SAND2015-9274PE

Exceptional service in the national interest

Sandia National Laboratories

Trilinos

IDEAS productivity

IDEAS xSDK: Providing Software Interoperability

Alicia Klinvex

U.S. DEPARTMENT OF ENERGY | NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

What is IDEAS xSDK?

Sandia National Laboratories

- IDEAS is a collaborative project between several DOE labs and universities
  - Focus: improving scientific productivity
- xSDK is our extreme-scale scientific software development kit
  - Composed of high-quality reusable CSE software components and libraries
  - **Goal: interoperability of IDEAS libraries**
    - Trilinos
    - PETSc
    - Hypre
    - SuperLU

Sample use case: Amanzi

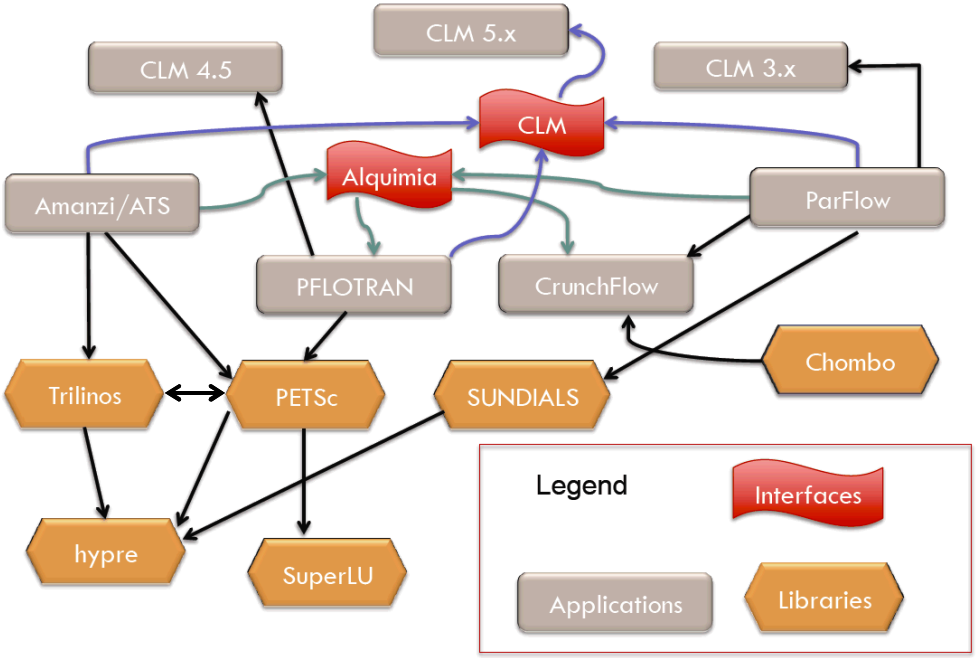
- A parallel flow and reactive transport simulator
- Used to analyze multiple DOE waste disposal sites
- Example application: modeling hydrological and biogeochemical cycling in the Colorodo River System
  - Carbon cycling is especially important because of its role in regulating atmospheric CO_2

Why does Amanzi need the xSDK?

Sandia National Laboratories

Goals of xSDK (1/3)

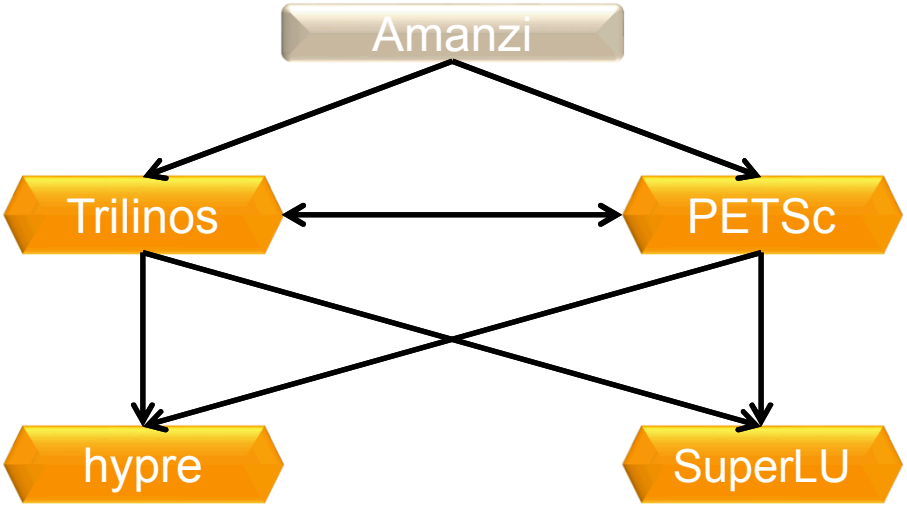
- Packages must be able to build together
  - Less trivial than it sounds due to
    - Namespace conflicts
    - Dependencies on different versions of the same TPL
    - Circular dependencies



