SAND2014-17387PE

Sparse Matrix Partitioning for Parallel Eigenanalysis of Large Static and Dynamic Graphs

Michael Wolf, Sandia National Laboratories

Ben Miller, MIT Lincoln Laboratory

IEEE HPEC 2014

September 10, 2014

Sandia National Laboratories

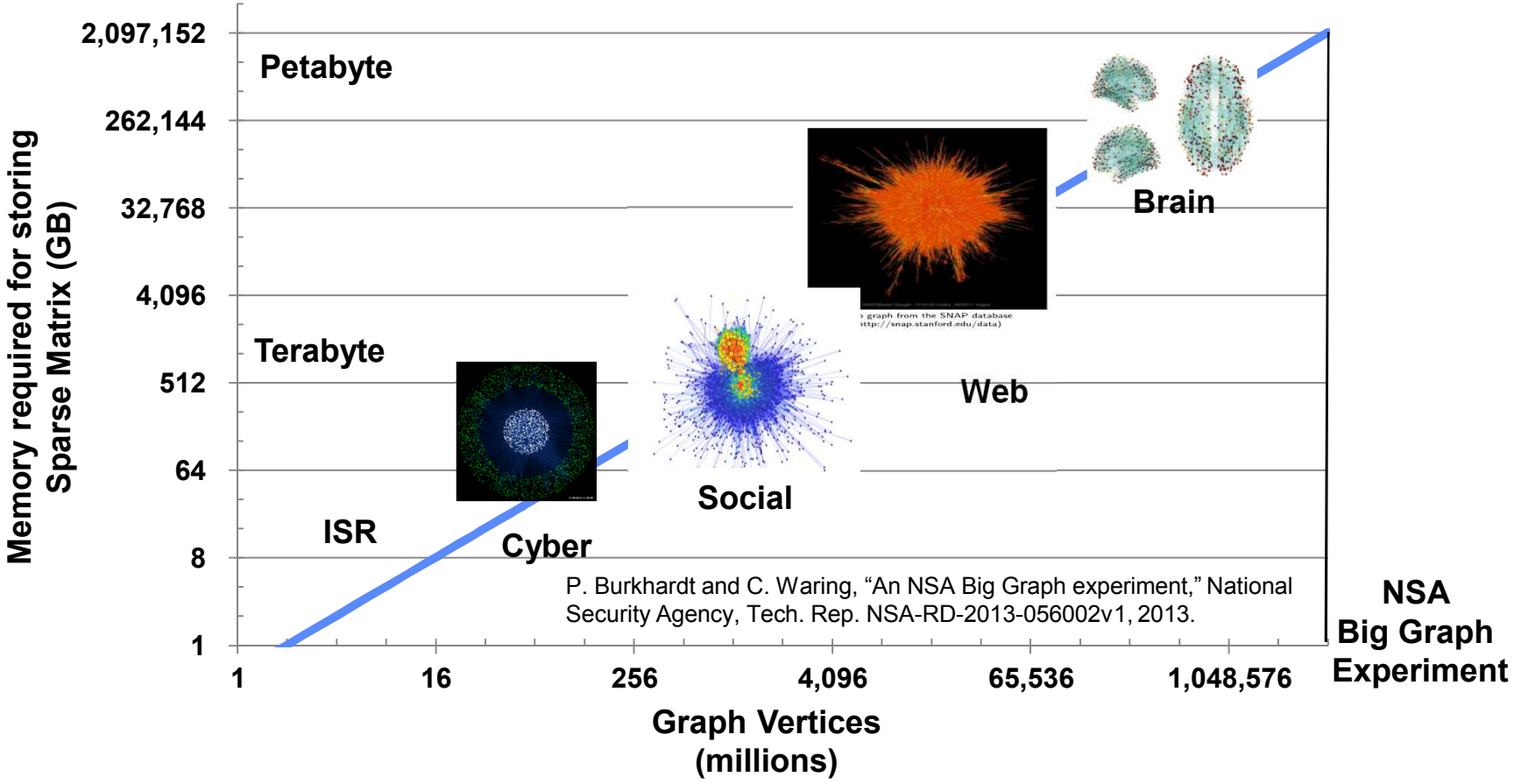
Exceptional service in the national interest

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. 2014-XXXXP

The Lincoln Laboratory portion of this work is sponsored by the Intelligence Advanced Research Projects Activity (IARPA) under Air Force Contract FA8721-05-C-0002. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA or the U.S. Government.

Big Data Challenge

Sandia National Laboratories

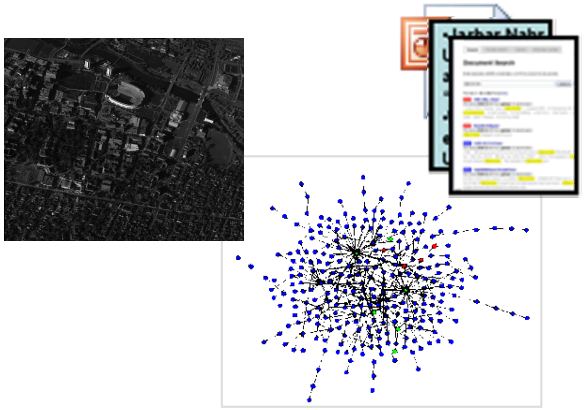


How do we address the data storage and compute challenges posed by the problem scales of interest to the DoD/IC community?

Example Applications of Graph Analytics

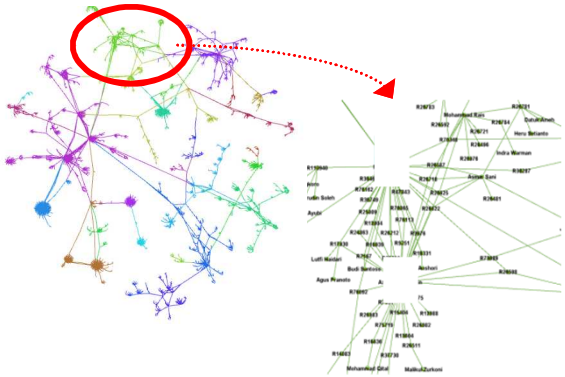
Sandia National Laboratories

ISR



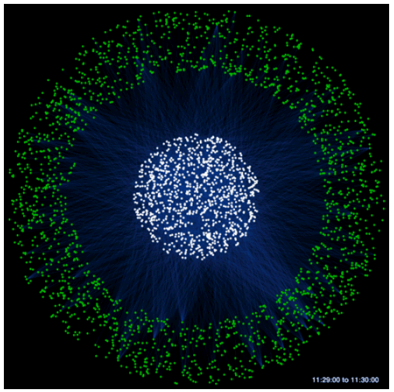
- Graphs represent entities and relationships detected through multiple sources
- 1,000s – 1,000,000s tracks and locations
- GOAL: Identify anomalous patterns of life

Social



- Graphs represent relationships between individuals or documents
- 10,000s – 10,000,000s individual and interactions
- GOAL: Identify hidden social networks

Cyber

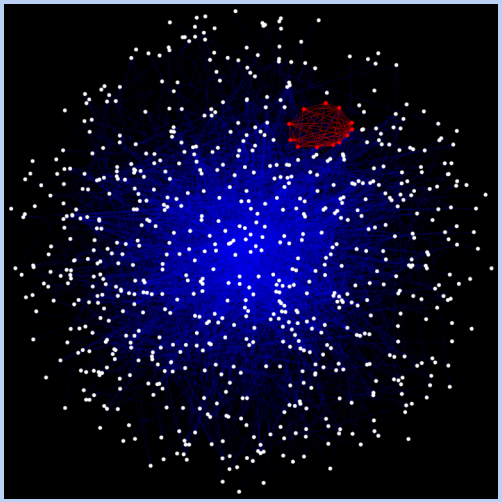


- Graphs represent communication patterns of computers on a network
- 1,000,000s – 1,000,000,000s network events
- GOAL: Detect cyber attacks or malicious software

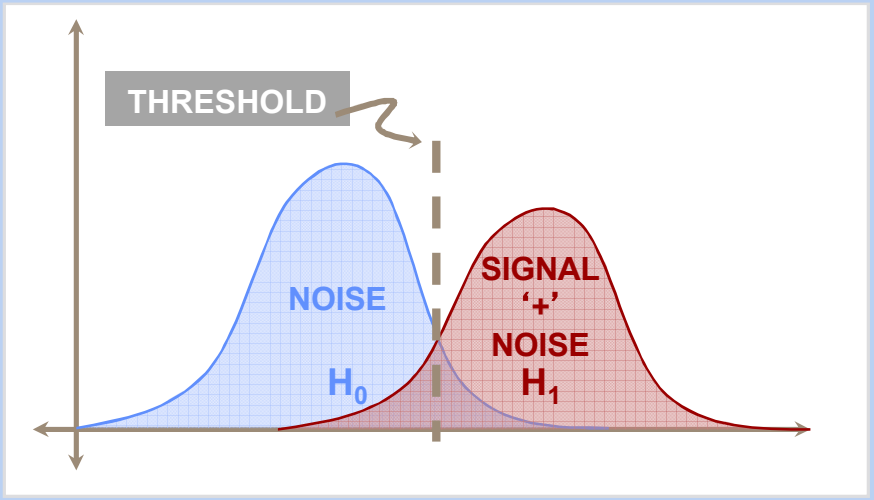
Cross-Mission Challenge:
Detection of subtle patterns in massive multi-source noisy datasets

Statistical Detection Framework for Graphs

Sandia National Laboratories



Graph Theory



Detection Theory

Develop fundamental graph signal processing concepts

Demonstrate in simulation

Apply to real data

SPG Processing Chain

Sandia National Laboratories

DIMENSIONALITY REDUCTION

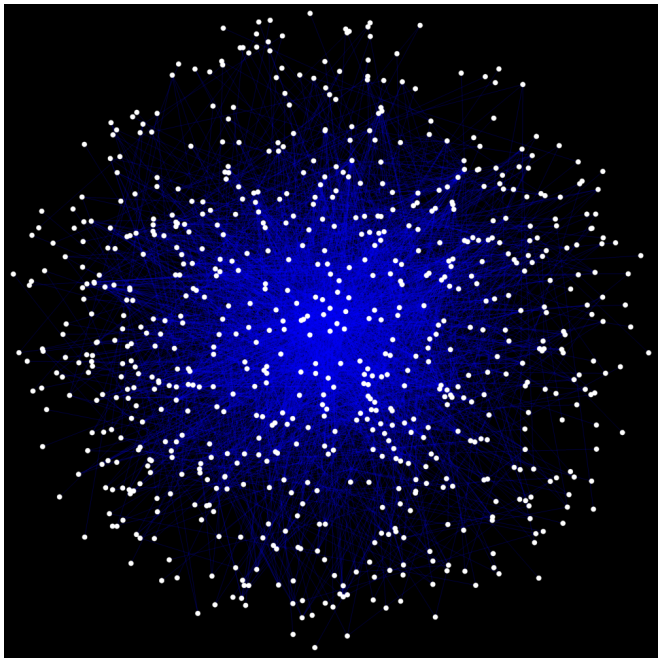
TEMPORAL INTEGRATION → GRAPH MODEL CONSTRUCTION → RESIDUAL DECOMPOSITION → COMPONENT SELECTION → ANOMALY DETECTION → IDENTIFICATION

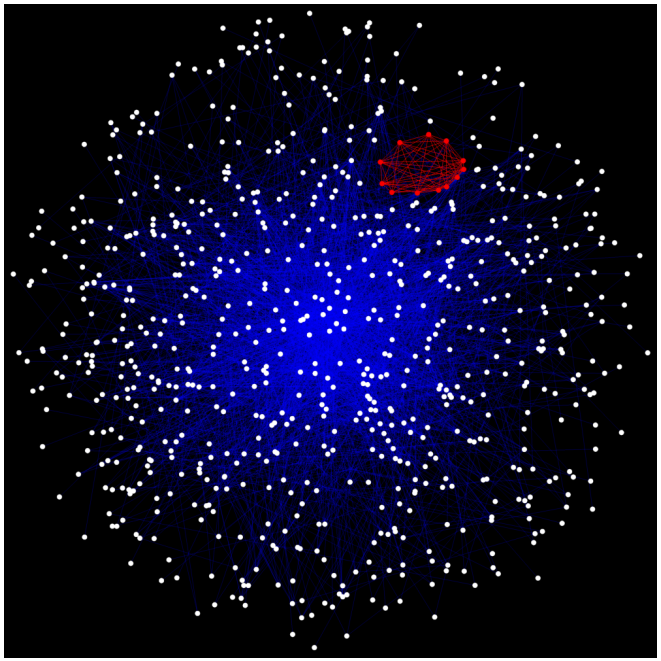
Input

- Graph
- No cue

Output

- Statistically anomalous subgraph(s)





Outline

- Anomaly Detection in Very Large Graphs
- Eigenanalysis and Performance Challenges
- Improving Sparse Matrix-Vector Multiplication (SpMV) Performance through Data Partitioning
- Partitioning: Dynamic Graphs and Sampling
- Summary

Computational Focus: Dimensionality Reduction

Sandia National Laboratories

Eigensystem

$$B = (A - E[A])$$

Solve:

$$Bx_i = \lambda_i x_i, i = 1, \square \ , m$$

Example: Modularity Matrix

$$E[A_s] = k\,k^T / (2|e|)$$

$|e|$ – Number of edges in graph $G(A)$

k – degree vector

k_i = degree(v_i), $v_i \in G(A)$

- Dimensionality reduction dominates computation
- Eigen decomposition is key computational kernel
- Parallel implementation required for very large graph problems
  - Fit into memory, minimize runtime

Need fast parallel eigensolvers

Detection Methods, Effectiveness, and Cost

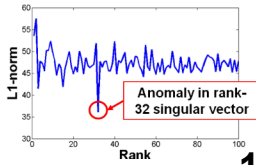
Sandia National Laboratories

Notional Comparison of Power and Effectiveness

Detection Power

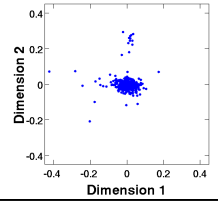
Computation Cost

Eigenvectors L_1 Norms



100s EVs

χ^2 in 2 Principal Components



2 EV

Spectral Norm

σ_1, λ_1

1 EV

- **More powerful methods require more computation**
- **For detection of subtle anomalies, need to calculate 100s of eigenvectors fast**

$O((|E|r+|V|r^2+r^3)h)^*$ **to compute r eigenvectors**

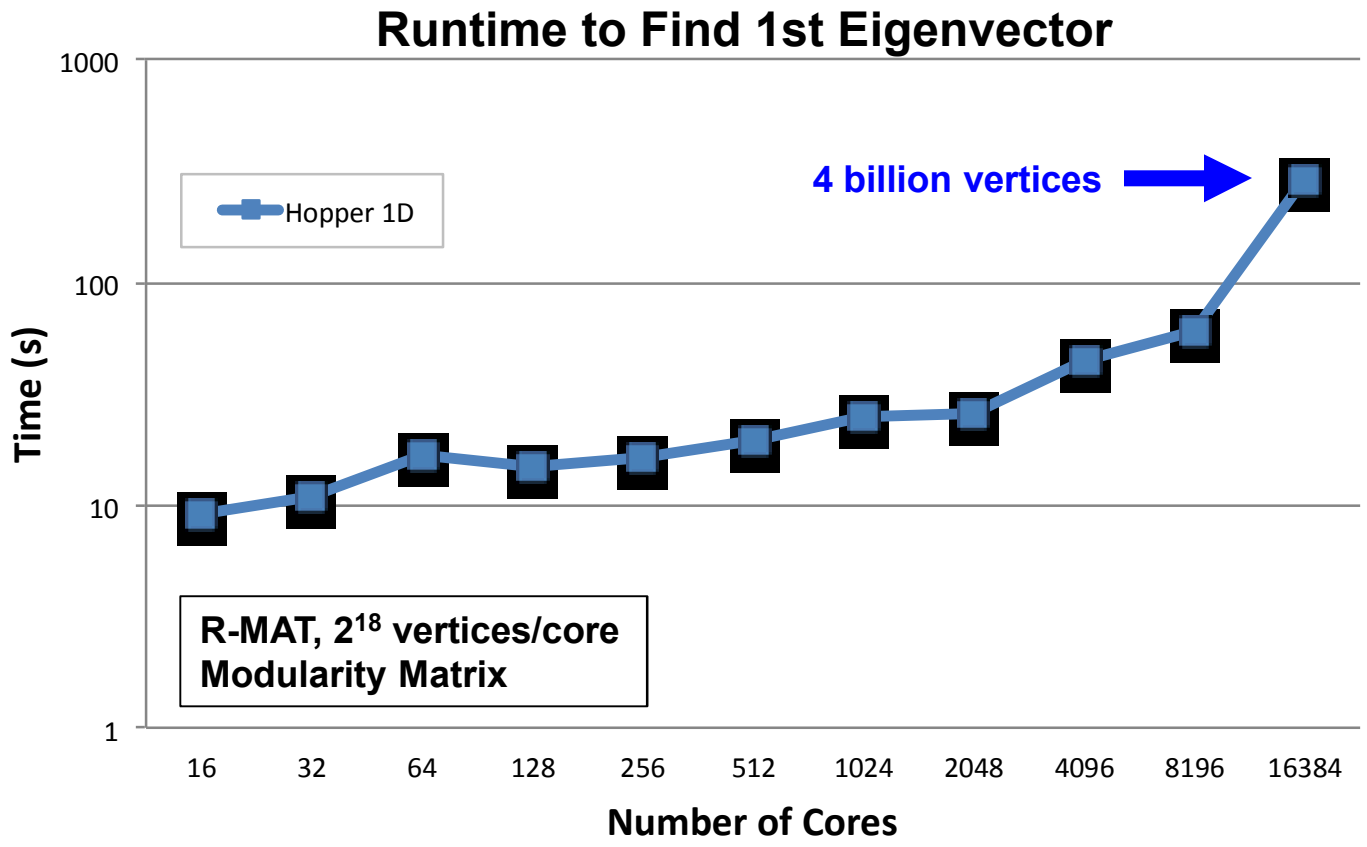
Parallel Implementation

Sandia National Laboratories

- Using Anasazi (Trilinos) Eigensolver
  - Block Krylov-Schur
  - Eigenpairs corresponding to eigenvalues with largest real component
  - User defined operators (don't form matrix explicitly)

- Initial Numerical Experiments
  - R-Mat (a=0.5, b=0.125, c=0.125, d=0.25)
    - Average nonzeros per row: 8
    - Number of rows: 2^{22} to 2^{32}
  - Two systems
    - LLGrid (MIT LL) – compute cluster (10 GB ethernet)
    - Hopper* (NERSC) -- Cray XE6 supercomputer
  - Initially: 1D random row distribution (good load balance)

Weak Scaling – Hopper*

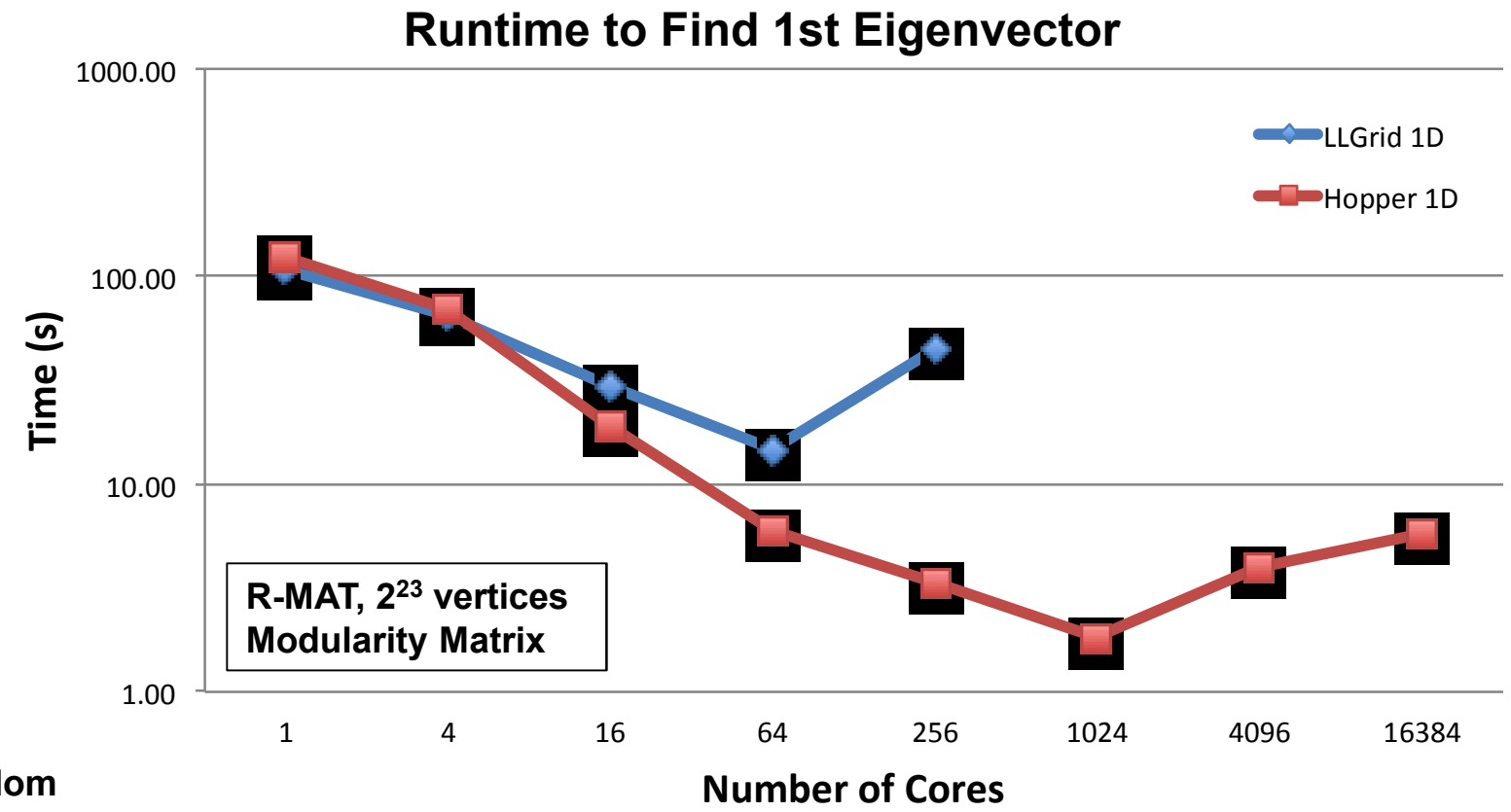


1D random partitioning

Solved system for up to 4 billion vertex graph

* This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Strong Scaling Results



