A Nested Dissection Partitioning Method for Parallel Sparse Matrix-Dense Vector Multiplication

HPEC 2013

9/11/2013

Erik Boman, *Michael Wolf**
Sandia National Laboratories

*Now at MIT Lincoln Laboratory

Sparse Matrix Partitioning Motivation

- Sparse matrix-dense vector multiplication (SpMV) is common kernel in many numerical computations
  - Iterative methods for solving linear systems
  - PageRank computation
  - Anomaly detection in graphs (spectral methods)
- Need to make parallel SpMV kernel as fast as possible
- Finding good data to processor mapping (partitioning) can greatly improve parallel performance

Parallel Sparse Matrix-Dense Vector Multiplication

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

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

- Partition matrix nonzeros
- Partition vectors

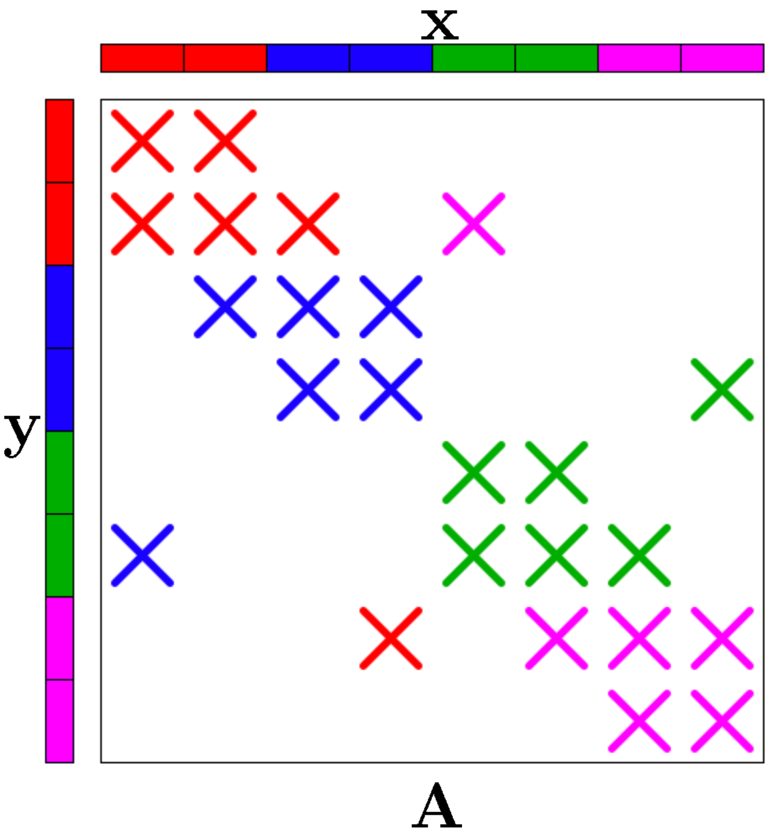
Objective

- Ideally we minimize total run-time of SpMV
- Settle for “easier” objective
  - Work balanced
  - Minimize total communication volume
  - NP-hard to find optimal solution (polynomial time heuristic algorithms)
- Can partition matrices in different ways
  - 1D
  - 2D
- Can model problem in different ways
  - Graph
  - Bipartite graph
  - Hypergraph

Parallel SpMV

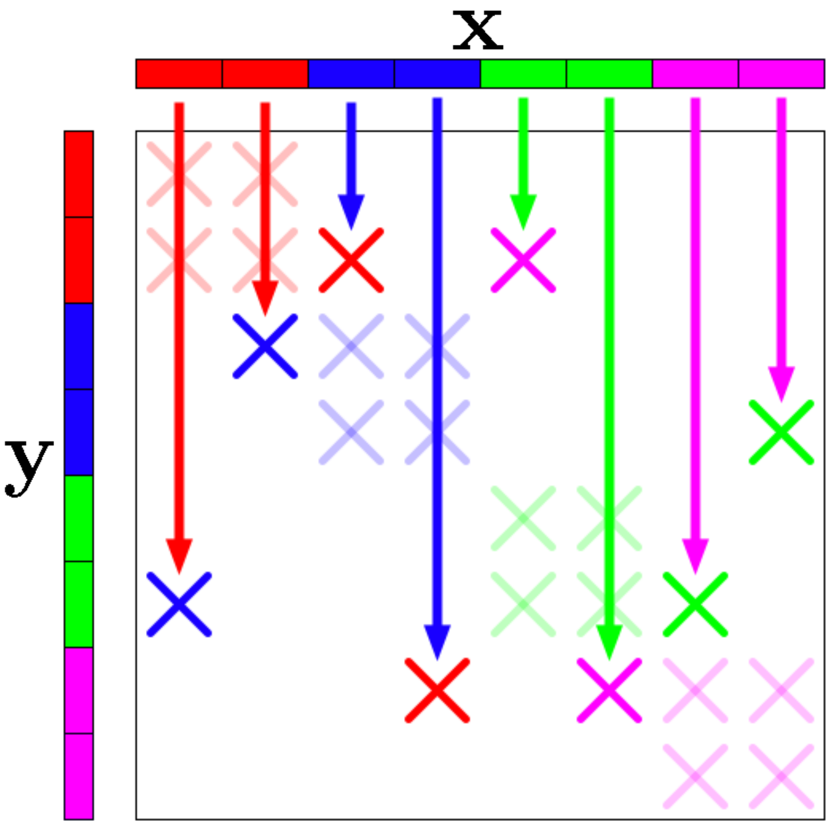
$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = \begin{bmatrix} 1 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 1 & 9 & 0 & 5 & 0 & 0 & 0 \\ 0 & 8 & 1 & 7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 1 & 8 & 0 & 0 \\ 4 & 0 & 0 & 0 & 3 & 1 & 3 & 0 \\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

