SAND2013-0978P

Exceptional service in the national interest

Sandia National Laboratories

Trilinos-based Software for Eigenanalysis of Graphs

Erik Boman, *Karen Devine*, Richard Lehoucq and Nicole Slattengren, SNL
David Hysom and Andy Yoo, LLNL

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.

Overview

- Trilinos-based capability for finding eigenvalues/eigenvectors of very large graphs
  - Improved parallel scalability through use of 2D matrix distribution
  - Able to analyze problems with > 2B vertices/edges through new Epetra64 capability
- Distributed-memory MPI-based results today
- Cray XMT implementation with MEGRAPHS also completed

Trilinos Computational Science Toolkit

- Heroux et al., SNL

- Trilinos Capabilities:
  - Scalable Linear & Eigen Solvers
  - Discretizations, Meshes & Load Balancing
  - Nonlinear, Transient & Optimization Solvers
  - Scalable I/O
  - Software Engineering Technologies & Integration

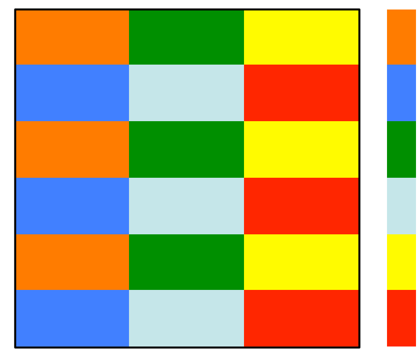
- Trilinos features:
  - Block-based data structures and algorithms
    - Block-based linear and eigen solvers use “multivector” data structures.
  - Toolkit/package-based design
    - Packages can be combined, but not all of Trilinos is needed to get work done.

- In this project, we use Trilinos’…
  - Distributed Matrix/Vector classes *Epetra*
  - Eigensolver package *Anasazi*
  - Linear solver package *Belos*
  - Preconditioning package *Ifpack*
  - Utilities package *Teuchos* (e.g., communicators, parameters, ref-counted pointers)

1D and 2D Matrix distributions

- 1D matrix distribution:
  - Entire rows (or columns) of matrix assigned to a processor
  - Same mapping used for vectors
  - Trilinos' default distribution
- 2D matrix distribution:
  - Block-based layout of matrix within processors
  - Long used in parallel direct solvers
  - Yoo et al. (SC'11) demonstrated benefit over 1D layouts for eigensolves on scale-free graphs

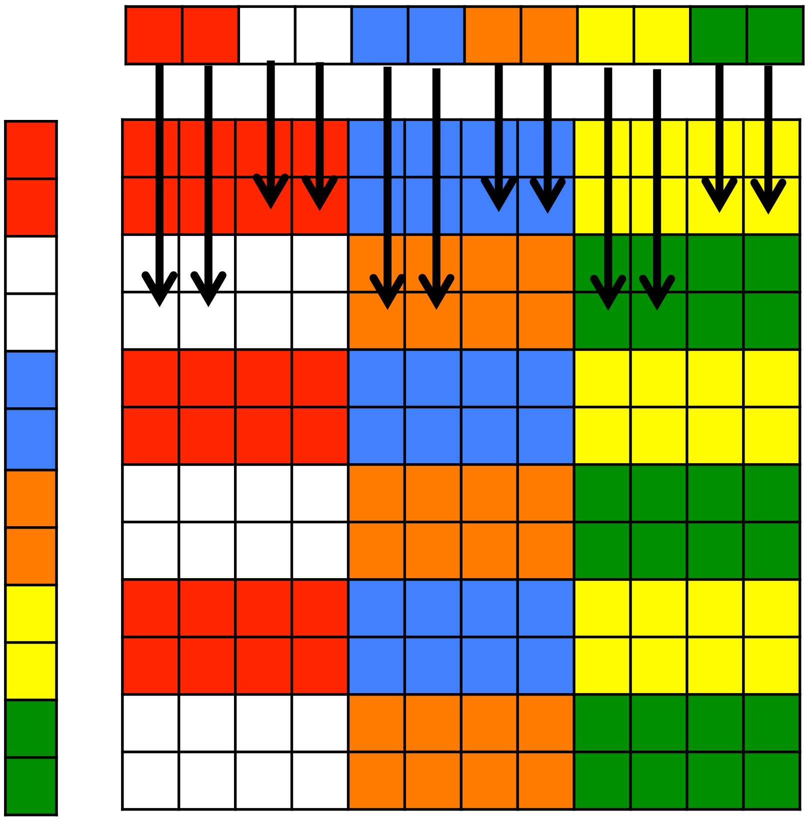
1D row-wise matrix distribution; 6 processes