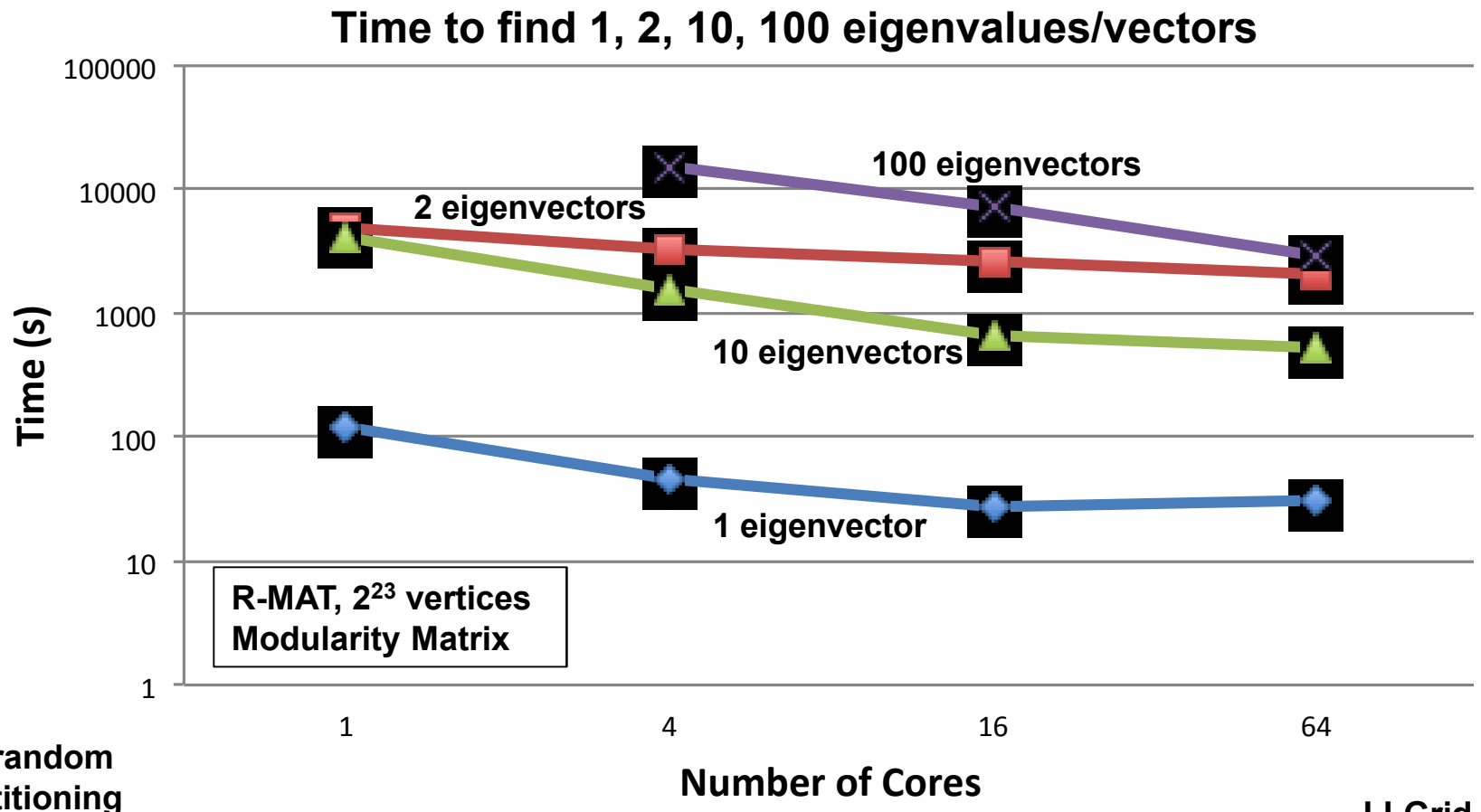
1D random partitioning

Scalability limited and runtime increases for large numbers of cores

* This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Finding Multiple Eigenvectors – LLGrid

Sandia National Laboratories



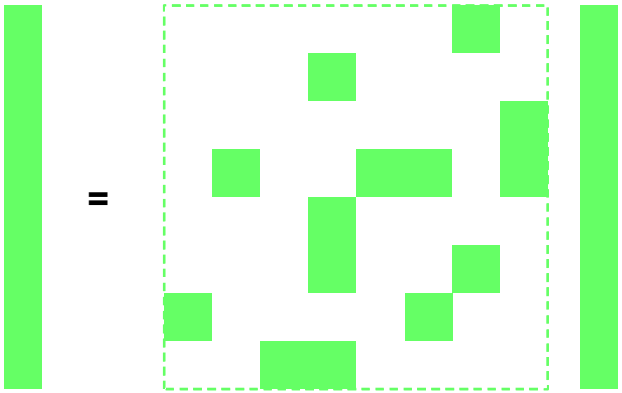
1D random partitioning

LLGrid system

Significant increase in runtime when finding additional eigenvectors

Sparse Matrix-Vector Multiplication

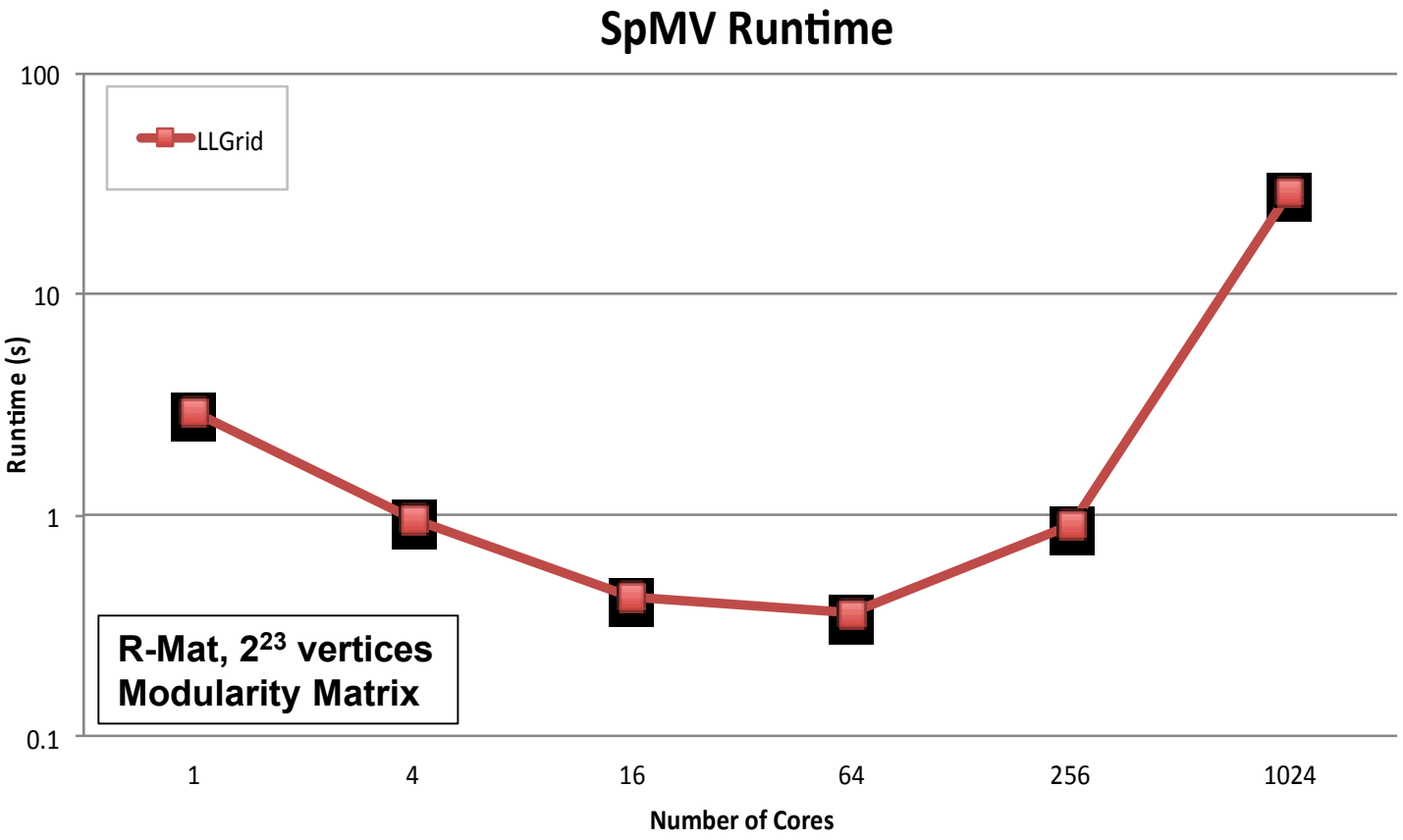
Sandia National Laboratories



- Sparse matrix-dense vector multiplication (SpMV) key computational kernel in eigensolver
- Performance of SpMV challenging for matrices resulting from power-law graphs
  - Load imbalance
  - Irregular communication
  - Little data locality
- Important to improve performance of SpMV

SpMV Strong Scaling -- LLGrid

Sandia National Laboratories



1D random partitioning

Scalability limited and runtime increases for large numbers of cores

Outline

- Anomaly Detection in Very Large Graphs
- Eigenanalysis and Performance Challenges
- Improving Sparse Matrix-Vector Multiplication (SpMV) Performance through Data Partitioning
- Partitioning: Dynamic Graphs and Sampling
- Summary

Data Partitioning to Improve SpMV

Sandia National Laboratories

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = \begin{bmatrix} 1 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 1 & 9 & 0 & 5 & 0 & 0 & 0 \\ 0 & 8 & 1 & 7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 1 & 8 & 0 & 0 \\ 4 & 0 & 0 & 0 & 3 & 1 & 3 & 0 \\ 0 & 0 & 0 & 6 & 0 & 9 & 1 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 4 \\ 3 \\ 1 \\ 4 \\ 2 \\ 1 \end{bmatrix}$$

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

- Partition matrix nonzeros
- Partition vector elements

Partitioning Objective

Sandia National Laboratories

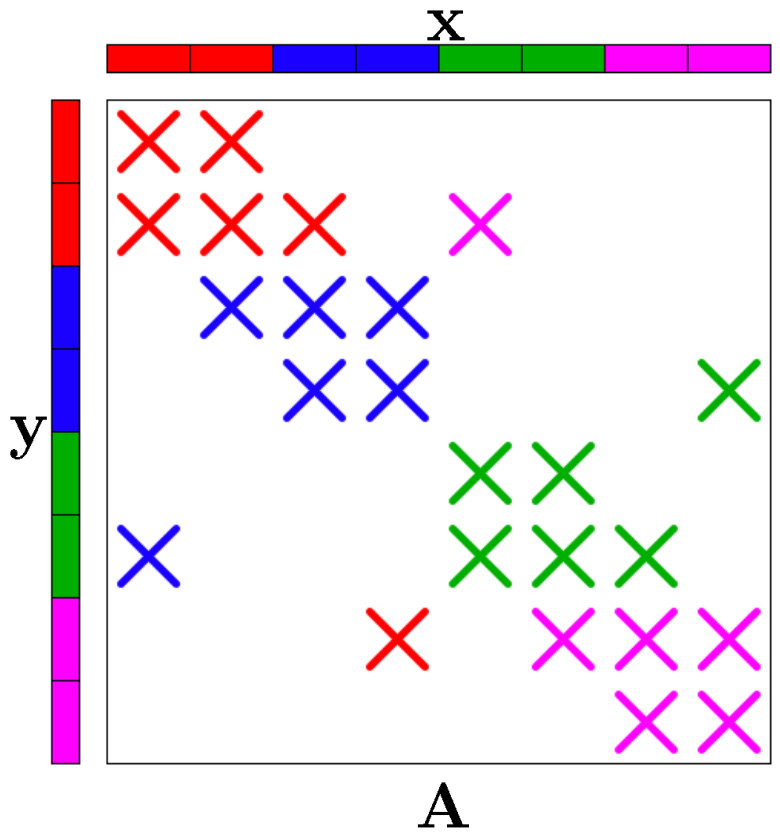
- Ideally we minimize total execution time of SpMV
- Settle for easier objectives
  - Balance computational work
  - Minimize communication metric
    - Total communication volume
    - Bound number of messages
- Can Partition matrices in different ways
  - 1D
  - 2D
- Can model problem in different ways
  - Graph
  - Bipartite graph
  - Hypergraph

Parallel SpMV

Sandia National Laboratories

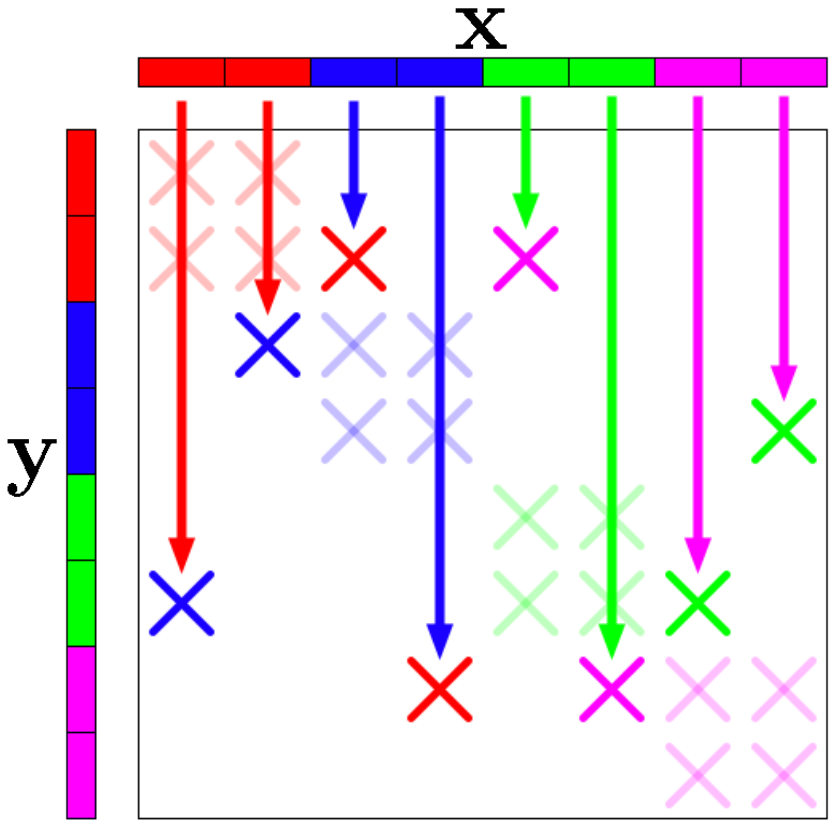
$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = \begin{bmatrix} 1 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 1 & 9 & 0 & 5 & 0 & 0 & 0 \\ 0 & 8 & 1 & 7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 1 & 8 & 0 & 0 \\ 4 & 0 & 0 & 0 & 3 & 1 & 3 & 0 \\ 0 & 0 & 0 & 6 & 0 & 9 & 1 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 4 \\ 3 \\ 1 \\ 4 \\ 2 \\ 1 \end{bmatrix}$$

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$



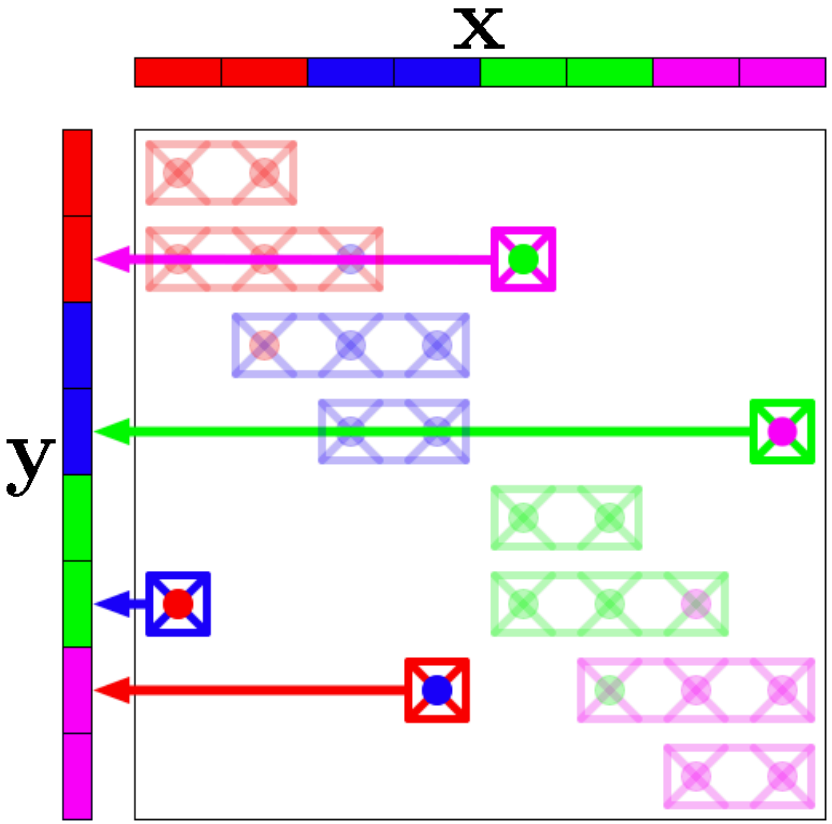
- Alternative way of visualizing partitioning

Parallel SpMV Communication



"fan-out"

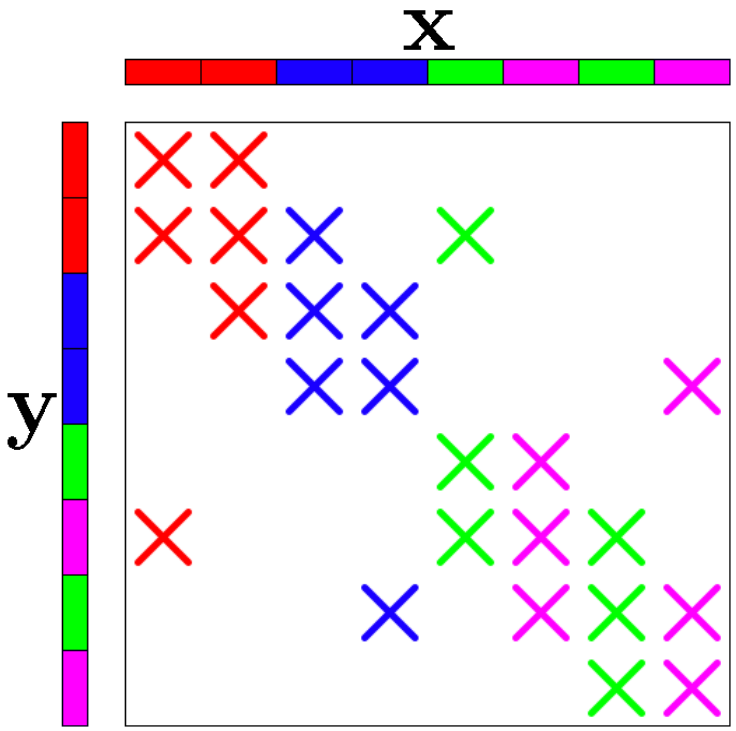
- x_j sent to remote processes that have nonzeros in column j



"fan-in"

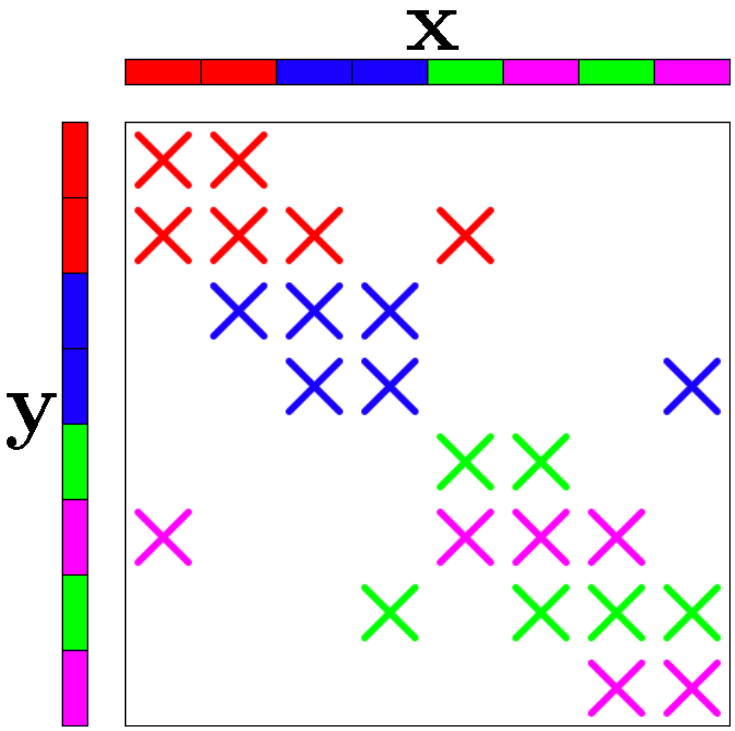
- Partial inner-products sent to process that owns vector element y_i