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

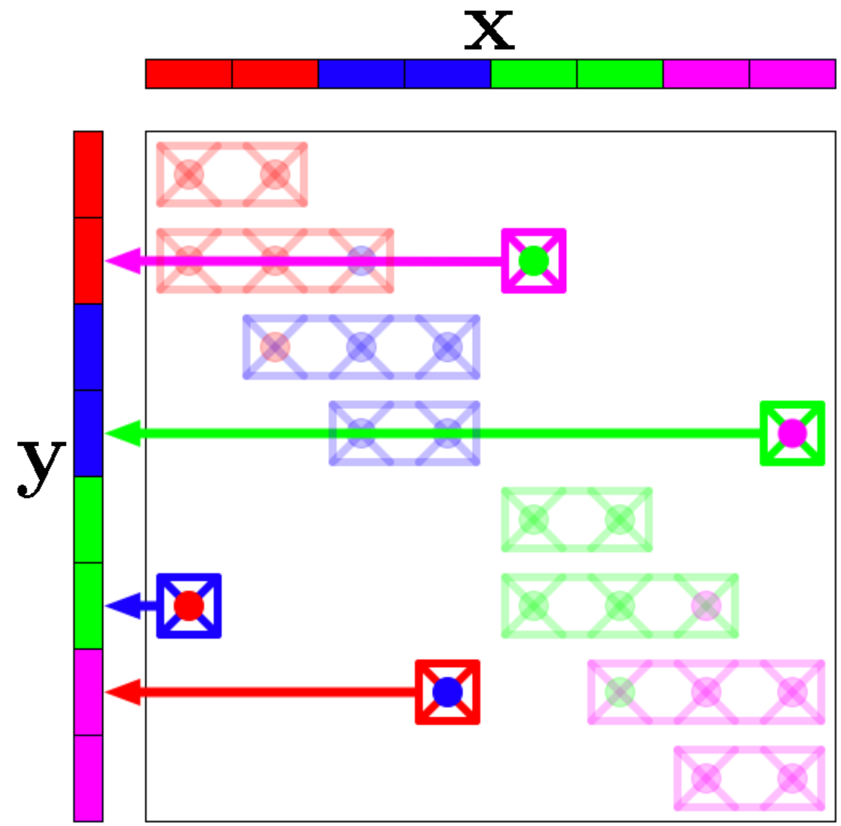


- Alternative way of visualizing partitioning

Parallel SpMV Communication

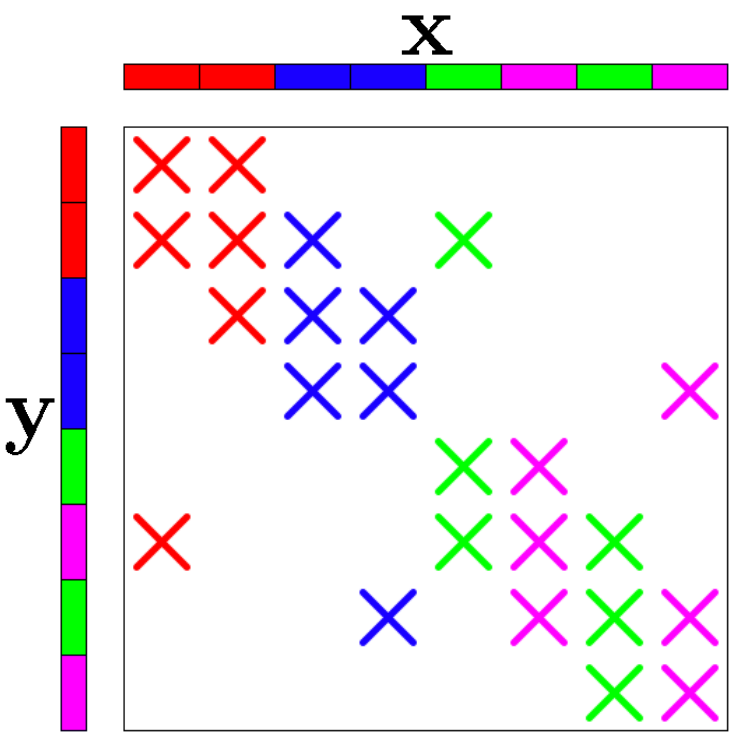


- x_j sent to remote processes that have nonzeros in column j

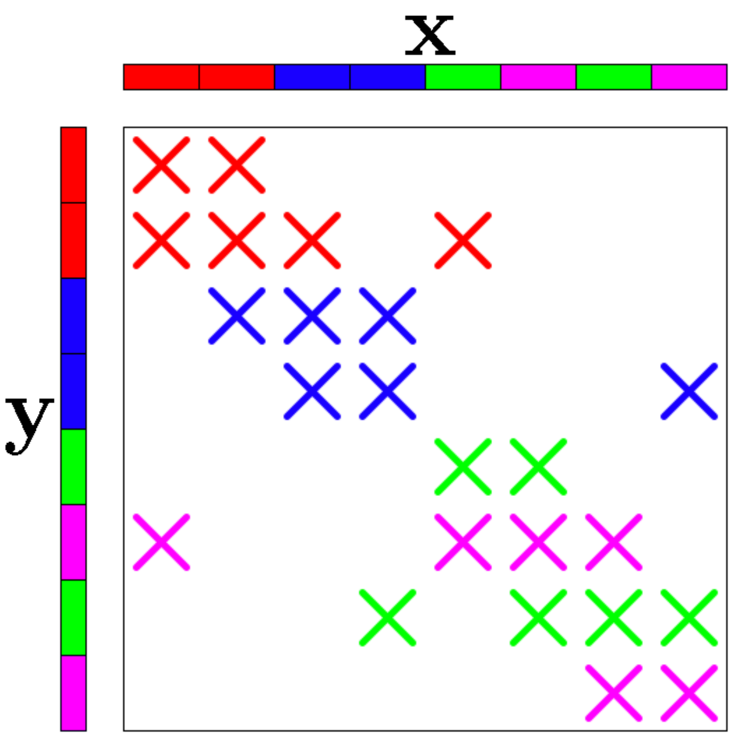


- Partial inner-products sent to process that owns vector element y_i

1D Partitioning



1D Column

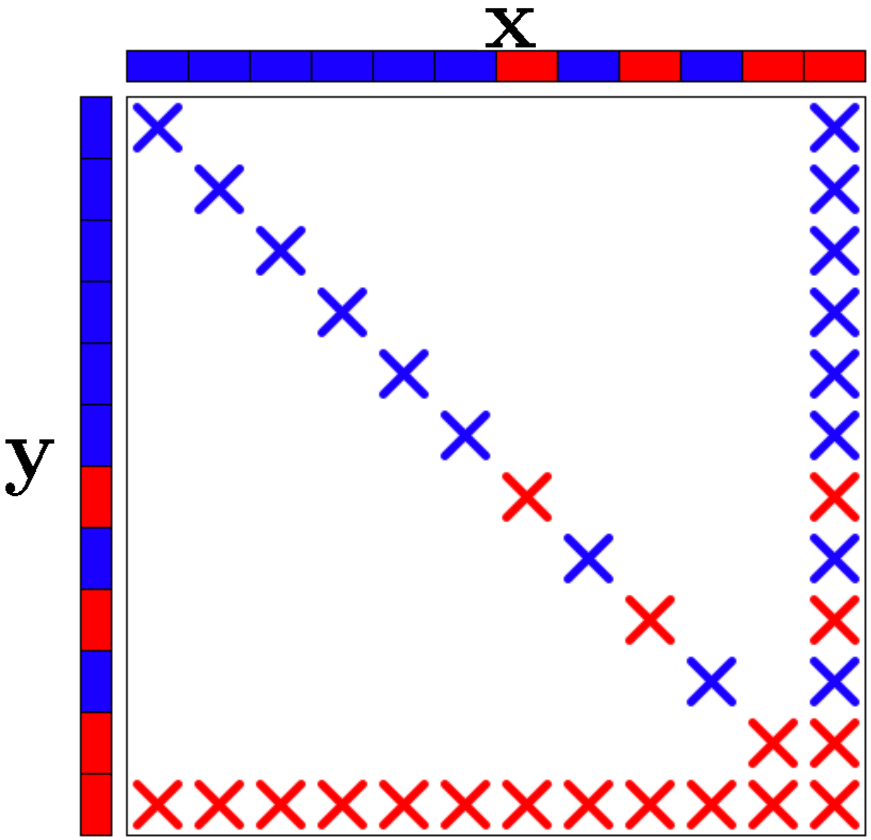


1D Row

Each process assigned nonzeros for set of columns

Each process assigned nonzeros for set of rows

When 1D Partitioning is Inadequate



"Arrowhead" matrix

n=12

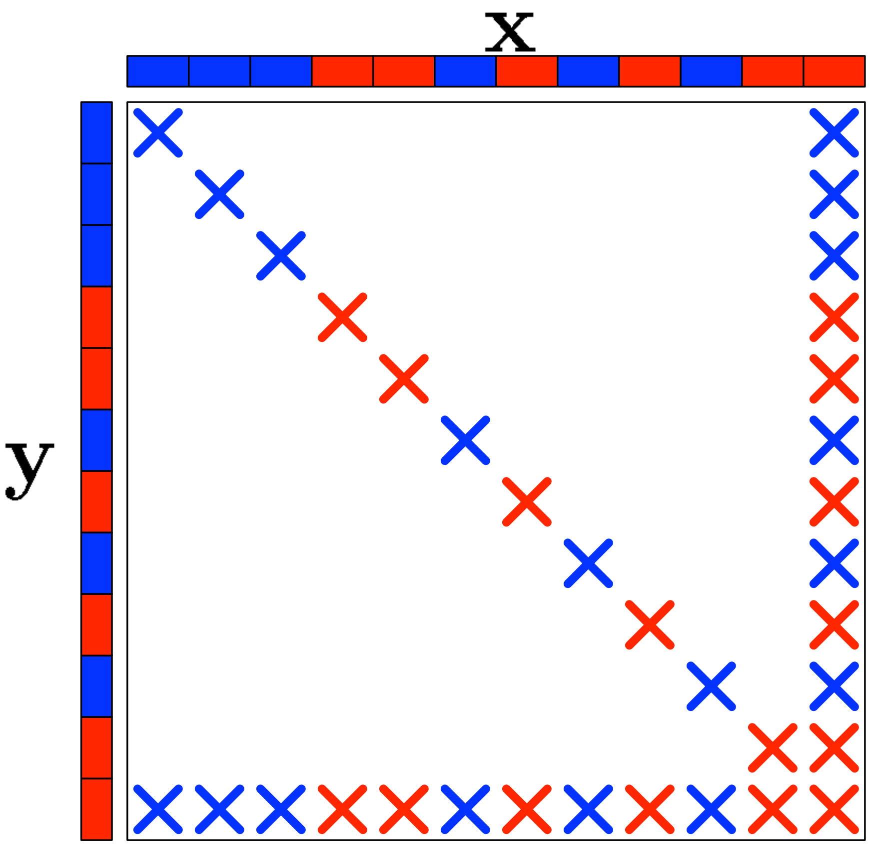
nnz=34 (18,16)

volume = 9

- For any 1D bisection of nxn arrowhead matrix:
  - nnz = 3n-2
  - Volume ≈ (3/4)n

1D partitioning of arrowhead matrix yields high volume for SpMV

When 1D Partitioning is Inadequate



"Arrowhead" matrix

n=12

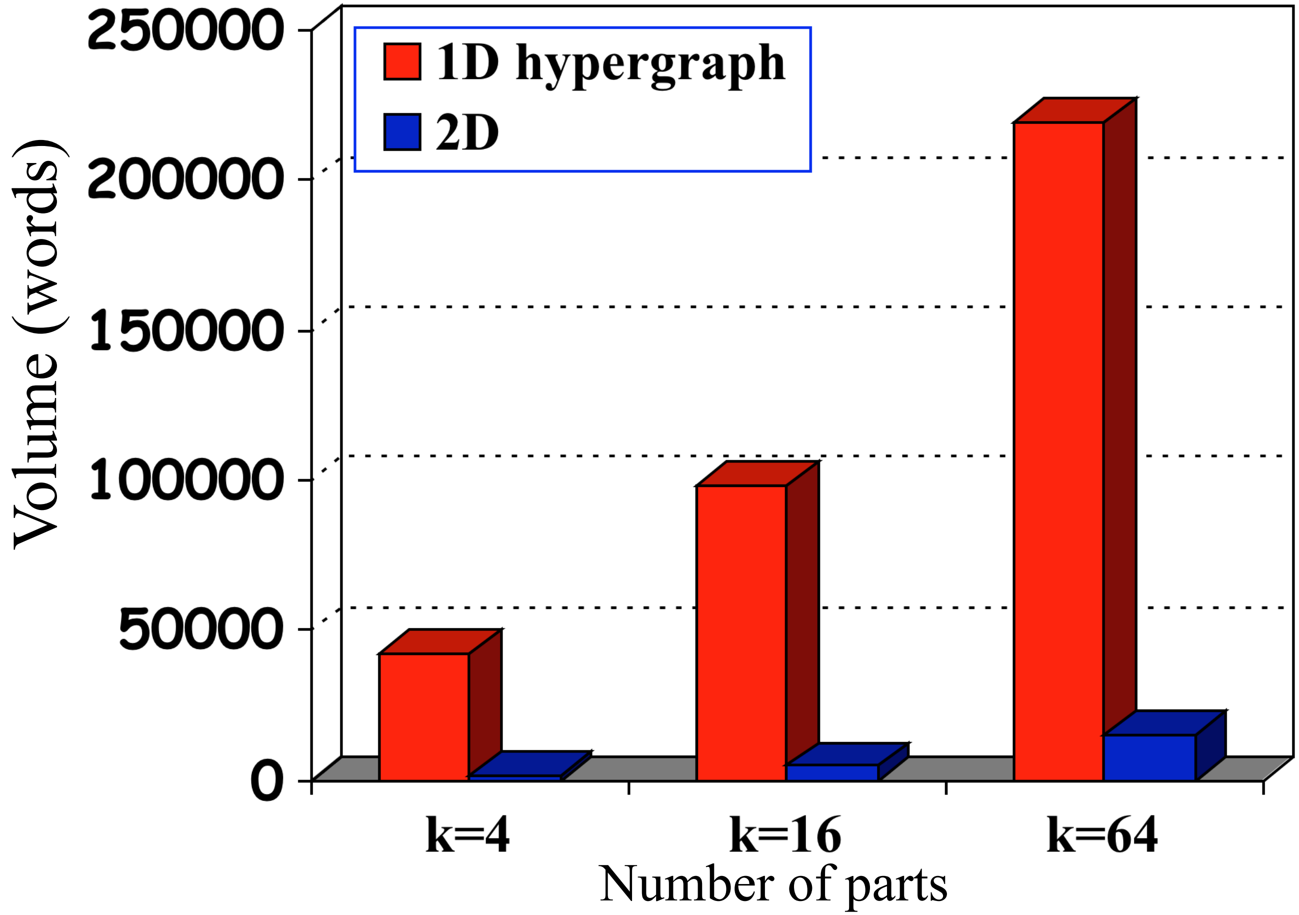
nnz=34 (16,18)