2D matrix distribution; 6 processes

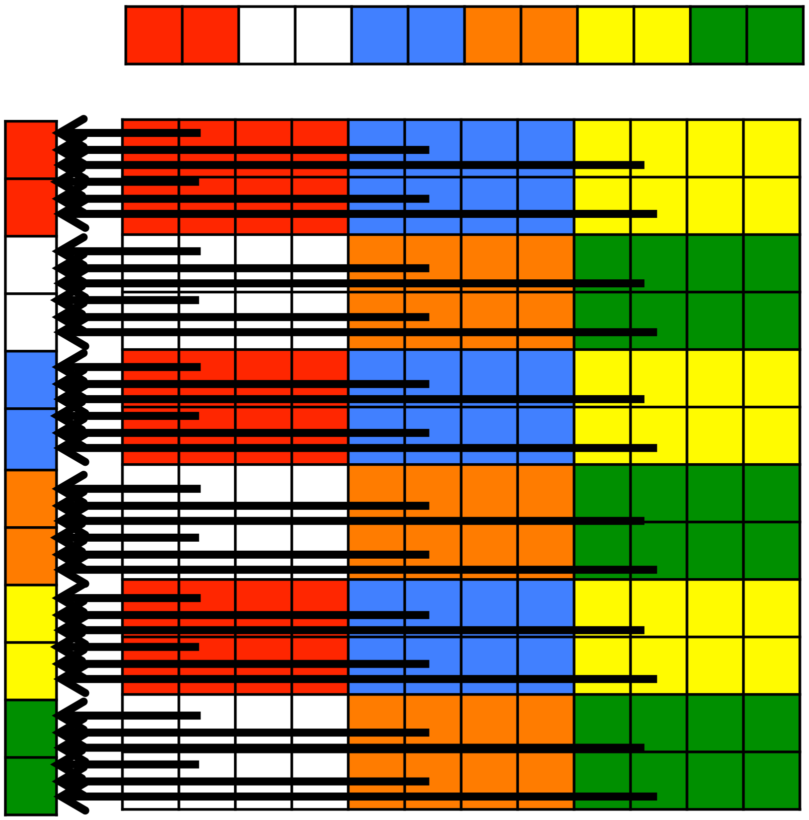
Benefit of 2D Matrix Distribution

- During matrix-vector multiplication, communication occurs only along rows or columns of processors.
  - Expand (vertical): Vector entries x_j sent to column processors to compute local product $y^p = A^p x$
  - Fold (horizontal): Local products y^p summed along row processors; $y = \Sigma y^p$
- In 1D, fold is not needed, but expand may be all-to-all.