1D Partitioning



1D Column

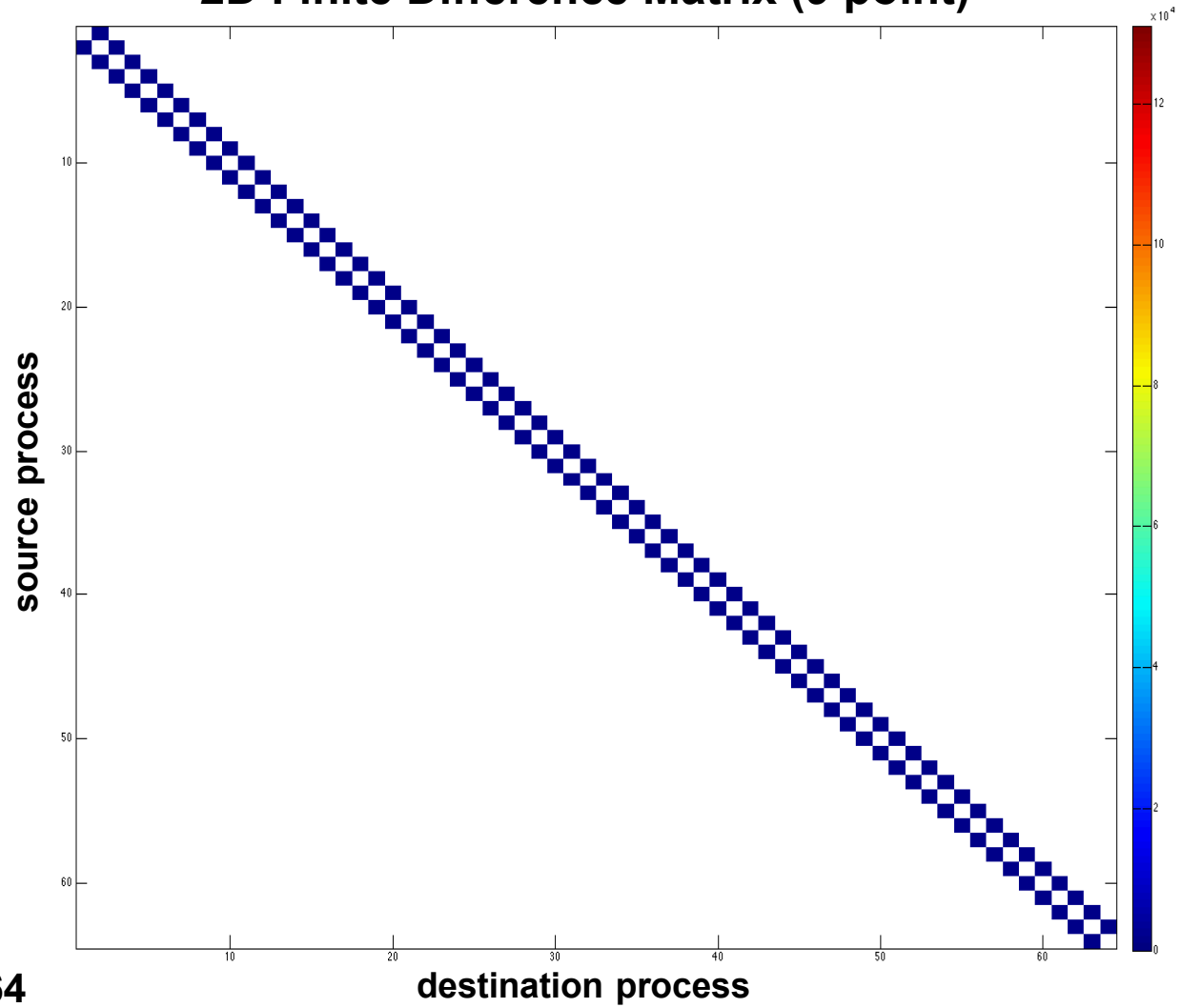
- Each process assigned nonzeros for set of columns



1D Row

- Each process assigned nonzeros for set of rows

Communication Pattern: 1D Block Partitioning



Number of Rows: 2^{23}
Nonzeros/Row: 9

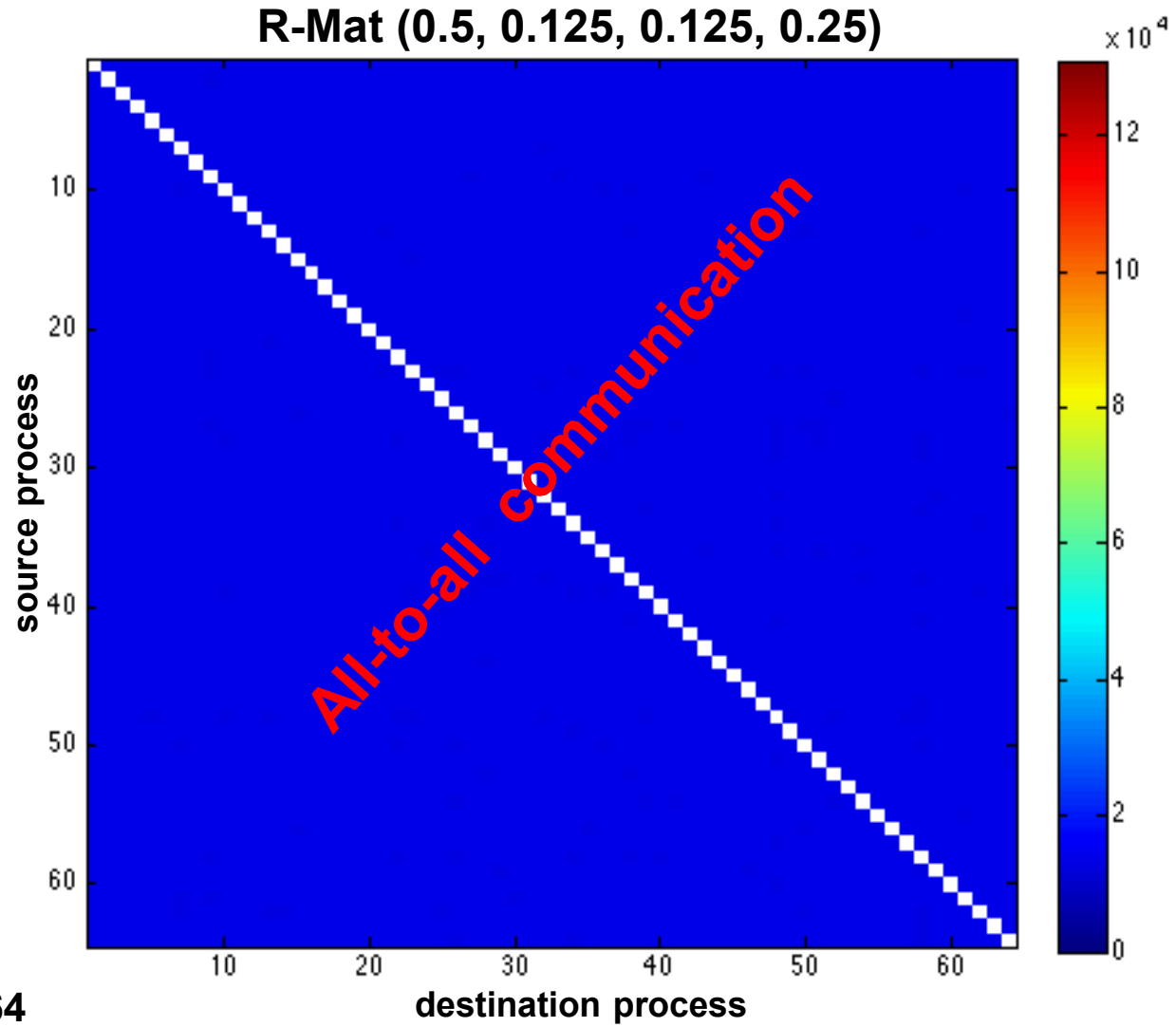
NNZ/process
min: 1.17E+06
max: 1.18E+06
avg: 1.18E+06
max/avg: 1.00

 # Messages (Phase 1)
total: 126
max: 2

Volume (Phase 1)
total: 2.58E+05
max: 4.10E+03

Nice properties:
Great load balance
Small number of messages
Low communication volume

Communication Pattern: 1D Random Partitioning



P=64

Number of Rows: 2^{23}
Nonzeros/Row: 8

NNZ/process
min: 1.05E+06
max: 1.07E+06
avg: 1.06E+06
max/avg: 1.01

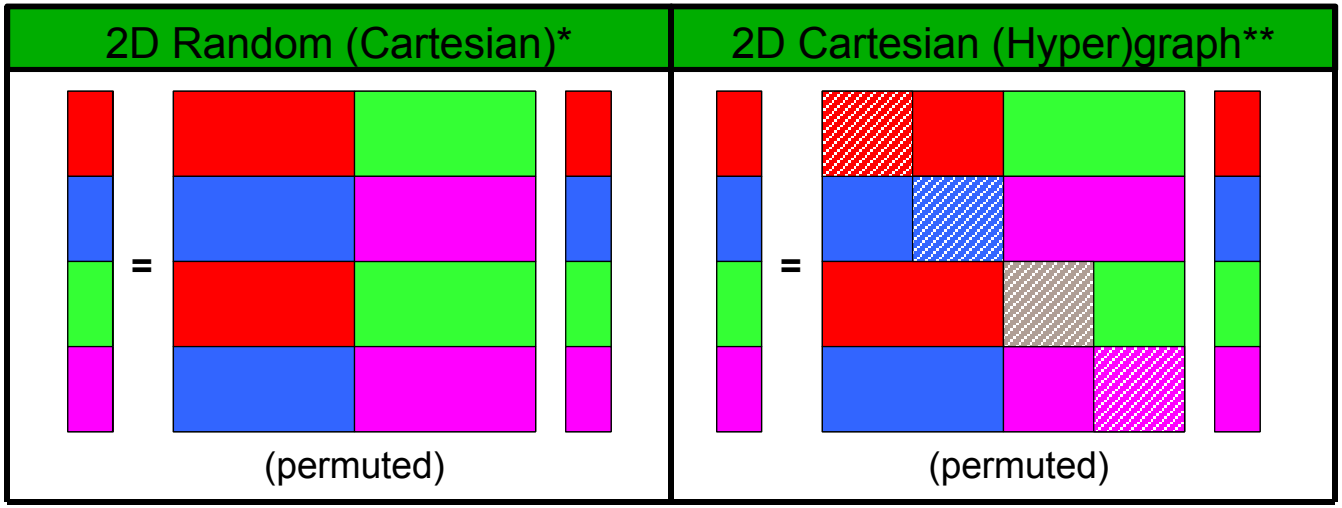
 # Messages (Phase 1)
total: 4032
max: 63

Volume (Phase 1)
total: 5.48E+07
max: 8.62E+05

Nice properties:
Great load balance

Challenges:
All-to-all communication

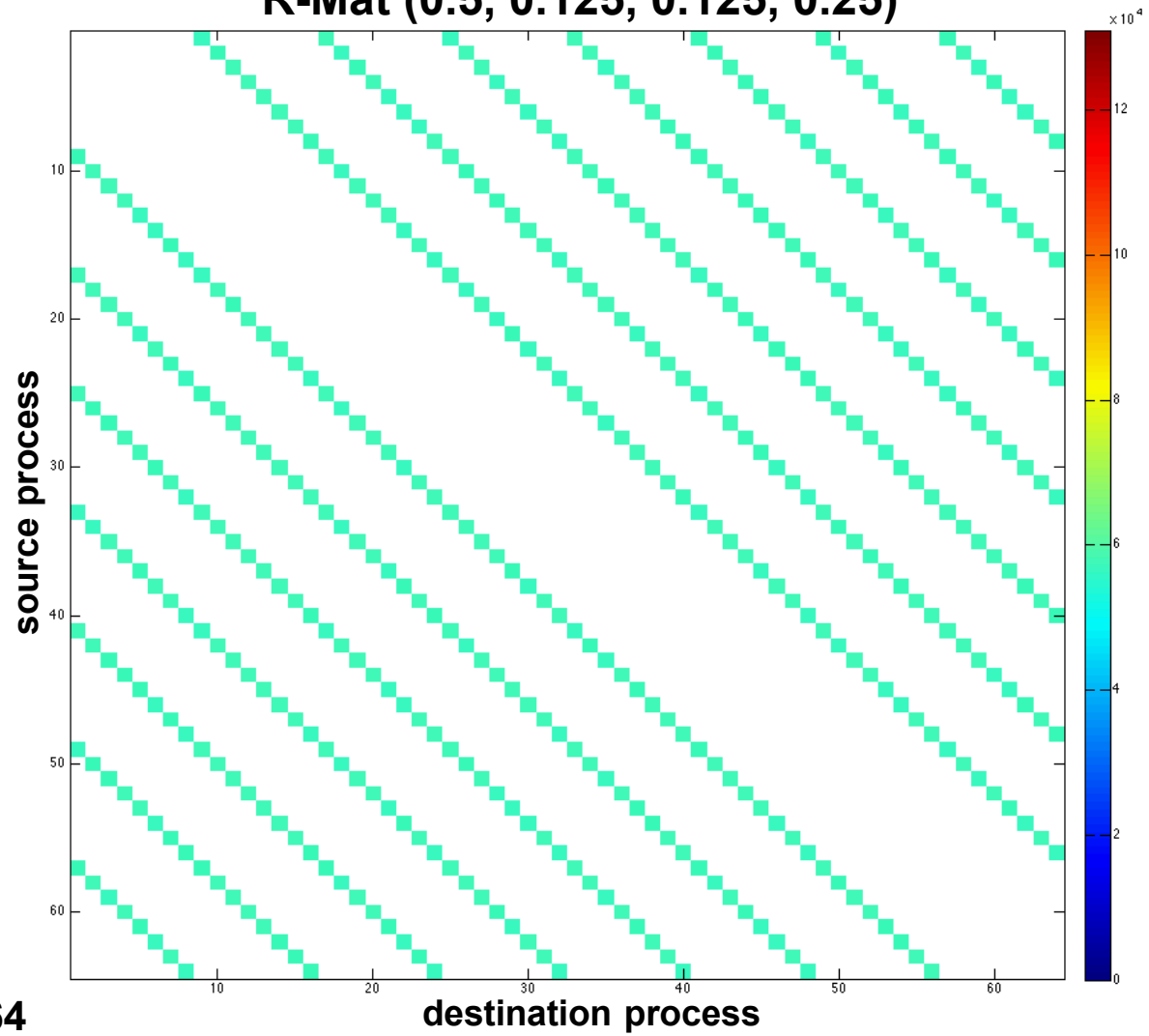
2D Partitioning



- 2D Partitioning
  - More flexibility: no particular part for entire row/column, more general sets of nonzeros
- Use flexibility of 2D partitioning to bound number of messages
  - Distribute nonzeros in permuted 2D Cartesian block manner
- 2D Random (Cartesian)*
  - Block Cartesian with rows/columns randomly distributed
  - Cyclic striping to minimize number of messages
- 2D Cartesian (Hyper)graph**
  - Replace random partitioning with hyper(graph) partitioning to minimize communication volume

*Hendrickson, et al.; Bisseling; Yoo, et al.
**Boman, Devine, Rajamanickam, "Scalable Matrix Computations on Large Scale-Free Graphs Using 2D Partitioning, SC2013.

Communication Pattern: 2D Random Partitioning Cartesian Blocks (2DR)



Number of Rows: 2^{23}
Nonzeros/Row: 8

NNZ/process
min: 1.04E+06
max: 1.05E+06
avg: 1.05E+06
max/avg: 1.01

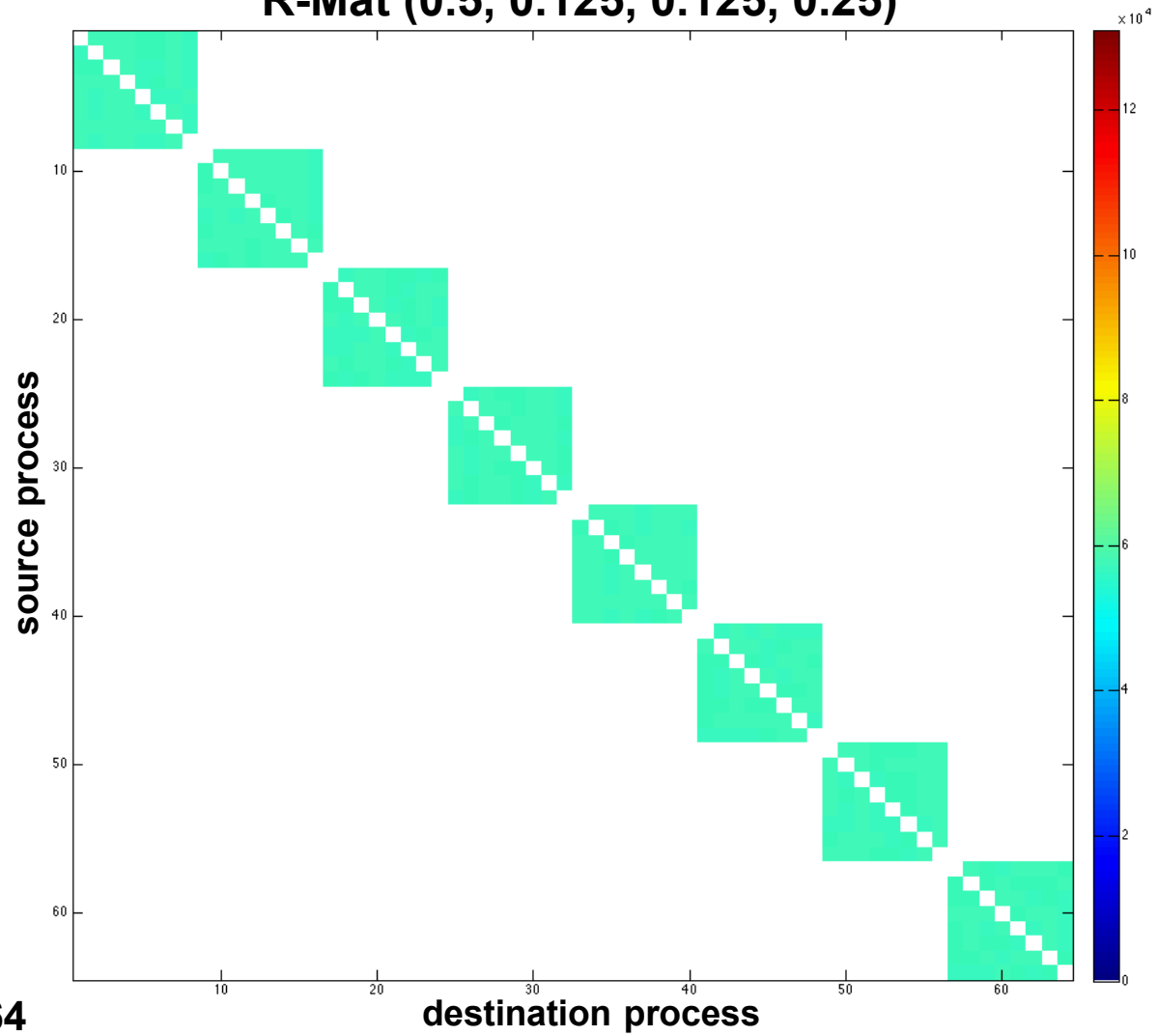
 # Messages (Phase 1)
total: 448
max: 7

Volume (Phase 1)
total: 2.57E+07
max: 4.03E+05

Nice properties:
No all-to-all communication
Total volume lower than 1DR

1DR = 1D Random

Communication Pattern: 2D Random Partitioning Cartesian Blocks (2DR)