volume = 2

- 2D partitioning
- O(k) volume partitioning possible

2D partitioning of arrowhead matrix reduces volume for SpMV

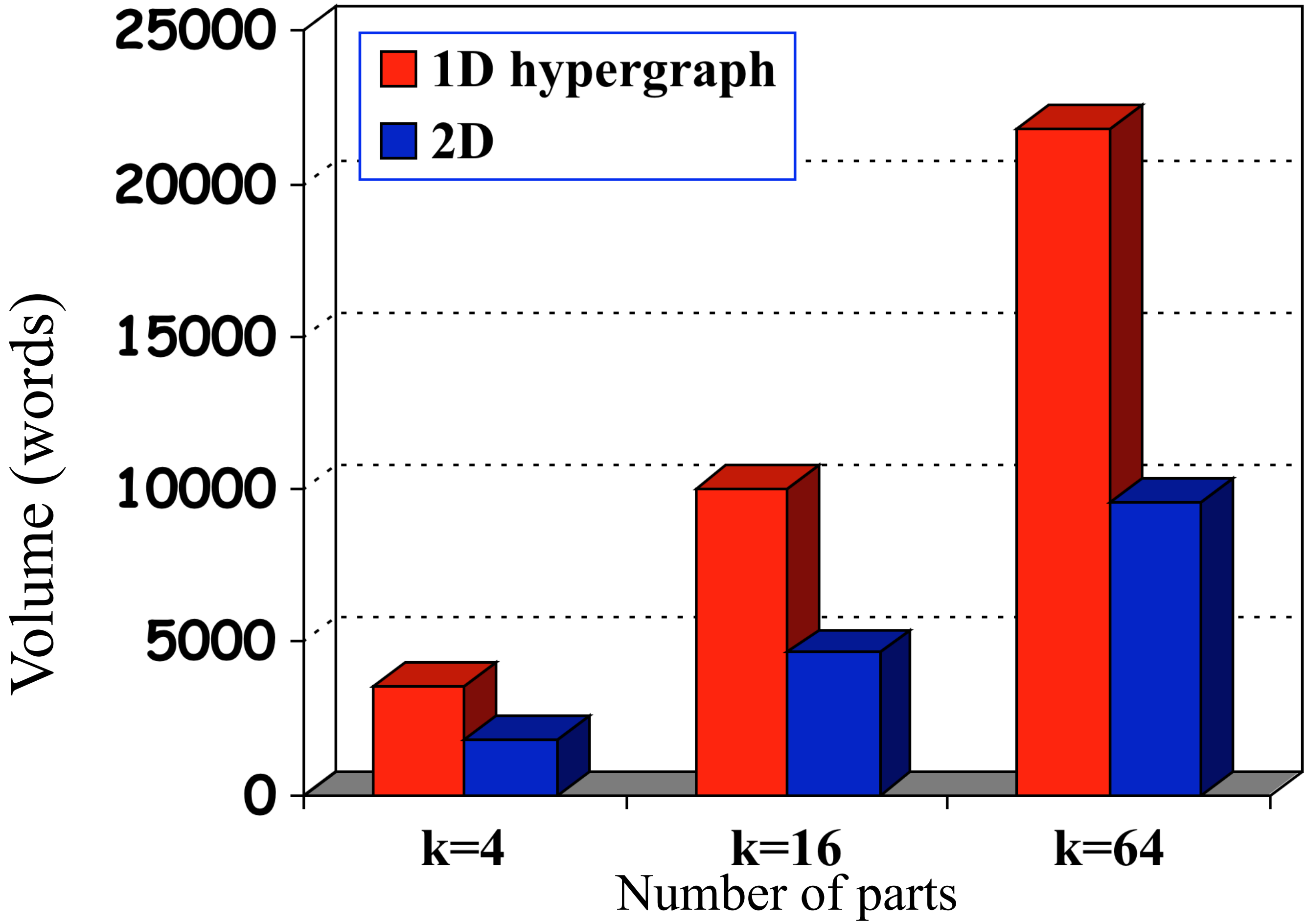
1D is Inadequate



c-73: nonlinear optimization (Schenk)

- UF sparse matrix collection
- n=169,422 nnz=1,279,274

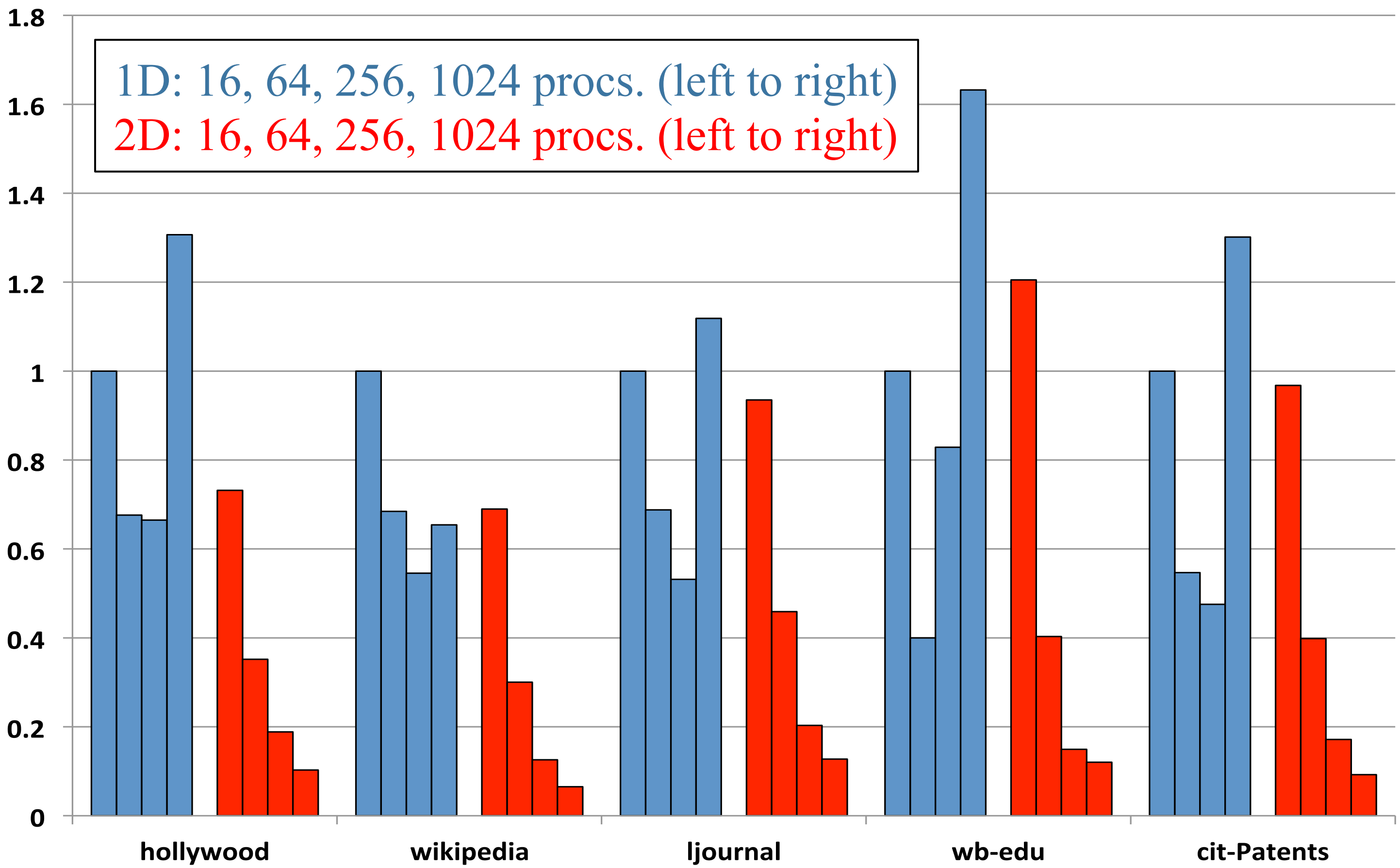
1D is Inadequate



asic680ks: Xyce circuit simulation (Sandia)

– n=682,712   nnz=2,329,176

1D vs 2D: Strong Scaling for “Scale Free” Networks

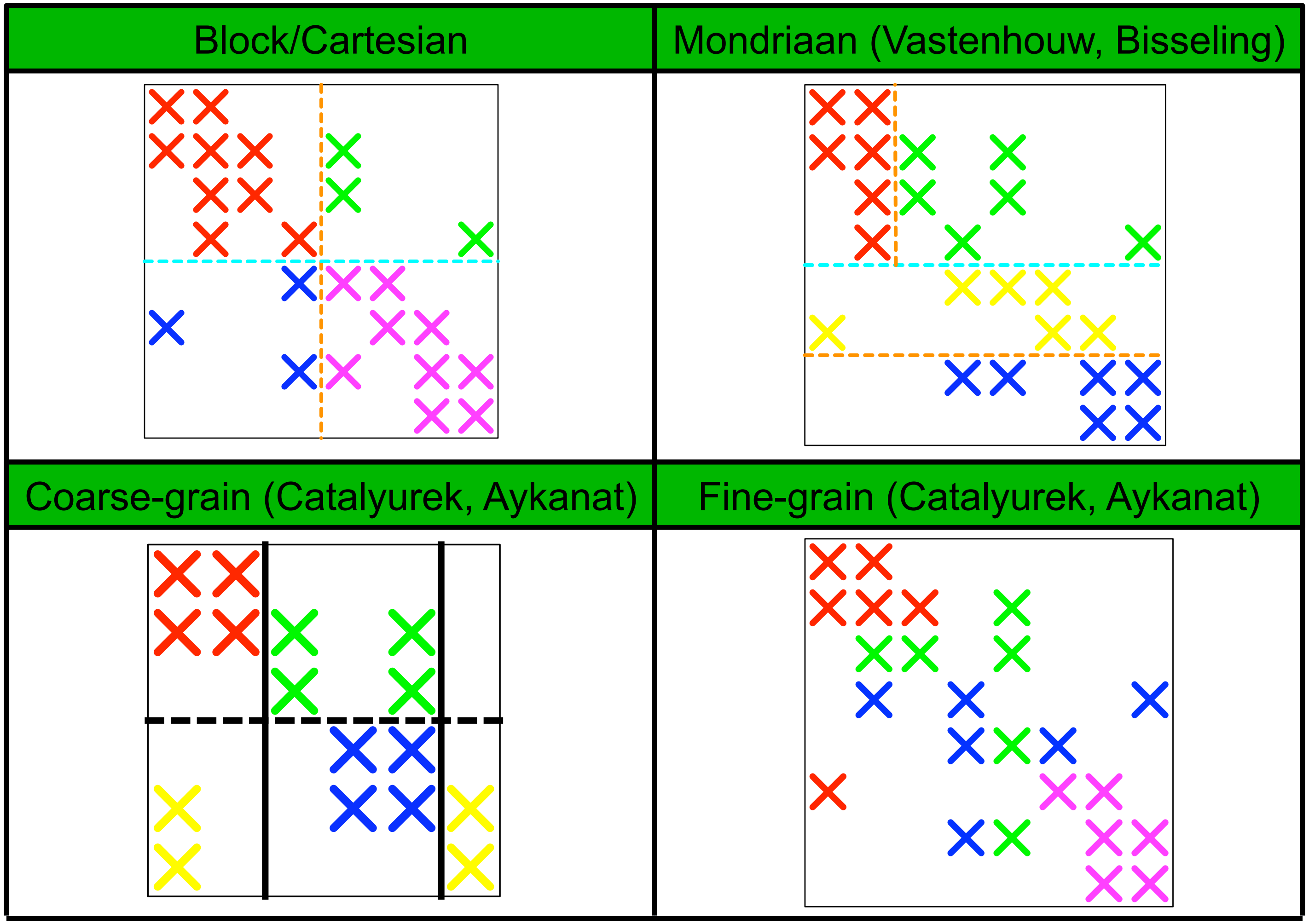


Runtime (relative to 16 processor/1D runtime) for SpMV using Trilinos with 1D and 2D distributions

SpMV with 1D distributions not scalable

Source: Boman, Devine, Rajamanickam; “Scalable Matrix Computations on Large Scale-Free Graphs using 2D Graph Partitioning,” Sparse Days, CERFACS, 2013.