Benefit of 2D Matrix Distribution

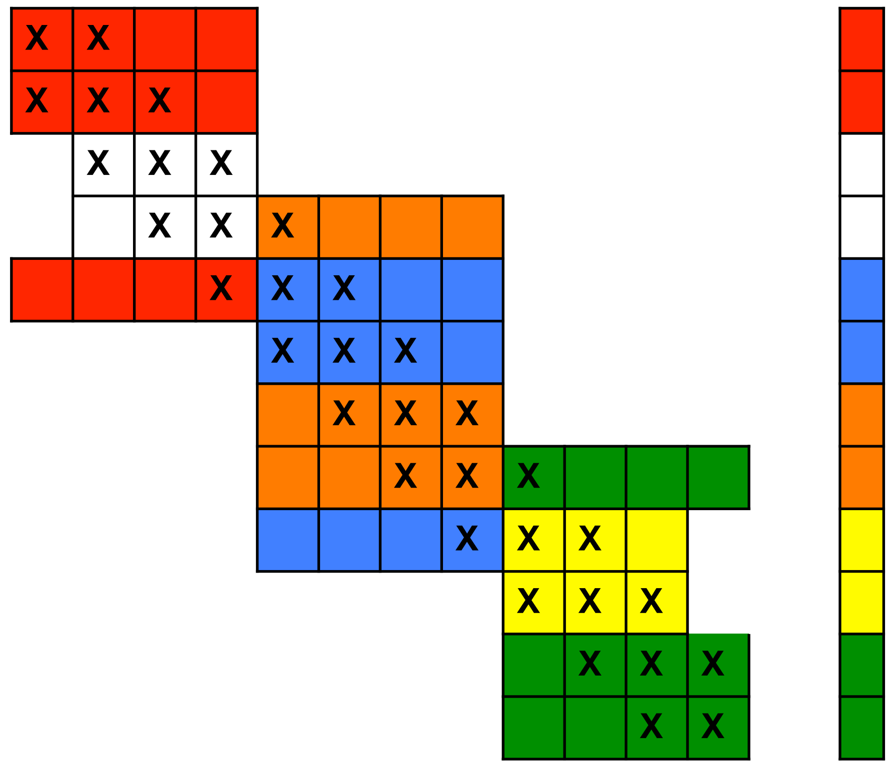
- During matrix-vector multiplication, communication occurs only along rows or columns of processors.
  - Expand (vertical): Vector entries x_j sent to column processors to compute local product $y^p = A^p x$
  - Fold (horizontal): Local products y^p summed along row processors; $y = \Sigma y^p$
- In 1D, fold is not needed, but expand may be all-to-all.



Trilinos Maps

- Maps describe the distribution of global IDs for rows/columns/vector entries to processors.
- Four maps needed in most general case:
  - Row map for matrix
  - Column map for matrix
  - Range map for vector
  - Domain map for vector
- Part of *Epetra* package

Rank 3 (Blue)
Row Map = {4, 5, 8}
Column Map = {4, 5, 6, 7}
Range/Domain Map = {4, 5}



1D vs 2D Strong Scaling Experiments

- Compare times for matrix-vector multiplication with 1D and 2D distributions
- Hera cluster at LLNL (AMD quad-core, quad-socket Opteron processors operating at 2.2/2.3 GHz)
- Matrices from the University of Florida matrix collection
- Symmetrized and largest connected component extracted

Name	Description	Number of Rows	Number of Nonzeros
Hollywood-2009	Hollywood movie actor network (Boldi, Rosa, Santini, Vigna)	1.1M	113M
Wikipedia-20070206	Links between wikipedia pages (Gleich)	3.5M	85M
Ljournal-2008	LiveJournal social network (Boldi, Rosa, Santini, Vigna)	5.6M	99M
Wb-edu	Links between *.edu webpages (Gleich)	8.9M	88M
Cit-Patents	Citation network among US patents (Hall, Jaffe, Trajtenberg)	3.8M	33M