Number of Rows: 2^{23}
Nonzeros/Row: 8

NNZ/process
min: 1.04E+06
max: 1.05E+06
avg: 1.05E+06
max/avg: 1.01

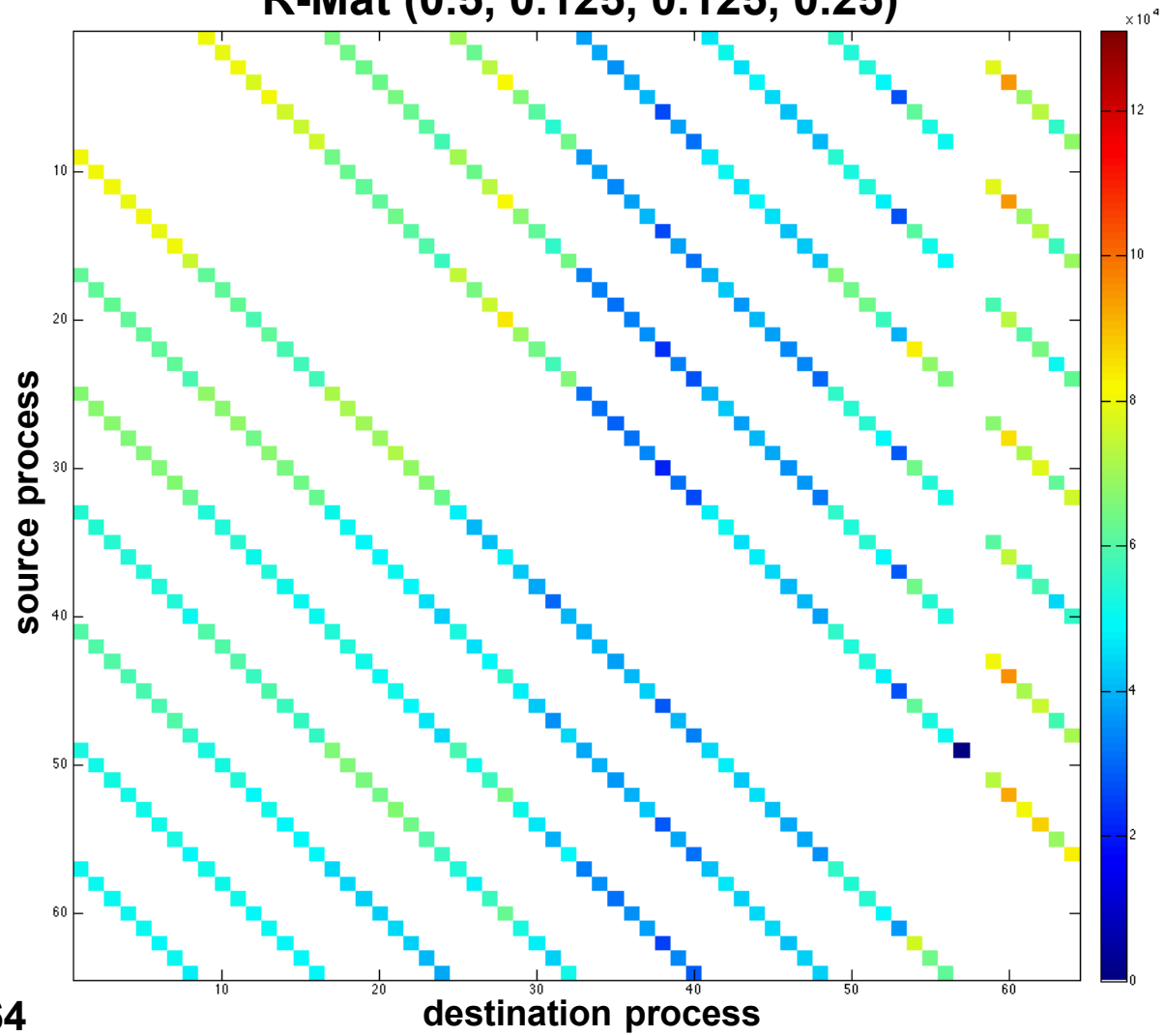
 # Messages (Phase 2)
total: 448
max: 7

Volume (Phase 2)
total: 2.57E+07
max: 4.03E+05

Nice properties:
No all-to-all communication
Total volume lower than 1DR

1DR = 1D Random

Communication Pattern: 2D Cartesian Hypergraph Partitioning



Number of Rows: 2^{23}
Nonzeros/Row: 8

NNZ/process
min: 5.88E+05
max: 1.29E+06
avg: 1.05E+06
max/avg: 1.23

 # Messages (Phase 1)
total: 448
max: 7

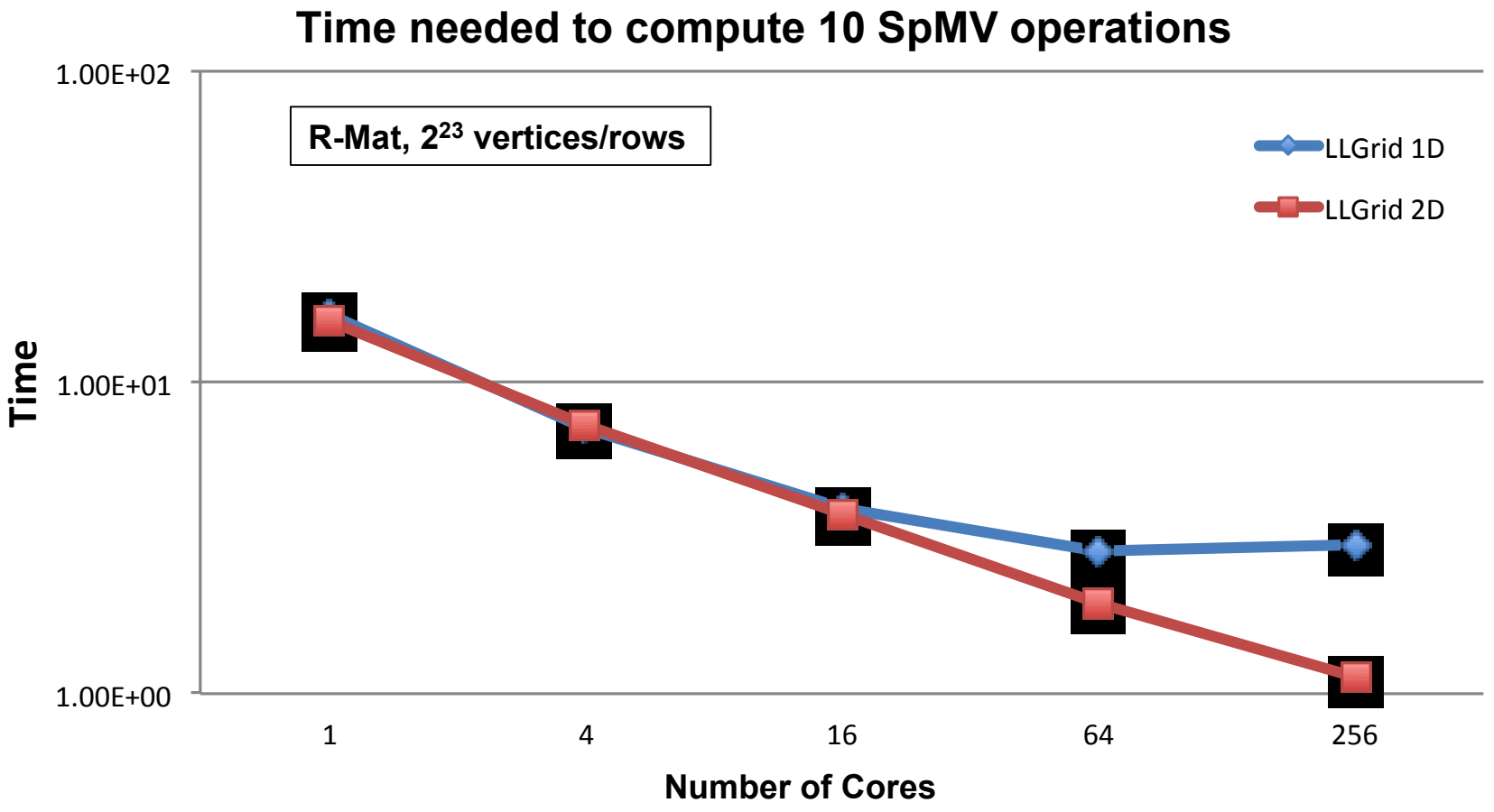
Volume (Phase 1)
total: 2.33E+07
max: 4.52E+05

Nice properties:
No all-to-all communication
Total volume lower than 2DR

Challenges:
Imbalance worse than 2DR

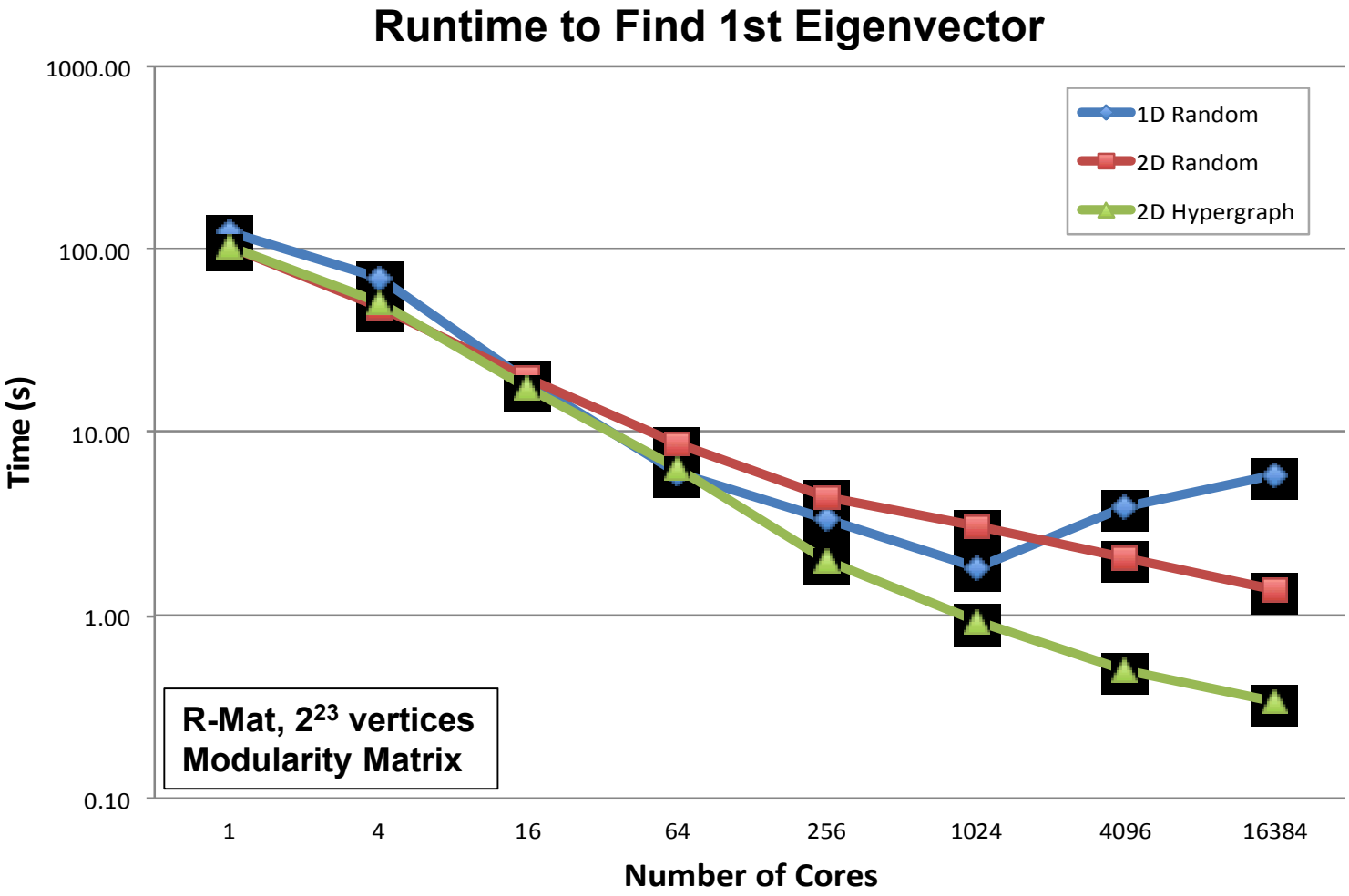
2DR = 2D Random Cartesian

Improved Results: SpMV – LLGrid



Simple 2D method shows improved scalability

Improved Results – NERSC Hopper*



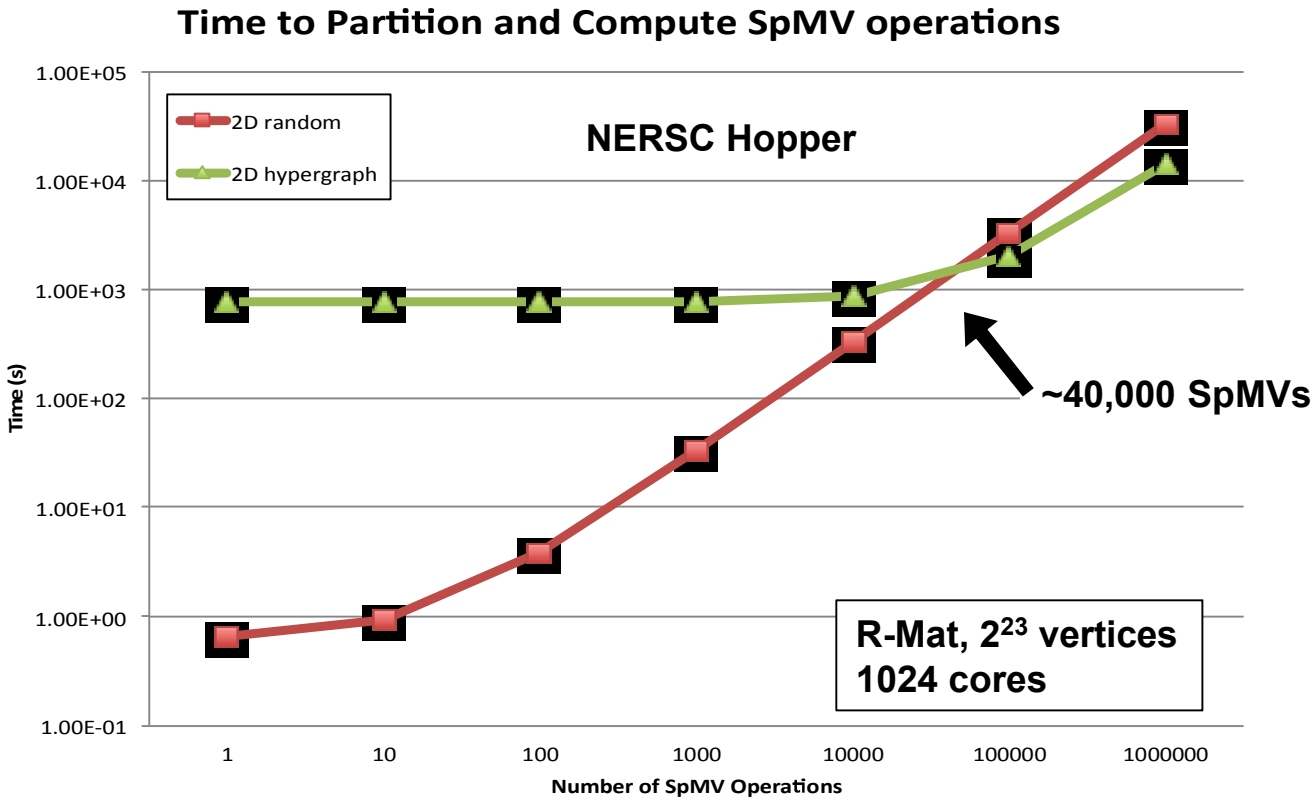
2D methods show improved scalability

* This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Outline

- Anomaly Detection in Very Large Graphs
- Eigenanalysis and Performance Challenges
- Improving Sparse Matrix-Vector Multiplication (SpMV) Performance through Data Partitioning
- Partitioning: Dynamic Graphs and Sampling
- Summary

Challenge with Hypergraph/Graph Partitioning

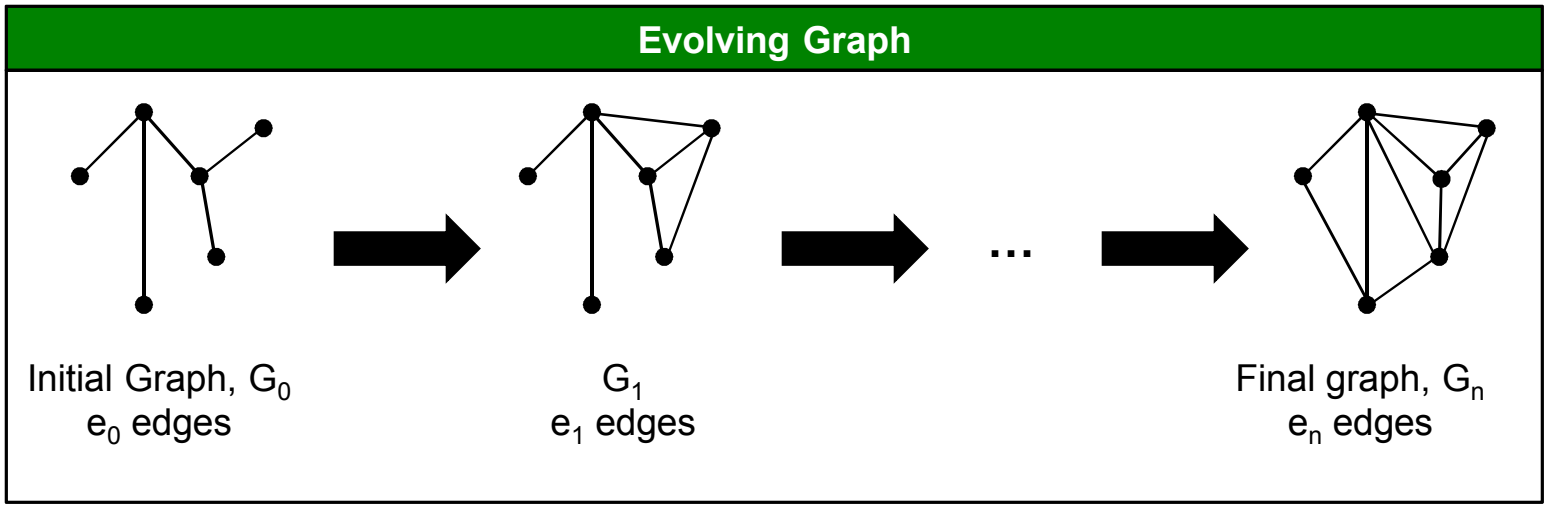


- High partitioning cost of graph/hypergraph methods must be amortized by computing many SpMV operations
- Detection* requires at most 1000s of SpMV operations
- Expensive partitions need to be effective for multiple graphs

*L1 norm method: computing 100 eigenvectors

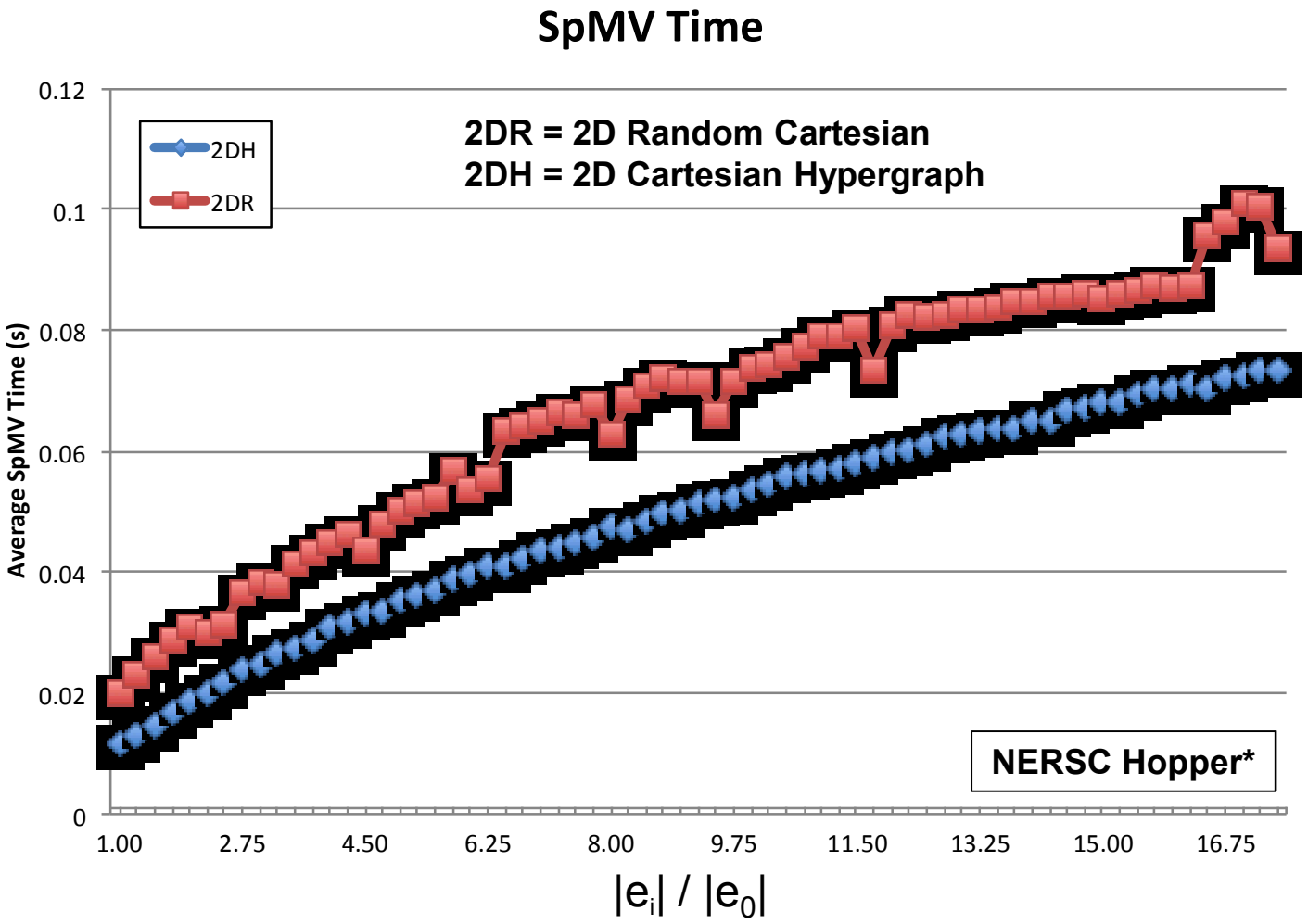
Wolf, Miller: "Sparse Matrix Partitioning for Parallel Eigenanalysis of Large Static and Dynamic Graphs," *2014 IEEE HPEC Proc.*

Experiment: Partitioning for Dynamic Graphs



- Key question: How long will a partition be effective?
- Initial experiment
  - Evolving R-Mat matrices: fixed number of rows, R-Mat parameters (a,b,c,d)
  - Start with a given number of nonzeros ($|e_0|$)
  - Iteratively add nonzeros until target number of nonzeros is reached ($|e_n|$)

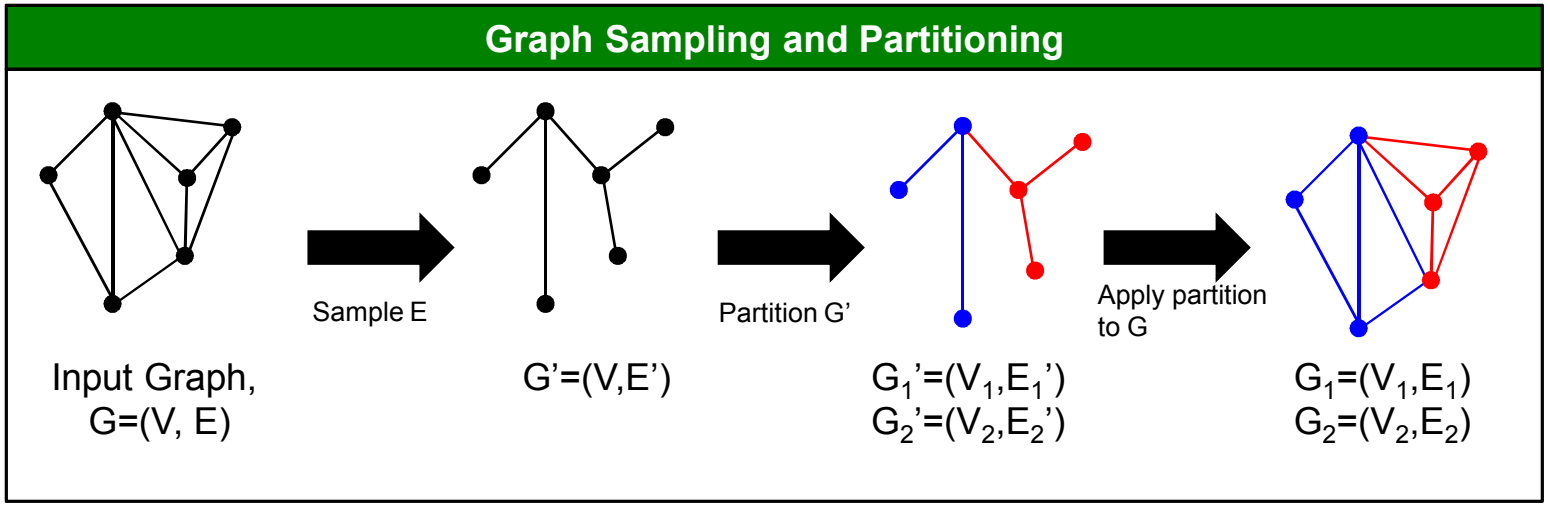
Results: Partitioning for Dynamic Graphs



Hypergraph partition surprising effective after more than 16x $|e_0|$ edges added