2D Partitioning

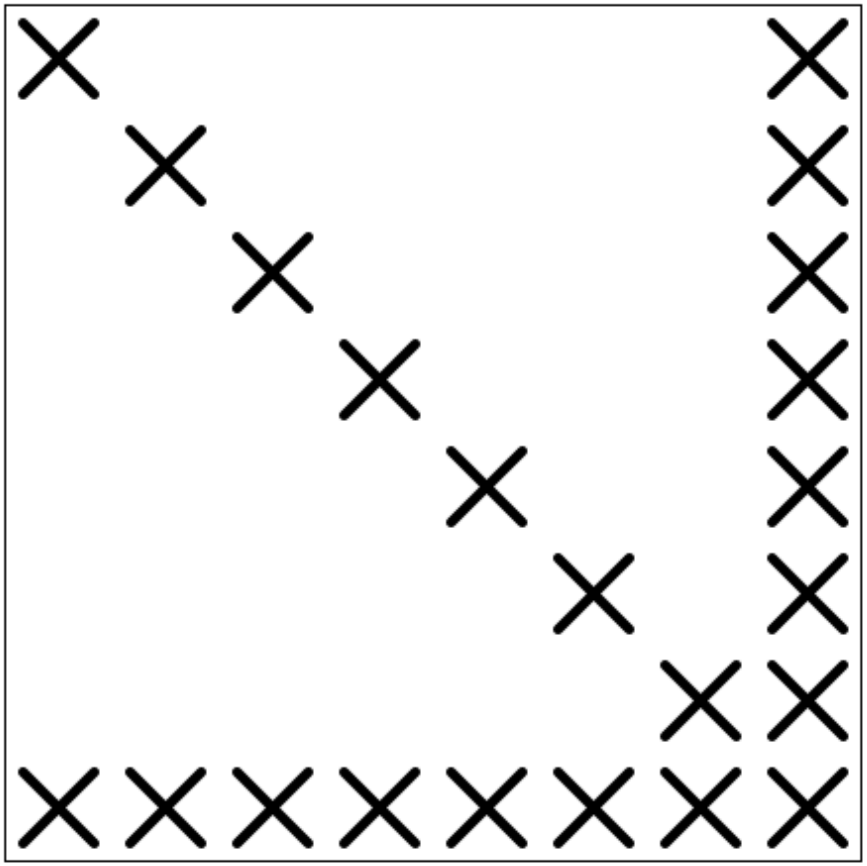


- More flexibility: no particular part for entire row or column
- More general sets of nonzeros assigned parts

2D Partitioning

- Fine-grain hypergraph
- Graph model for symmetric 2D partitioning
- **Nested dissection symmetric partitioning method**
  - New 2D method

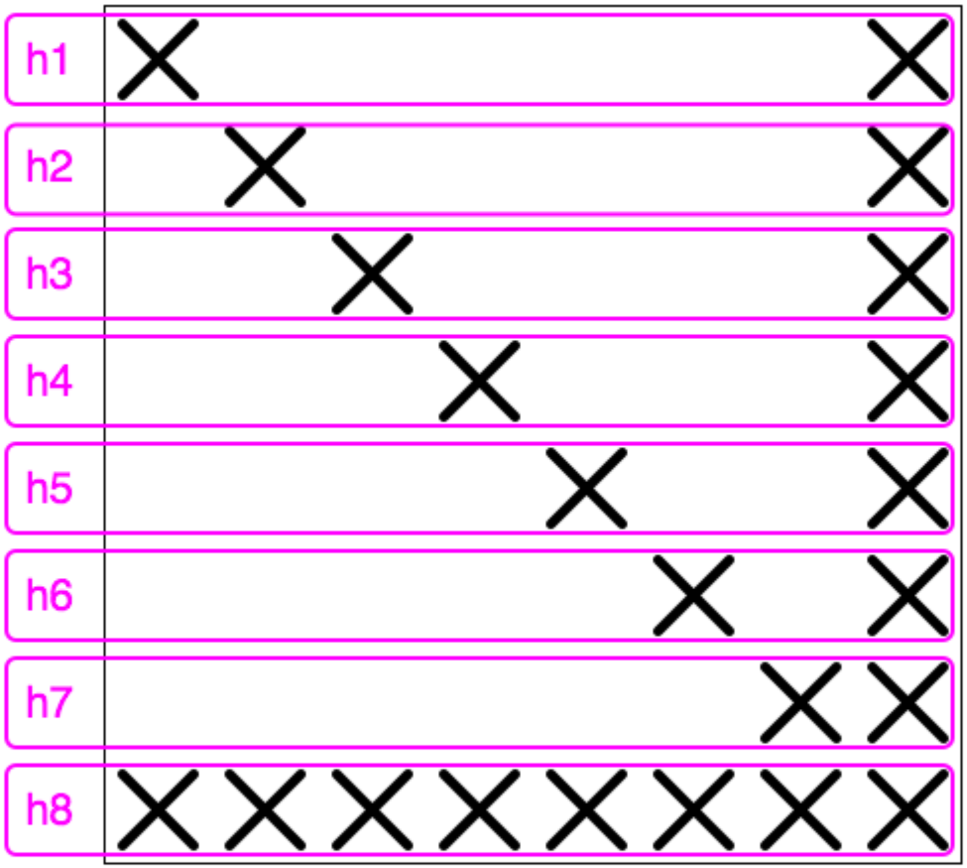
Fine-Grain (FG) Hypergraph Model



- Catalyurek and Aykanat (2001)
- Each nonzero partitioned independently
- Good quality partitions
- Significantly slower than 1D methods

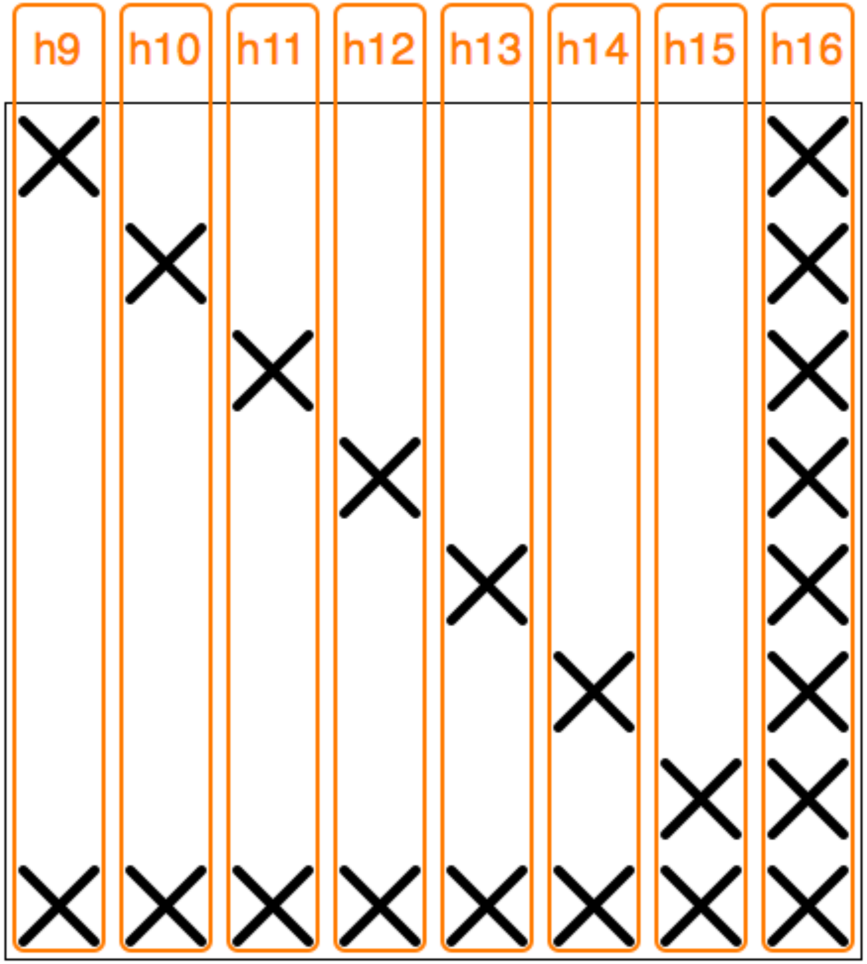
Nonzeros represented by vertices in hypergraph

