Partitioning for Parallel Sparse Matrix-Vector Multiplication

August 7, 2007

Michael Wolf
University of Illinois at Urbana-Champaign
(Org. 1415)

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy's National Nuclear Security Administration
under contract DE-AC04-94AL85000.

Parallel Computing

- Motivation: large scientific problems
  - Memory on single processor too small
  - Runtime too long
- Need to distribute data across multiple processors
- Parallel sparse matrix-vector multiplication
  - Distribute matrices
  - Distribute vectors

Parallel Matrix-Vector Multiplication

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

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

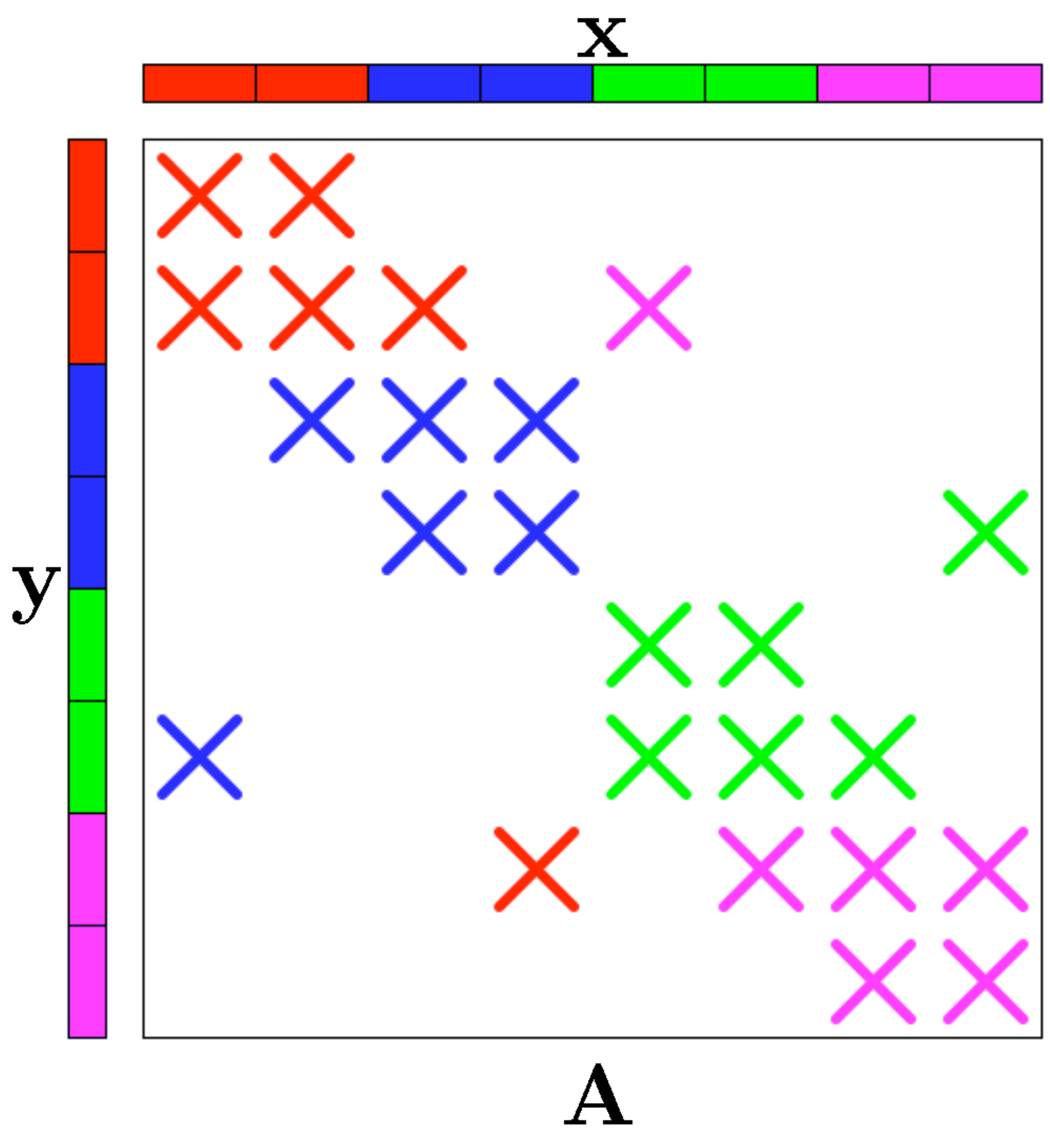
- Vectors partitioned identically

Sandia National Laboratories

Objective

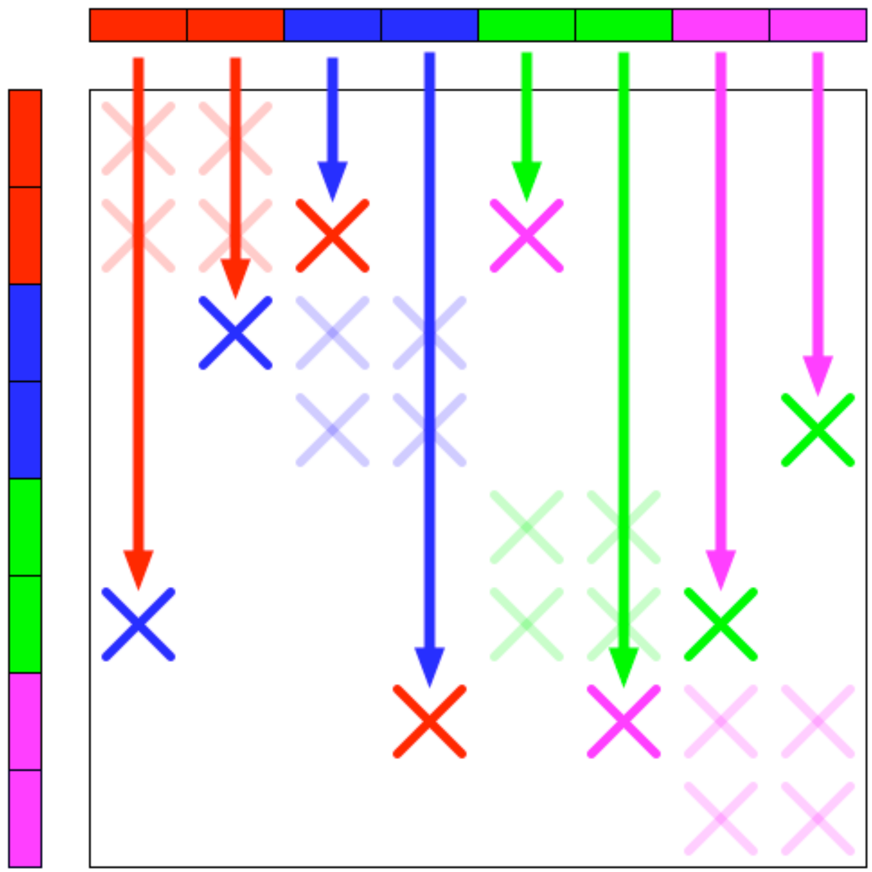
- Ideally we minimize total run-time
- Settle for easier objective
  - Work balanced
  - Minimize total communication volume
- Can partition matrices in different ways
  - 1-D
  - 2-D
- Can model problem in different ways
  - Graph
  - Bipartite graph
  - Hypergraph

Parallel Matrix-Vector Multiplication



Sandia National Laboratories

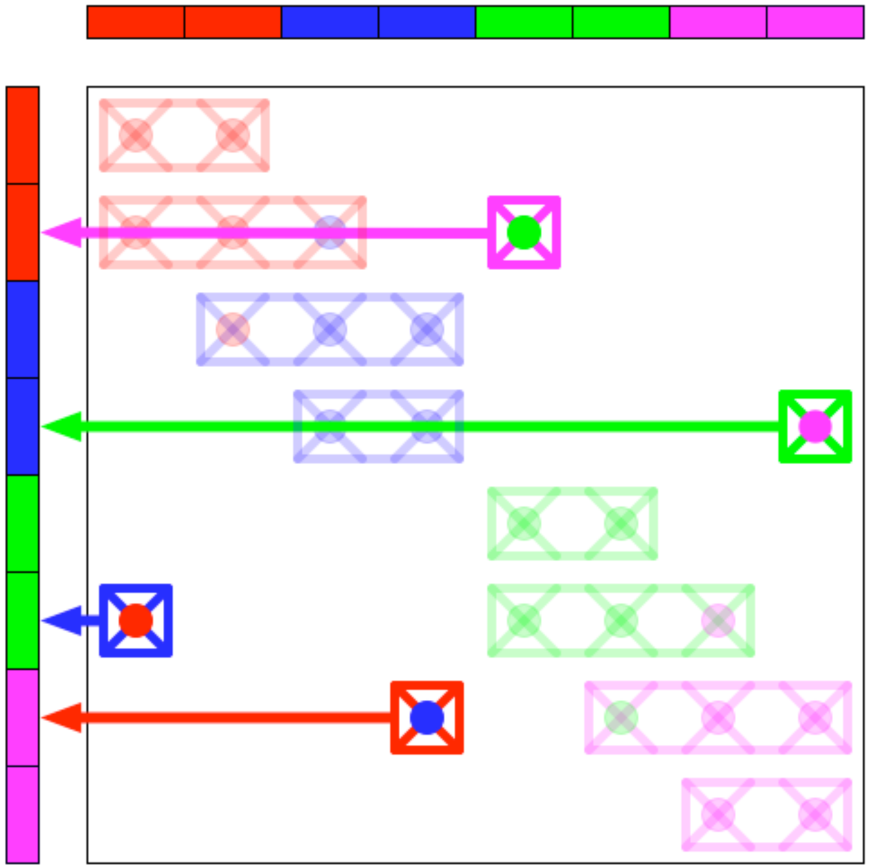
Parallel Mat-Vec Multiplication Communication