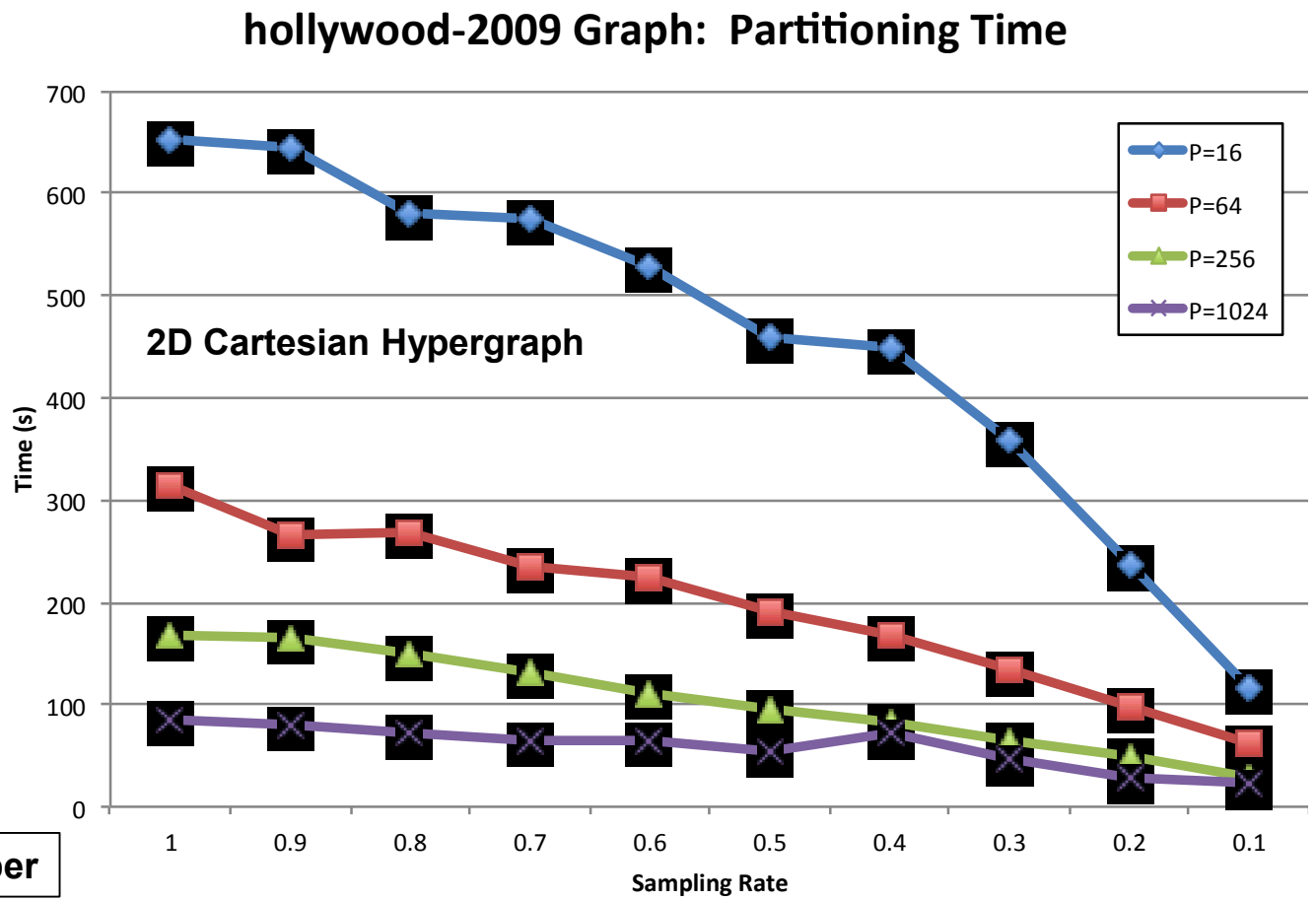
* This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Sampling and Partitioning for Web/SN Graphs



- Idea: Partition sampled graph to reduce partitioning time
- Steps:
  1. Produce smaller graph G' by sampling edges in graph G (uniform random sampling), keep vertices same
  2. Partition G' (2D Cartesian Hypergraph)
  3. Apply partition to G

hollywood-2009* Graph



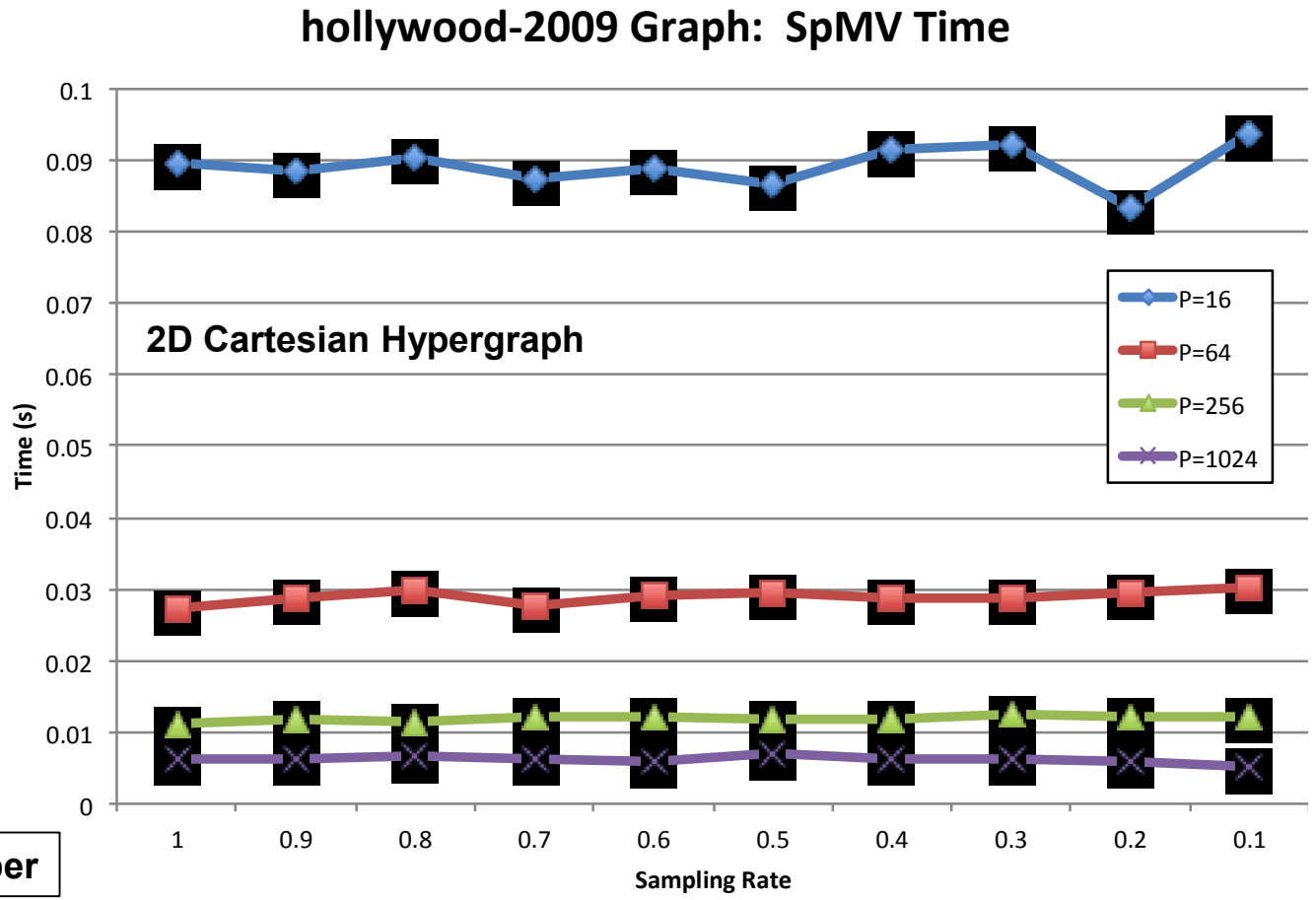
NERSC Hopper

Edge sampling greatly reduces partitioning time

*The University of Florida Sparse Matrix Collection

Wolf, Miller: *2014 IEEE HPEC Proc.*

hollywood-2009* Graph



NERSC Hopper

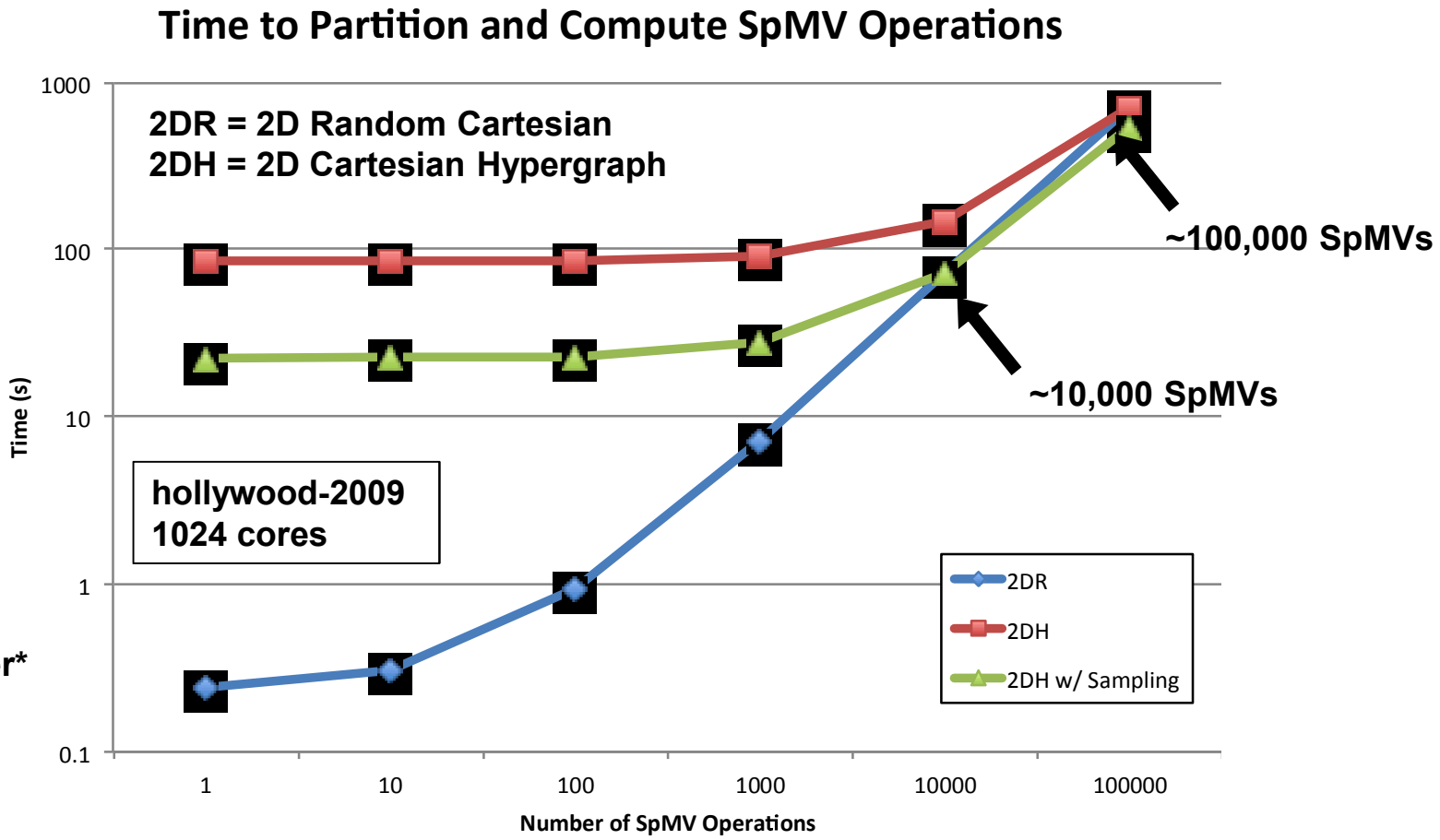
Resulting SpMV time does not increase for modest sampling

*The University of Florida Sparse Matrix Collection

Wolf, Miller: *2014 IEEE HPEC Proc.*

Challenge with Hypergraph Partitioning Revisited

Sandia National Laboratories



NERSC Hopper*

Sampling reduces overhead of hypergraph partitioning (fewer SpMVs needed to amortize partitioning cost)

* This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Wolf, Miller: *2014 IEEE HPEC Proc.*

Outline

- Anomaly Detection in Very Large Graphs
- Eigenanalysis and Performance Challenges
- Improving Sparse Matrix-Vector Multiplication (SpMV) Performance through Data Partitioning
- Partitioning: Dynamic Graphs and Sampling
- Summary

Summary

Sandia National Laboratories

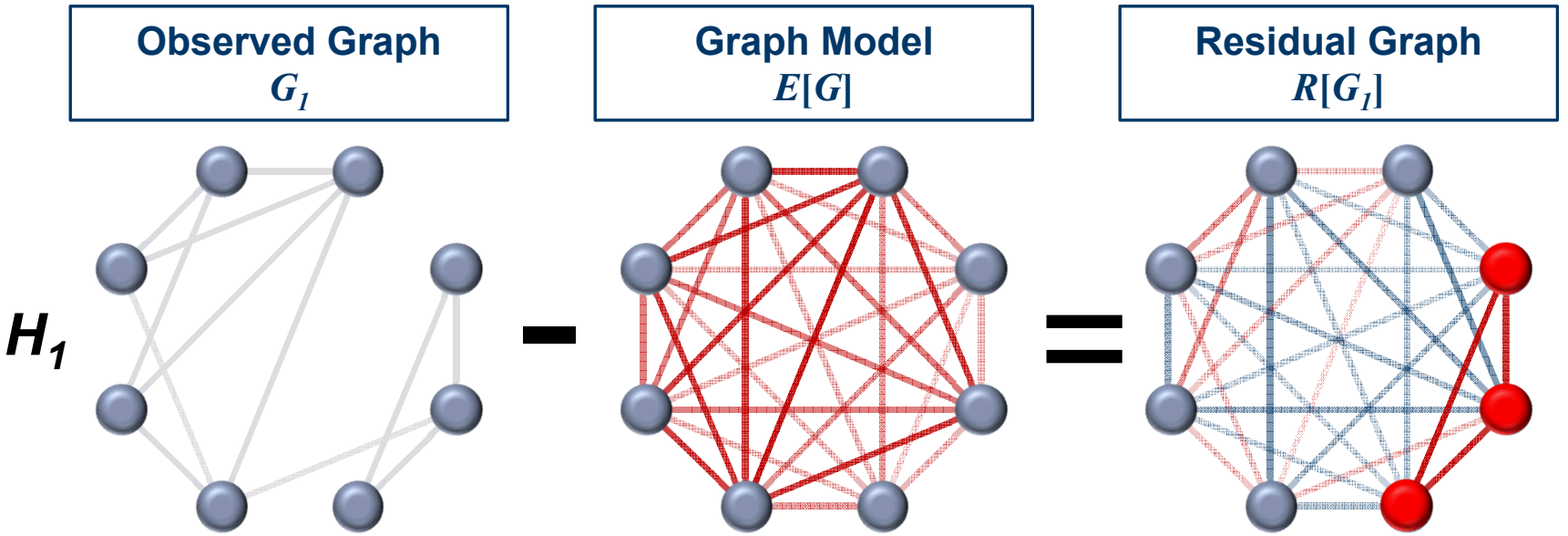
- Outlined HPC approach to processing big data
  - Signal processing for graphs
  - Statistical framework for anomaly detection in graphs
- Key component is eigensolver for dimensionality reduction
- Solving eigensystems resulting from power law graphs challenging
  - Load imbalance
  - Poor data locality
- SpMV key computational kernel
  - 1D data partitioning limits performance due to all-to-all communication
  - 2D data partitioning can be used to improve scalability
- Sampling can improve hypergraph-based partitioning performance for web/SN graphs

Extra

Sandia National Laboratories

Residuals Example: Anomalous Subgraph

Sandia National Laboratories



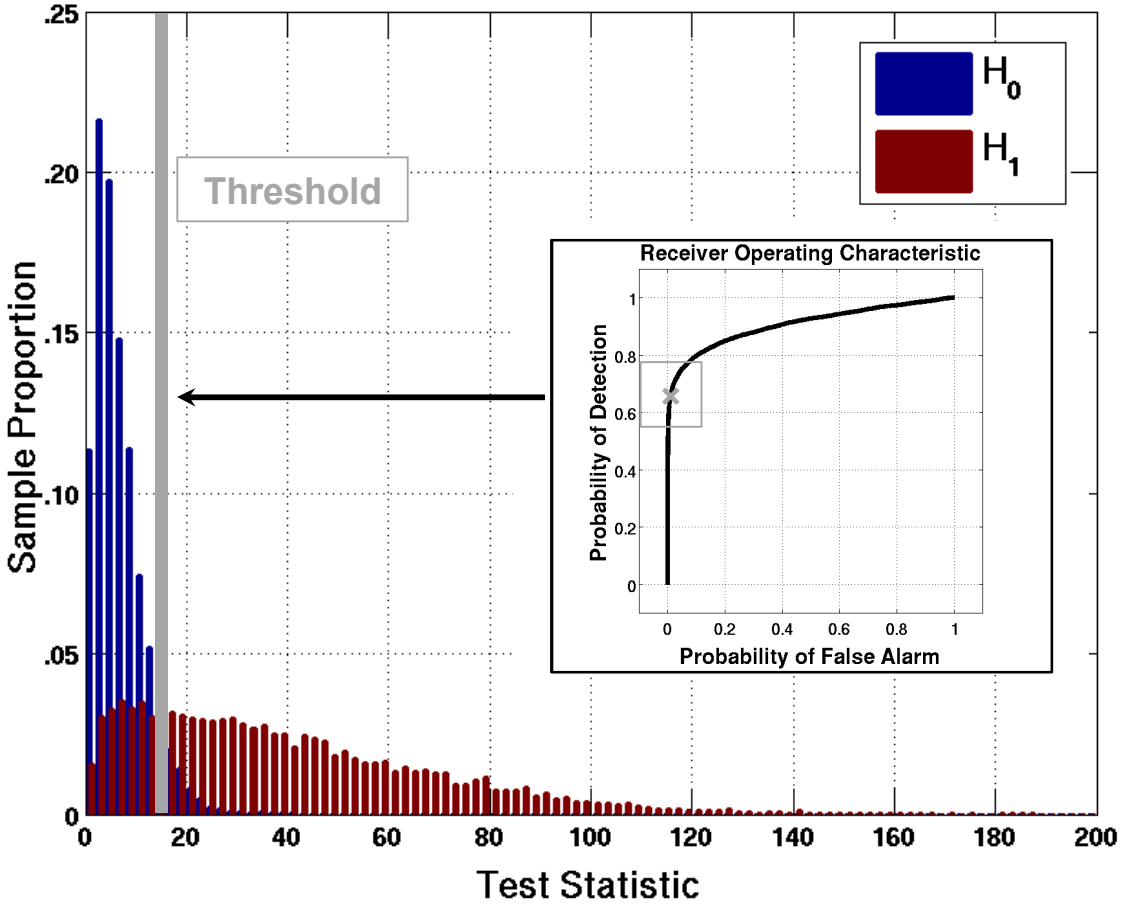
- Residual graph represents the difference between the observed and expected
- *Coordinated* vertices (subsets of vertices connected by edges with large edge weights) in residual graph will produce much stronger signal than uncoordinated vertices

Detection framework is designed to detect coordinated deviations from the expected topology

Anomaly Detection: Setup Phase

Sandia National Laboratories

Temporal Integration → Graph Model Construction → Residual Decomposition → Component Selection → **Anomaly Detection** → Identification



H_0 – Null hypothesis, no signal
H_1 – Alternative hypothesis, signal

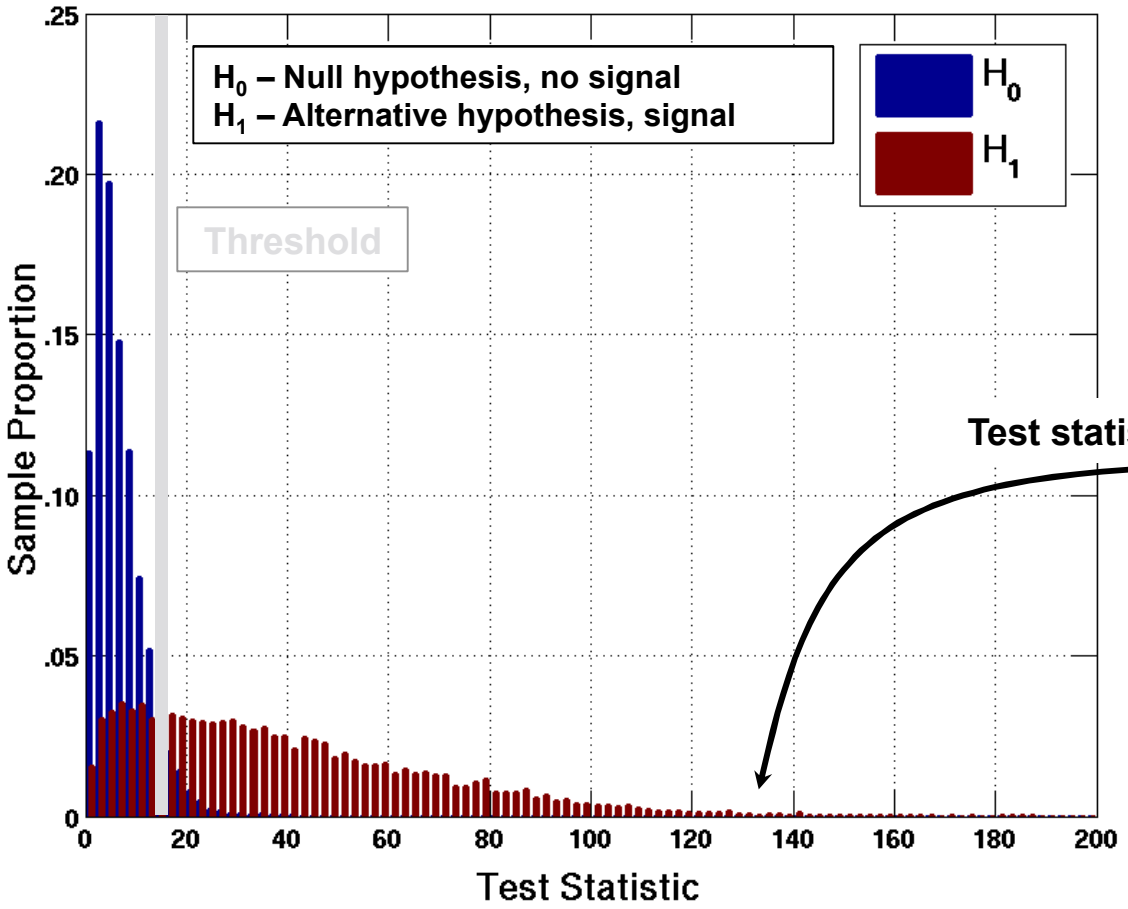
Detection Setup

1. Monte-Carlo simulations to generate density functions
2. ROC-curve generated from density function

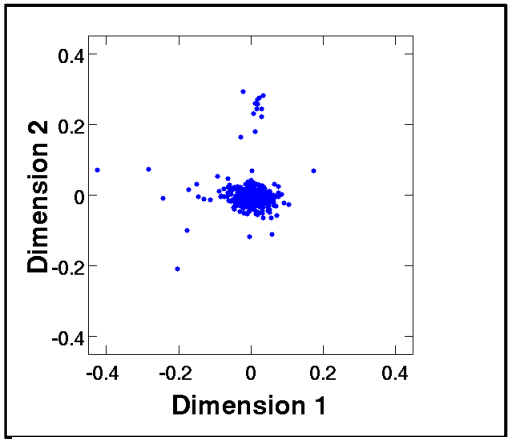
Anomaly Detection

Sandia National Laboratories

Temporal Integration → Graph Model Construction → Residual Decomposition → Component Selection → **Anomaly Detection** → Identification



Test statistic calculated for observed graph:



$$\chi^2_{\max} = \max_{\theta} \chi^2 \left(\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} [u_1\ u_2]^T \right)$$

