
  - User or generated data sets.
  - Compilation and execution on webserver.

Sample Page

- Intended use:
  - Installation on departmental intranet.
  - Allows first time users easy access.
  - Supports “just-in-time” use of solvers.



Fortran Interface

- Presently Trilinos has no full-featured Fortran interface.
- Plans in place to develop OO Fortran 9X interface.
- Will Develop as part of SciDAC TOPS-2 effort.
- Interested in a target Fortran application for collaboration.

Formal Software Engineering

CSE Software is not IT Software

Sandia National Laboratories

SQA/SQE

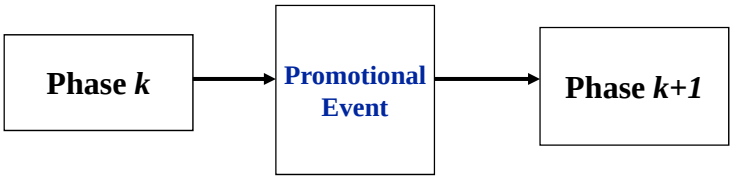
- Software Quality Assurance/Engineering is important.
- Not sufficient to say, “We do a good job.”
- Trilinos facilitates SQA/SQE development/processes for packages:
  - 10 of 30 ASC SQE practices are directly handled by Trilinos (no requirements on packages).
  - Trilinos provides infrastructure support for the remaining 20.
  - Trilinos Dev Guide Part II: Specific to ASC requirements.
  - Trilinos software engineering policies provide a ready-made infrastructure for new packages.
  - Trilinos philosophy:
    Few *requirements*. Instead mostly *suggested practices.* Provides package with option to provide alternate process.

Sandia National Laboratories

Trilinos Service	SQE Practices Impact
Yearly Trilinos User Group Meeting (TUG) and Developer Forum: Once a year gathering for tutorials, package feature updates, user/developer requirements discussion and developer training.	— All Requirements steps: gathering, derivation, documentation, feasibility,etc. — User and Developer training.
Monthly Trilinos leaders meetings: Trilinos leaders, including package development leaders, key managers, funding sources and other stakeholders participate in monthly phone meetings to discuss any timely issues related to the Trilinos Project.	—Developer Training. —Design reviews. —Policy decisions across all development phases.
Trilinos and package mail lists: Trilinos lists for leaders, announcements, developers, users, checkins and similar lists at the package level support a variety of communication. All lists are archived, providing critical artifacts for assessments and audits.	—Developer/user/client communication. —Requirements/design/testing artifacts. —Announcement/documenting of releases.
Trilinos and Trilinos3PL source repositories: All source code, development and user documentation is retained and tracked. In addition, reference versions of all external software, including BLAS, LAPACK, Umfpack, etc. are retained in Trilinos3PL.	—Source management. —Versioning. —Third-party software management.
Bugzilla Products: Each package has its own Bugzilla Product with standard components.	—Requirements/faults capturing and tracking.
Trilinos configure script and M4 macros: The Trilinos configure script and related macros support portable installation of Trilinos and its packages	—Portability. —Software release.
Trilinos test harness: Trilinos provides a base testing plan and automated testing across multiple platforms, plus creation of testing artifacts. Test harness results are used to derive a variety of metrics for SQE.	—Pre-checkin and regression testing. —Software metrics.

Sandia National Laboratories

"Promotional" Model



- Lower formality
- Fewer Artifacts
- Lean/Agile

- Higher formality
- Sufficient Artifacts
- Bullet proof
- Maintainable

Sandia National Laboratories

Trilinos Software Lifecycle Model

- Three phases:
  - Research.
  - Production Growth.
  - Production Maintenance.
- Each phase contains its own lifecycle model.
- Promotional events:
  - Required for transition from one phase to next.
  - Signify change in behaviors and attitude.
- Phase assigned individually to each package.

Willenbring, J. M., Heroux, M. A., and Heaphy, R. T. 2007. The Trilinos Software Lifecycle Model. In *Proceedings of the 3rd international Workshop on Software Engineering For High Performance Computing Applications* (May 20 - 26, 2007). International Conference on Software Engineering. IEEE Computer Society, Washington, DC, 6.

A Few More Useful Things

Stratimikos

- New package in Trilinos 7.0.
- Single point of access to Trilinos preconditioners/solvers:
  - Uniform interface to all preconditioners.
  - Uniform interface to all solvers, linear and non-linear.
  - Selection of preconditioner/solver via parameter list.
- Simplest way to access the suite of Trilinos capabilities
- Simple driver code available on website.

Dynamic External Package Support

- New directory Trilinos/packages/external.
- Supports seamless integration of externally developed packages via package registration.
- Your package: “WorldsBestPreconditioner”
  - Understands configure/make.
  - Can have its own options: --enable-superfast-mode
- Copy source into Trilinos/packages/external.
- In Trilinos/packages/external, type: ./CustomizeExternal.csh WorldsBestPreconditioner
- Build Trilinos in the usual way using configure/make.
  - Include arguments such as --enable-superfast-mode: They will be passed down to your package.

The Trilinos Tutorial Package

- Trilinos is large (and still growing…)
  - More than 900K code lines
  - Many packages
    - a lot of functionalities…
    - … and also a lot to learn
- Trilinos tutorial is also delivered as a package: Didasko.

Sandia National Laboratories

What DIDASKO is not

- DIDASKO, as a tutorial, cannot cover the most advanced features of Trilinos:
  - Only the “stable” features are covered
- DIDASKO is **not** a substitute of each package’s documentation and examples, which remain of fundamental importance

Sandia National Laboratories

Covered Packages

- At present, we cover

Epetra	Triutils	IFPACK
Teuchos	AztecOO	ML
NOX	Amesos	EpetraExt
Anasazi	Tpetra	

Some Upcoming Efforts

- Exploiting new architecture opportunities:
  - Multicore especially.
- Redefining the scope of Trilinos beyond solvers:
  - CSE enabling technologies distribution.
- Expanding accessibility:
  - Fortran Interfaces.
  - Easier installation (esp. on Windows).

Challenges

- Support and perfective maintenance of existing software:
  - Presently done through growth, changes in system architecture.
  - Not sustainable long term.
- Adaptation of software engineering research to CSE software.
  - Imposition of standard IT approach worse than ineffective.
  - Must have counter-proposals of our own.

Sandia National Laboratories

Trilinos Availability/Information

- Trilinos and related packages are available via LGPL.
- Current release (7.0) is "click release". Unlimited availability.
- Trilinos Release 8: September 2007.
- More information:
  - http://trilinos.sandia.gov
  - http://software.sandia.gov
  - Additional documentation at my website: http://www.cs.sandia.gov/~mheroux.
- 5th Annual Trilinos User Group Meeting: November 6-8, 2007 at Sandia National Laboratories, Albuquerque, NM, USA.

Sandia National Laboratories

Summary

- Trilinos is a large, growing, modular software delivery framework:
  - Some initial investment in understanding design is typical (especially if unfamiliar with C++, configure/make, OO principles)
  - Modularity allows focusing on your scope of interest.
- Interested in new collaborations.