Goals of xSDK (2/3)

Sandia National Laboratories

- User must be able to
  - Use PETSc datatypes (Mat, Vec) with Trilinos linear and nonlinear solvers
  - Use Trilinos datatypes with PETSc linear and nonlinear solvers
  - Use Hypre preconditioners with Trilinos linear solvers
  - Use SuperLu as an Amesos(2) solver



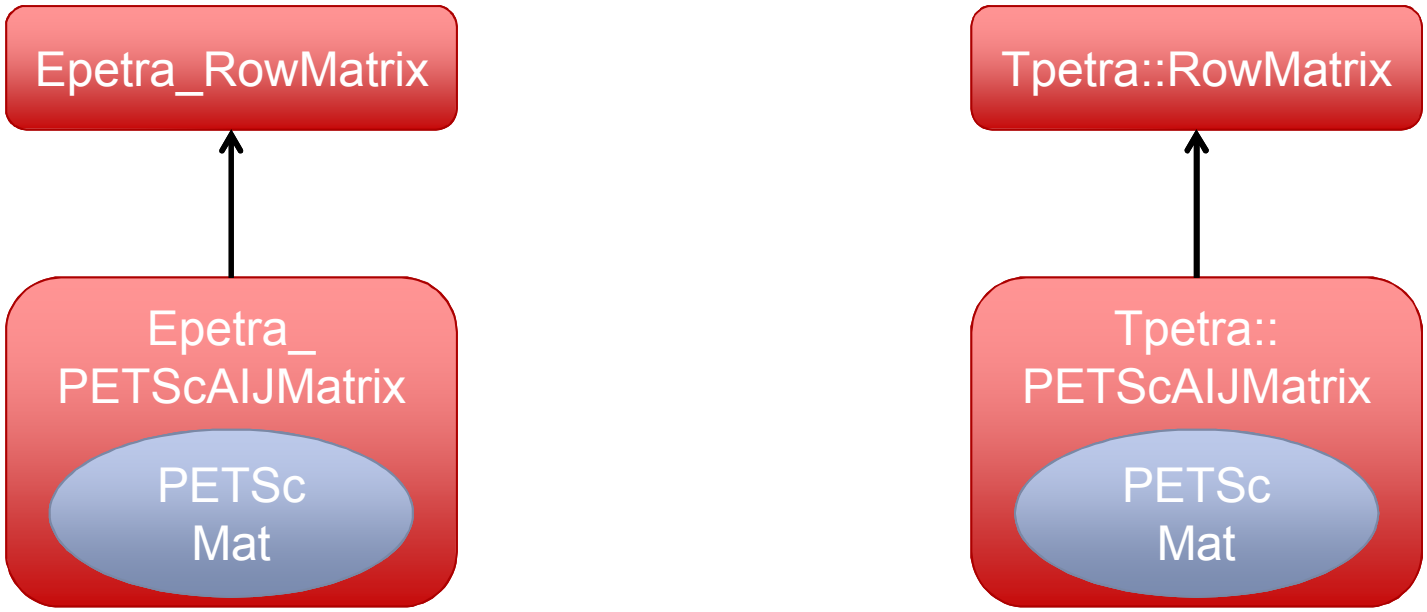
Goals of xSDK (3/3)

Sandia National Laboratories

- Interfaces must
  - Be regularly tested
  - Have documentation and examples
  - Be easy to use
  - Have long-term support

Using PETSc data structures with Trilinos

- **Have PETSc Mat and want to use Trilinos linear solvers, eigensolvers, preconditioners, etc.**
- Patched the existing Epetra-based interface
  - PETSc's Mat is wrapped in a subclass of Epetra_RowMatrix
- Created a new Tpetra-based interface