Test statistic value significantly larger than test statistic value threshold corresponding to 1% false alarm rate

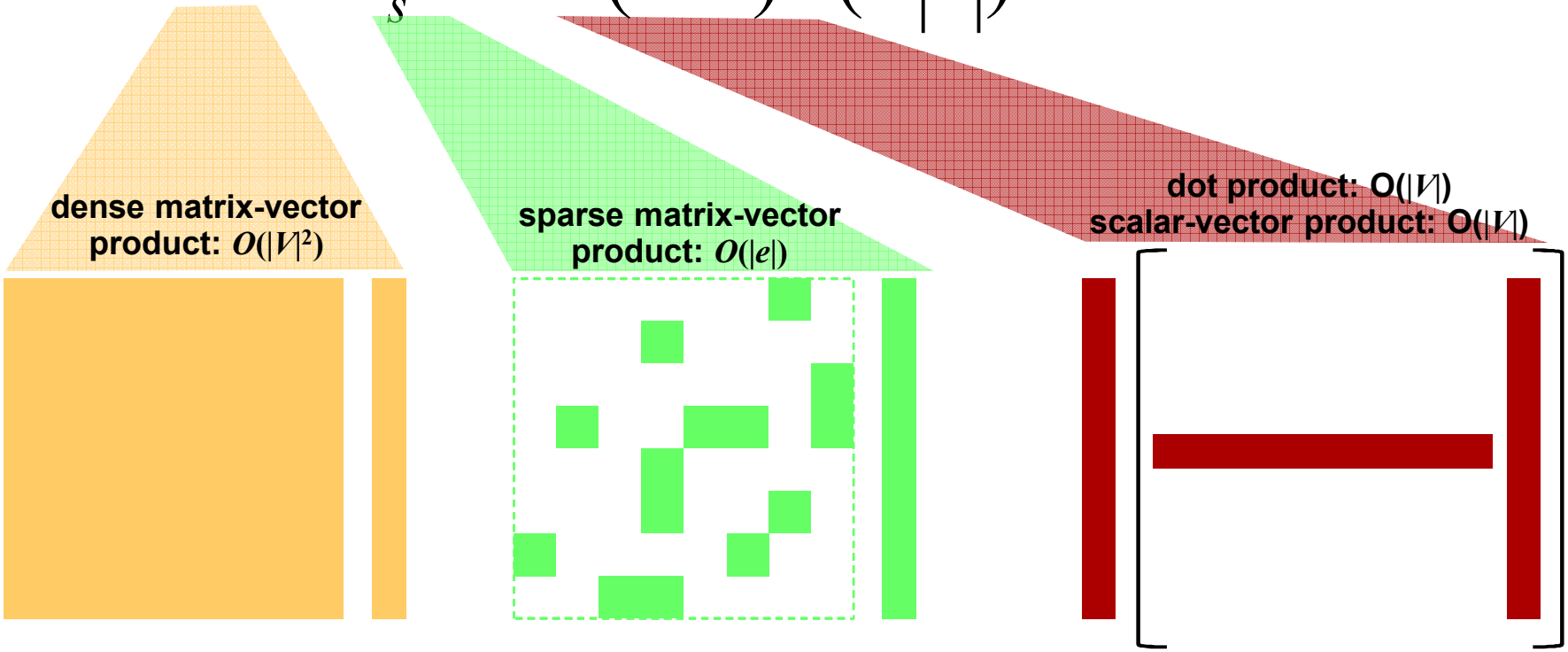
Modularity Matrix: Computation Breakdown

Sandia National Laboratories

Matrix-vector multiplication is at the heart of eigensolver algorithms

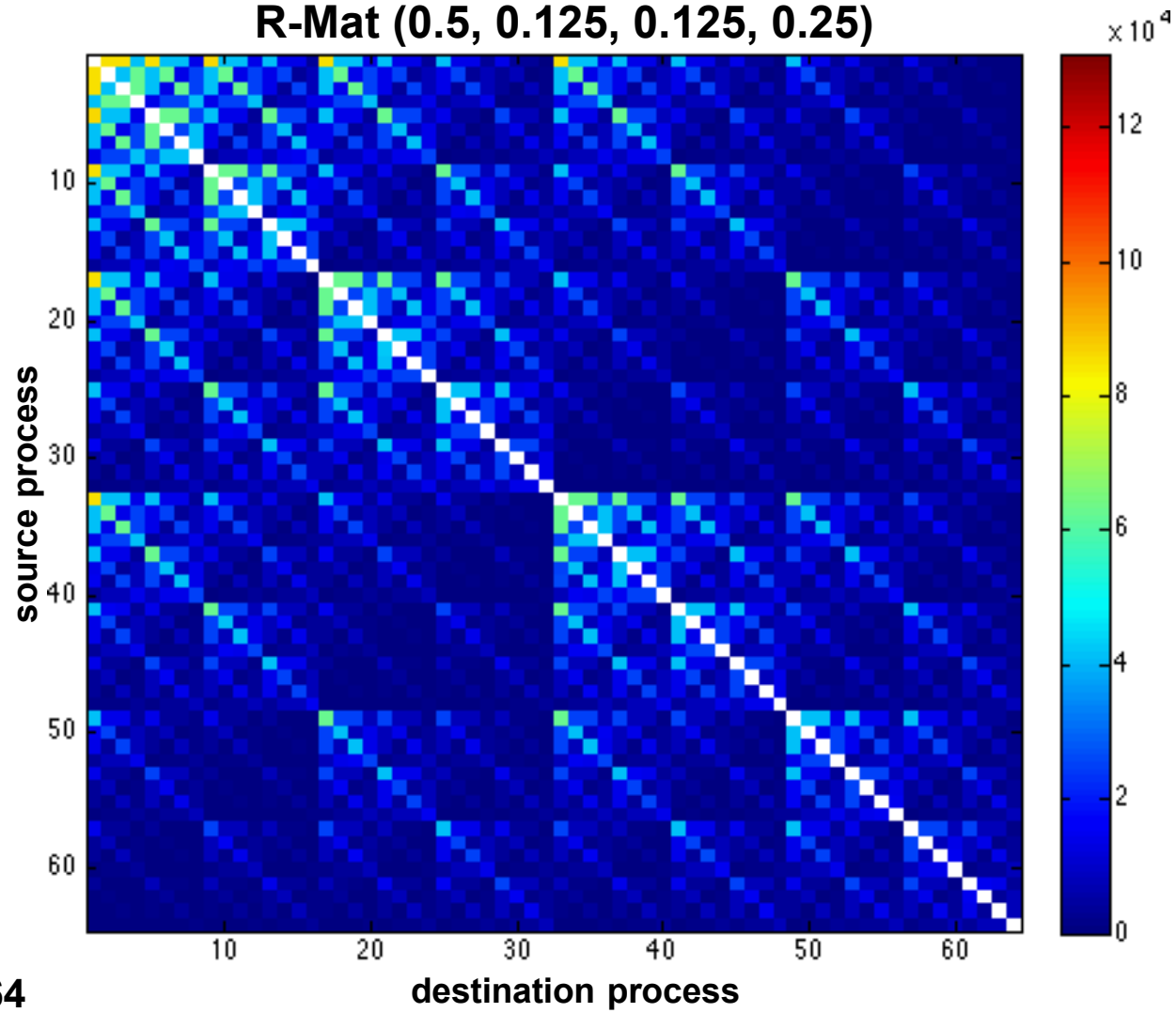
Operator apply:

$$Bx = A_s x - k\,(k^T x)/(2|e|)$$



***Bx* can be computed without storing *B* (modularity matrix)**

Communication Pattern: 1D Block Partitioning



P=64

Number of Rows: 2^{23}
Nonzeros/Row: 8

NNZ/process
min: 1.88E+05
max: 4.00E+06
avg: 1.06E+06
max/avg: 3.78

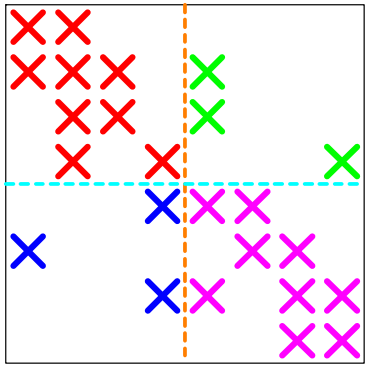
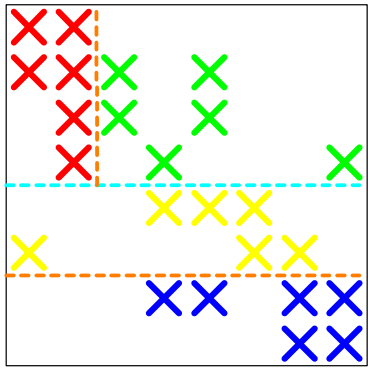
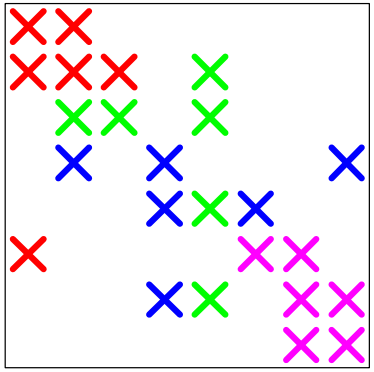
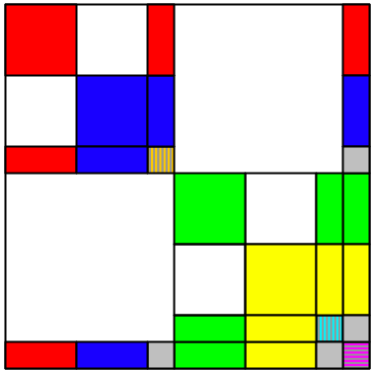
 # Messages (Phase 1)
total: 4032
max: 63

Volume (Phase 1)
total: 4.02E+07
max: 1.48E+06

Challenges:
Poor load balance
All-to-all communication

2D Partitioning

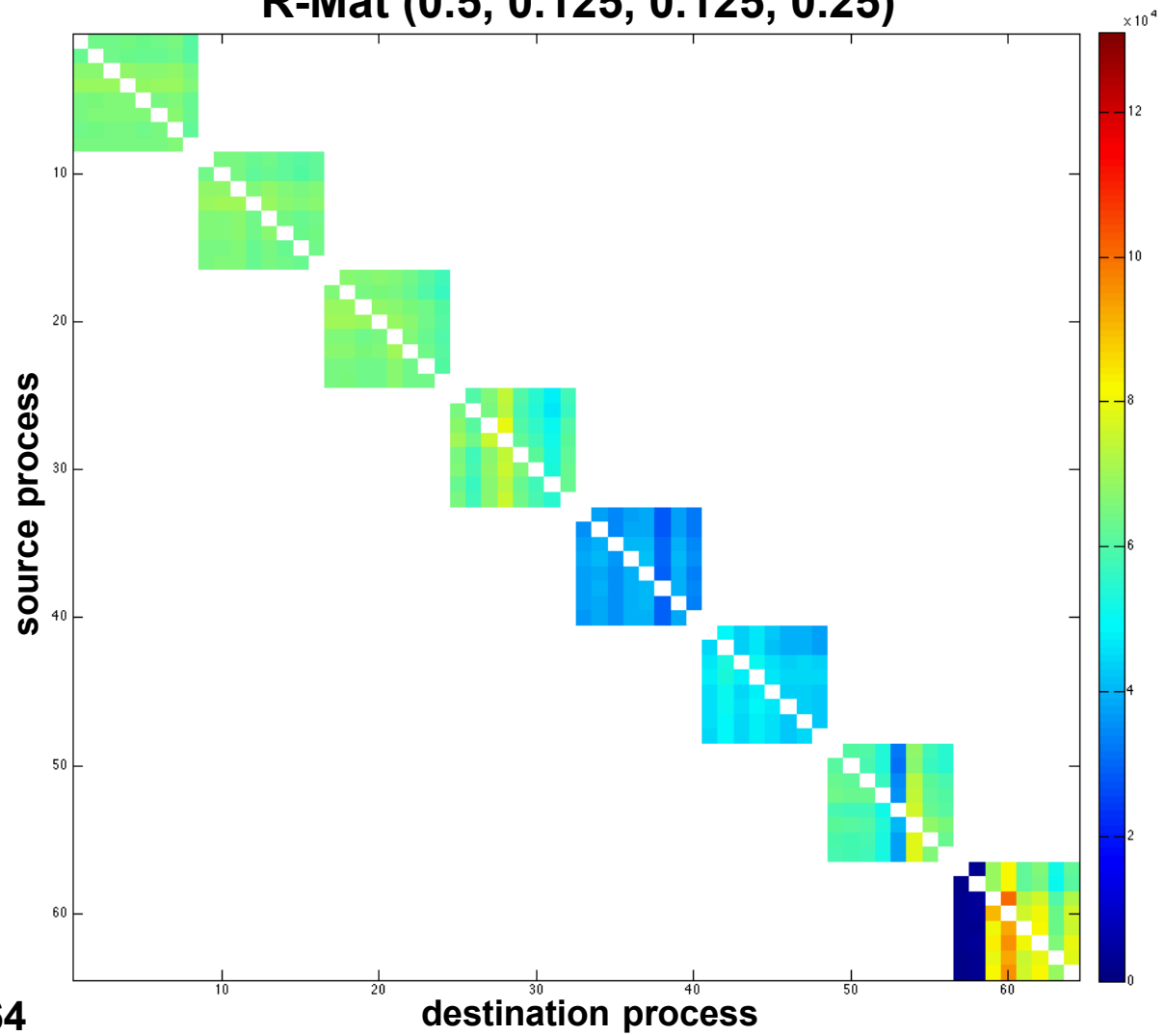
Sandia National Laboratories

Block/Cartesian	Mondriaan (Vastenhouw, Bisseling)
	
Fine-grain (Catalyurek, Aykanat)	**Nested-dissection (Boman, Wolf)***
	

- More flexibility: no particular part for entire row or column
- More general sets of nonzeros assigned parts

*Boman and Wolf, "A Nested Dissection Partitioning Method for Parallel Sparse Matrix-Vector Multiplication," IEEE HPEC 2013.

Communication Pattern: 2D Cartesian Hypergraph Partitioning



Number of Rows: 2^{23}
Nonzeros/Row: 8

NNZ/process
min: 5.88E+05
max: 1.29E+06
avg: 1.05E+06
max/avg: 1.23

 # Messages (Phase 2)
total: 448
max: 7

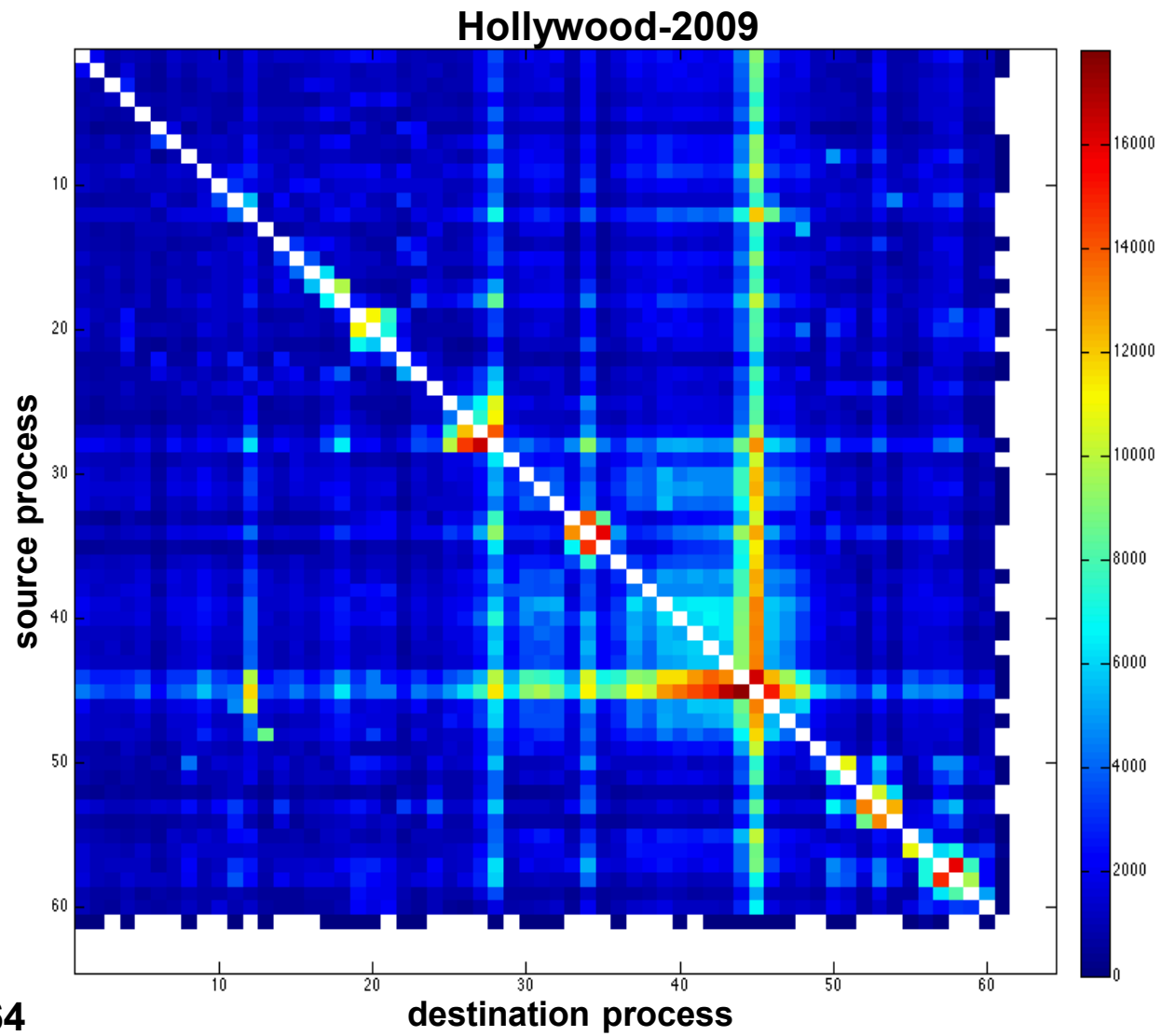
Volume (Phase 2)
total: 2.54E+07
max: 4.80E+05

Nice properties:
No all-to-all communication
Total volume lower than 2DR

Challenges:
Imbalance worse than 2DR

2DR = 2D Random Cartesian

Communication Pattern: 1D Block Partitioning



P=64

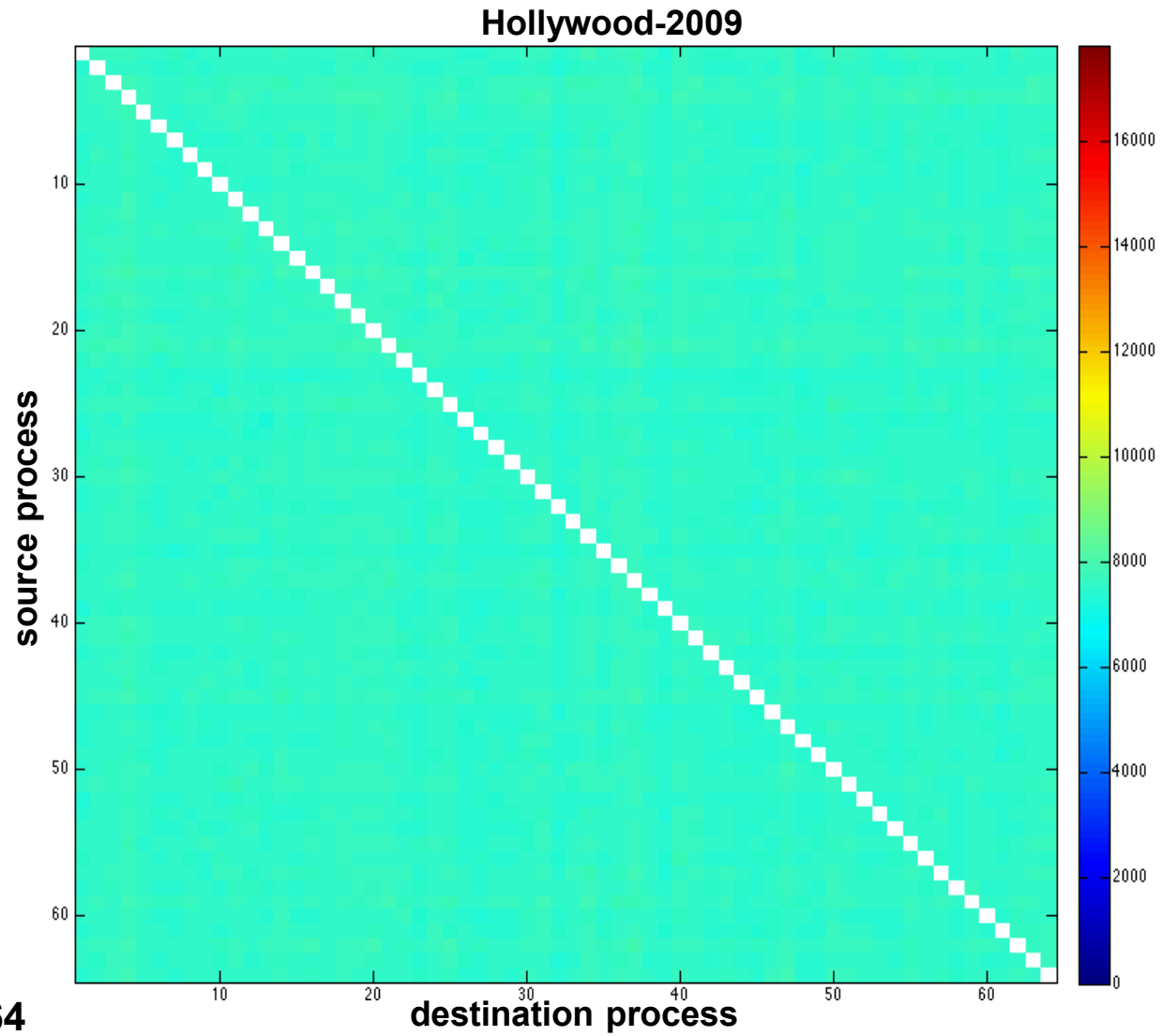
Number of Rows: 1.1M
Number of Nonzeros: 113.9M

