SAND2012-4160P

Exceptional service in the national interest

Sandia National Laboratories

Eigensolvers on HPC Platforms

Erik Boman, *Karen Devine*, Rich Lehoucq, Nicole Lemaster

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.

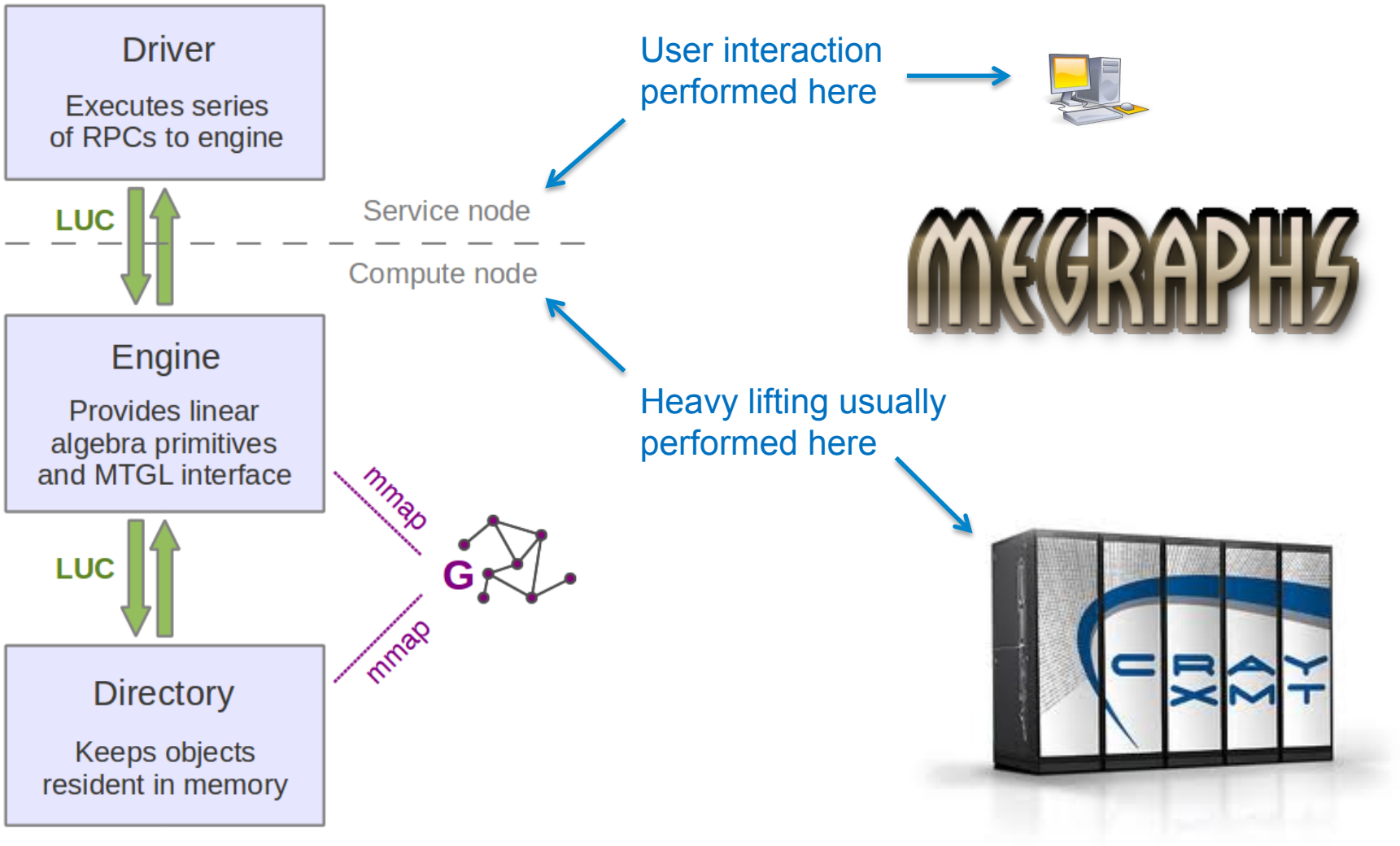
Goals

Sandia National Laboratories

- Integrate SNL's Trilinos eigensolver package Anasazi with the Megraphs graph framework for Cray-XMT
- Develop preconditioners and eigenanalysis theory to advance use of eigensolvers for graph problems
- Collaborate closely with LLNL in eigensolver comparisons and theoretical development

