SAND2013-9833C

Exceptional service in the national interest

Sandia National Laboratories

Using 2D Matrix Distributions in Trilinos

Karen Devine, Erik Boman, Sivasankaran Rajamanickam
Sandia National Laboratories

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.

Motivation

Sandia National Laboratories

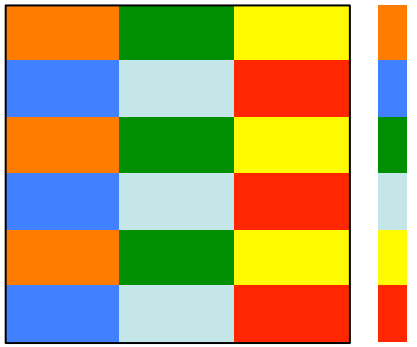
- Default row-based distribution in Trilinos is satisfactory for many scientific applications.
  - Nice, regular matrix structure.
  - Roughly equal number of nonzeros per row.
  - Data dependencies localized by finite difference stencil, finite element support, particle field radius, etc.
- Matrix structure for many graph applications is not so “nice.”
  - Scale-free graphs: web graphs, social networks, etc.
  - Power-law degree distribution in graphs leads to large variation in number of nonzeros per row (load imbalance)
  - No locality in data dependencies (high communication costs)
  - Row-based distributions can lead to All-to-All communication patterns.
    - Max. number of messages per process is 80-100% of number of processors.

1D and 2D Matrix distributions

Sandia National Laboratories

- 1D matrix distribution:
  - Entire rows (or columns) of matrix assigned to a processor
  - Same mapping used for vectors
  - Epetra's default distribution
- 2D Cartesian matrix distribution:
  - Block-based layout of matrix within processors
  - In effect, partitioning the nonzeros of the matrix (i.e., the edges of the graph).
  - Long used in parallel direct solvers
  - Also works for sparse matrices (Hendrickson et al. '95, Bisseling '04)
  - Yoo et al. (SC'11) demonstrated benefit over 1D layouts for eigensolves on scale-free graphs