"fan-out"

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

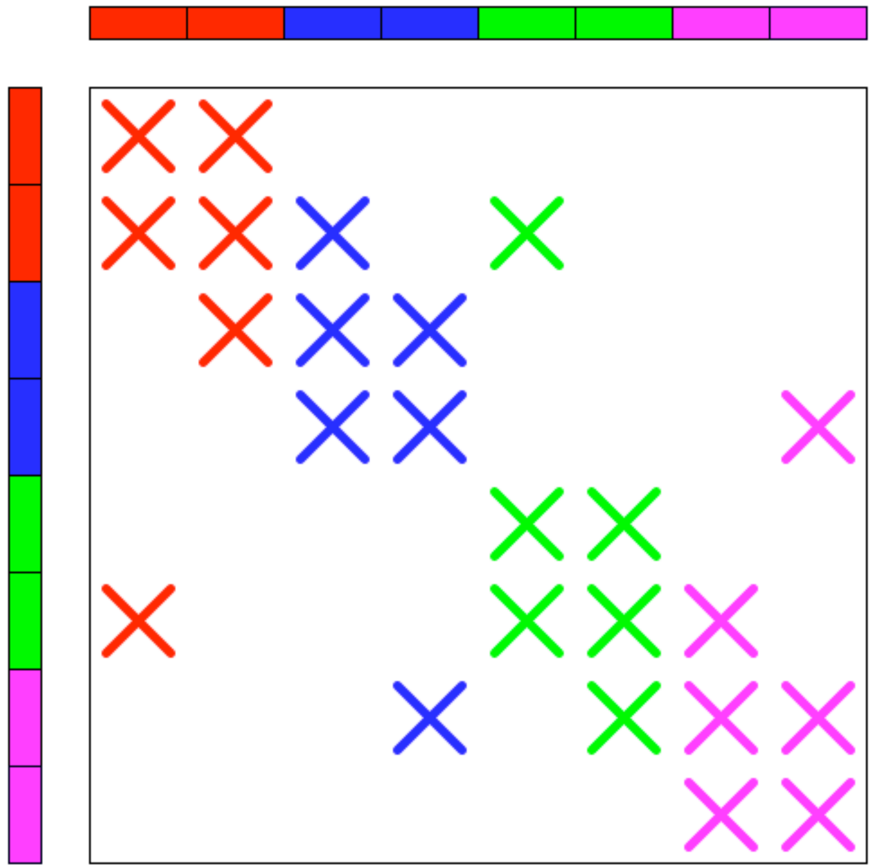
Parallel Mat-Vec Multiplication Communication



“fan-in”

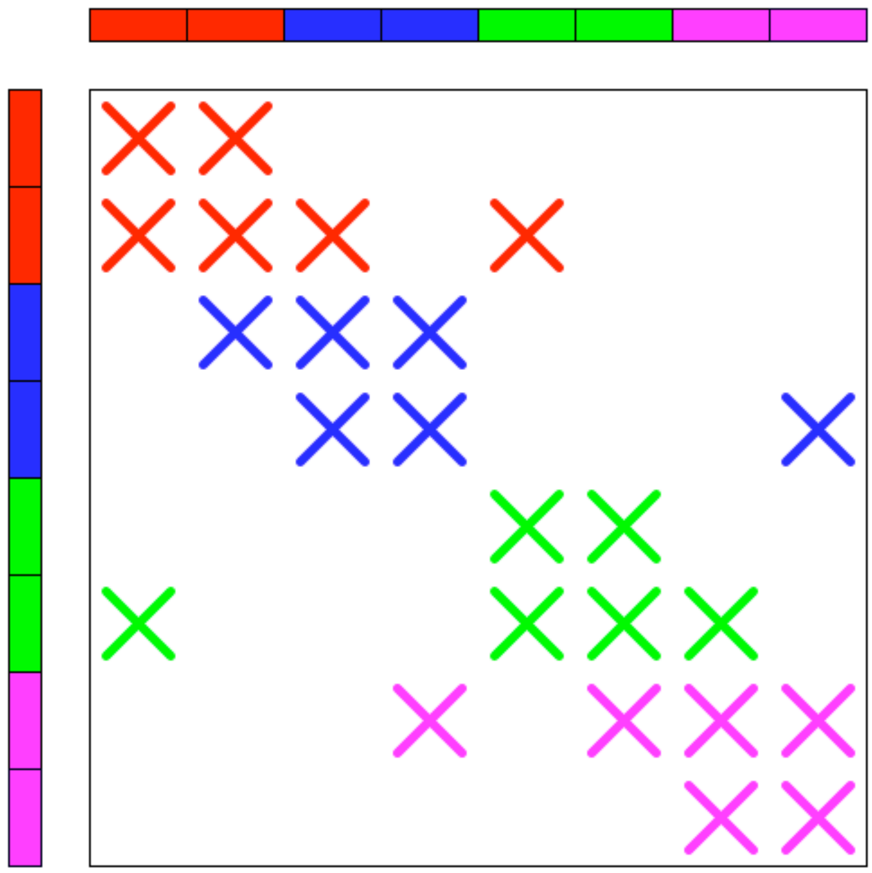
- Send partial inner-products to process that owns corresponding vector element y_i

1-D Column Partitioning



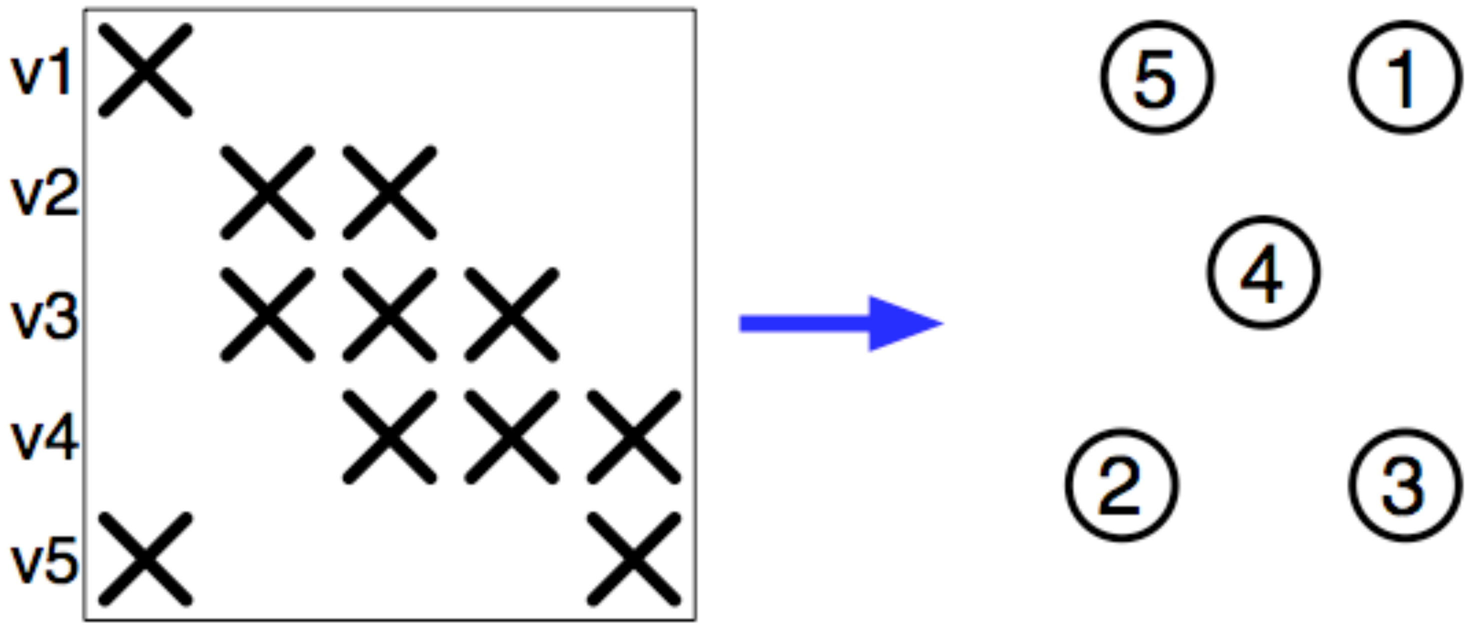
- Each process assigned nonzeros for set of columns

1-D Row Partitioning



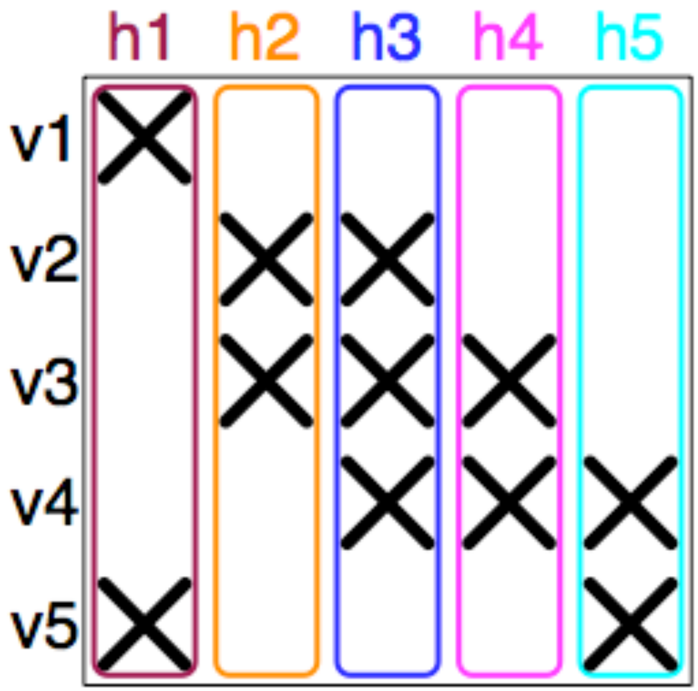
- Each process assigned nonzeros for set of rows