Using PETSc data structures with Trilinos

Sandia National Laboratories

- Tpetra::PETScAIJMatrix wrapper allows us to use PETSc matrices anywhere a Tpetra::RowMatrix would have been appropriate
  - Anasazi, Belos, Ifpack2, etc.
  - Raw data is **wrapped, not copied**

Using PETSc data structures with Trilinos

Sandia National Laboratories

- What this looks like to the user:

```
Mat petscA;
...
```

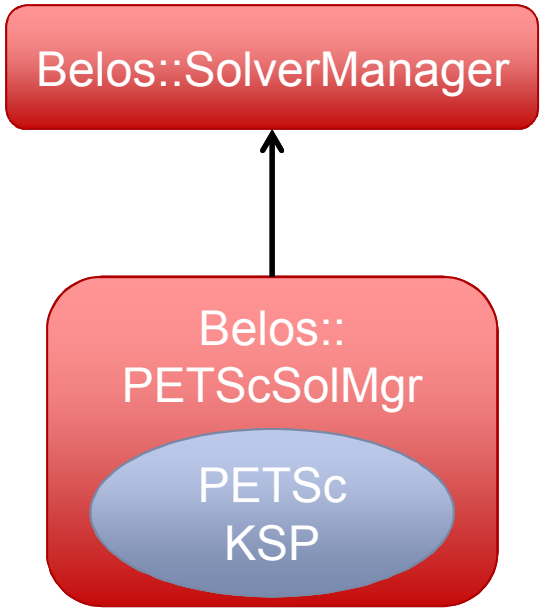
End of PETSc code

```
Tpetra::PETScAIJMatrix tpA(petscA);
Tpetra::MultiVector X;  ...
Anasazi::BasicEigenproblem problem(tpA,X);
ParameterList pl; ...
Anasazi::BlockKrylovSchurSolMgr
solver(problem, pl);
solver.solve();
```

Using Trilinos data structures with PETSc linear solvers

Sandia National Laboratories

- **Have Tpetra::Operator and want to use PETSc KSP linear solvers**
- We created a new Belos::SolverManager subclass that wraps PETSc's linear solvers (KSP)



Using Trilinos data structures with PETSc linear solvers

Sandia National Laboratories

- What calling a PETSc linear solver with Tpetra objects looks like to the user:

```
Tpetra::Operator A;
Tpetra::MultiVector X, B;
ParameterList pl;
…
Belos::LinearProblem problem(A,X,B);
Belos::PETScSolMgr solver(problem,pl);
solver.solve();
```

- **The user never sees a single line of PETSc code**

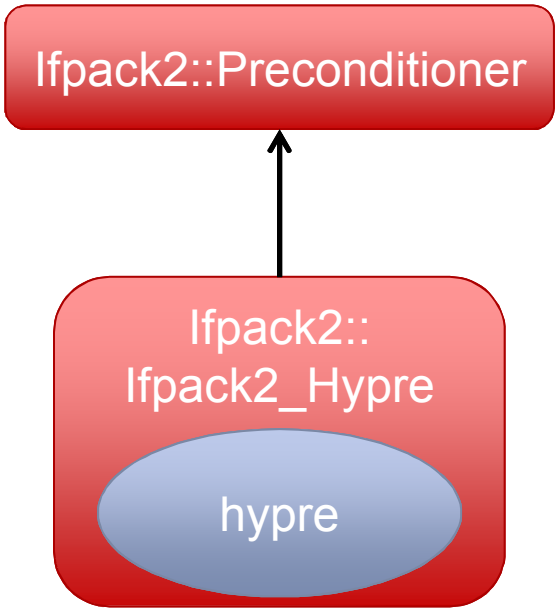
How does this interface benefit PETSc and Trilinos users in general?