1D row-wise matrix distribution; 6 processes

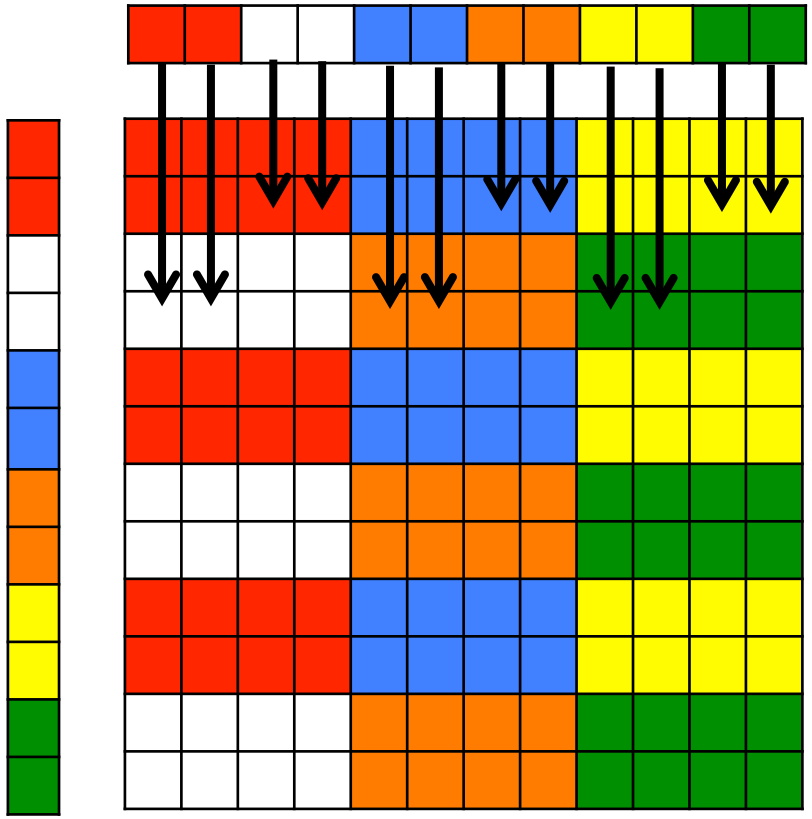


2D matrix distribution; 6 processes

Benefit of 2D Matrix Distribution

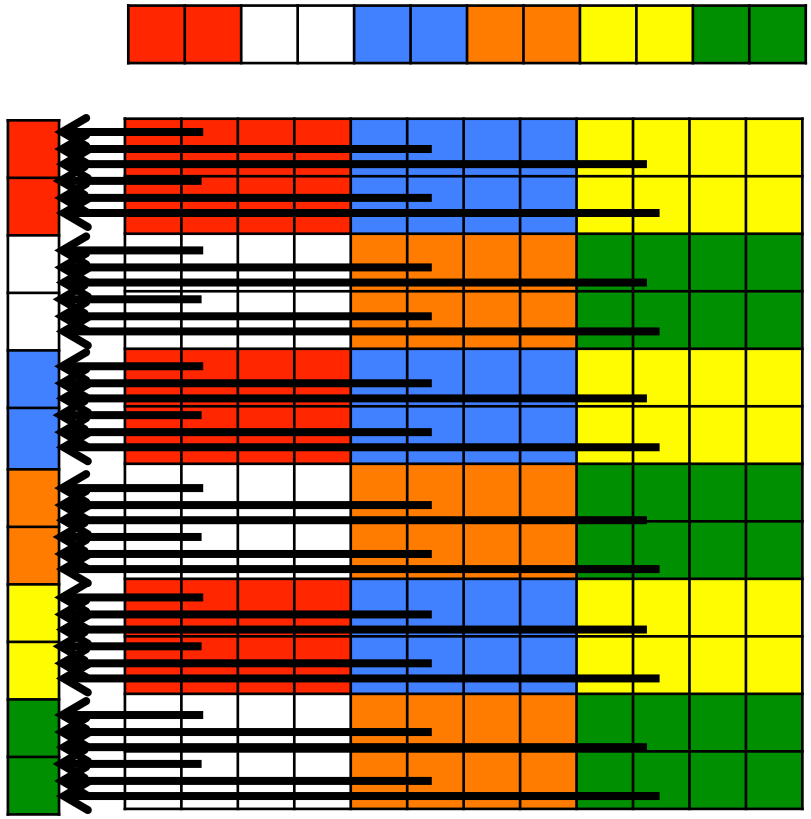
Sandia National Laboratories

- 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; $y = \sum y^p$
- 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; $y = \sum y^p$
- In 1D, fold is not needed, but expand may be all-to-all.

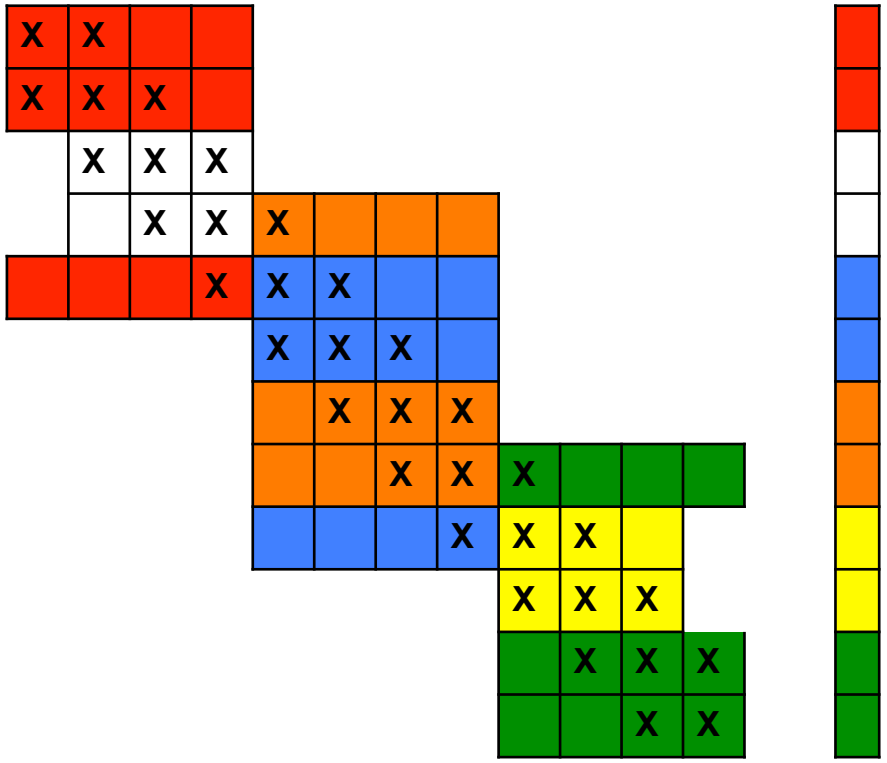


Epetra Maps Support 2D Distributions

Sandia National Laboratories

- In Petra Object Model, maps describe the distribution of global IDs to processors.
  - Row & column maps for matrix
  - Range & domain maps for vector
- Expand done by matrix's Importer; Fold done by matrix's Exporter.
- Important to keep the row map sparse during matrix construction
  - Memory for dense row maps doesn't scale; 2D layout has N/sqrt(P) rows per processor.
  - For locally empty rows, dense row maps still communicate zero-valued entries in fold.
- Let Epetra compute the Column map in FillComplete.
  - Similar reasons as above.