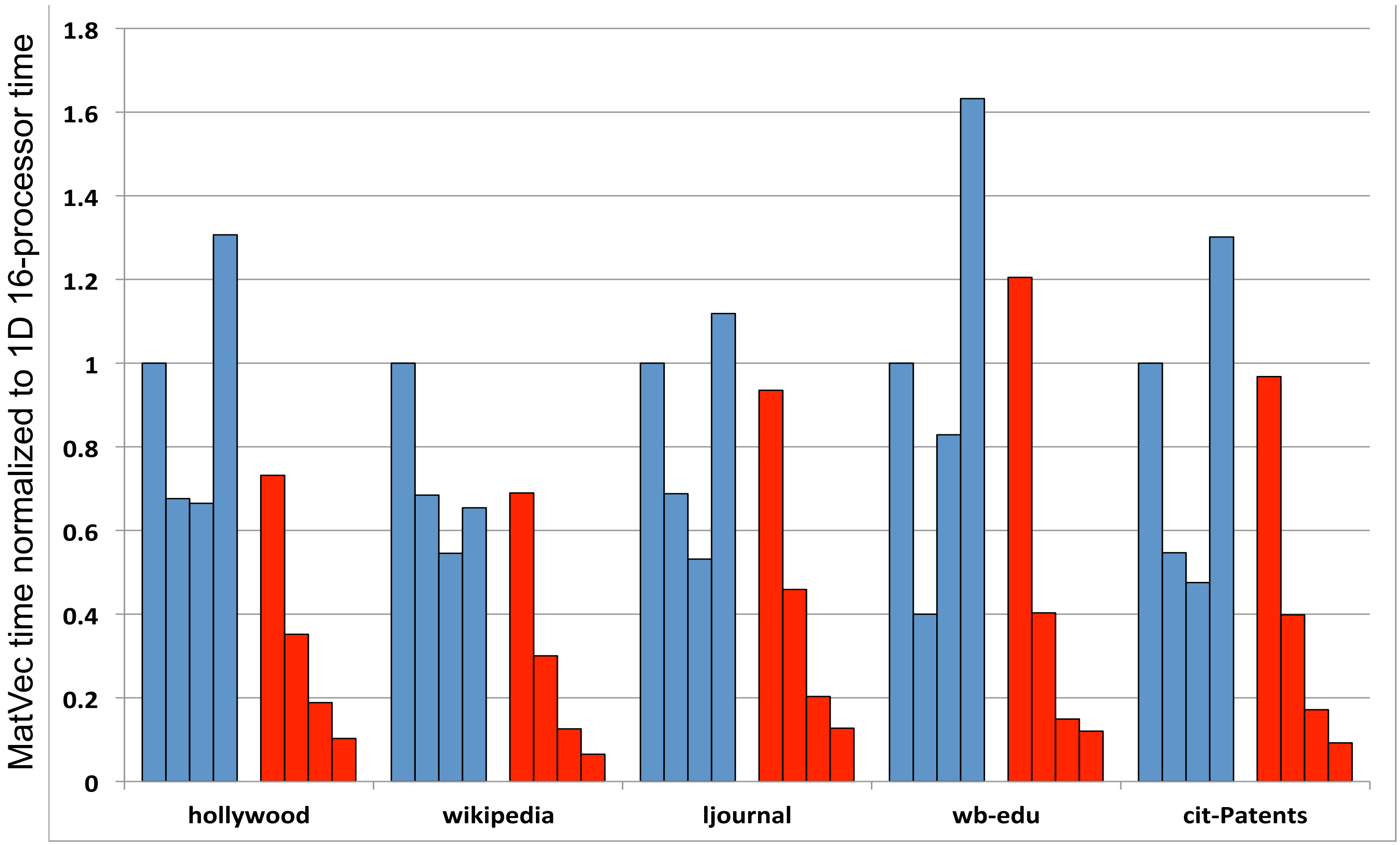
1D vs 2D Strong Scaling experiments

For each matrix:

Blue = Trilinos 1D Matrix Distribution on 16, 64, 256, 1024 processors (left to right)