NNZ/process
min: 2.31E+04
max: 1.26E+07
avg: 1.78E+06
max/avg: 7.08

 # Messages (Phase 1)
total: 3660
max: 60

Volume (Phase 1)
total: 7.68E+06
max: 3.99E+05

Communication Pattern: 1D Random Partitioning



P=64

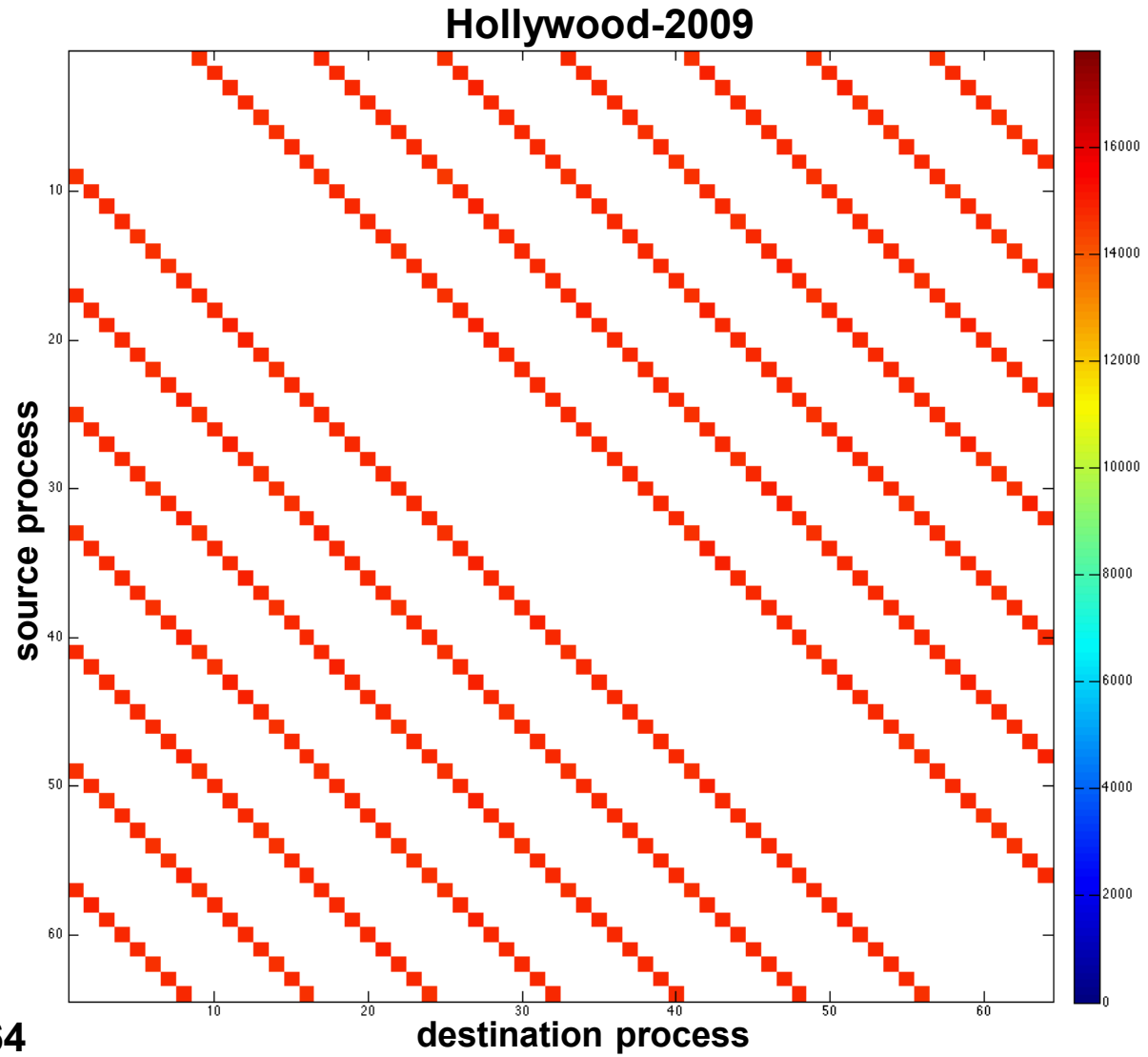
Number of Rows: 1.1M
Number of Nonzeros: 113.9M

NNZ/process
min: 1.70E+06
max: 1.87E+06
avg: 1.78E+06
max/avg: 1.05

 # Messages (Phase 1)
total: 4032
max: 63

Volume (Phase 1)
total: 3.06E+07
max: 4.87E+05

Communication Pattern: 2D Random Partitioning (Cartesian Blocks)



P=64

Number of Rows: 1.1M
Number of Nonzeros: 113.9M

NNZ/process
min: 1.75E+06
max: 1.82E+06
avg: 1.78E+06
max/avg: 1.02

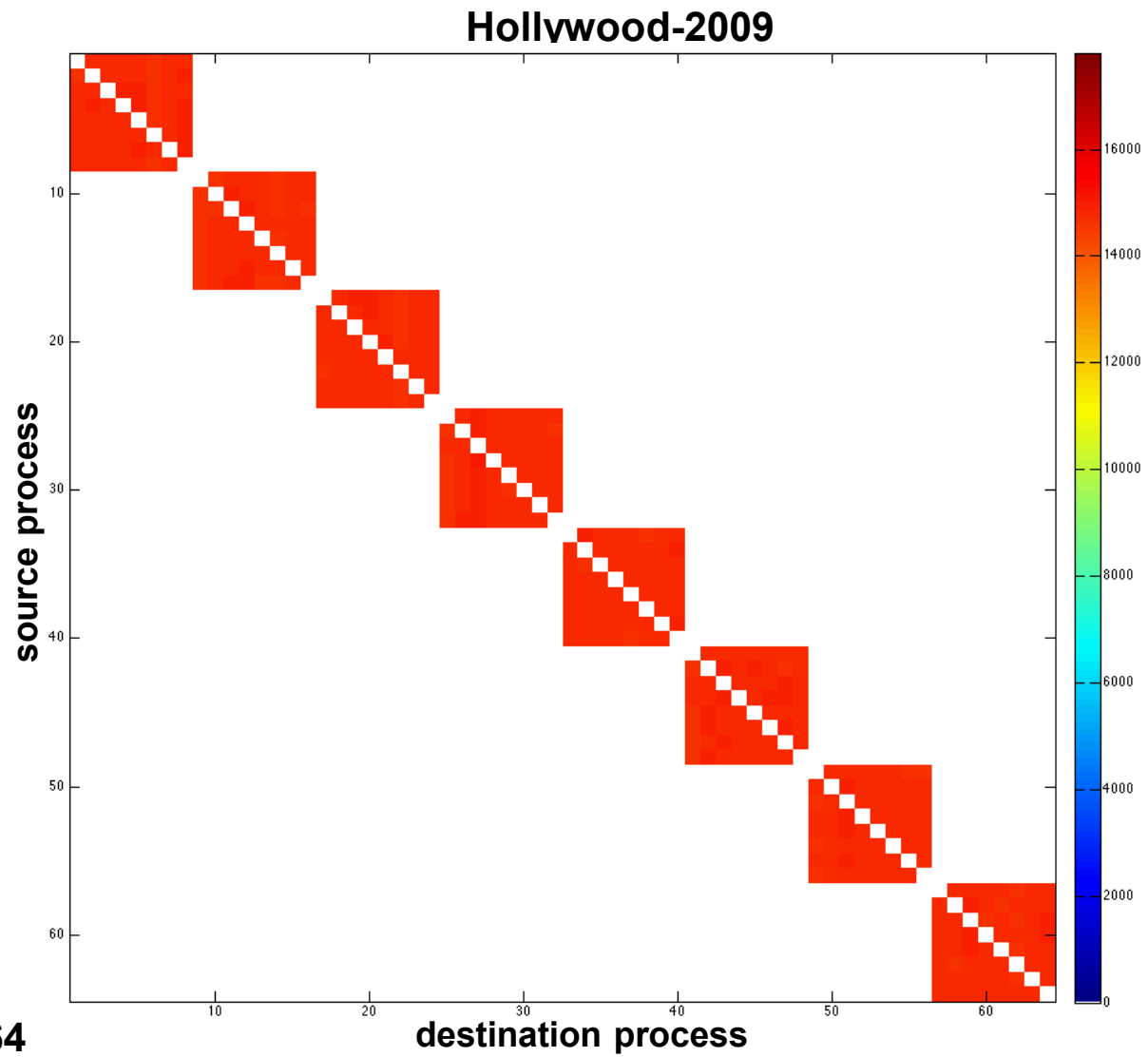
 # Messages (Phase 1)
total: 448
max: 7

Volume (Phase 1)
total: 6.63E+06
max: 1.04E+05

Nice properties:
No all-to-all communication
Total volume lower than 1DR

1DR = 1D Random

Communication Pattern: 2D Random Partitioning (Cartesian Blocks)



P=64

Number of Rows: 1.1M
Number of Nonzeros: 113.9M

NNZ/process
min: 1.75E+06
max: 1.82E+06
avg: 1.78E+06
max/avg: 1.02

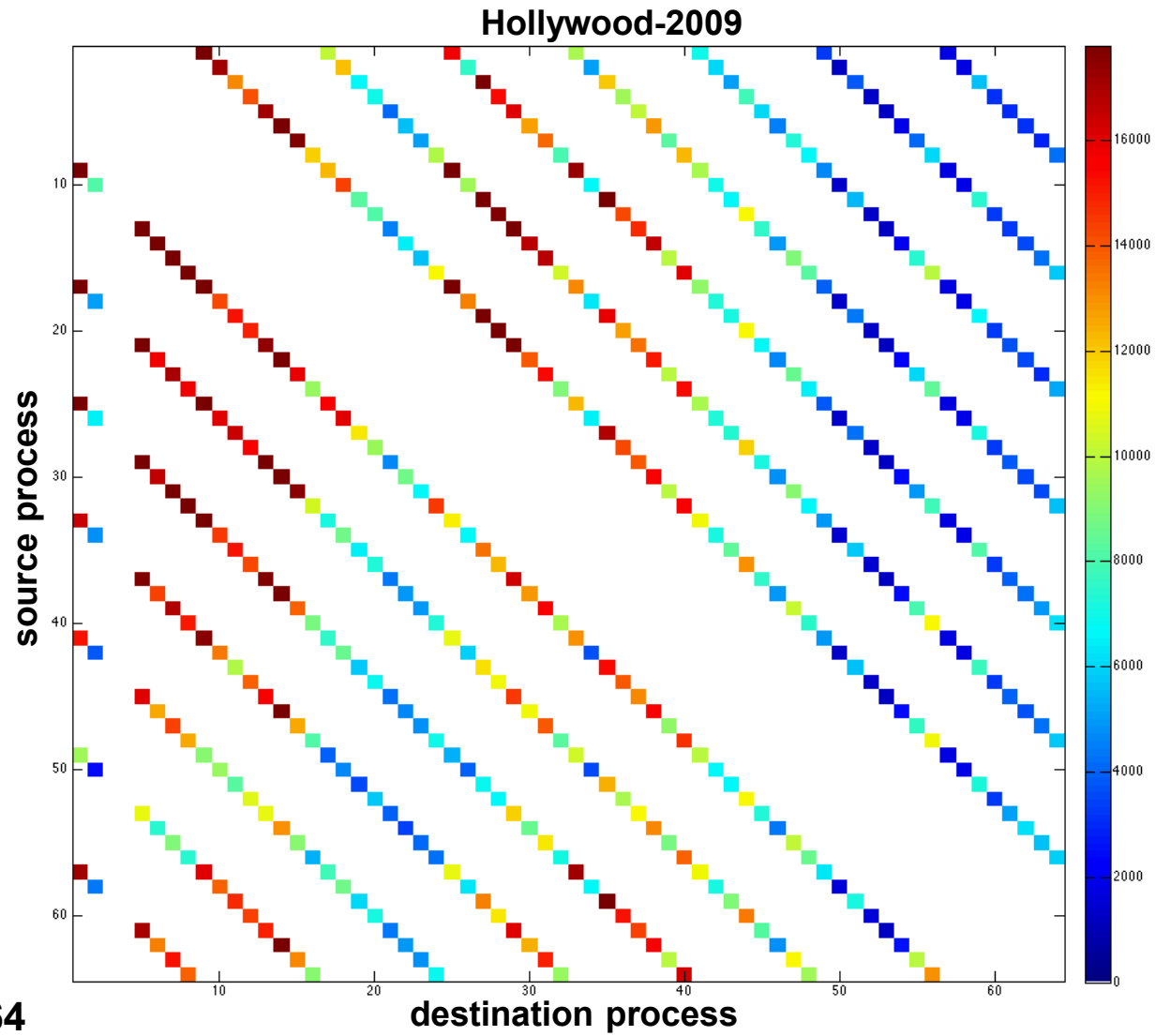
 # Messages (Phase 2)
total: 448
max: 7

Volume (Phase 2)
total: 6.63E+06
max: 1.04E+05

Nice properties:
No all-to-all communication
Total volume lower than 1DR

1DR = 1D Random

Communication Pattern: 2D Cartesian Hypergraph Partitioning



P=64

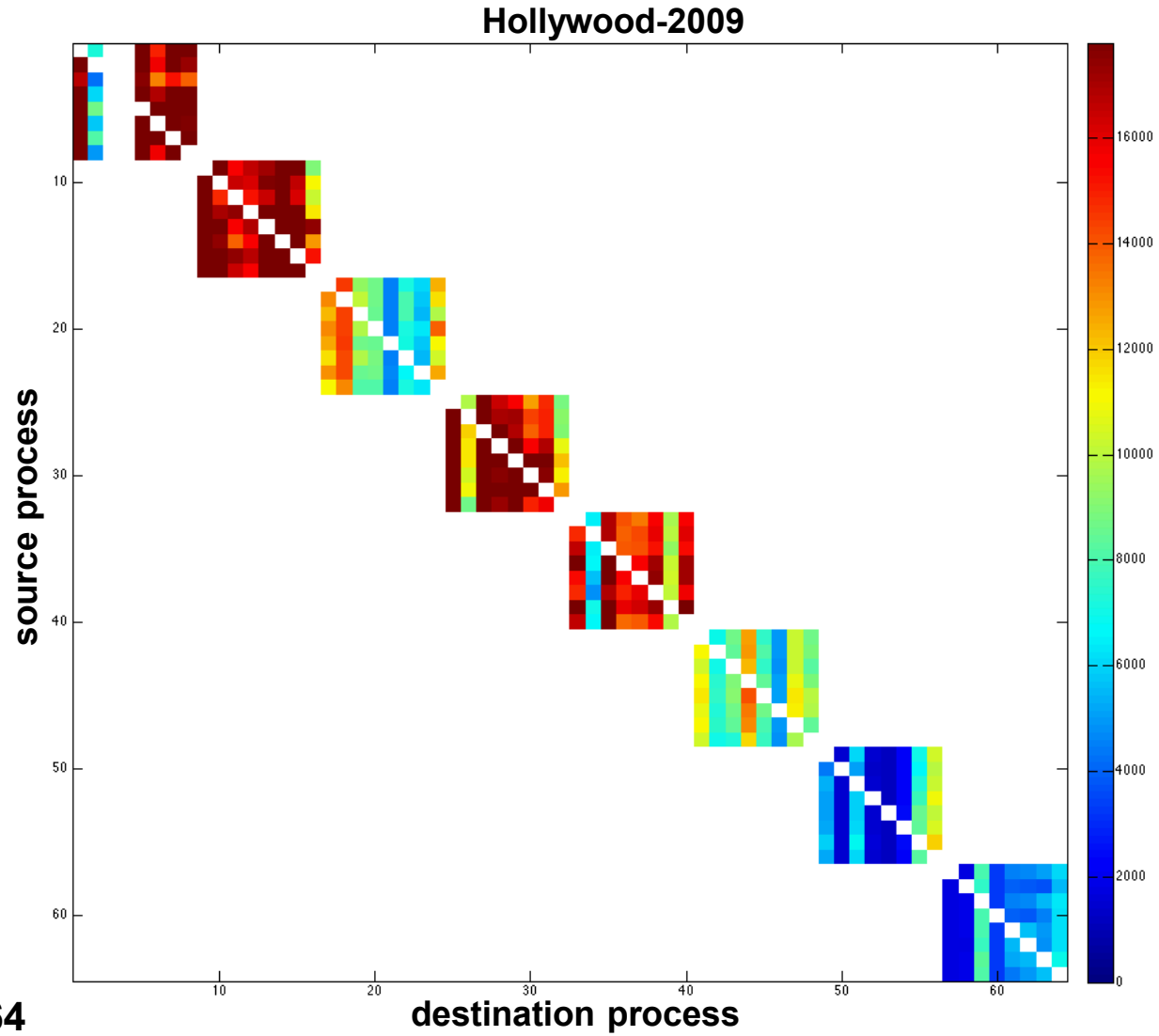
Number of Rows: 1.1M
Number of Nonzeros: 113.9M

NNZ/process
min: 5.16E+05
max: 2.61E+06
avg: 1.78E+06
max/avg: 1.47

 # Messages (Phase 1)
total: 439
max: 7

Volume (Phase 1)
total: 4.17E+06
max: 9.38E+04

Communication Pattern: 2D Cartesian Hypergraph Partitioning



P=64

Number of Rows: 1.1M
Number of Nonzeros: 113.9M

NNZ/process
min: 5.16E+05
max: 2.61E+06
avg: 1.78E+06
max/avg: 1.47

 # Messages (Phase 2)
total: 444
max: 7

Volume (Phase 2)
total: 5.06E+06
max: 1.49E+05

Parallel Implementation

Sandia National Laboratories

- Using Anasazi (Trilinos) Eigensolver
- 64 bit global ordinals
  - Necessary for graphs with 2^{31} vertices or more
- User defined operators
  - Modularity matrix
  - Moving average filter
  - Apply defined efficiently for particular operator
- Block Krylov-Schur method
  - Symmetric
  - Eigenvalues with largest real component
  - Blocksize=1

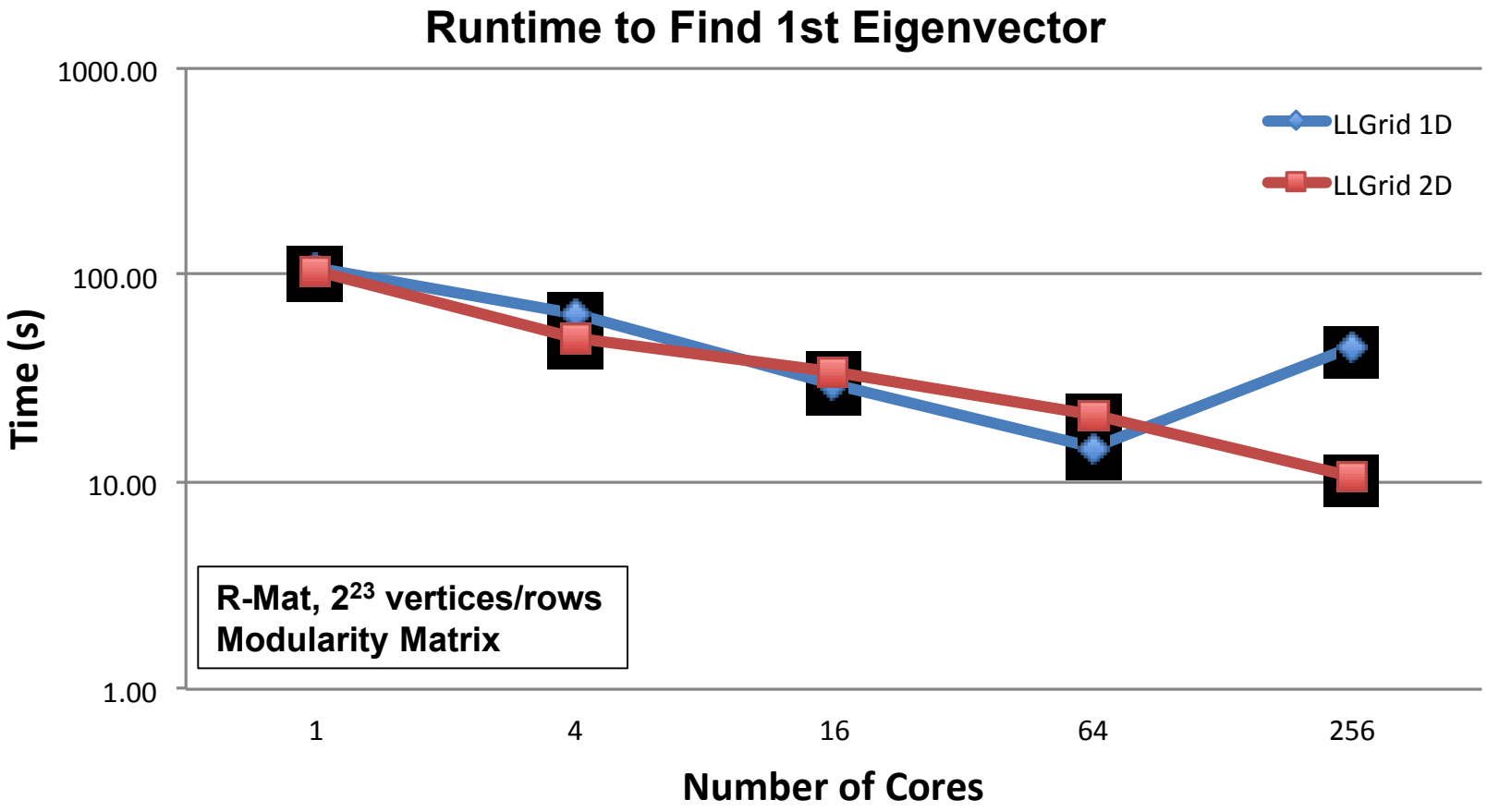
Initial Numerical Experiments

Sandia National Laboratories

- Matrices
  - R-Mat (a=0.5, b=0.125, c=0.125, d=0.25)
    - Average nonzeros per row: 8
    - Number of rows: 2^{22} to 2^{32}
- Two systems
  - LLGrid (MIT LL)
    - 274 compute nodes (8,768 cores)
    - Node: two 16-core AMD Opteron 6274 (2.2 GHz)
    - Network: 10 GB Ethernet
  - Hopper* (NERSC)
    - Cray XE6
    - 6,384 nodes (153,216 cores)
    - Node: two 12-core AMD 'MagnyCours' (2.1 GHz)
    - Network: 3D torus (Cray Gemini)
- Initially: 1D random row distribution (good load balance)