Rank 2 (Blue)
Row Map = {4, 5, 8}
Column Map = {4, 5, 6, 7}
Range/Domain Map = {4, 5}



Example: Common Parallel Distributions for Scale-Free Graphs

Sandia National Laboratories

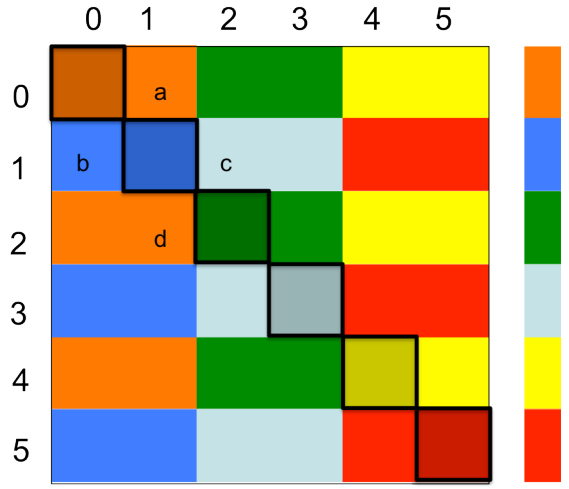
- Randomization can restore load balance for 1D and 2D distributions.
- 2D distributions have fewer messages per processor, making them faster than 1D (despite higher communication volume).

liveJournal matrix (4M rows; 73M nonzeros) on 1024 processes				
Method	Imbalance in nonzeros (Max/Avg per proc)	Max # Messages per SpMV	Comm. Vol. per SpMV (doubles)	100 SpMV time (secs)
1D-Block	12.8	1023	34.5M	2.14
1D-Random	1.3	1023	55.3M	1.52
2D-Block	11.4	62	43.4M	0.95
2D-Random	1.0	62	64.2M	0.43

New idea: Graph Partitioning + 2D

Sandia National Laboratories

- 2D Cartesian block-based layouts ignore structure of the graph.
- (Hyper)Graph partitioning (e.g., Zoltan, ParMETIS, Scotch) balances work (nonzeros per process) while attempting to minimize total communication volume.
  - Doesn't significantly reduce the number of messages.
- Idea: Apply (hyper)graph partitioning and 2D distribution together
  - Compute row-based partition of matrix using ParMETIS or Zoltan
  - Apply 2D distribution to a logical permutation based on the (hyper)graph partition
- Advantages:
  - Balance the number of nonzeros per process
  - Exploit structure in the graph to reduce communication volume
  - Reduce the number of messages via 2D distribution



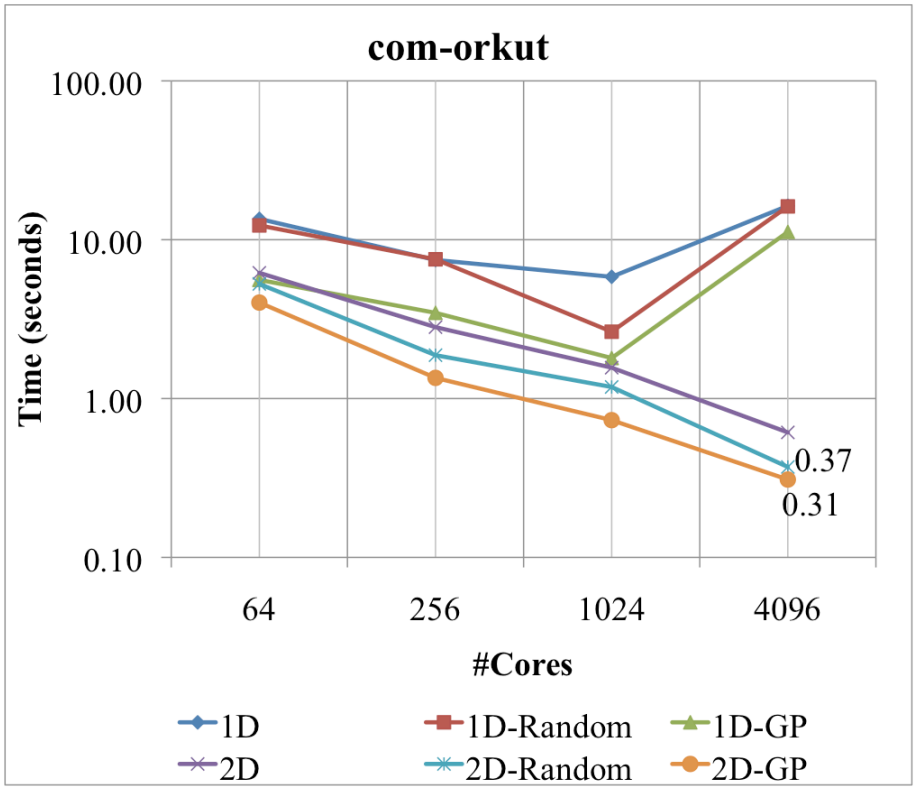
"Scalable Matrix Computations on Large Scale-Free Graphs using 2D Graph Partitioning."
E. Boman, K. Devine, S. Rajamanickam. To appear, Proceedings of SC13