Fine-Grain Hypergraph Model



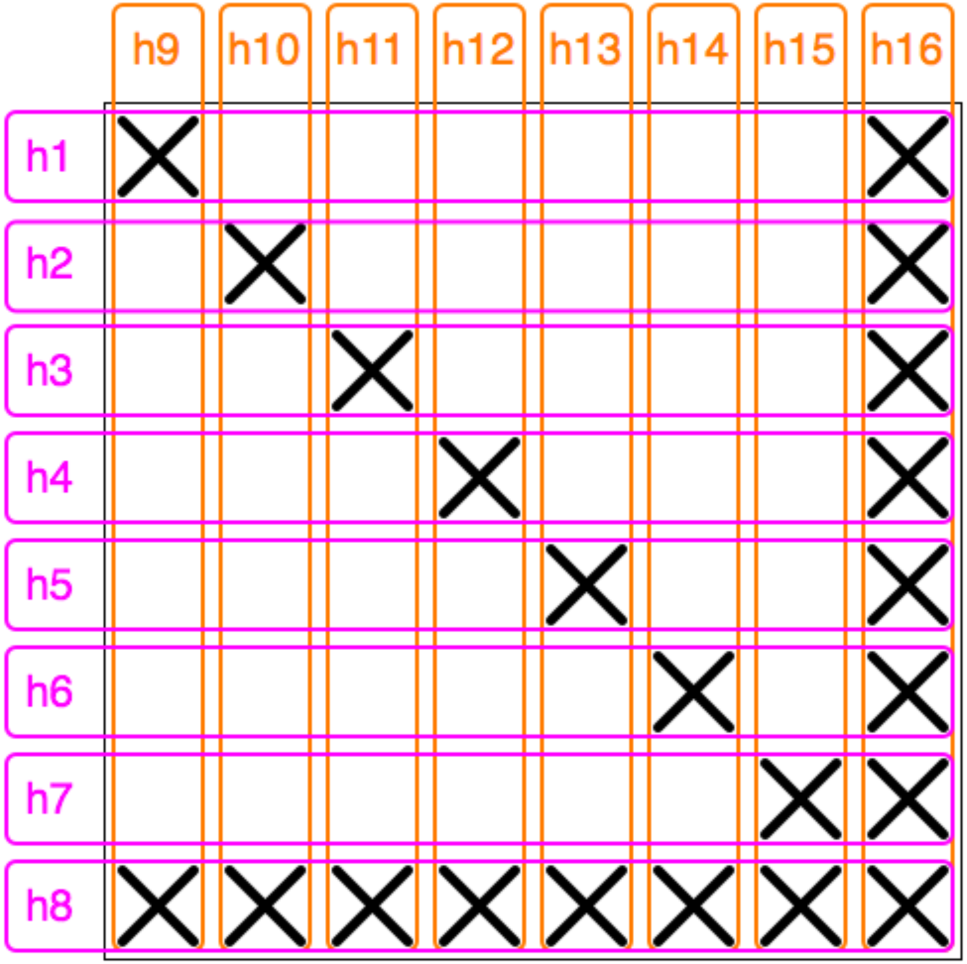
- Rows represented by hyperedges
- Hyperedge - set of one or more vertices

Fine-Grain Hypergraph Model



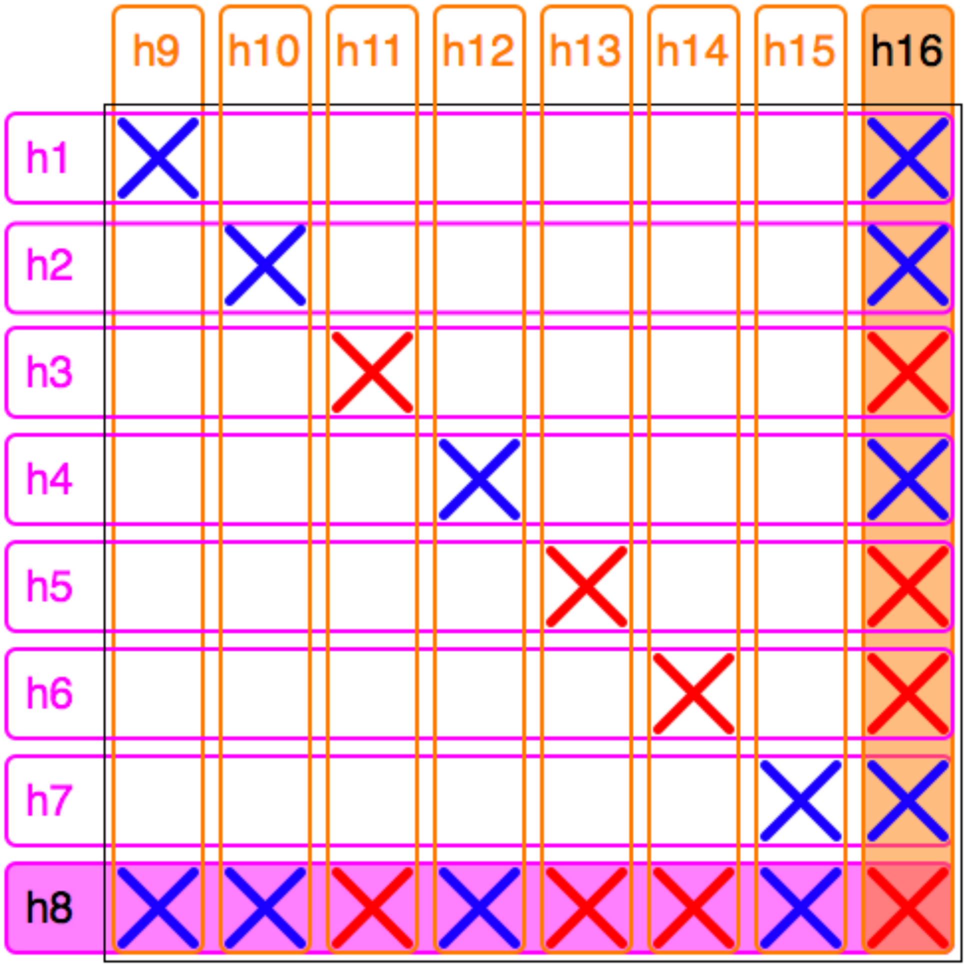
- Columns represented by hyperedges

Fine-Grain Hypergraph Model



- 2n hyperedges

Fine-Grain Hypergraph Model



k=2, volume = cut = 2

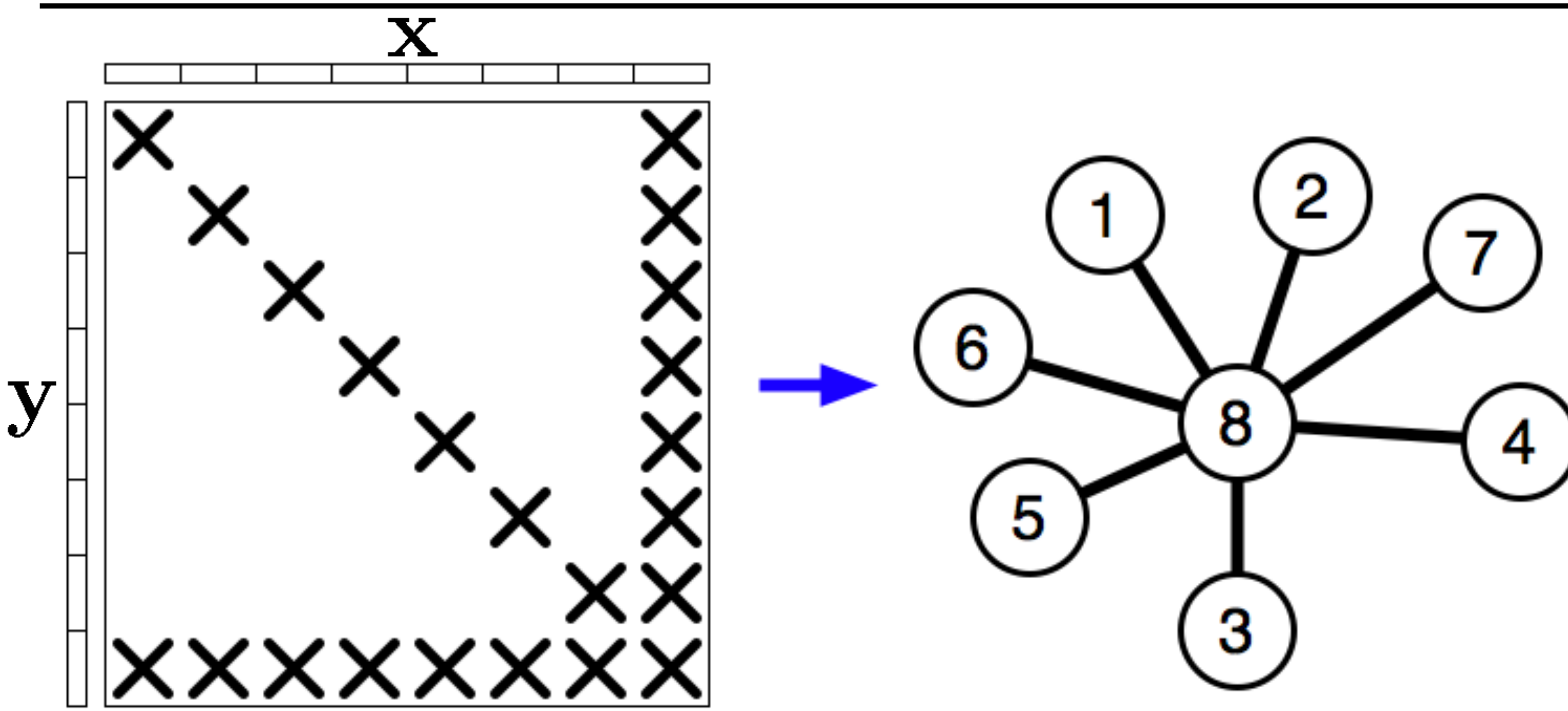
- Partition vertices into k equal sets
- For k=2
  - Volume = number of hyperedges cut
- Minimum volume partitioning when optimally solved
- Larger NP-hard problem than 1D

Objective: minimize hyperedge cut, subject to load balance constraint

Graph Model for Symmetric 2D Partitioning

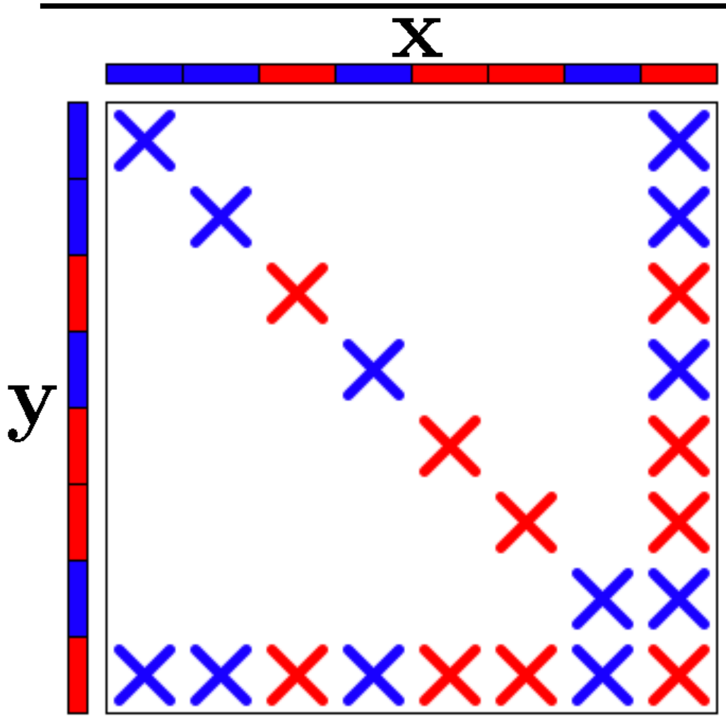
- Exact model of communication for symmetric matrix partitioning
- Given matrix A with symmetric nz structure
- Symmetric partition
  - a(i,j) and a(j,i) assigned same part
  - Input and output vectors have same distribution
- Corresponding graph G(V,E)
  - Vertices correspond to vector elements, diagonal nonzero
  - Edges correspond to off-diagonal nonzeros