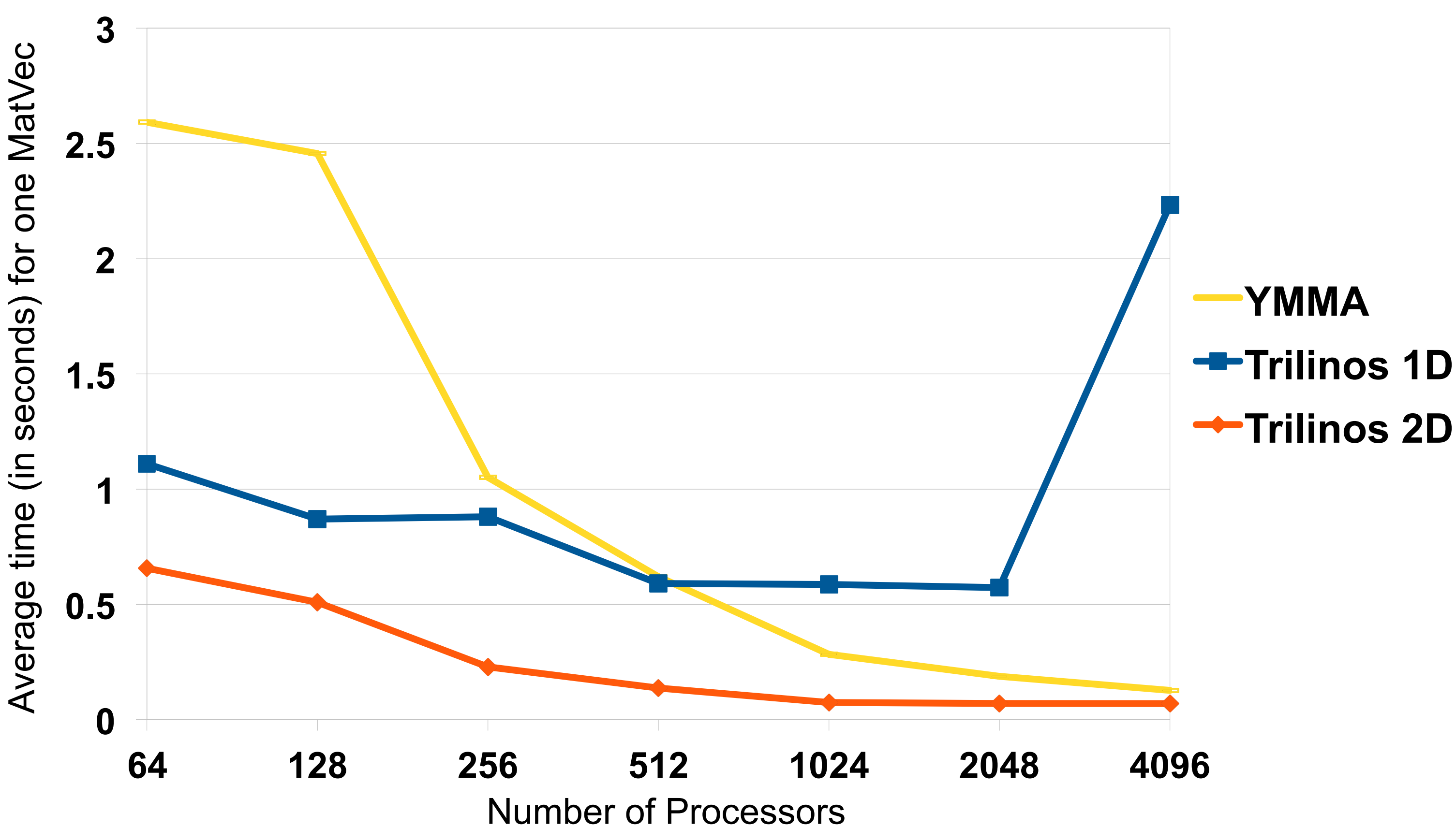
Red = Trilinos 2D Matrix Distribution on 16, 64, 256, 1024 processors (left to right)

Times are normalized to the 1D 16-processor runtime for each matrix.