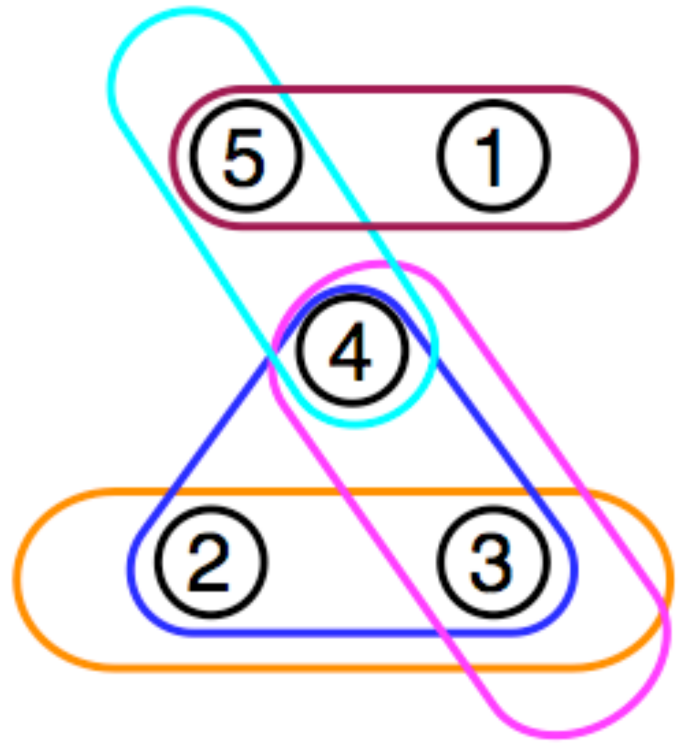
Hypergraph Model of 1-D (row) Partitioning



- Nonzero pattern can be unsymmetric
- Rows represented by vertices in hypergraph

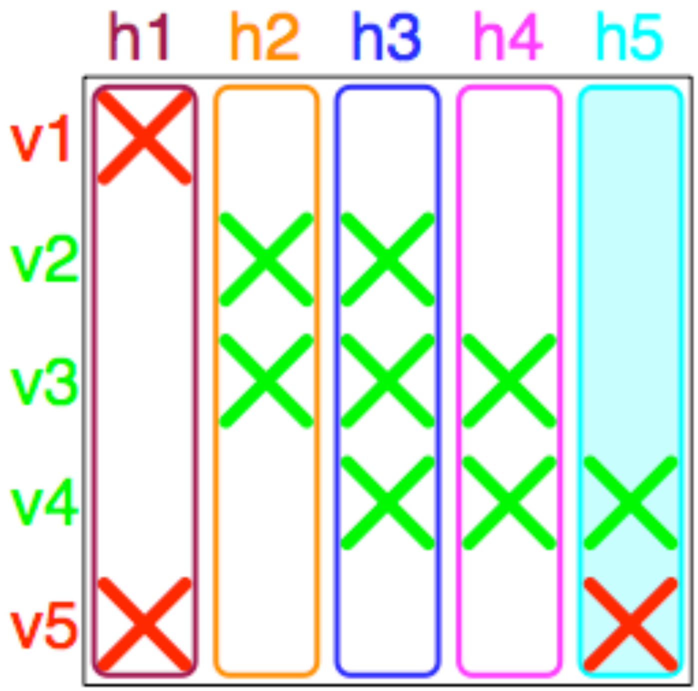
Hypergraph Model of 1-D (row) Partitioning



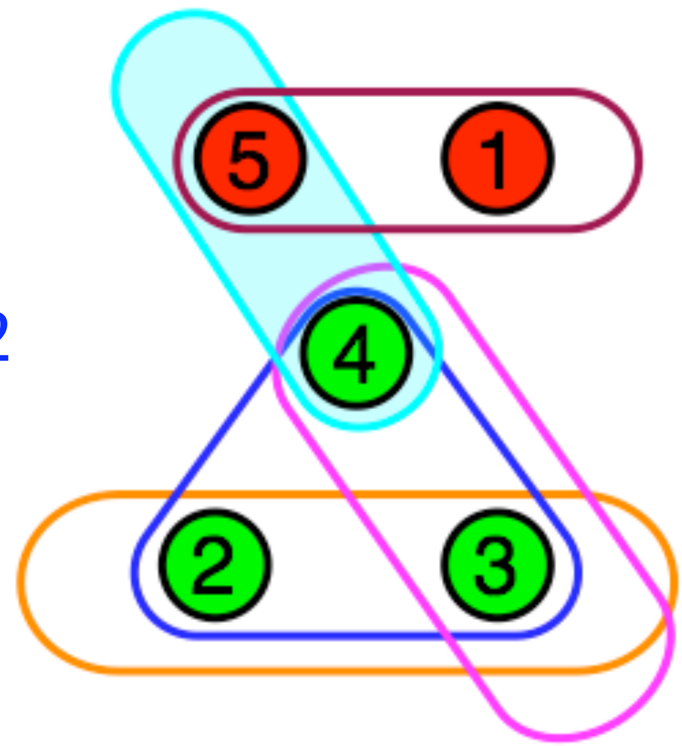


- Columns represented by hyperedges in hypergraph

Hypergraph Model of 1-D (row) Partitioning

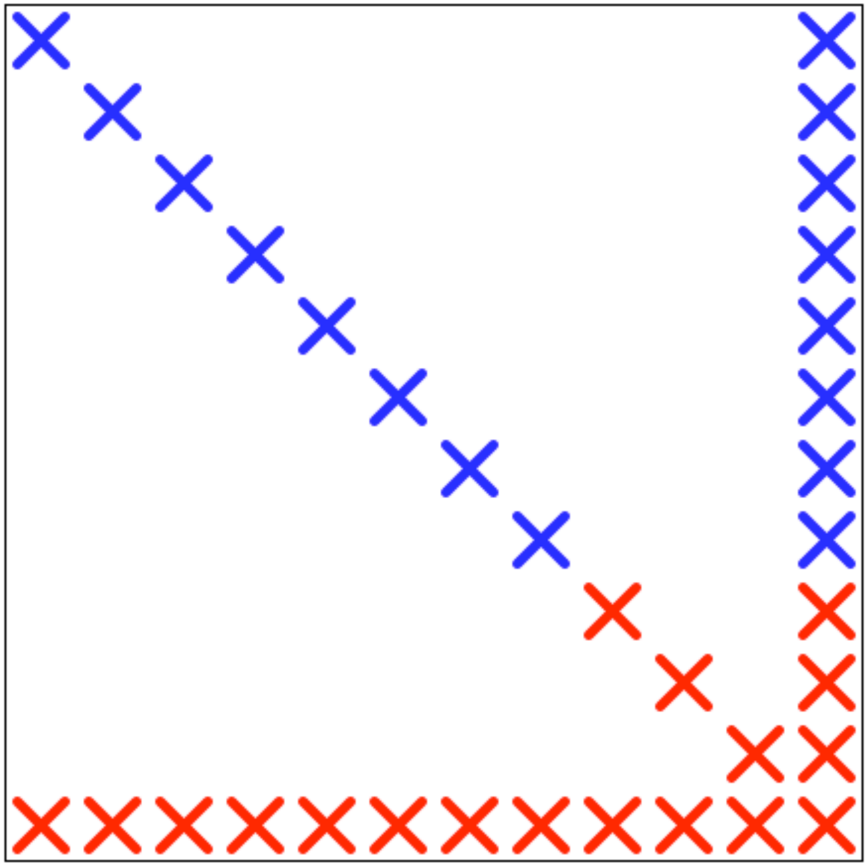


k=2



- Partition vertices into k equal sets
- Hyperedge cut = communication volume
  - Aykanat and Catalyurek (1996)
- NP-hard to solve optimally