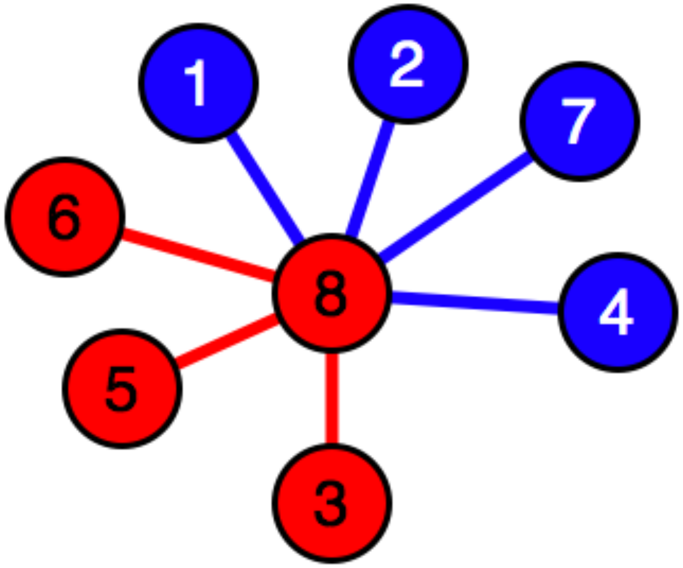
Graph Model for Symmetric 2D Partitioning



- Corresponding graph G(V,E)
  - Vertices correspond to vector elements, diagonal nonzeros
  - Edges correspond to off-diagonal nonzeros

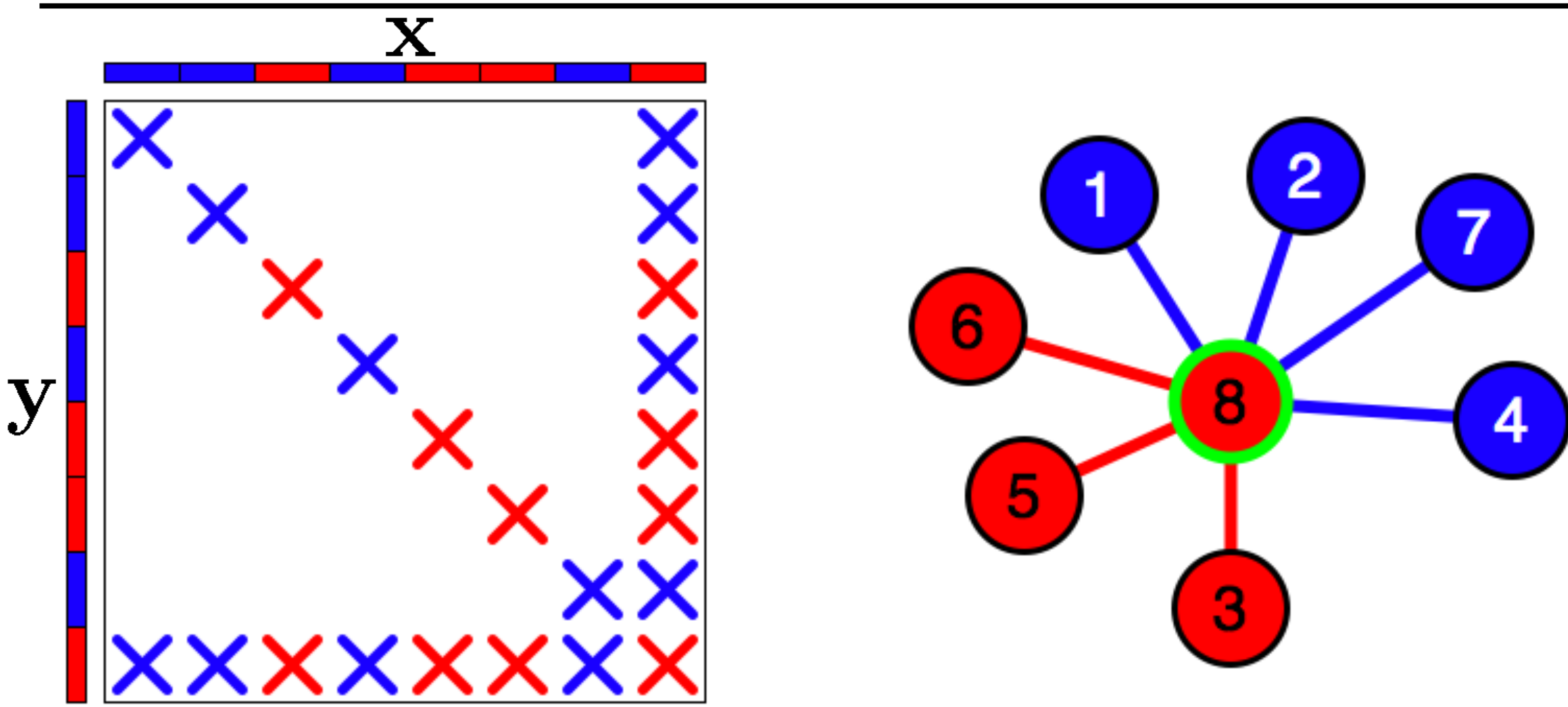
Graph Model for Symmetric 2D Partitioning





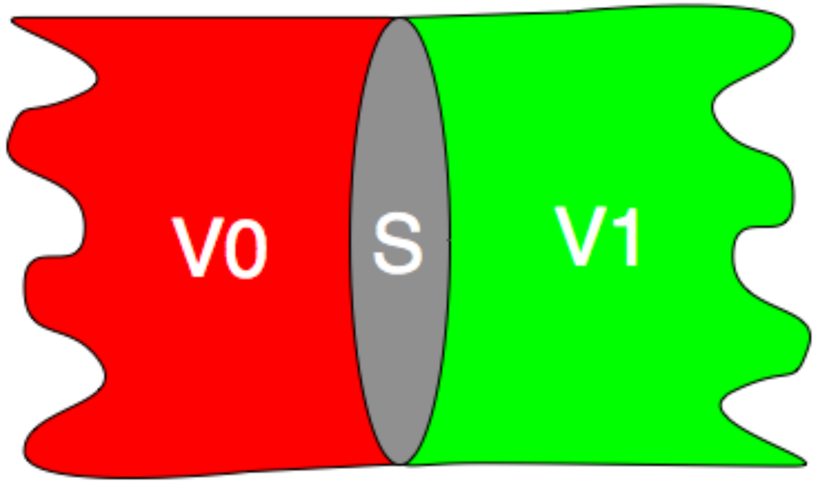
- Symmetric 2D partitioning
  - Partition both V and E
  - Gives partitioning of both matrix and vectors

Communication in Graph Model



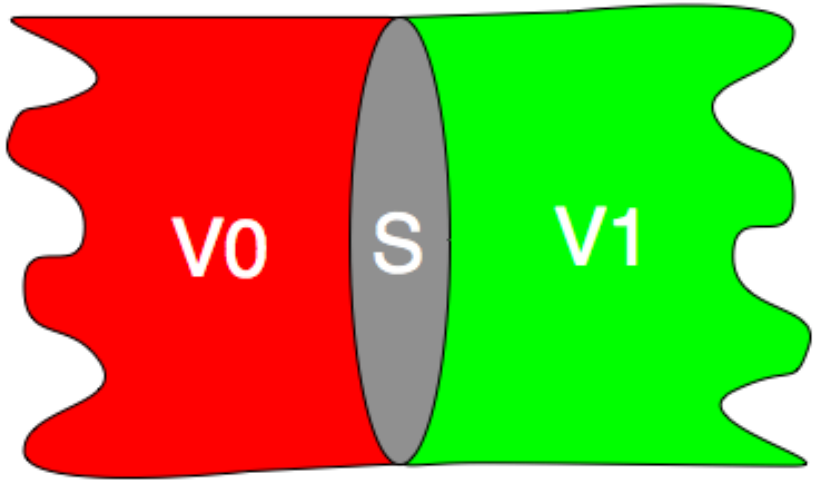
- Communication is assigned to vertices
- Vertex incurs communication iff incident edge is in different part
- Want small vertex separator -- $S=\{V_8\}$
- For bisection, volume = 2 |S|

Nested Dissection Partitioning - Bisection



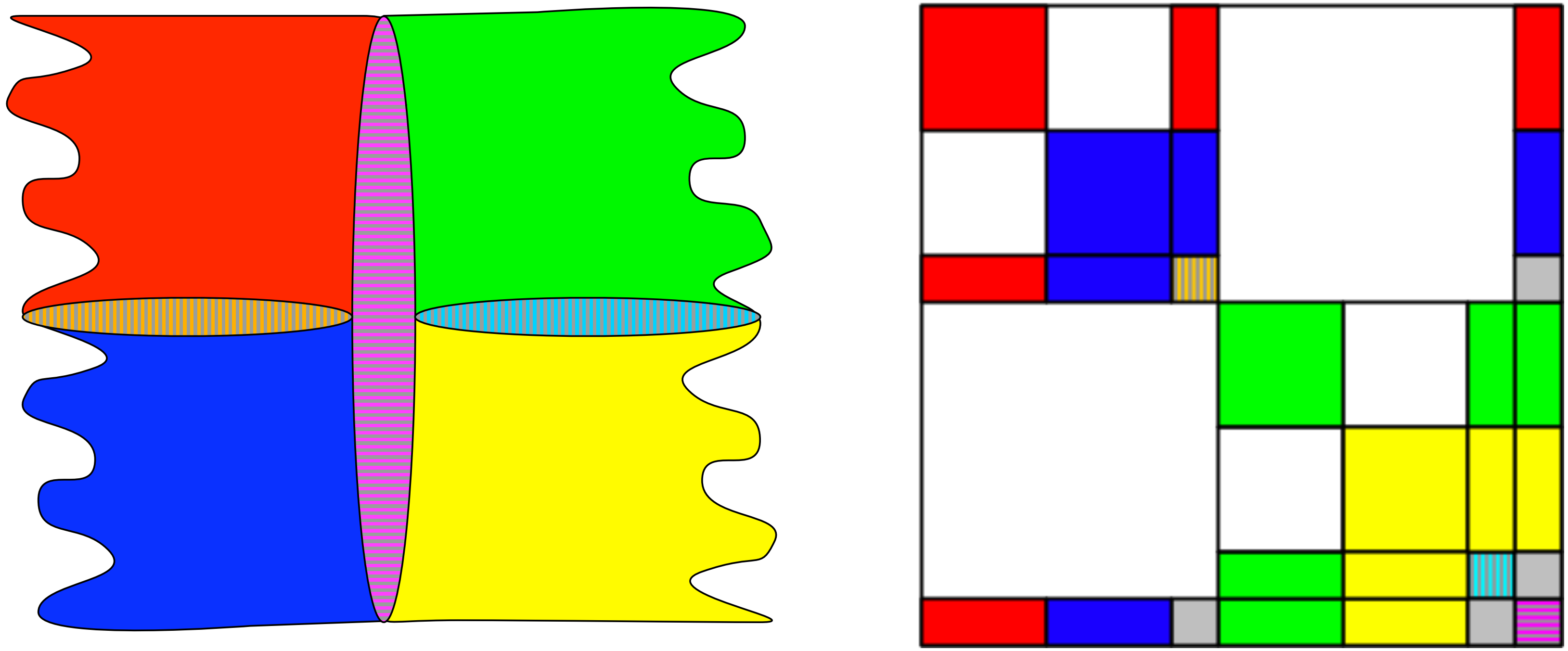
- Suppose A is structurally symmetric
- Let G(V,E) be graph of A
- Find small, balanced separator S
  - Yields vertex partitioning V = (**V0**,**V1**,S)
- Partition the edges such that
  - **E0** = {edges incident to a vertex in **V0**}
  - **E1** = {edges incident to a vertex in **V1**}