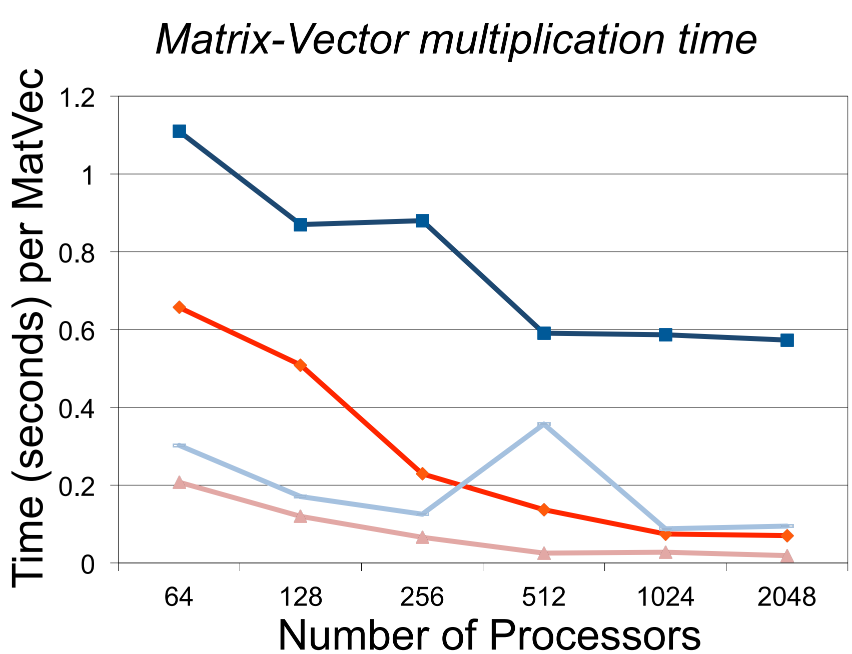
Strong Scaling: Larger Graph

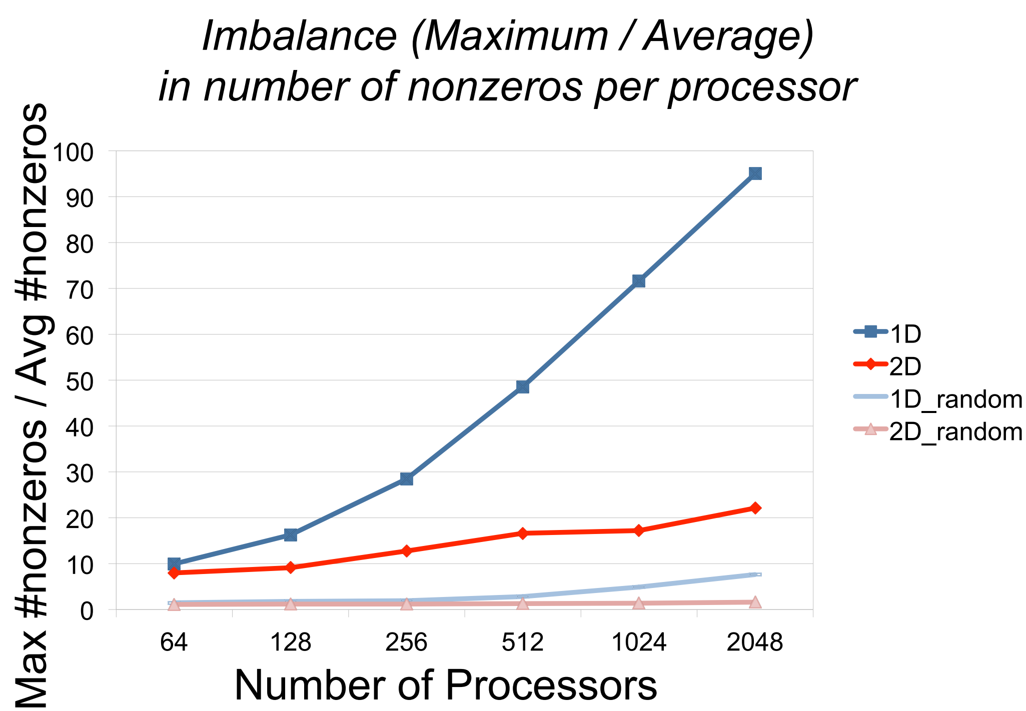
- Twitter graph: 27M vertices, 1B nonzeros (avg. degree 36)
- Trilinos 1D and 2D matrix distribution
- YMMA 2D matrix distribution integrated with PETSc (Yoo et al., SC'11)