Modular Environment for Graph Research and Analysis with Persistent Hierarchical Storage

Sandia National Laboratories

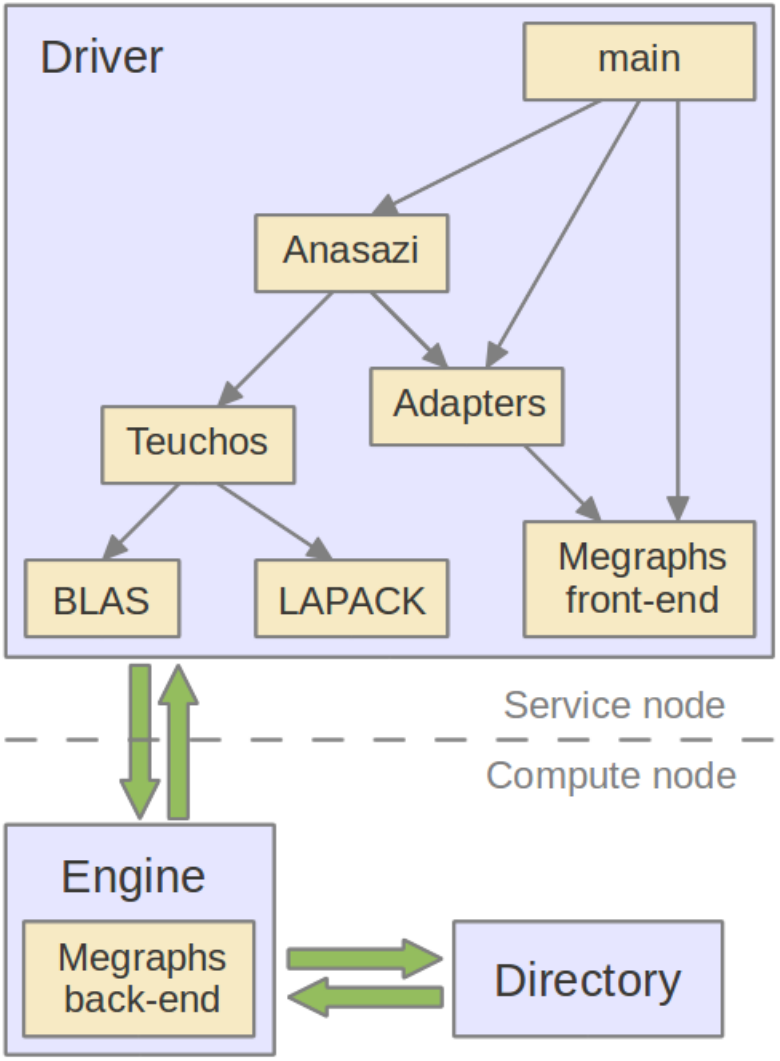


Anasazi Eigensolver in the Trilinos Solver Toolkit

Sandia National Laboratories

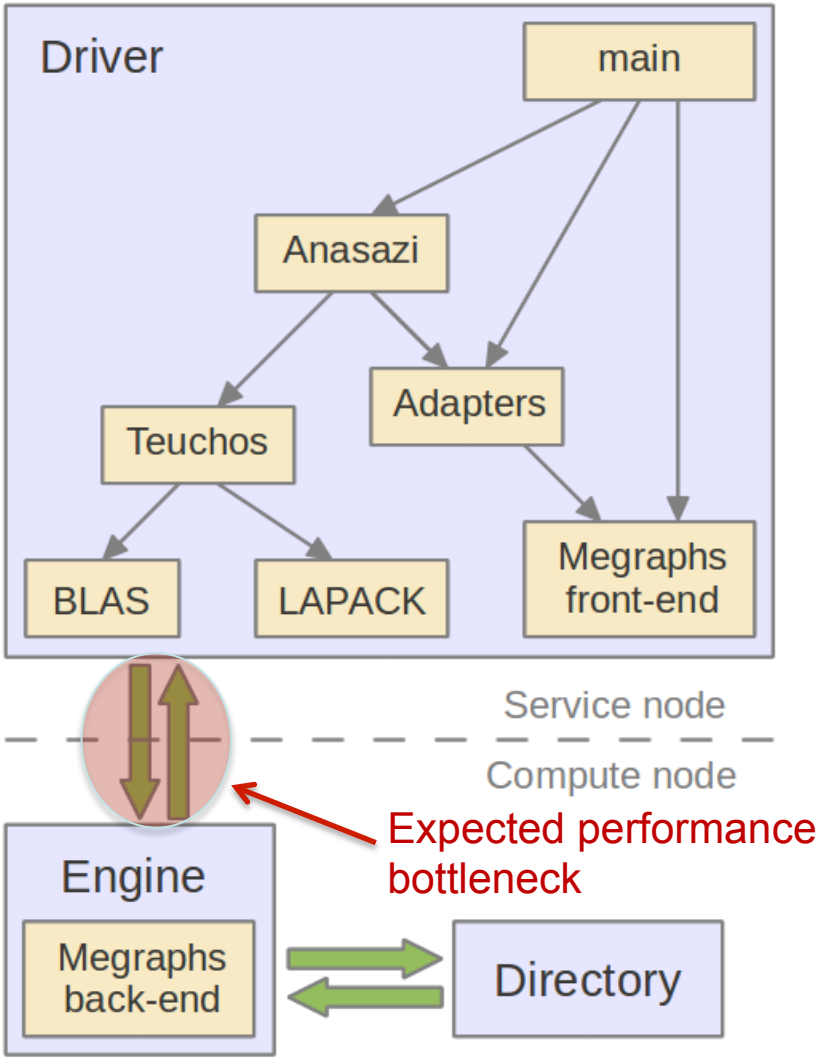
- Generic framework for large-scale iterative eigensolvers
- Block-based eigensolvers: Solves $AX = X\Lambda$ or $AX = BX\Lambda$
  - Reliably determine multiple and/or clustered eigenvalues
  - Achieve better cache locality for operator-vector products
  - Example applications: Modal/stability/bifurcation analysis, commute time
- Four eigensolvers:
  - *LOBPCG Locally Optimal Block Preconditioned Conjugate Gradient (Knyasev, 2002; Hetmaniuk & Lehoucq, 2006)*
  - Block Krylov-Schur (a block extension of Stewart, 2000)