Nested Dissection Partitioning - Bisection



- Vertices in S and corresponding edges
  - Can be assigned to either part
  - Can use flexibility to maintain balance
- Communication Volume = 2*|S|
  - Regardless of S partitioning
  - |S| in each phase

Nested Dissection (ND) Partitioning Method



- Recursive bisection to partition into >2 parts
- Use nested dissection!

Nested dissection used to obtain symmetric 2D partitioning

Extension to Nonsymmetric Matrices

- Bipartite graph gives exact model of communication volume
  - Trifunovic and Knottenbelt (2006)
- Apply nested dissection method to A'
  (adjacency matrix for bipartite graph)
  - Use same algorithm as for symmetric case

$$A' = \left[\begin{array}{cc} 0 & A \\ A^T & 0 \end{array} \right]$$

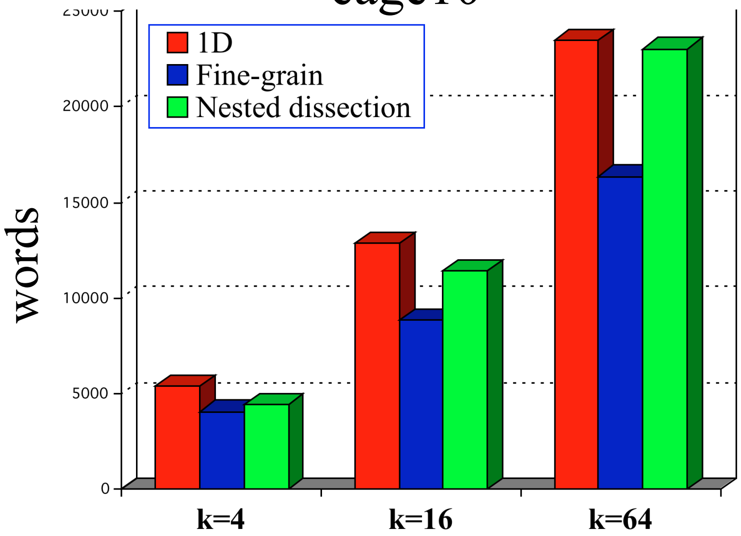
Nested dissection partitioning easily extended to nonsymmetric matrices

Numerical Experiments

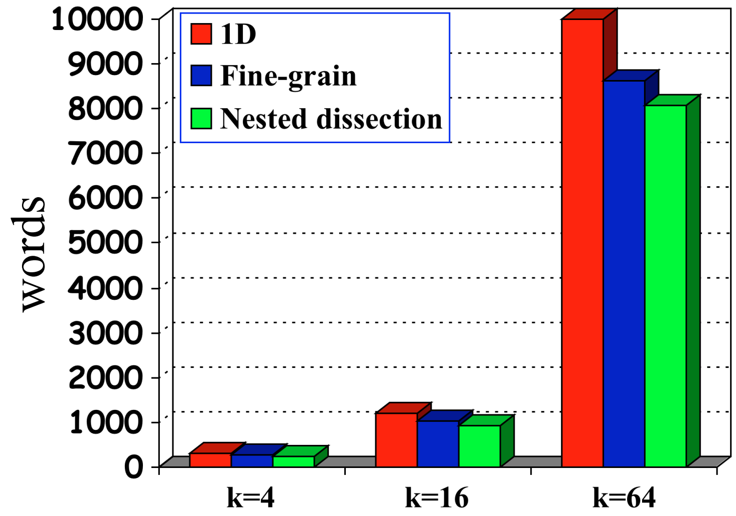
- Structurally symmetric matrices
- k = 4, 16, 64 parts using
  - 1D hypergraph partitioning
  - Fine-grain hypergraph partitioning (2D)
    - Good quality partitions but slow
  - **Nested dissection partitioning (2D)**
- Hypergraph partitioning for all methods
  - Zoltan (Sandia) with PaToH (Catalyurek)
  - Allows "fair" comparison between methods
- Vertex separators derived from edge separators
  - MatchBox (Purdue: Pothen, et al.)
- Heuristic used to partition separators