2D-GP: Graph partitioning with 2D Distribution

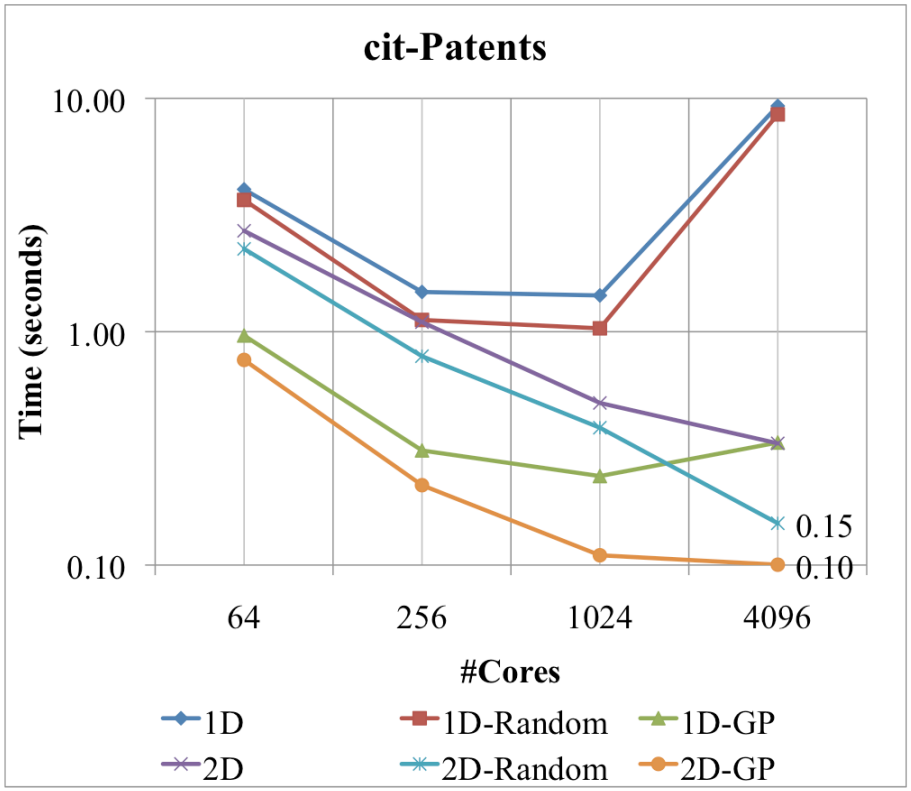
Sandia National Laboratories

liveJournal matrix (4M rows; 73M nonzeros) on 1024 processes				
Method	Imbalance in nonzeros (Max/Avg per proc)	Max # Messages per SpMV	Comm. Vol. per SpMV (doubles)	100 SpMV time (secs)
1D-Block	12.8	1023	34.5M	2.14
1D-Random	1.3	1023	55.3M	1.52
1D-GP	1.2	1011	18.9M	0.53
2D-Block	11.4	62	43.4M	0.95
2D-Random	1.0	62	64.2	0.43
2D-GP	1.4	62	22.4M	0.22