When 1-D Partitioning is Inadequate



"Arrowhead" matrix

n=12

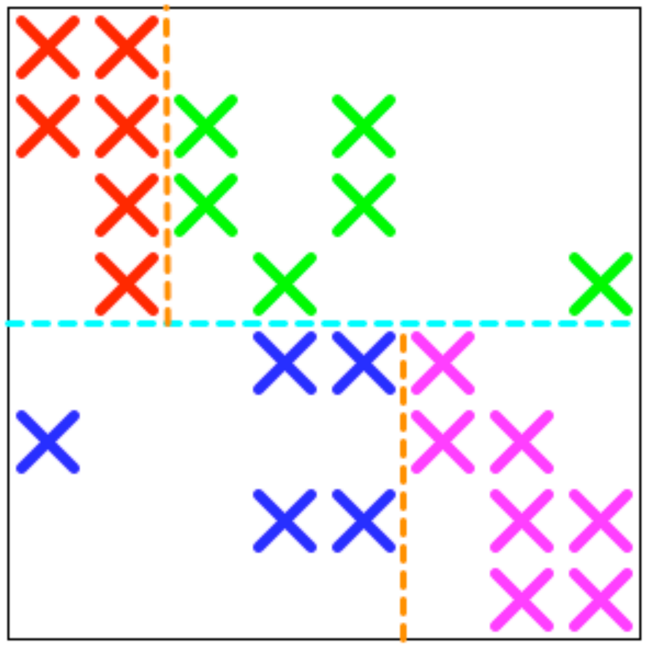
nnz=30

volume = 9

- For nxn matrix for any 1-D bisection:
  - nnz = 3n-2
  - Volume ≈ 3/4*n

Sandia National Laboratories

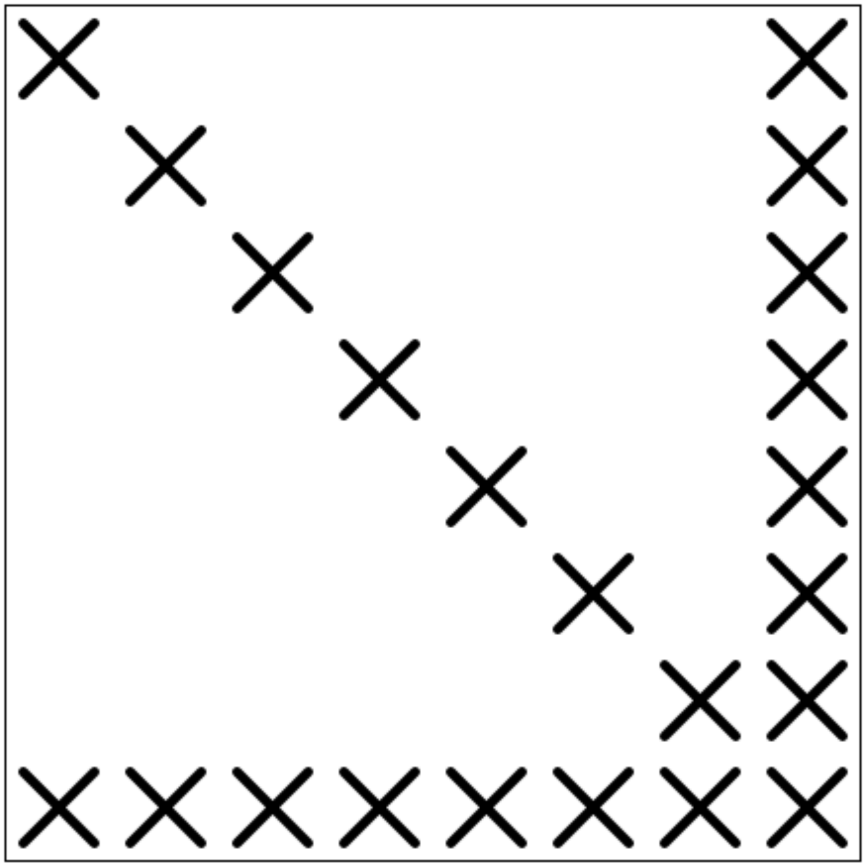
2-D Partitioning Methods



Mondriaan

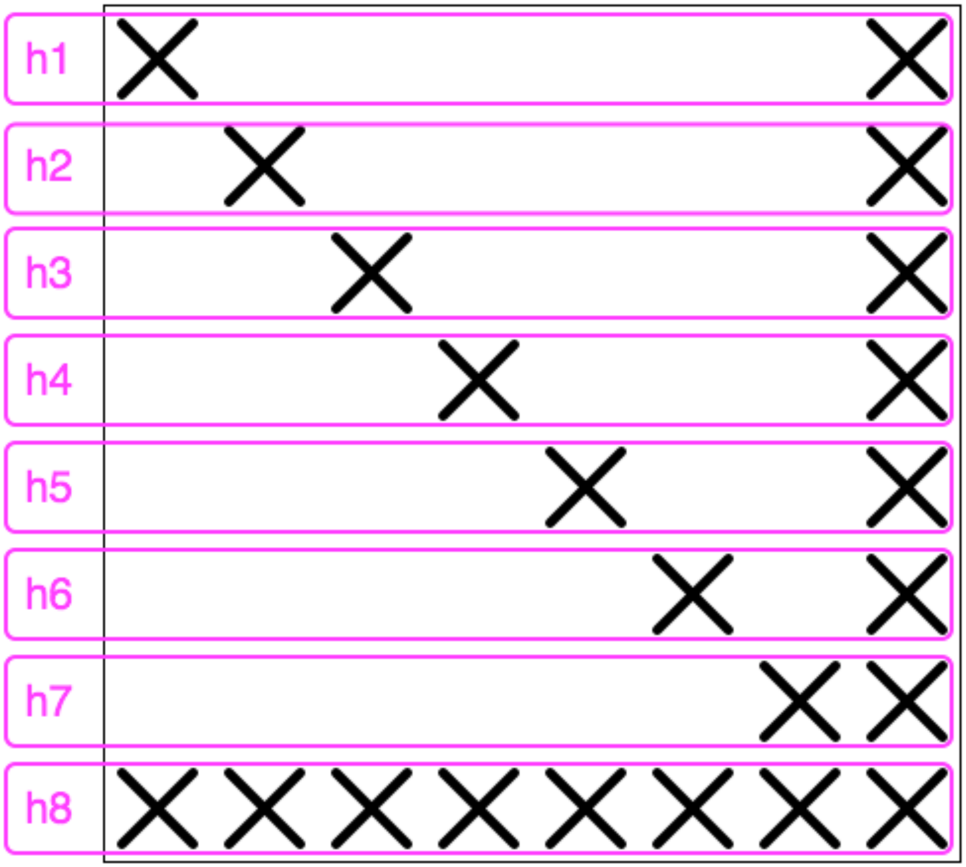
- More flexibility in partitioning
- Mondriaan
  - Fairly fast
  - Generally gives good partitions

2-D Method: Fine-grain Hypergraph Model



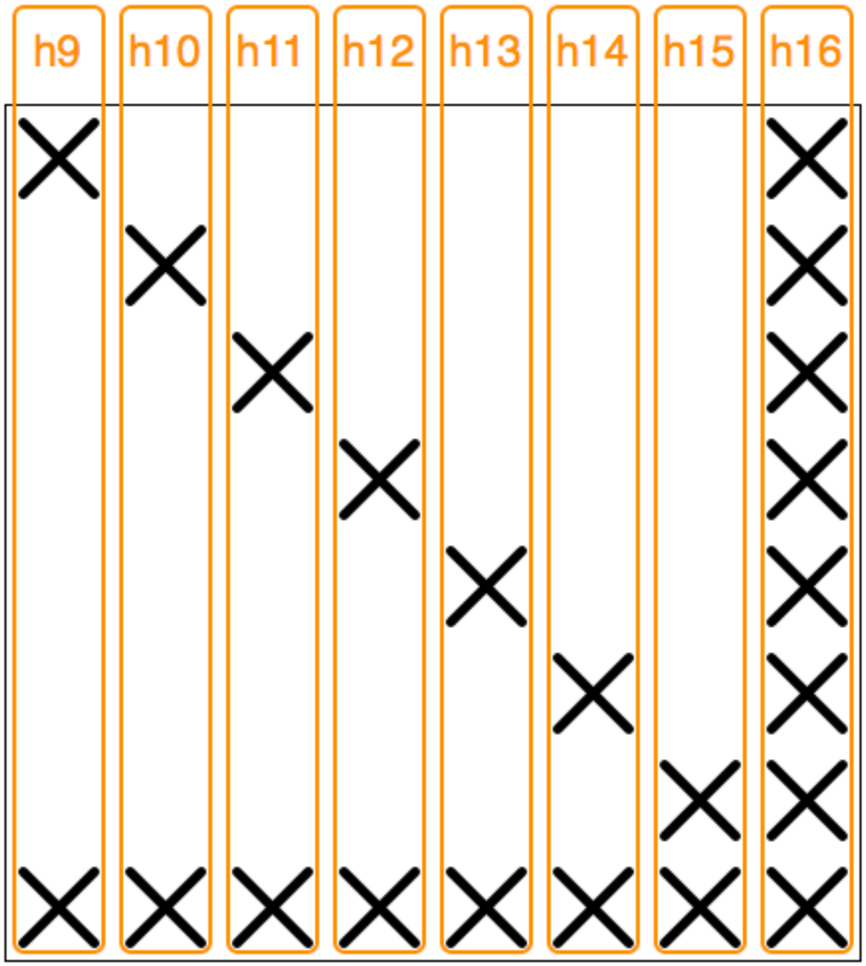
- Catalyurek and Aykanat (2001)
- Assign each nz separately
- Nonzeros represented by vertices in hypergraph