Sandia National Laboratories

- Gives Trilinos users access to more linear solvers
- Gives PETSc users access to Kokkos for in-node parallelism
  - Can use Kokkos for OpenMP, CUDA, etc
  - Matrix-vector multiplication currently use Trilinos code
  - Vector operations currently use PETSc code (but we're working on it)

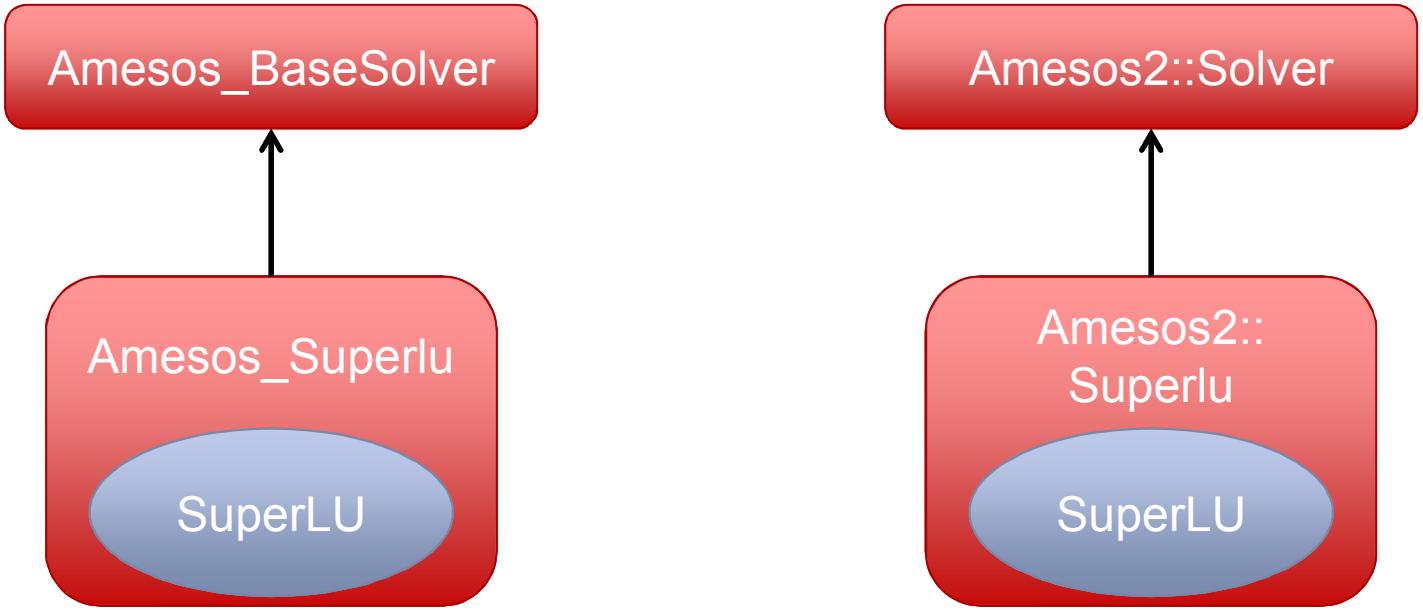
Using Trilinos data structures with hypre

- There were two existing Epetra->hypre interfaces
  - Ifpack
  - EpetraExt
  - They were recently patched by Trilinos user Denis Davydov
- New Ifpack2/Tpetra interface
  - Hypre is wrapped in a subclass of Ifpack2::Preconditioner
- **User never sees hypre**



Using Trilinos data structures with SuperLU

- We have an Amesos (Epetra) interface and an Amesos2 (Tpetra) interface
- Separate interfaces for SuperLU and SuperLU_Dist



Issues

Sandia National Laboratories

- Trilinos has multiple vector abstractions
  - Anasazi
  - Belos
  - Thyra
- Belos doesn't allow the user to specify a status test
- There is no automated testing for the PETSc and hypre interfaces
- Ifpack2 makes certain assumptions about the Tpetra::Maps describing the data distribution that still need to be addressed

Future work

Sandia National Laboratories

- Automated tests
  - Hypre interface
  - PETSc solver interface
  - SuperLU interface?
- Interoperability of PETSc and Trilinos nonlinear solves
- Documentation on Trilinos abstraction layers
- More examples demonstrating interoperability of packages

Summary

Sandia National Laboratories

- xSDK will allow application developers to use IDEAS libraries together
- This effort has positive unintended consequences for Trilinos users in general (even if you don't think you care about interoperability)
  - Better support and documentation for Trilinos abstraction layers
  - Makes additional solvers and preconditioners available