  - Block Davidson (Arbenz, Hetmaniuk, Lehoucq, Tuminaro, 2005)
  - IRTR Implicit Riemannian Trust Region (Absil, Baker, Gallivan, 2006)
- Software written in templated C++
  - Distributed with Trilinos' Epetra and Tpetra matrix/vector class adapters
  - Templated interface allows use of alternate matrix/vector classes (e.g., Megraphs) .

Hybrid Approach to Megraphs/Anasazi Integration



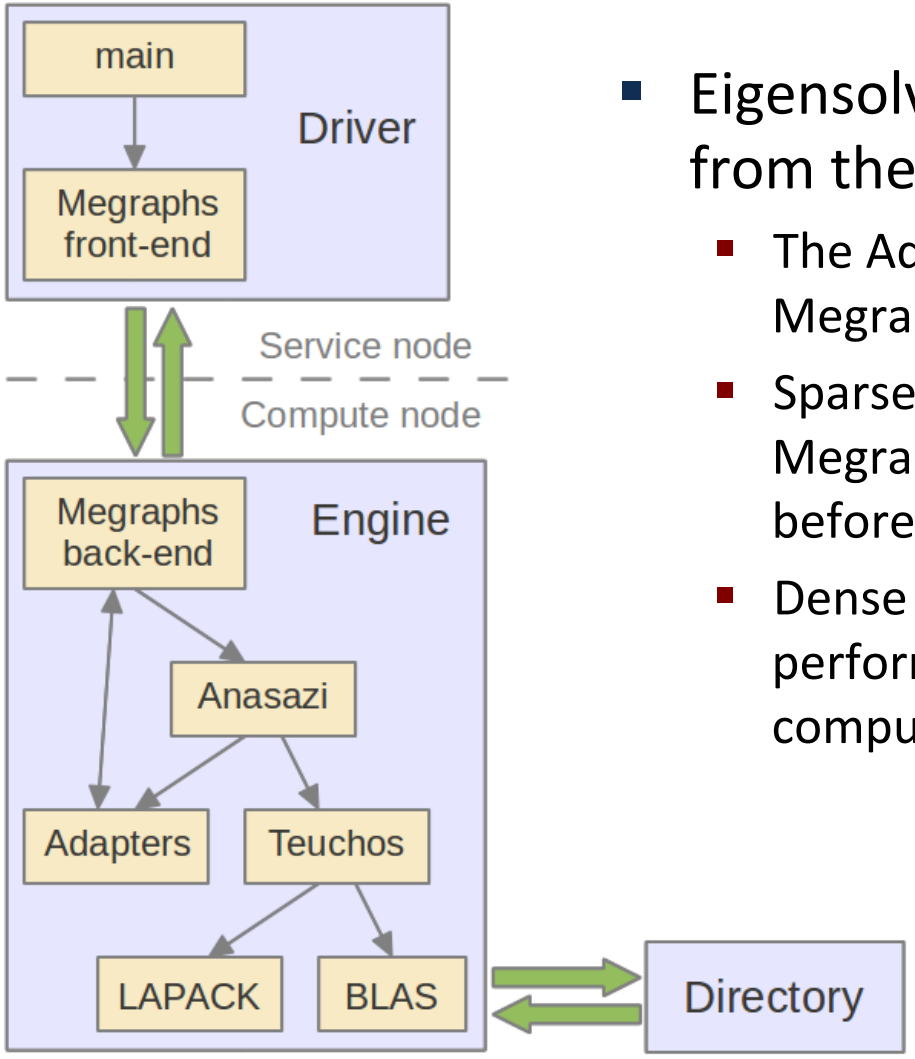
- Eigensolve is orchestrated by Anasazi from the service node:
  - The Adapters allow Anasazi to make use of Megraphs kernels via the Megraphs front-end.
  - Dense matrix operations are performed by BLAS/LAPACK on the XMT service node.
  - Sparse matrix operations are performed by Megraphs on the XMT compute nodes.