2-D Method: Fine-grain Hypergraph Model



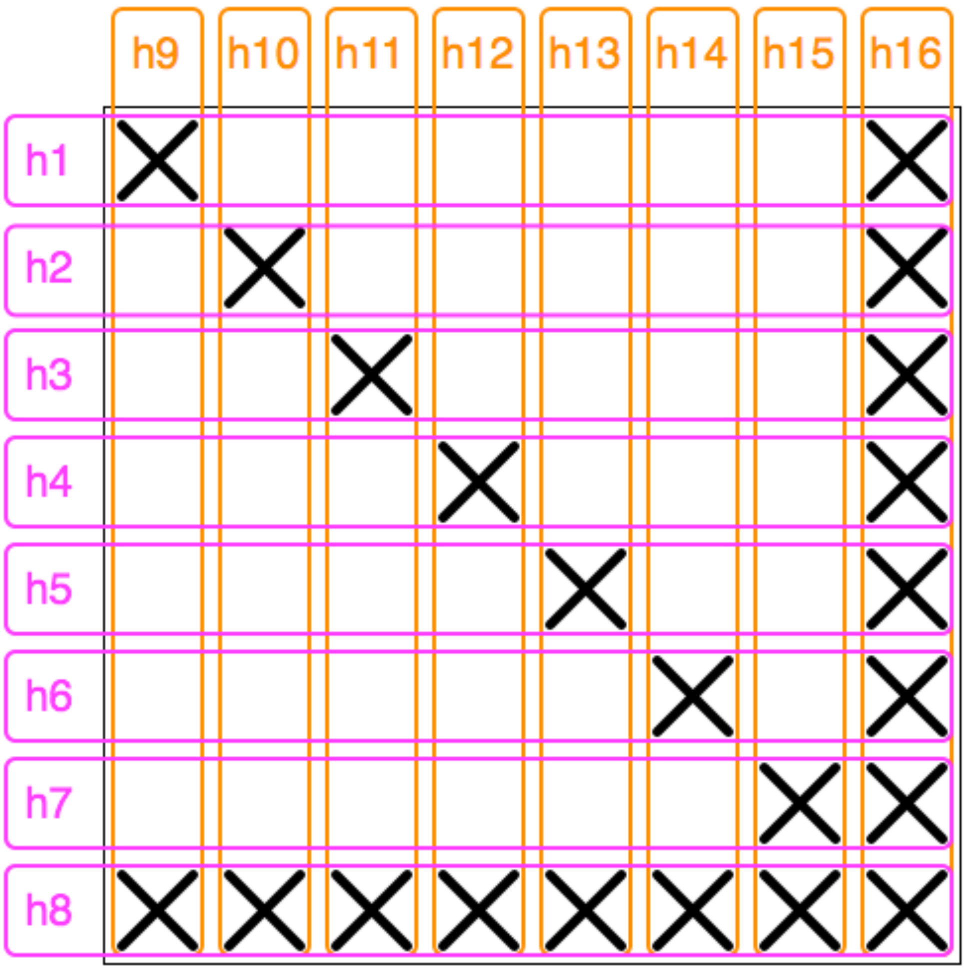
- Rows represented by hyperedges

2-D Method: Fine-grain Hypergraph Model



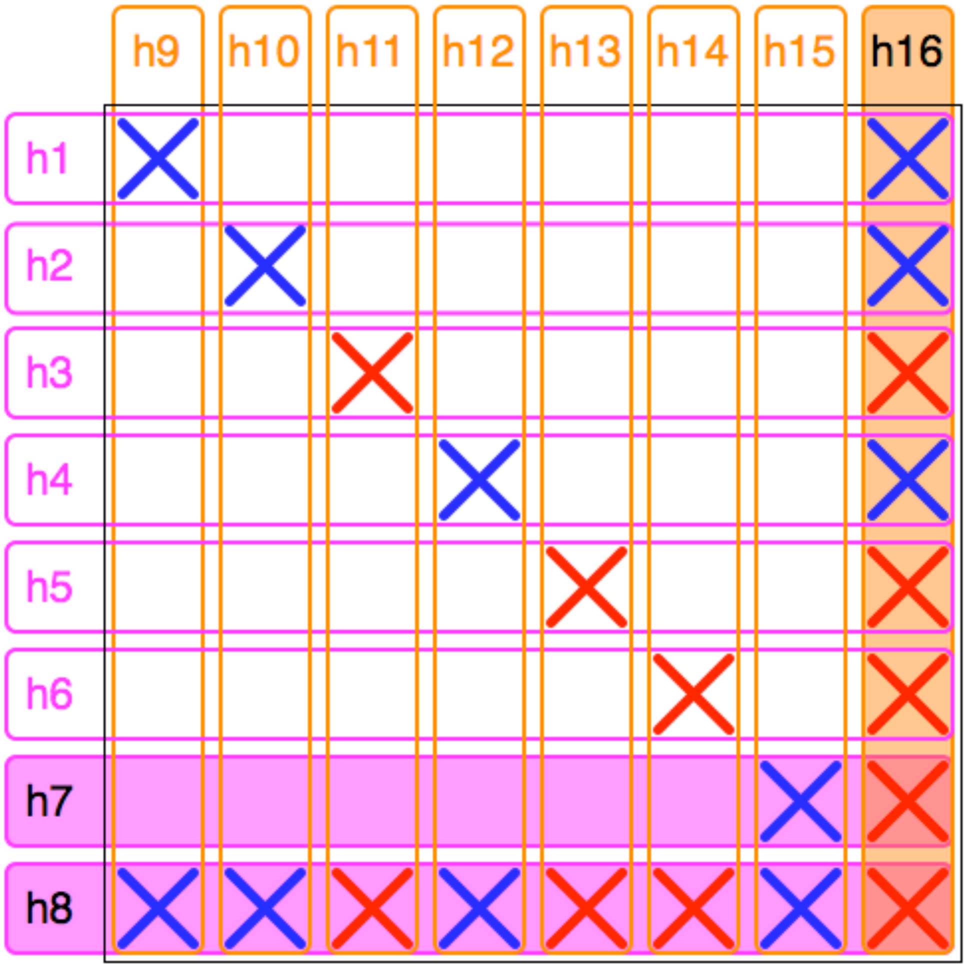
- Columns represented by hyperedges

2-D Method: Fine-grain Hypergraph Model



- 2n hyperedges

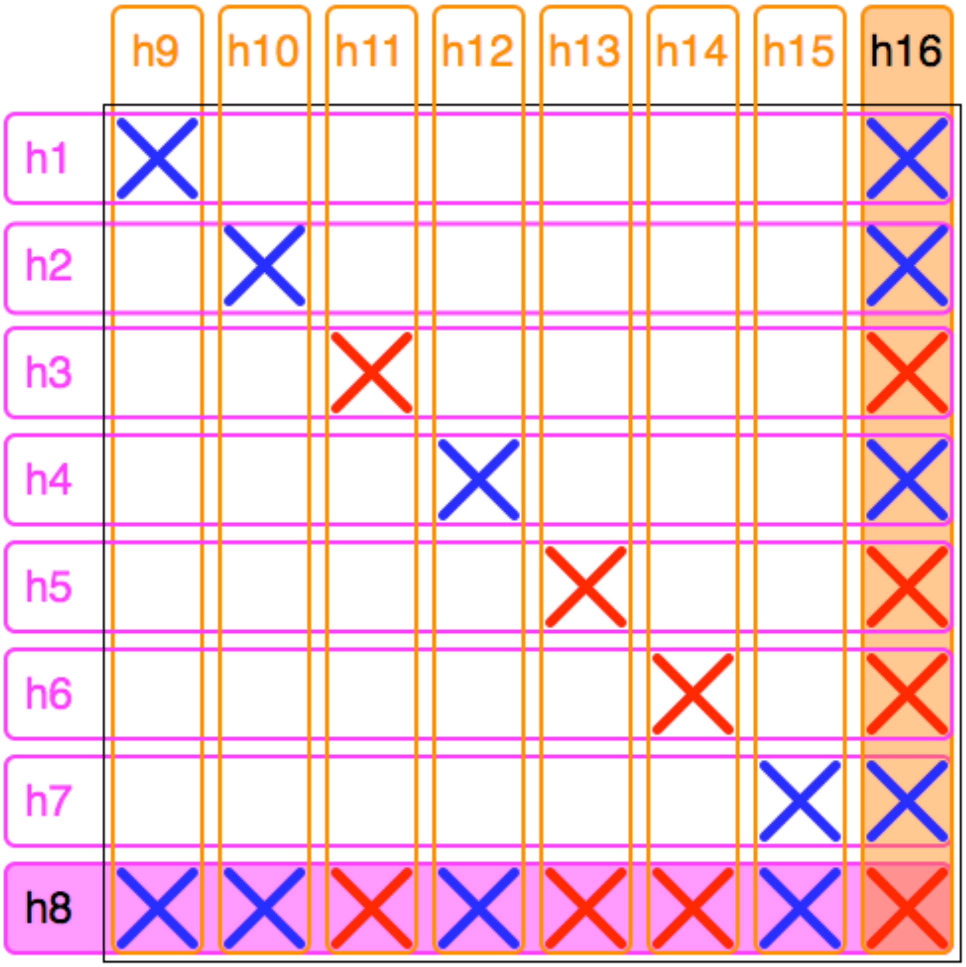
2-D Method: Fine-grain Hypergraph Model



k=2, volume = 3

- Partition vertices into k equal sets
- Volume = hypergraph cut
- Minimum volume partition when optimally solved
- Larger NP-hard problem