Another performance trick

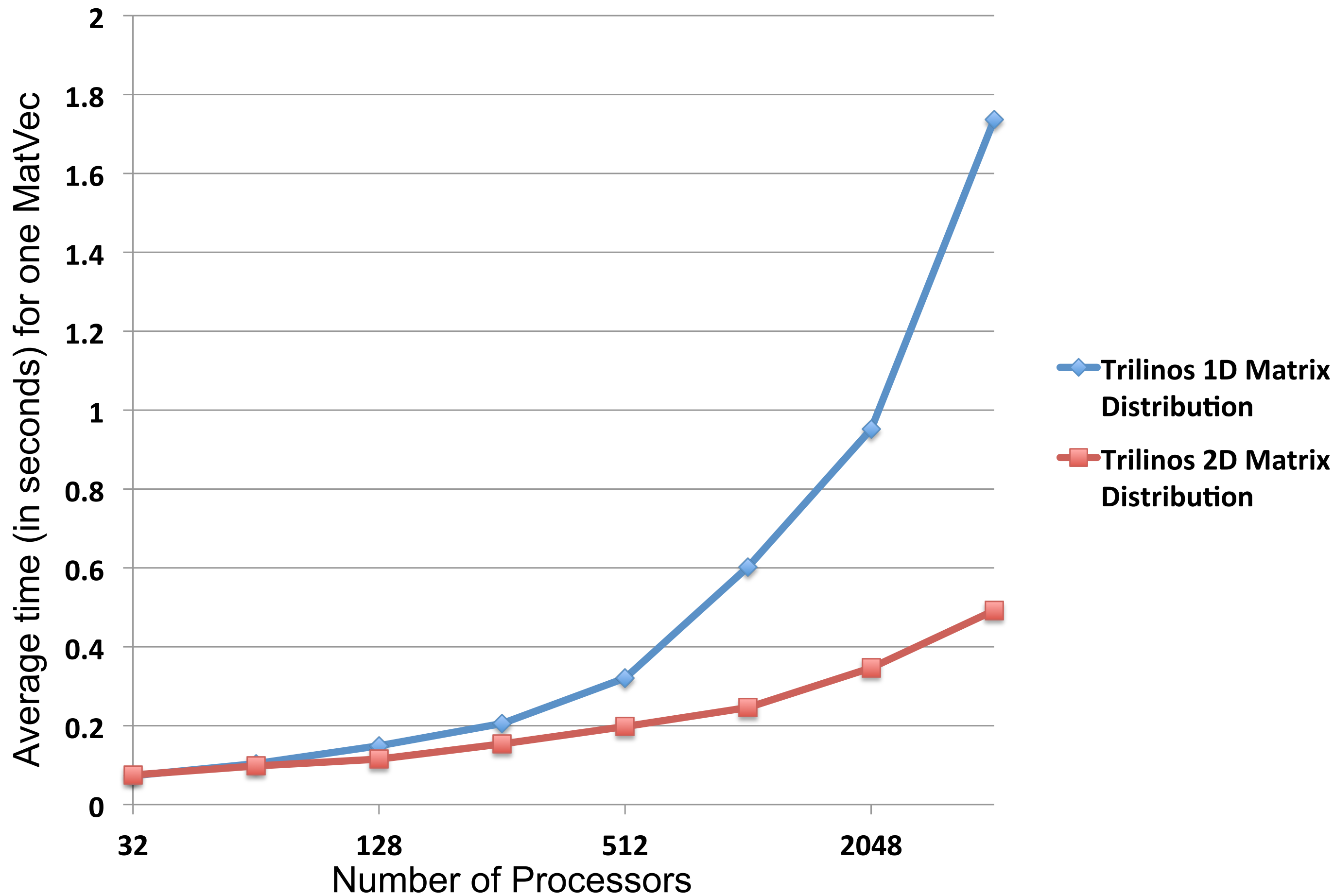
- On input, randomly permute matrix rows/columns
  - Eliminates any inherent structure in input file (e.g., high degree nodes first)
  - Gives better balance in number of nonzeros per processor for 1D and 2D.
- Twitter graph, 27M rows, 1B nonzeros