Hybrid Approach to Megraphs/Anasazi Integration



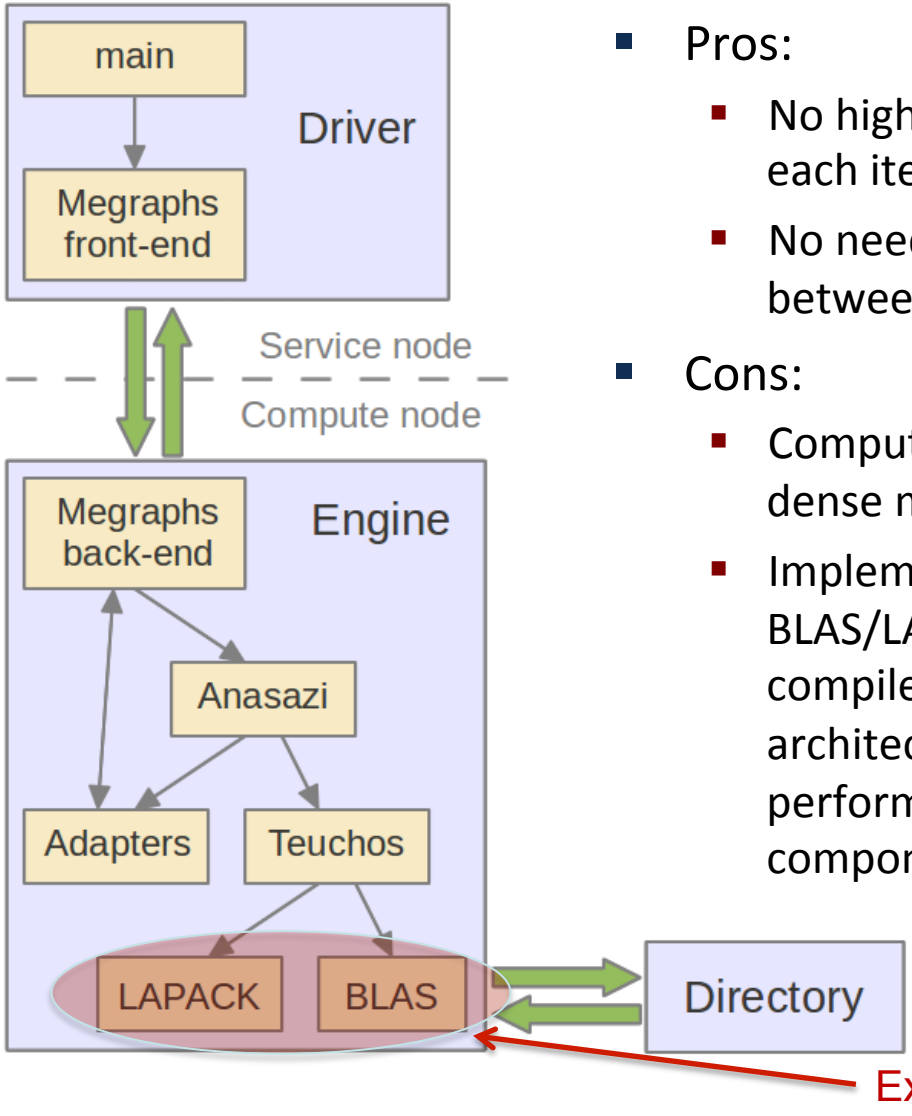
- Pros:
  - Takes advantage of the service node's floating point performance and the compute nodes' ability to hide memory latency.
  - Straight-forward implementation (the service node is just a Linux box).
- Cons
  - High-latency communication is required on each iteration of the eigensolve.
  - Dense matrices must be transferred between the service and compute nodes on each iteration.