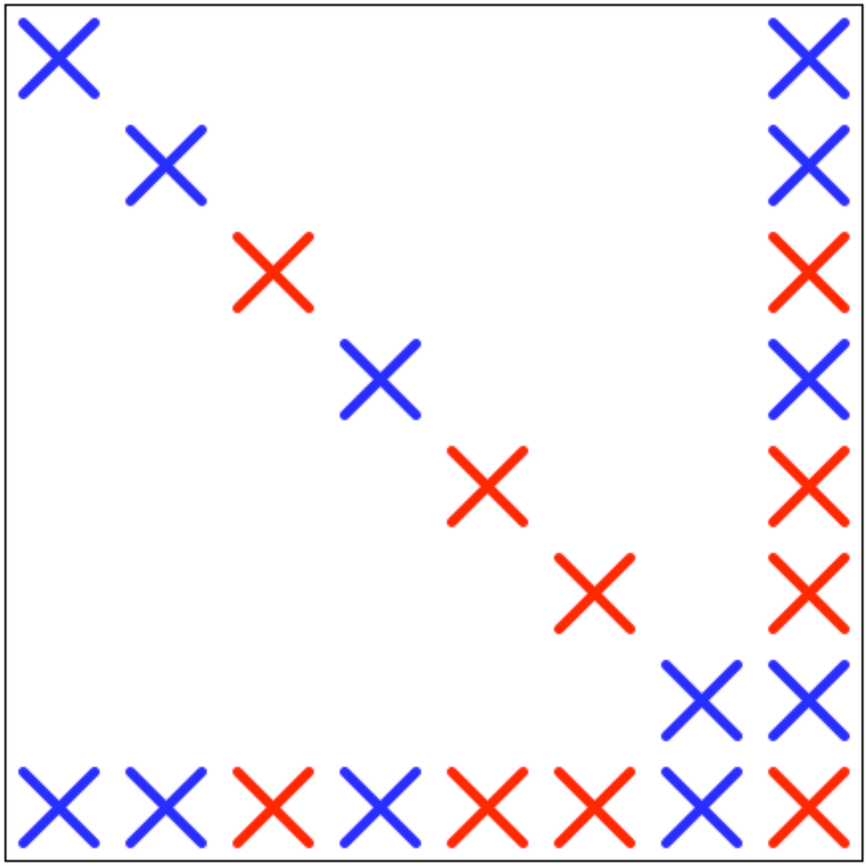
2-D Method: Fine-grain Hypergraph Model



Volume = 2

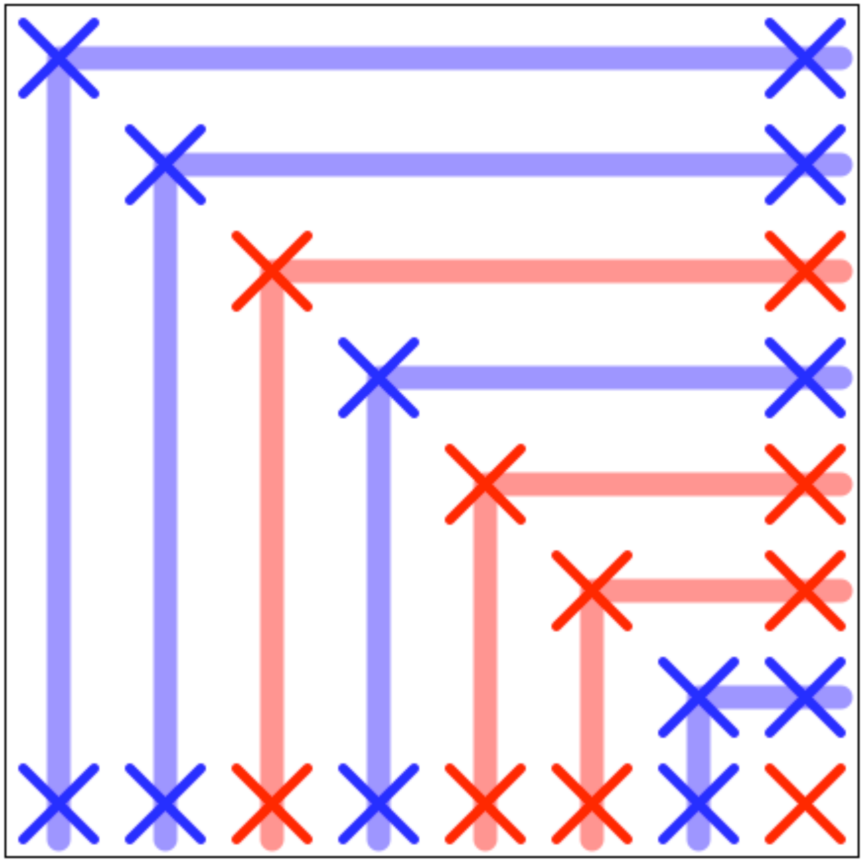
- Loosening load-balancing restriction we can obtain a nontrivial partition of minimum cut

New 2-D Method: “corner” partitioning



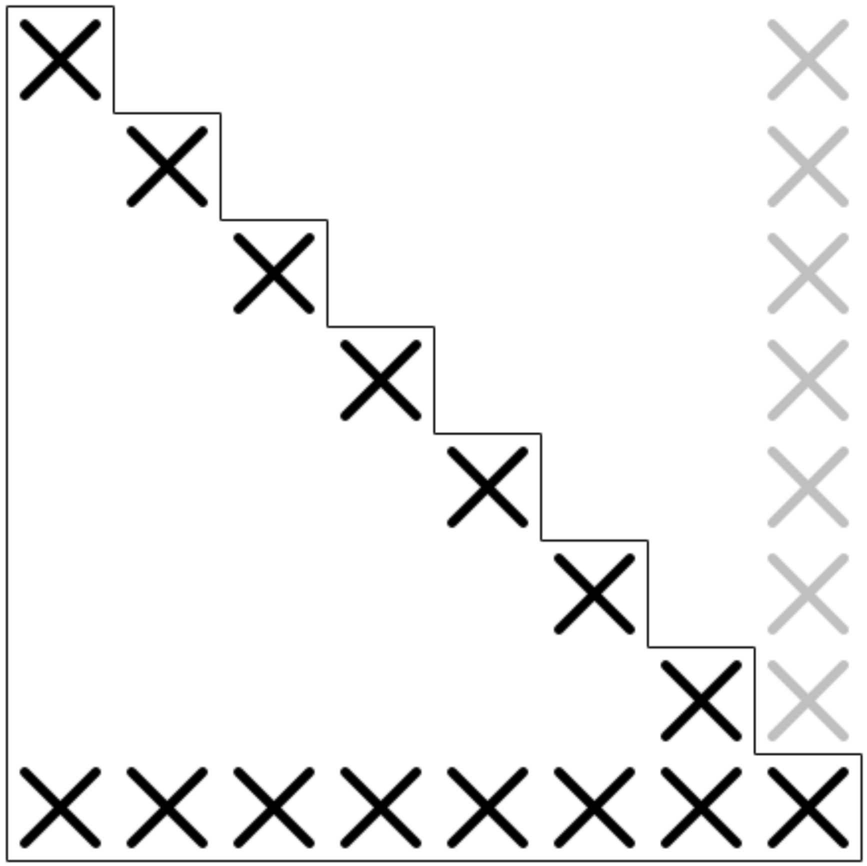
- Optimal partitioning of arrowhead matrix suggests new partitioning method

New 2-D Method: “corner” partitioning



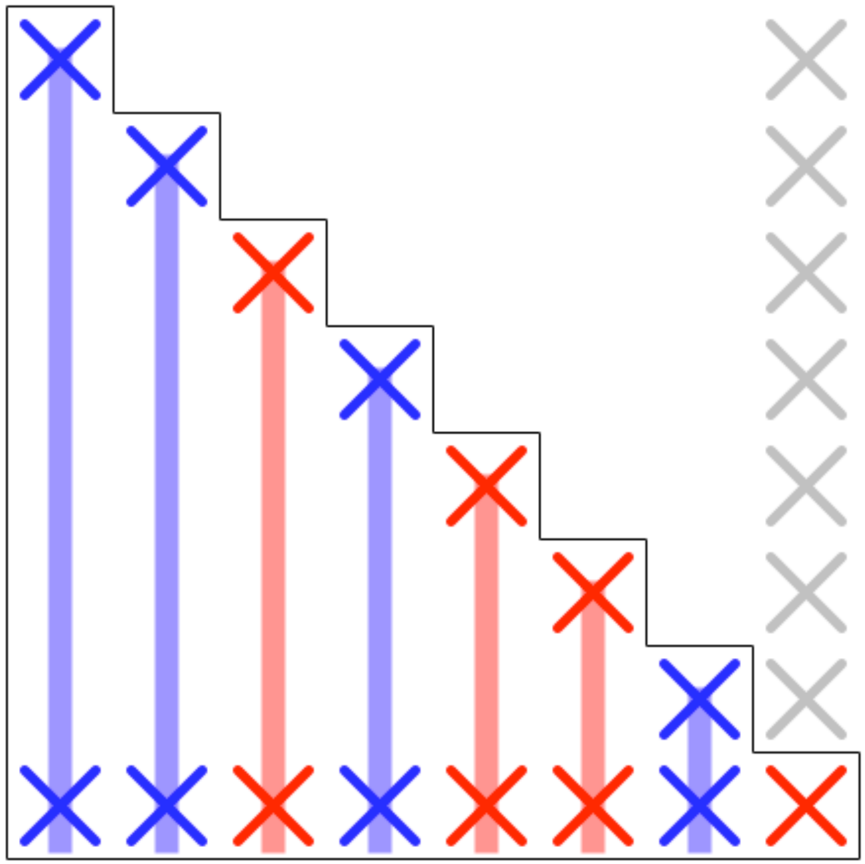
- 1-D partitions reflected across diagonal