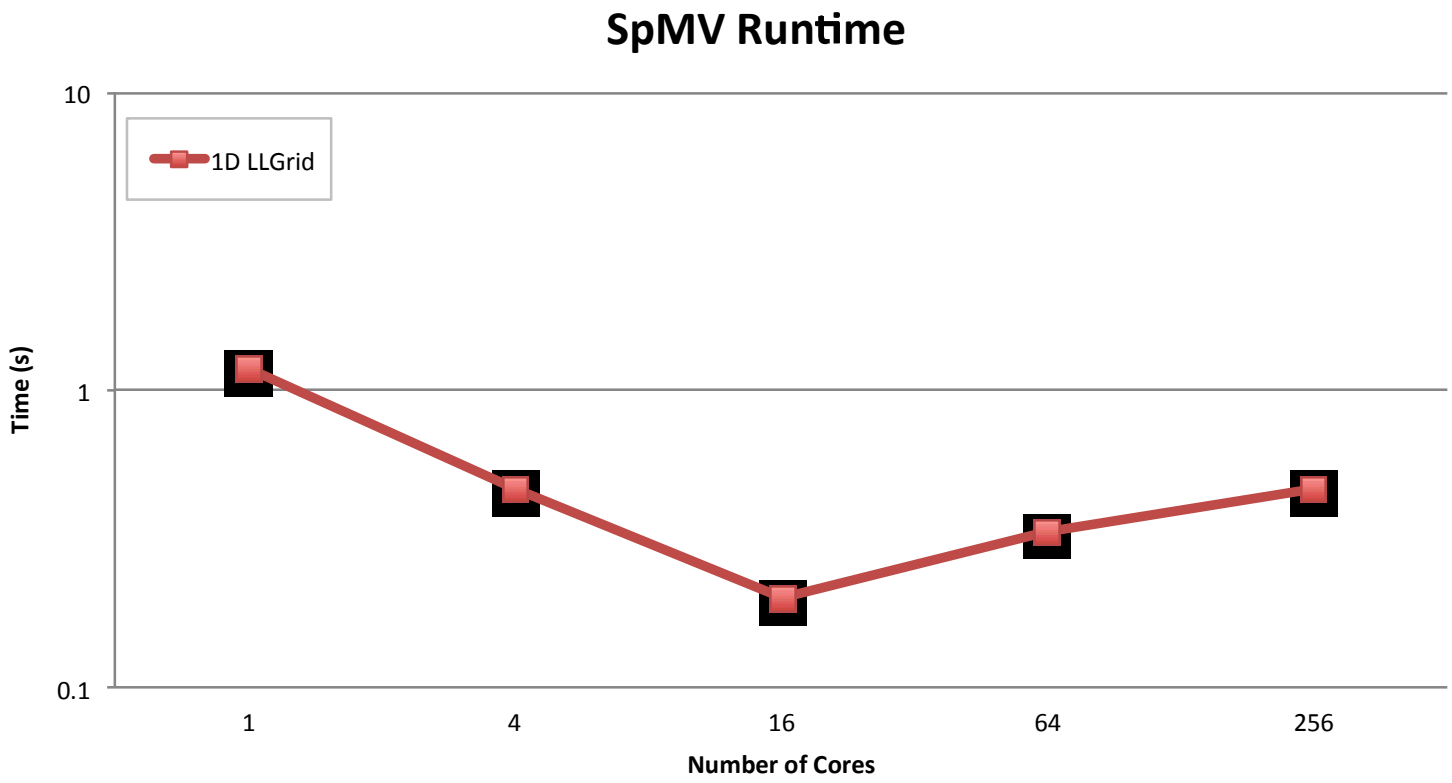
* This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Improved Results – LLGrid



Simple 2D method shows improved scalability

Strong Scaling Results: Hollywood

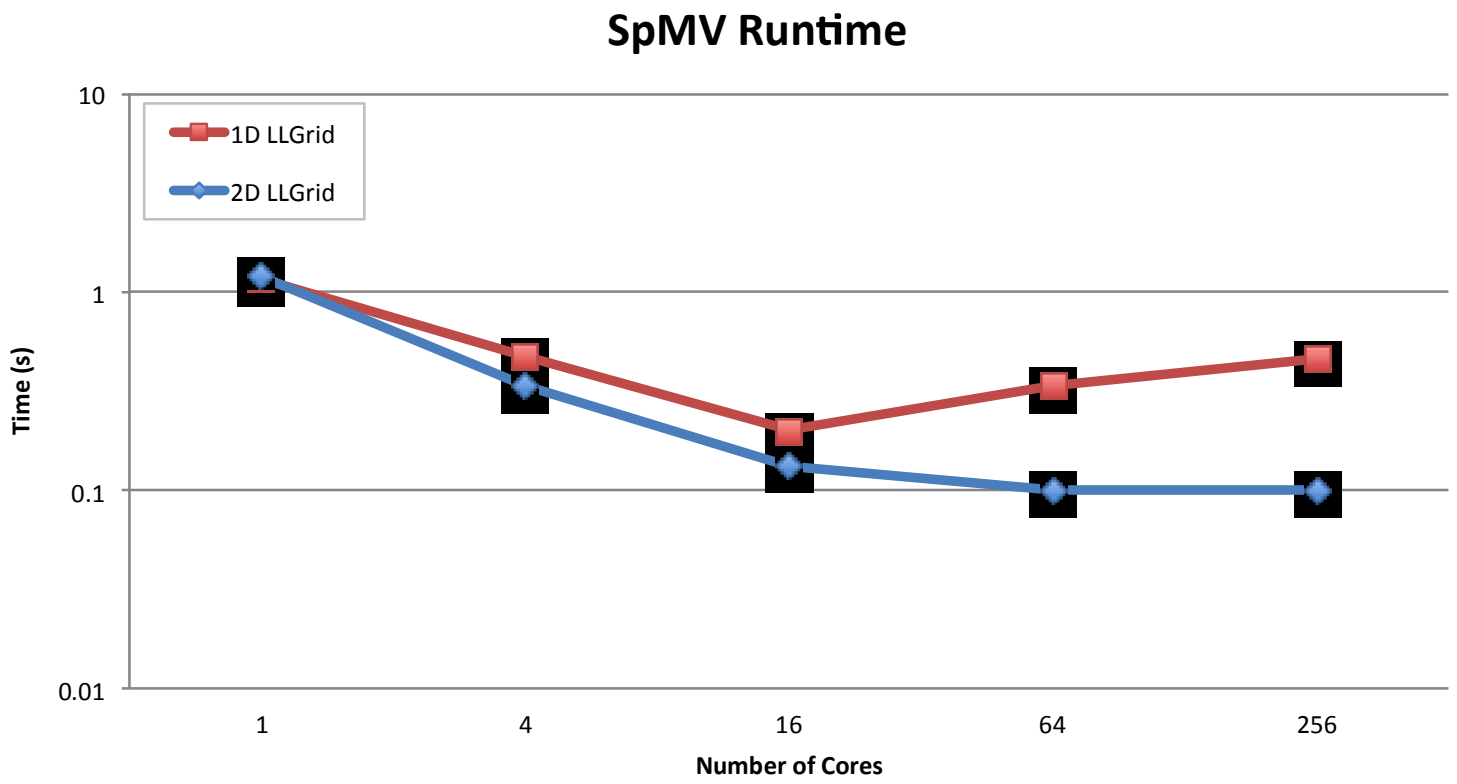


1D random partitioning

Scalability limited and runtime increases for large numbers of cores

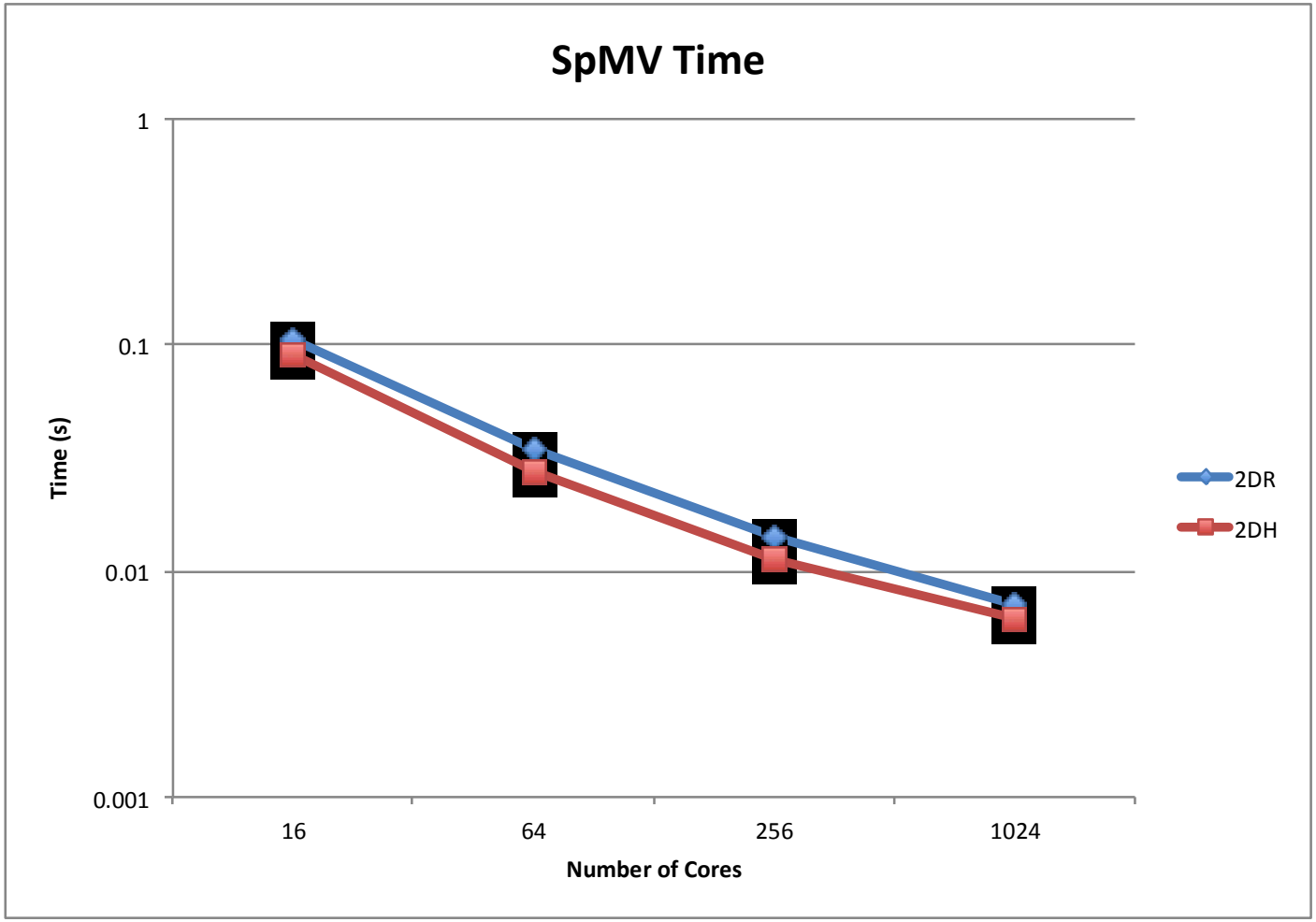
SpMV: Strong Scaling Results - Hollywood

Sandia National Laboratories



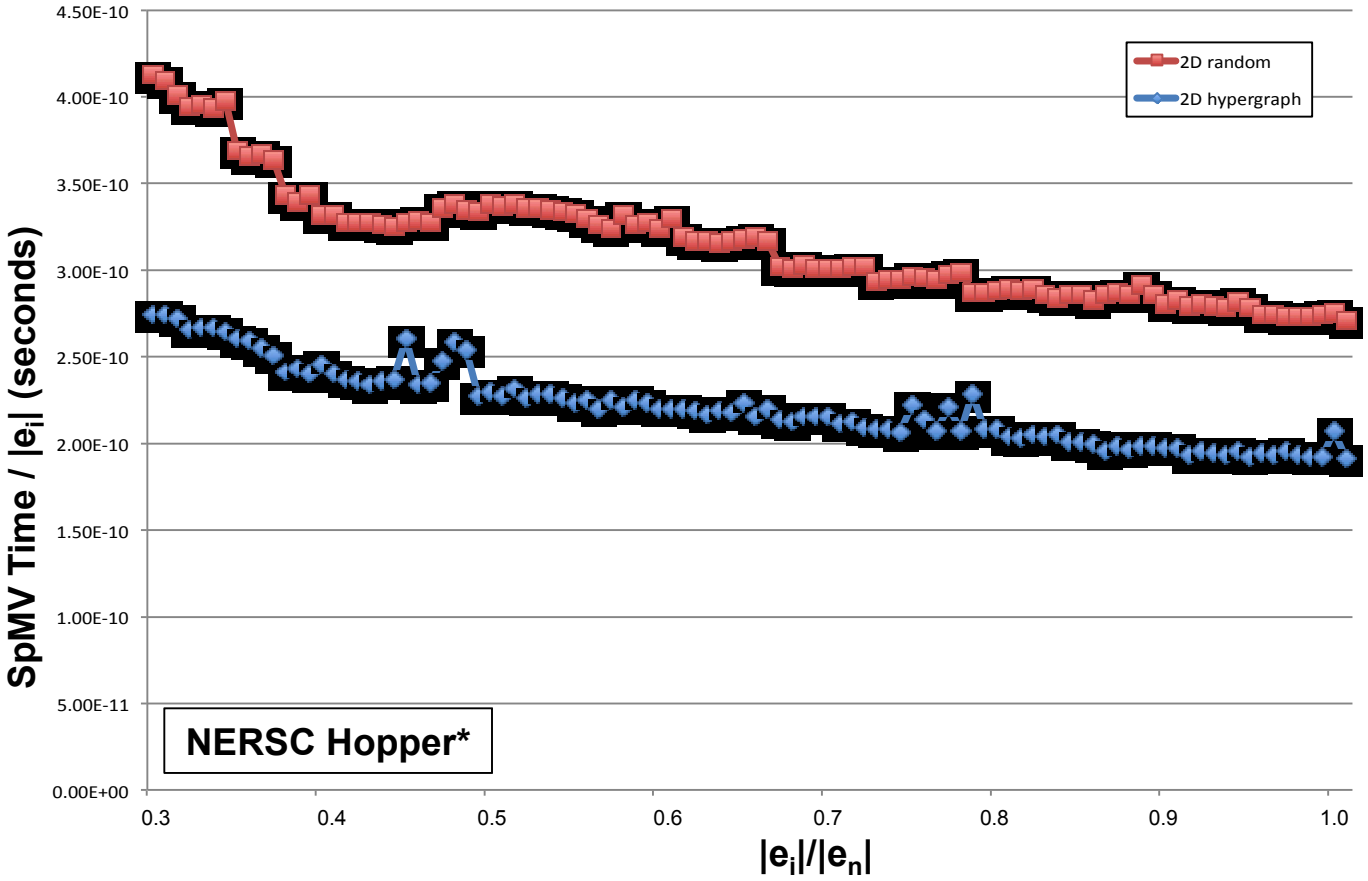
Simple 2D method shows improved scalability

SpMV: Strong Scaling Results - Hollywood



Hypergraph partitioning further reduces runtime

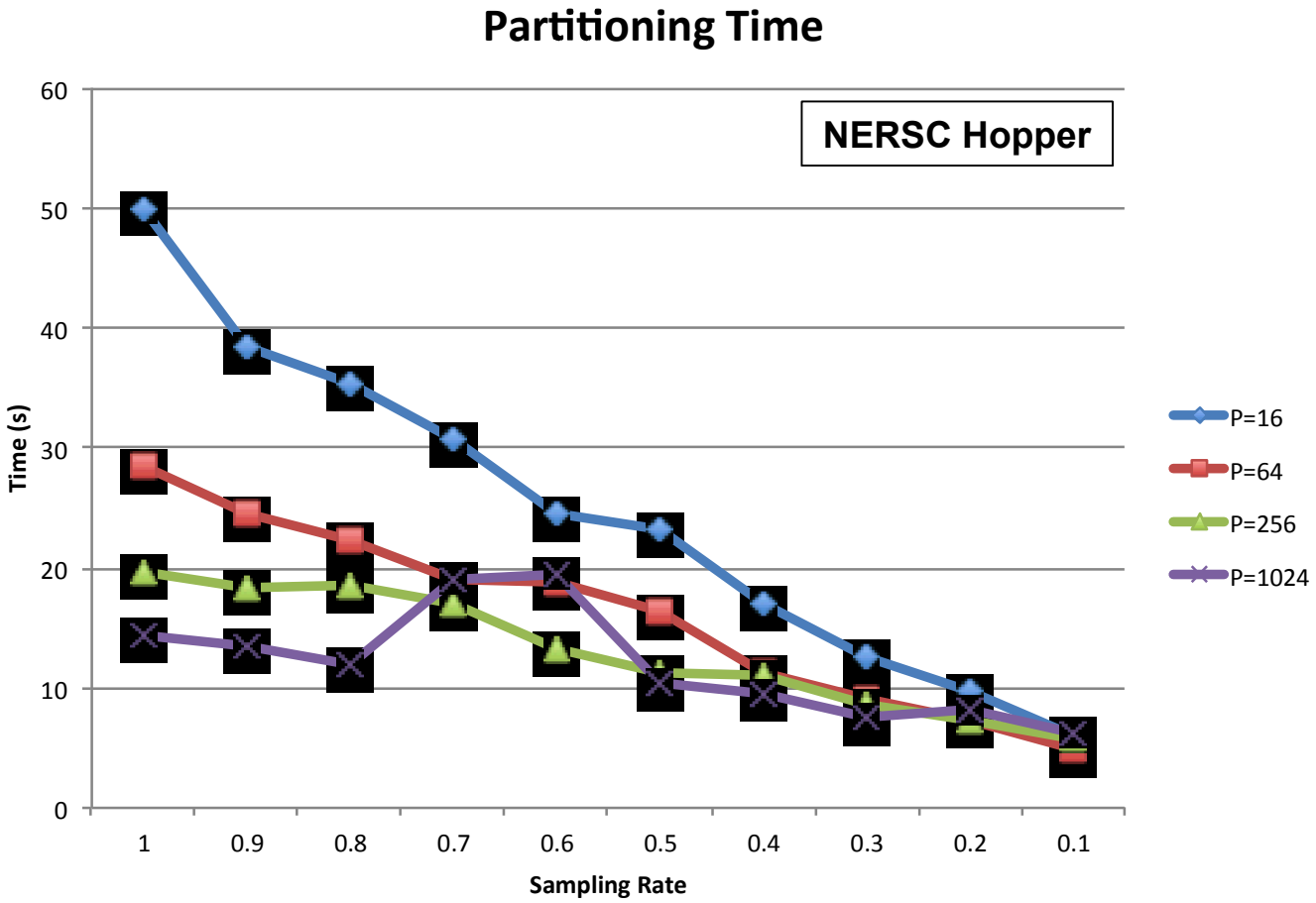
Results: Partitioning for Dynamic Graphs



- $|e_0| = 0.3\ |e_n|$
- 2D hypergraph surprisingly effective as edges are added to graph

* This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

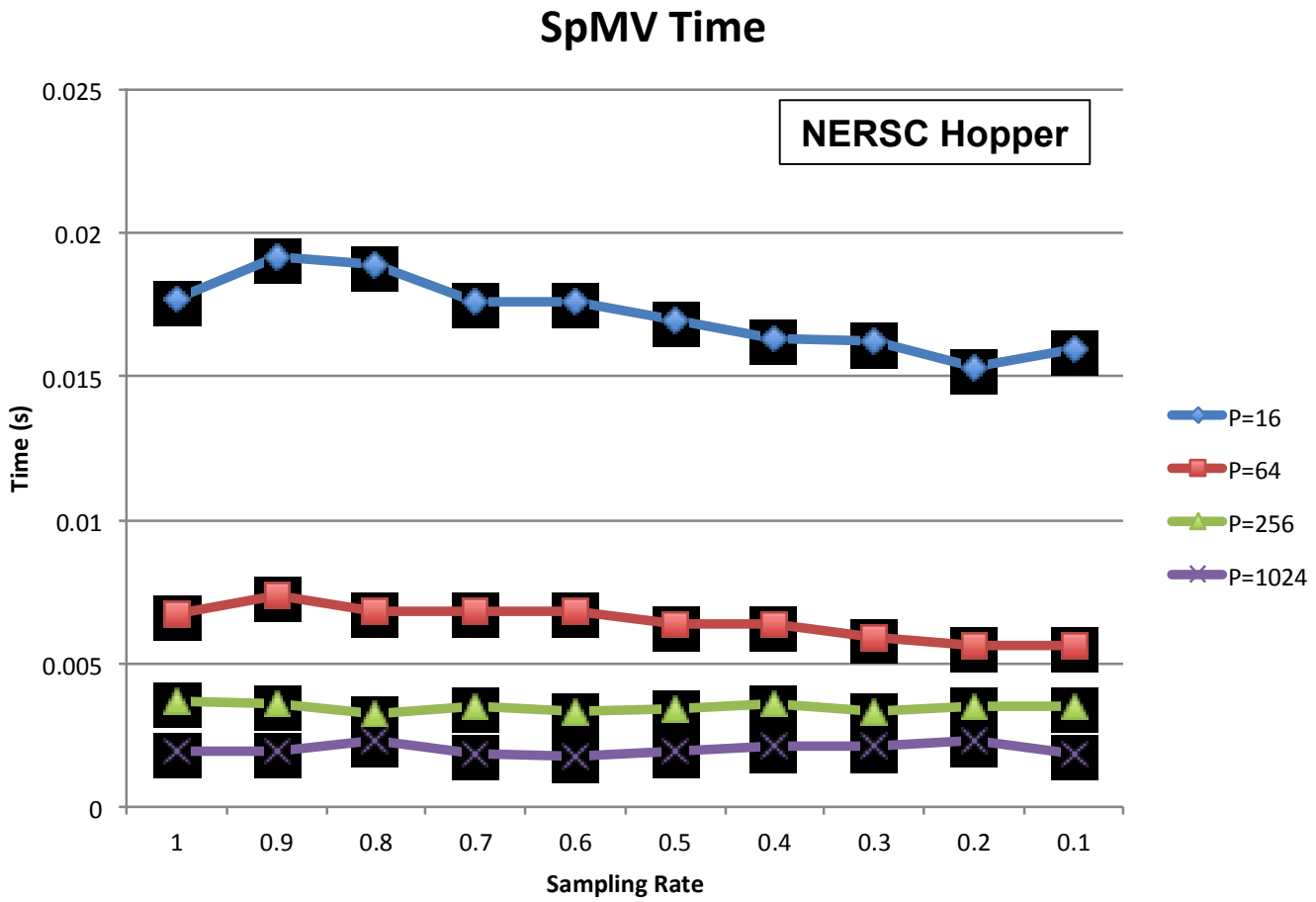
com-youtube* Graph



Edge sampling greatly reduces partitioning time

*SNAP: J. Yang and J. Leskovec. Defining and Evaluating Network Communities based on Ground-truth. ICDM, 2012.

com-youtube* Graph



Resulting SpMV time does not increase for modest sampling

*SNAP: J. Yang and J. Leskovec. Defining and Evaluating Network Communities based on Ground-truth. ICDM, 2012.