Full Compute-Node Approach to Integration



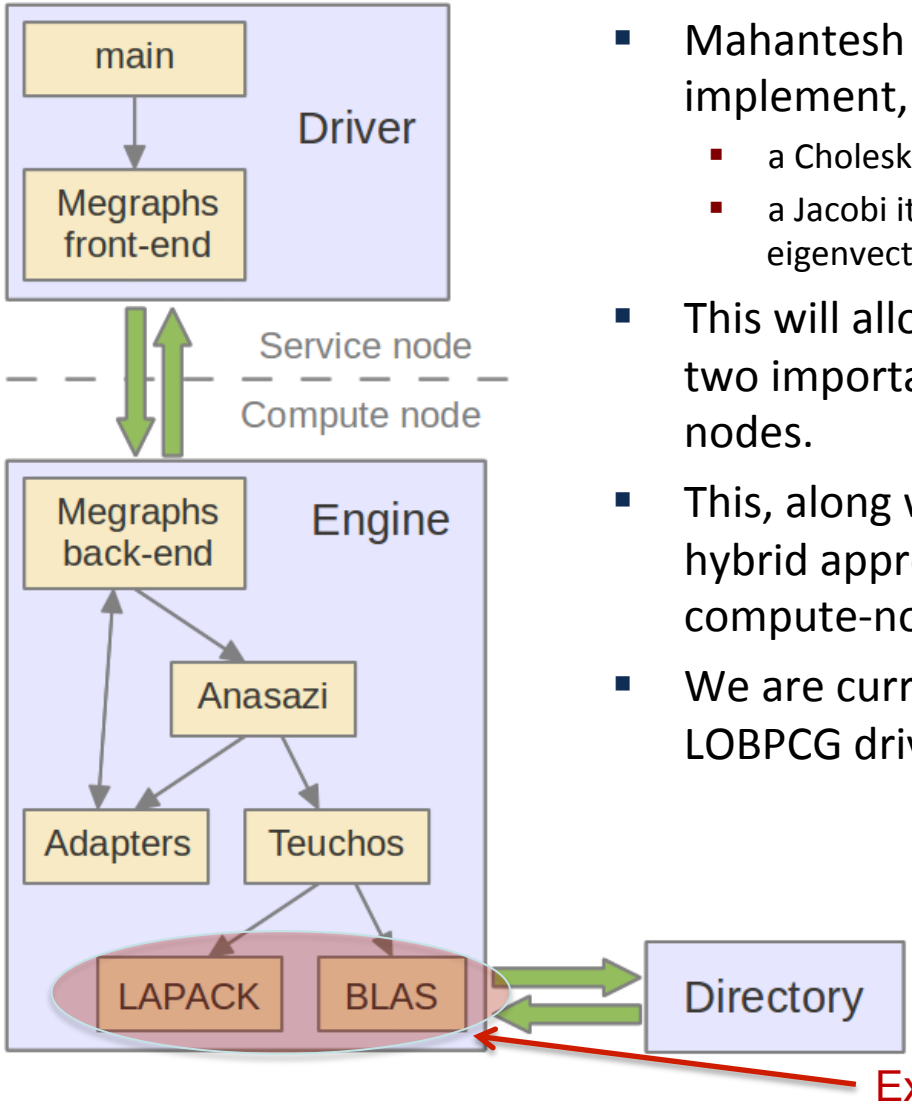
- Eigensolve is orchestrated by Anasazi from the compute nodes:
  - The Adapters allow Anasazi to make use of Megraphs kernels directly.
  - Sparse matrix operations are performed by Megraphs on the XMT compute nodes, as before.
  - Dense matrix operations are now performed by BLAS/LAPACK on the XMT compute nodes.