New 2-D Method: "corner" partitioning



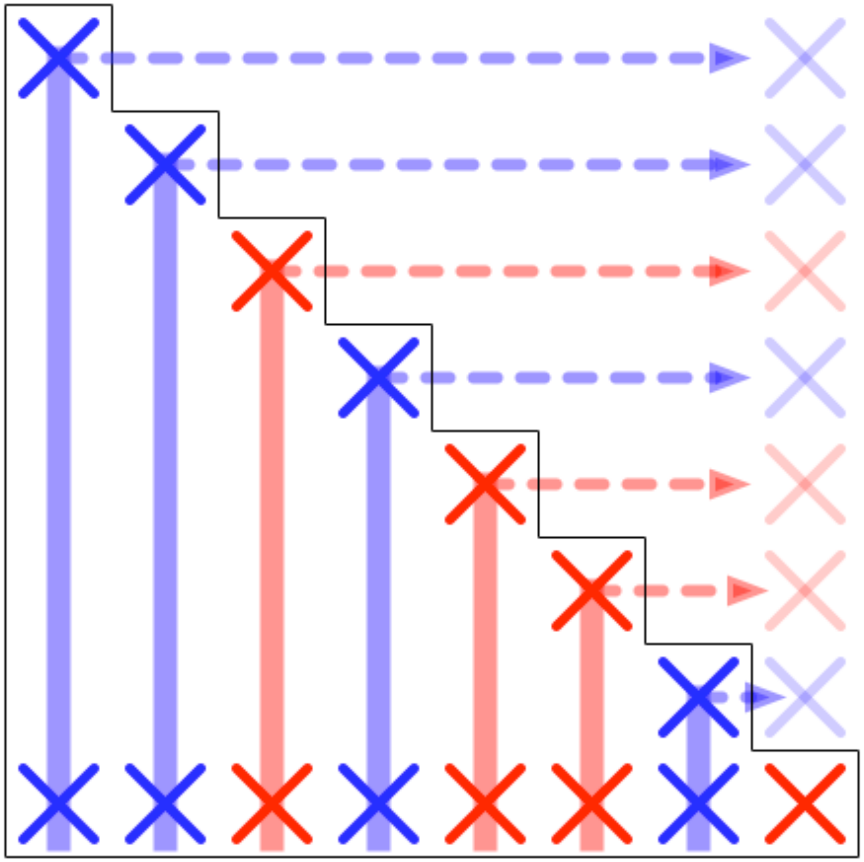
- Take lower triangular part of matrix

New 2-D Method: “corner” partitioning



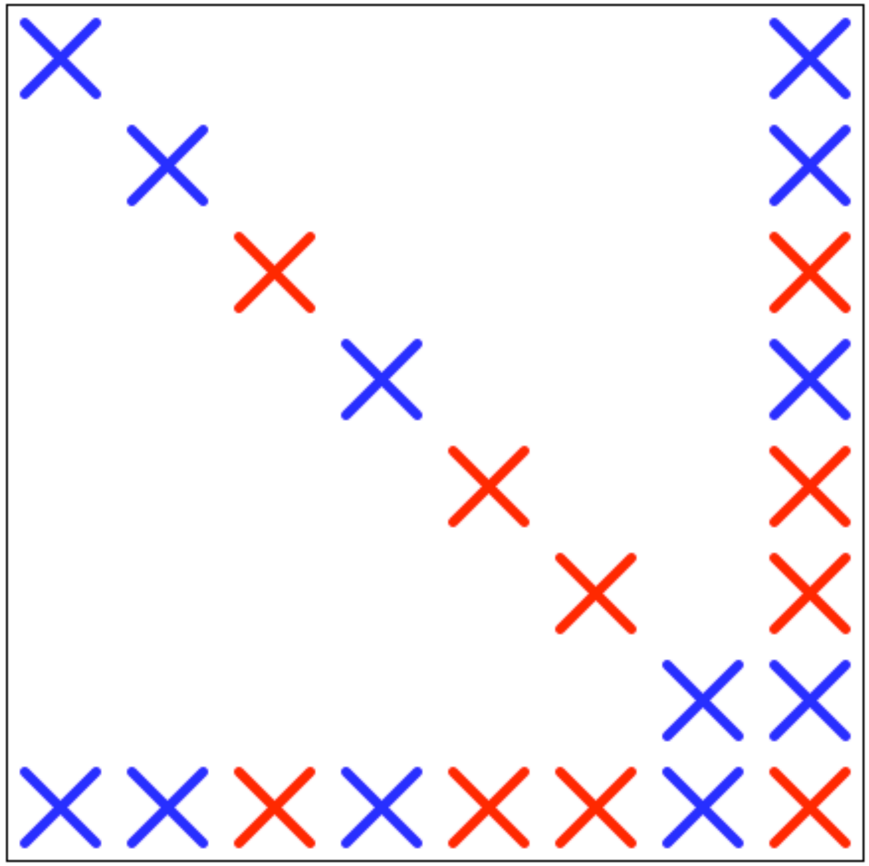
- 1-D (column) hypergraph partition of lower triangular matrix

New 2-D Method: "corner" partitioning



- Reflect partition symmetrically across diagonal

New 2-D Method: “corner” partitioning



Volume = 2

- Optimal partition

Comparison of Methods -- Arrowhead Matrix

p	1D column	Mondriaan	Corner	Fine grain
2	29101	29102	**2***	**2***
4	40001	29778	**6***	**6***
16	40012	37459	**30***	**30***
64	40048	39424	**126***	**126***

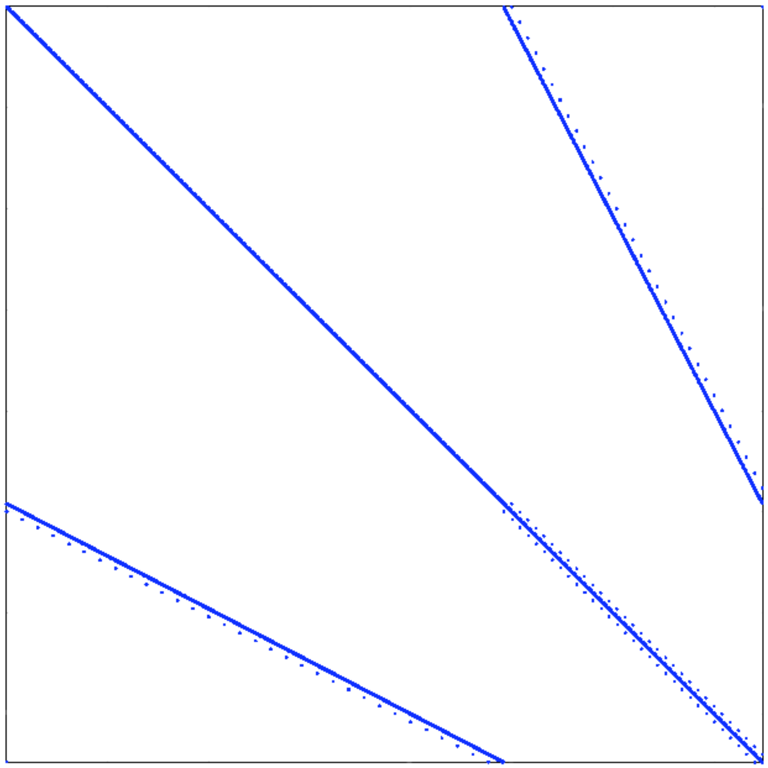
Order n

2(p-1)