Weak scaling

- Preferential attachment graph generator (Yoo, Henderson)
  - 100K vertices per processor; Average degree = 2
  - Number of nonzeros = 4 * 100K * number of processors
  - No randomization needed; the input is perfectly balanced.



Anasazi Eigensolver in the Trilinos Solver Toolkit

- Baker, Hetmaniuk, Lehoucq, Thornquist; ACM TOMS 2009

- 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) .

Benefit in Anasazi

- Compare 2D vs 1D matrix distribution in Anasazi solve
- Use Anasazi's Block Krylov Schur method to find ten largest eigenvalues of the normalized Laplacian matrix (tol=0.0001)
- No randomization used.

Matrix	Average solution time (seconds) on 256 processors		Average solution time (seconds) on 1024 processors	
	1D	2D	1D	2D
Hollywood-2009 (1.1M rows; 113M nonzeros)	107	20	58	13
Twitter (27M rows; 1B nonzeros)	865	374	570	131

Epetra64 capability

- Initial Epetra implementation (1998) used “int” as data type for global row/column numbers and data sizes.
  - Limits use to < 2B rows/cols and < 2B nonzeros
- Epetra64 (2012) extension to Epetra allows (nearly) same application to be used with “long long” global row/column numbers and data sizes
  - Enables execution for problems with > 2B rows/cols and/or > 2B nonzeros
  - Overloaded functions take “int” or “long long” arguments to create maps with appropriate data sizes.
  - A few minor interface changes needed for 64-bit IDs.
- We are giving Epetra64 its first big work-out.

Epetra64 testing in progress

- Success on simple generated matrix with 1B rows, 3B nonzeros (1D and 2D).
- Biggest challenge: Reading in a large matrix in a time- and memory-scalable way.
- Currently tracking an Epetra performance bottleneck in constructing the matrix.
  - Suspect it is due to load imbalance, as it goes away with randomization of input.
- More results to come…

Our Anasazi Testing Platform

- Driver program for testing/evaluating various eigensolvers, parameters, inputs, parallel distributions, scalability

- Enables use of all Anasazi eigensolvers: BKS, BD, LOBPCG, IRTR
- Uses Trilinos' IFPACK preconditioners: Jacobi, SGS, IC, ILU, KLU, Support tree
- Finds smallest or largest eigenvalues and corresponding eigenvectors.
- Constructs matrices from Matrix-Market input:
  - Combinatorial Laplacian
  - Normalized Laplacian
  - Signless Laplacian
  - Adjacency Matrix
- Creates 1D and 2D matrix distributions.
- Runs in parallel (distributed memory with MPI) or serial.
  - Similar program available for shared memory (e.g., XMT, UV) using MEGRAPHS.

- Options set through command-line arguments:
  - `anasazi.exe --file=big.mtx --use2D --matrix=Laplacian --normalize --nev=25 --tol=0.001 --method=LOBPCG`