Strong scaling



Orkut social network
3.1M rows; 237M nonzeros
Max nonzeros/row = 33K

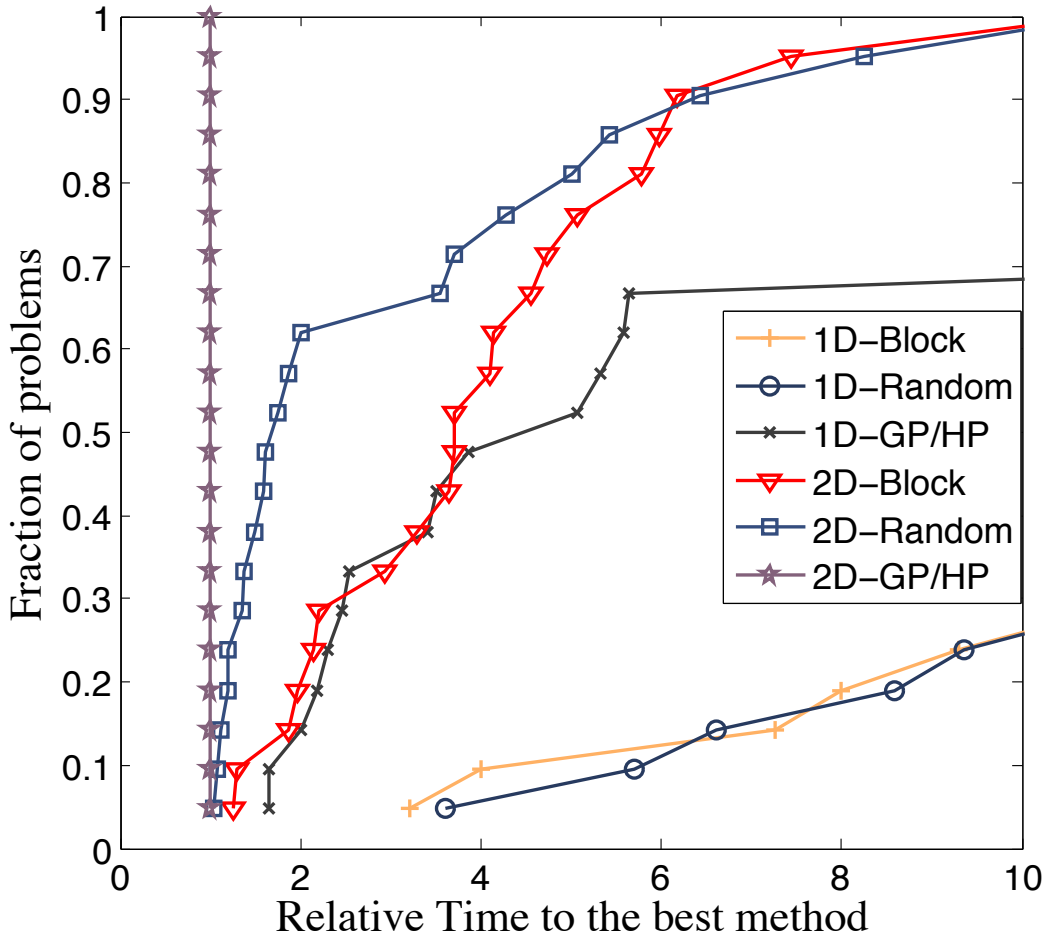


Patent citations network
3.8M rows; 37M nonzeros
Max nonzeros/row = 1K

Performance comparison

- 10 scale-free graphs from SNAP and UF matrix collections:
  1.1M - 67.5M rows;
  36M-1.6B nonzeros
- 1024-4096 processes

- 2D layout has clear benefits.
  - Reduces max number of messages per process
- 2D-GP/HP gives best results.
  - Low number of messages, low communication volume, low imbalance



Conclusions

Sandia National Laboratories

- 2D distribution has definite benefit, especially at high process counts.
  - Reduces max number of messages per process
- Randomization can be effective to restore load balance.
  - But can increase communication volume
- (Hyper)graph partitioning can maintain load balance while keeping communication volume low.
  - More effective for scale-free graphs than thought
- Combining 2D distribution with (hyper)graph partitioning gives best results.
  - Low number of messages, low communication volume, low imbalance

Questions for the audience

Sandia National Laboratories

- Will 2D distributions work as seamlessly in TPetra?
- How does one efficiently redistribute matrices into 2D distributions?
- Which preconditioners work for 2D distributions?
- How does one precondition matrices from scale-free graphs?
  - Boman and Deweese have some ideas.
- Does Trilinos' development path extend beyond scientific applications and into data-centric applications?
  - How do we get it done? Staffing? Funding?