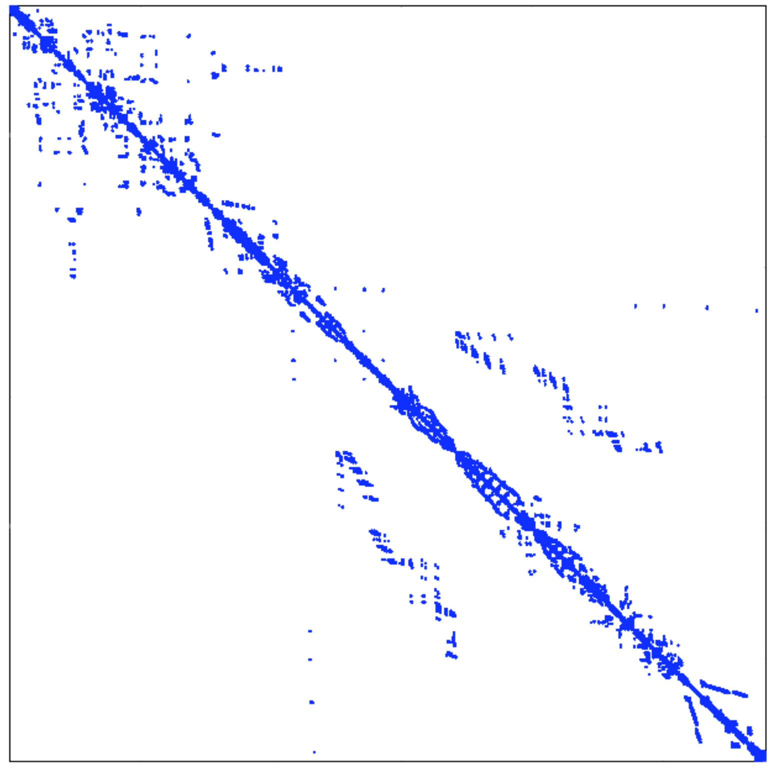
- n = 40,000
- nnz = 119,998

*optimal

Comparison of Methods -- "Real" Matrices



finan512

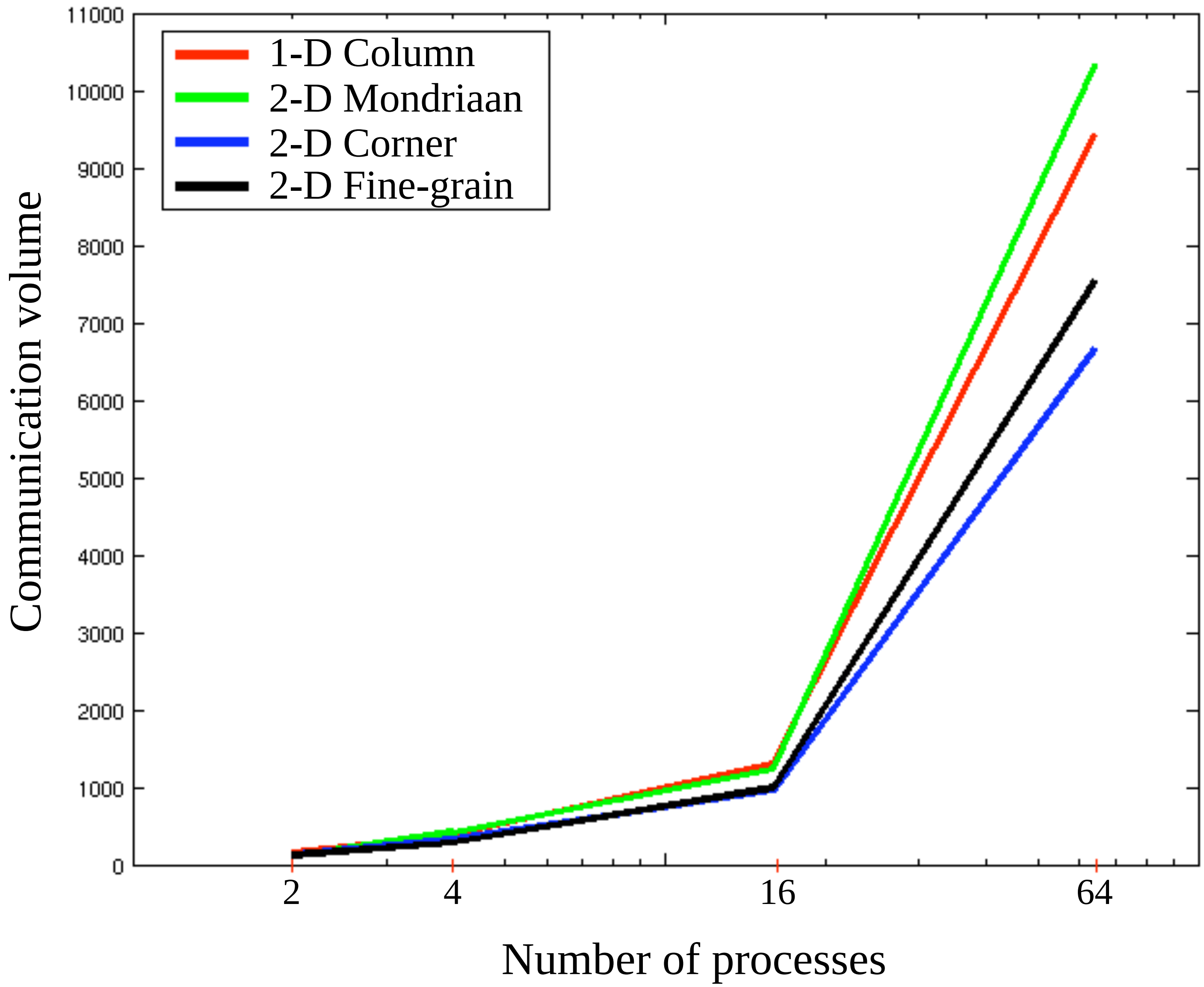


bcsstk30

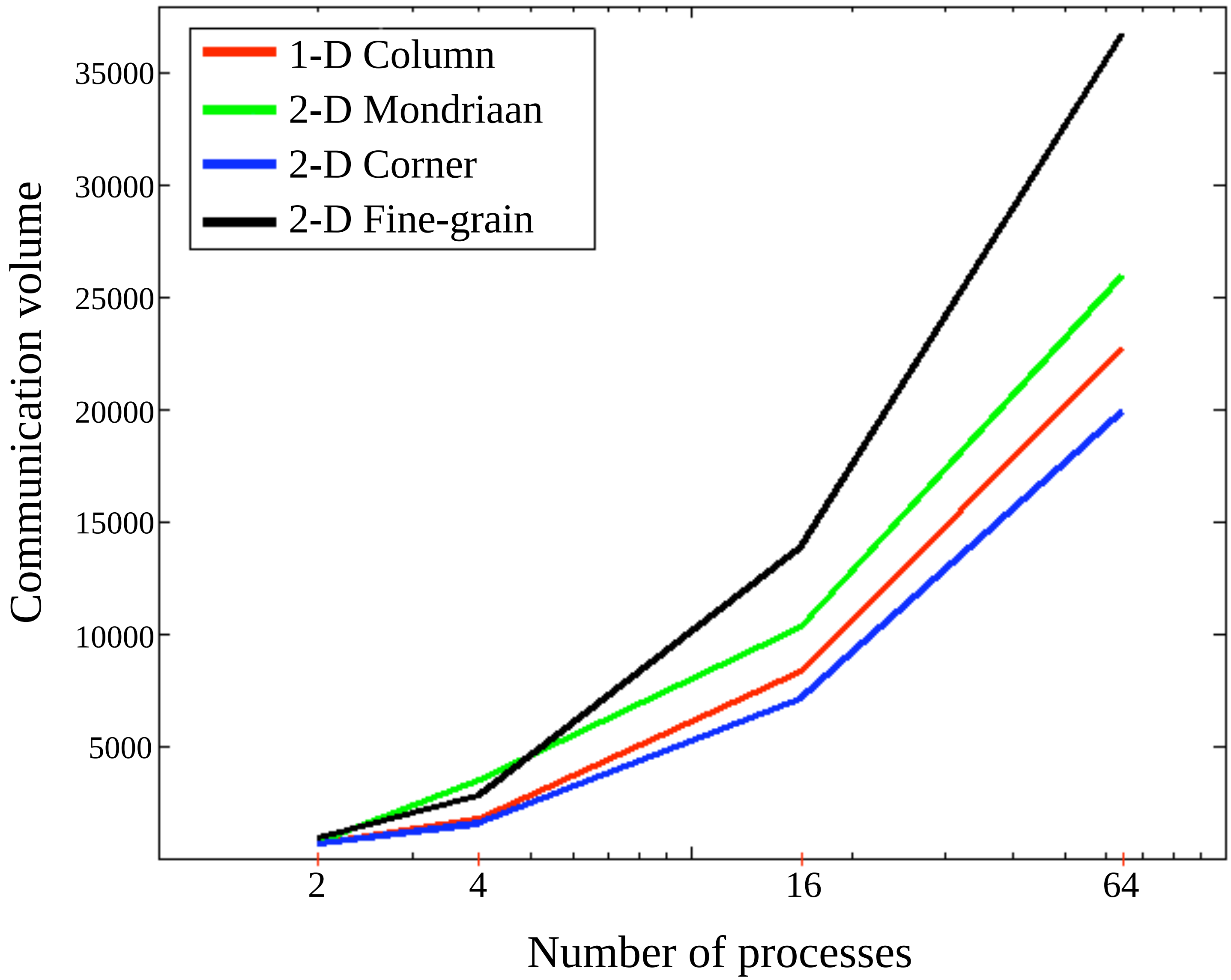
Portfolio optimization

Structural Engineering

Comparison of Methods -- finan512 Matrix



Comparison of Methods -- bcsstk30 Matrix



Sandia National Laboratories

Summary

- Many models for reducing communication in matrix-vector multiplication
- 1-D partitioning inadequate for many partitioning problems
- New method of 2-D matrix partitioning
  - Improvement for some matrices
  - Faster than fine-grain method

Future Work

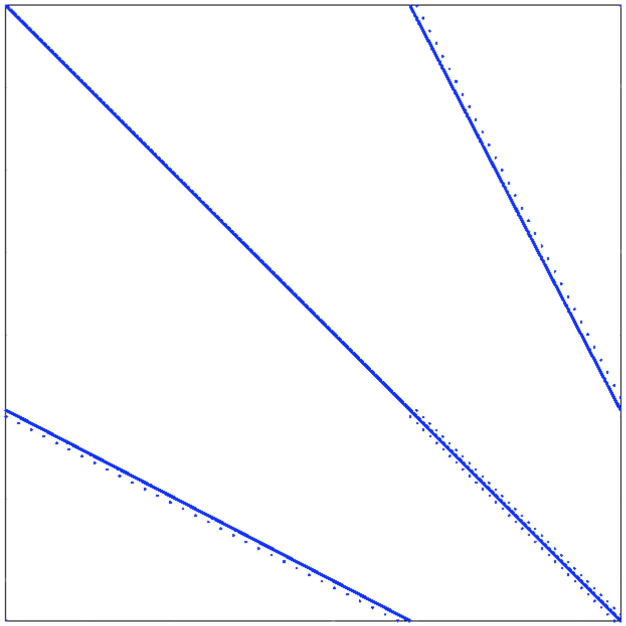
- Better intuition for “corner” partitioning method
  - Optimal for arrowhead matrix
  - Good for finan512, bcsstk30 matrices
  - When a good method?
- Reordering of matrix rows/columns for “corner” partitioning method
  - Unlike 1-D graph/hypergraph, dependence on ordering
  - Find optimal ordering/partition
  - Extend utility of method

Acknowledgements

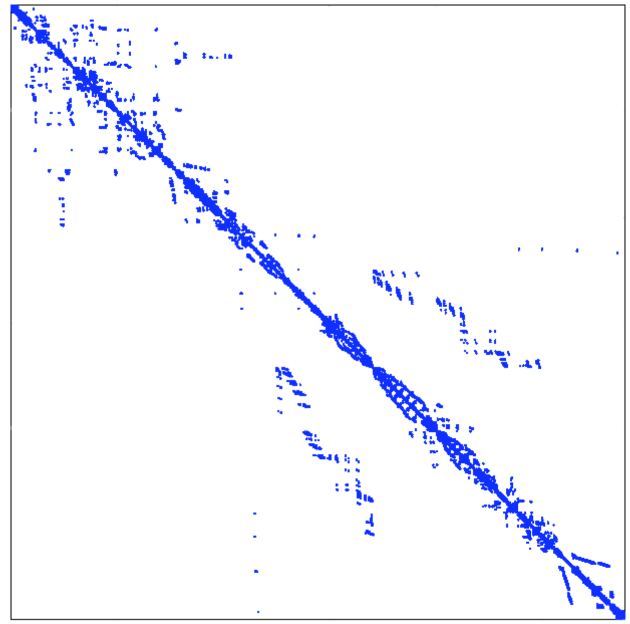
- Dr. Erik Boman
  - Technical advisor
- Dr. Bruce Hendrickson
  - Row/column reordering work
- Zoltan
  - Used Zoltan for 1-D hypergraph partitioning

Extra

Comparison of Methods -- "Real" Matrices



finan512



bcsstk30

Portfolio optimization

matrix	rows	nonzeros
finan512	74,752	596,992
bcsstk30	28,924	2,043,492

Structural Engineering

Sandia National Laboratories