Full Compute-Node Approach to Integration



- Pros:
  - No high-latency communication required on each iteration of the eigensolve.
  - No need to repeatedly transfer dense matrices between the service and compute nodes.
- Cons:
  - Compute nodes were not designed for efficient dense matrix operations.
  - Implementation is very labor-intensive: no BLAS/LAPACK library provided; no Fortran compiler available; architectural peculiarities; architecture-specific parallelization and performance optimization are required for all components.

Full Compute-Node Approach to Integration



- Mahantesh Halappanavar (PNNL) has agreed to implement, for the XMT compute nodes,
  - a Cholesky factorization and forward/backward solves and
  - a Jacobi iteration for computing the eigenvalues/ eigenvectors of a symmetric matrix.
- This will allow us to ascertain the performance of two important dense matrix kernels on the compute nodes.
- This, along with performance measurements of the hybrid approach, will allow us to decide if the full compute-node approach is worth pursuing.
- We are currently debugging the adapters and an LOBPCG driver needed for the hybrid approach.

Preconditioning Eigensolvers

Sandia National Laboratories

- Interested in support-graph preconditioners for solving eigenproblems on graph data.
- Experiments to determine benefit of existing preconditioners:
  - Integrated Trilinos' Ifpack preconditioner package with Anasazi eigensolvers
  - Computed five smallest eigenvalues (tol=10^{-5}) of combinatorial Laplacians of public real-world graphs
  - Ran Anasazi's LOBPCG method on serial Linux workstation

Graph	n	m	No pre.	Jacobi	SGS	IC(0)
pgp_cc	10,680	24,316	33.0 s (2444 it.)	6.9 s (315 it.)	5.7 s (201 it.)	4.4 s (362 it.)
dblp-ubc	93,156	178,145	523 s (3769 it.)	78.6 s (344 it.)	83.9 s (230 it.)	55.3 s (385 it.)
flickr_cc	513,969	3,190,452	19,240 s (22,328 it)	1540 s (673 it.)	Failed	363 s (367 it.)

t

Results courtesy of Erik Boman, SNL

SNL/LLNL Collaboration

- Preconditioners:
  - SNL will develop serial support-graph preconditioners.
  - LLNL will adapt AMGe for graph problems.
  - Future work: Use support-graph preconditioner as inexact solve (smoother) within 2-level AMGe to obtain a scalable, parallel algorithm.
- Tutorials:
  - Rich Lehoucq & Karen Devine, Anasazi how-to document.
  - Erik Boman, Support Graph Preconditioning (March 23).
  - Panayot Vassilevski, Element Based Algebraic Multigrid (April 20).
- Comparison of Eigensolvers and Platforms:
  - Agreed on common experiments for comparing eigensolvers and platforms.
  - Sharing data sets from SNL's BTER generator and LLNL's preferential attachment generator.
  - Sharing software installations on LLNL and SNL platforms.
  - Will work together to run and analyze experiments.

Evaluation of Eigensolvers on HPC Platforms: SNL & LLNL

Sandia National Laboratories

- Phase 0: Software development
  - Drivers for various solver libraries, input formats, analyses, performance tests
  - Megraphs/Anasazi integration for Cray XMT
- Phase 1: Find smallest eigenvalues & corresponding eigenvectors of graph Laplacians
  - Number of eigenvalues sought: 10, 20, 50 , 100
  - Input graphs: Yoo's preferential attachment, Kolda's BTER, public domain (e.g., twitter, flickr, snap); 1M-100M vertices
  - Platforms: hera 800-node AMD quad-core cluster at LLNL; Cray XMT as Megraphs/ Anasazi matures.
  - Solvers: Anasazi, SLEPc, hypre's LOBPCG; block vs non-block methods
  - Metrics: robustness for various data, ability to find multiplicities, run time, scalability
- Later phases:
  - Experiments on Dark Storm and sponsor platforms with sponsor data
  - Experiments with very large graphs (more than 2B edges/vertices)
  - More detailed comparisons of MPI clusters vs XMT
  - Effects of preconditioning

Continuing work

- Evaluation of Anasazi/Megraphs integration on XMT
- Implementation/evaluation of support-graph preconditioners
  - With summer student Kevin Deweese (UCSB, advisor = John Gilbert)
- Collection/analysis of HPC eigensolver comparison results
- Further investigation of …
  - Relationships between classical, graph and nonlocal Laplacians
  - Variational characterization of relationships for commute time and statistical ranking