Communication Volume - Symmetric Matrices

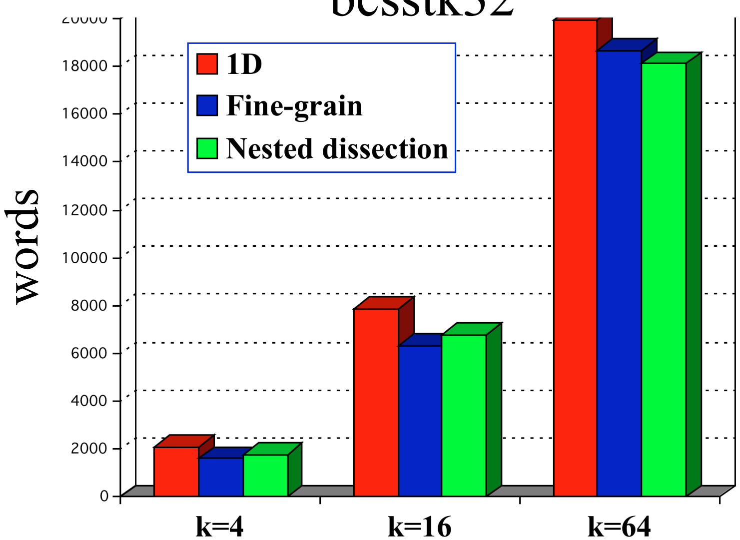
cage10



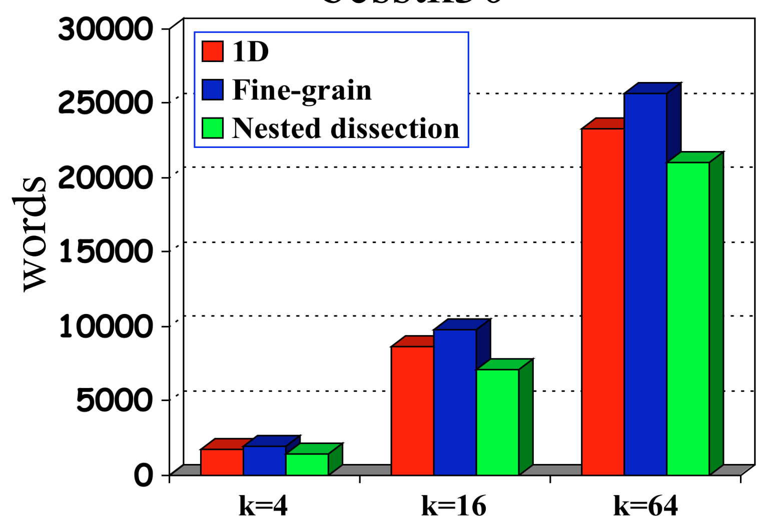
finan512



bcsstk32

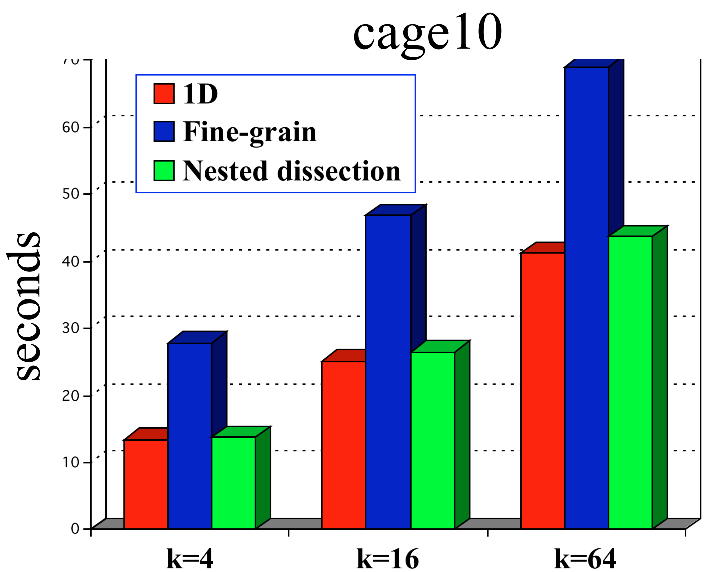


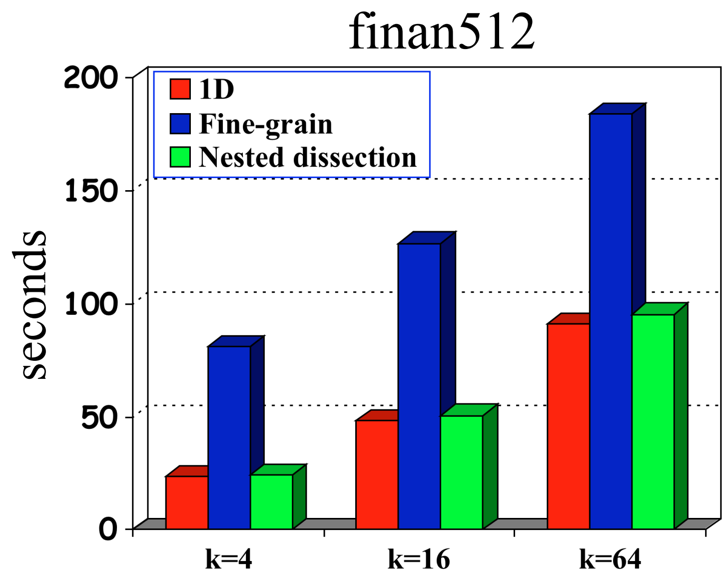
bcsstk30

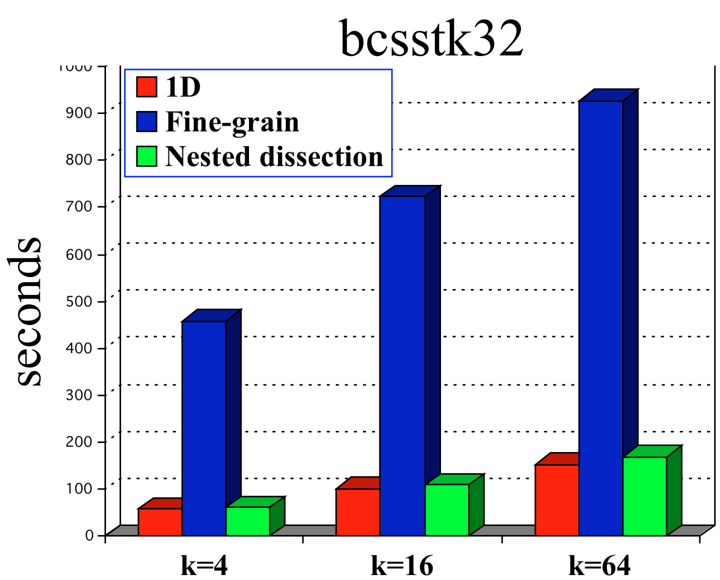


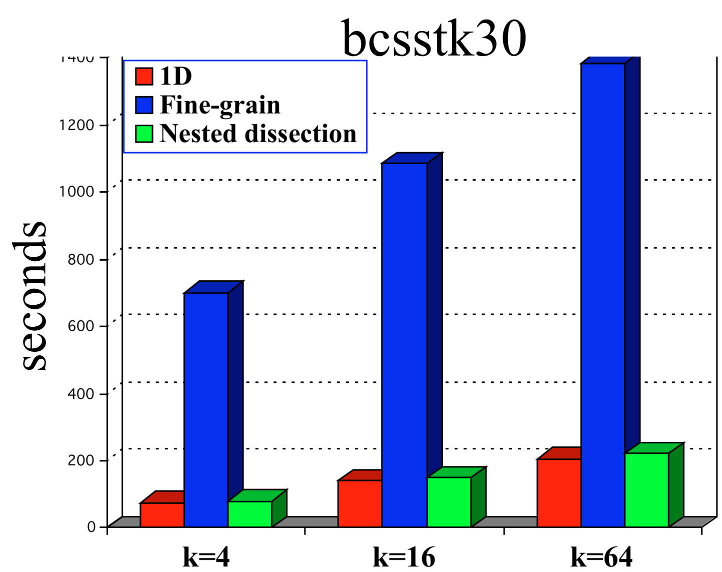
Test matrices from Rob Bisseling (Utrecht)

Runtimes of Partitioning Methods

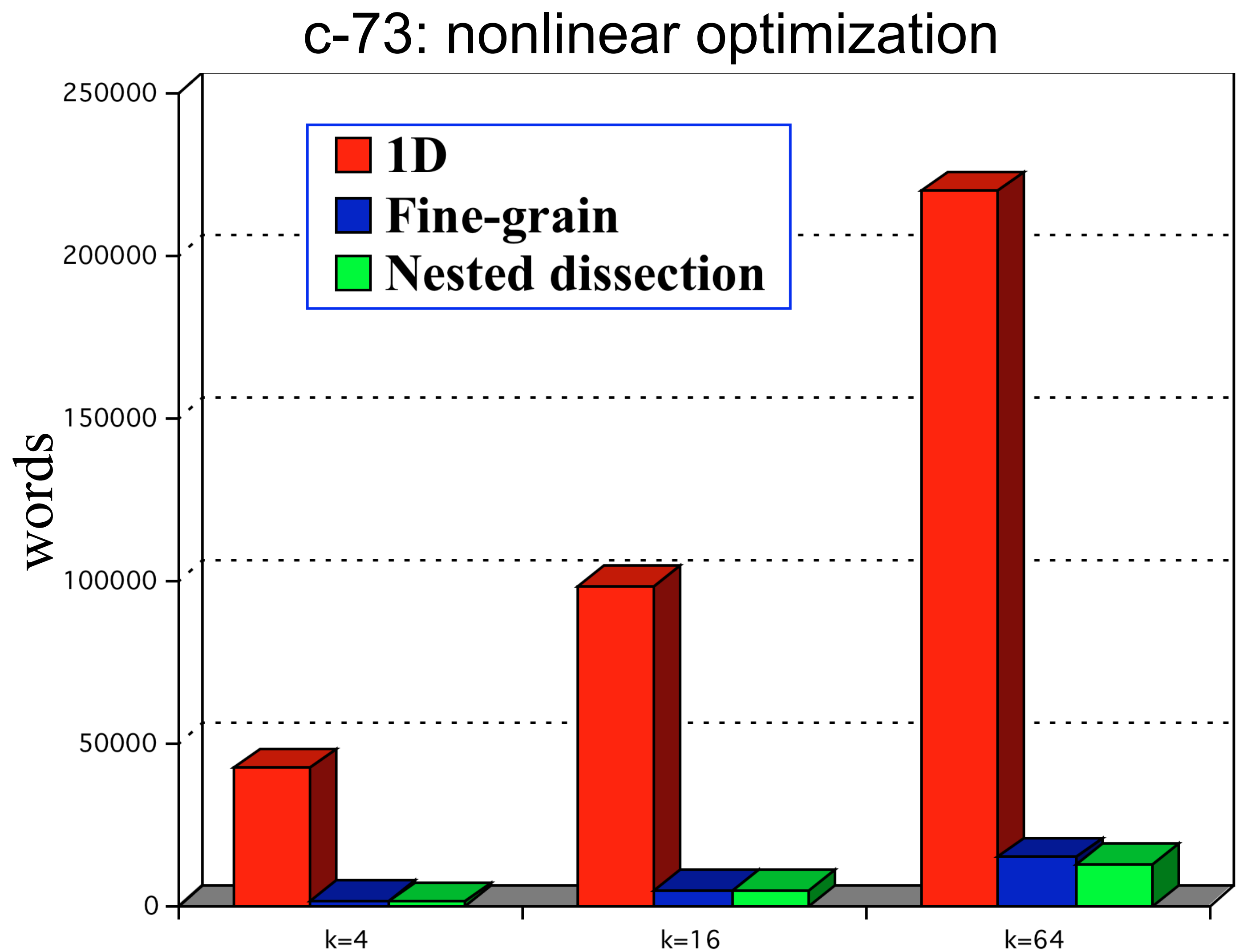




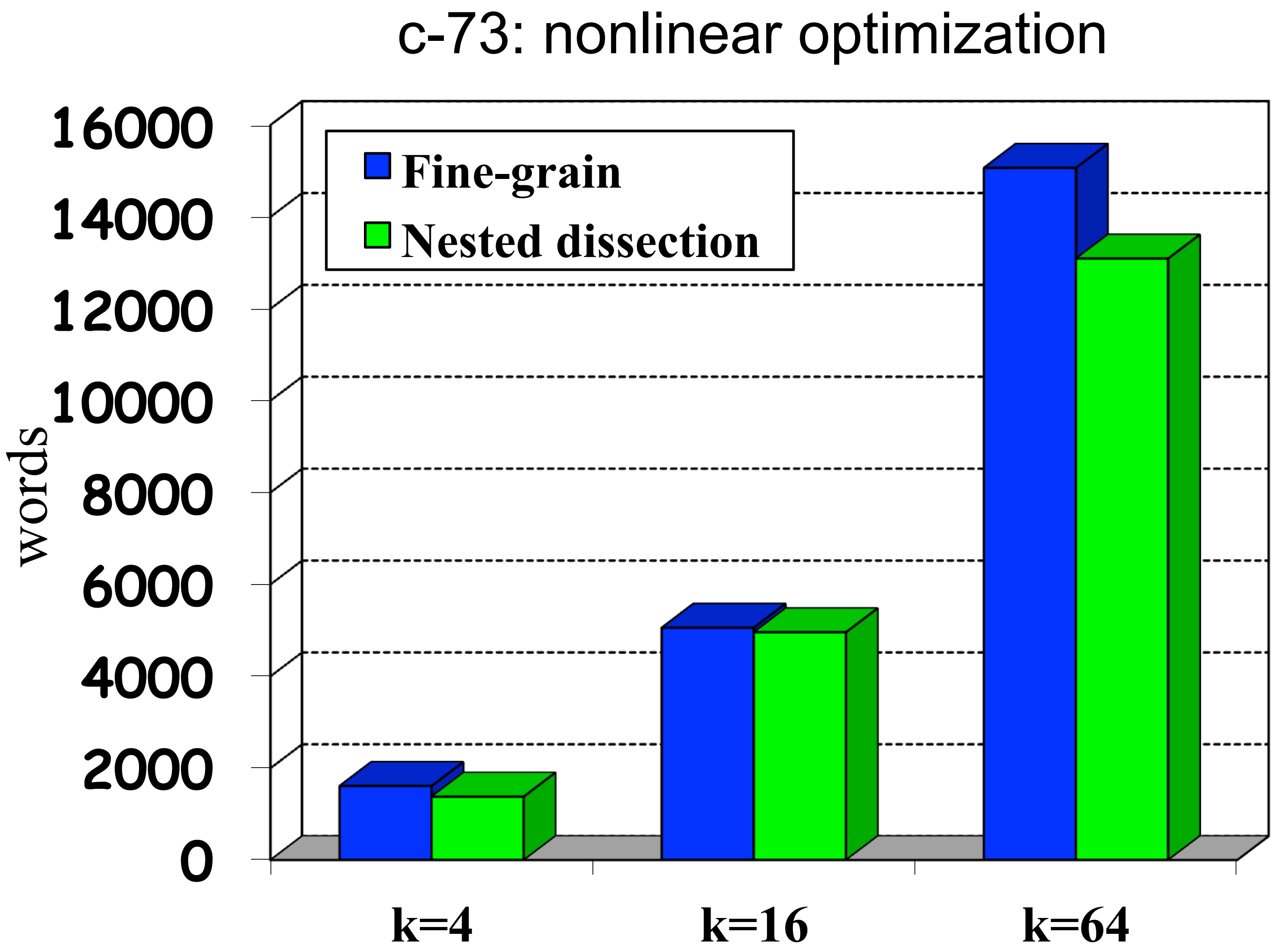




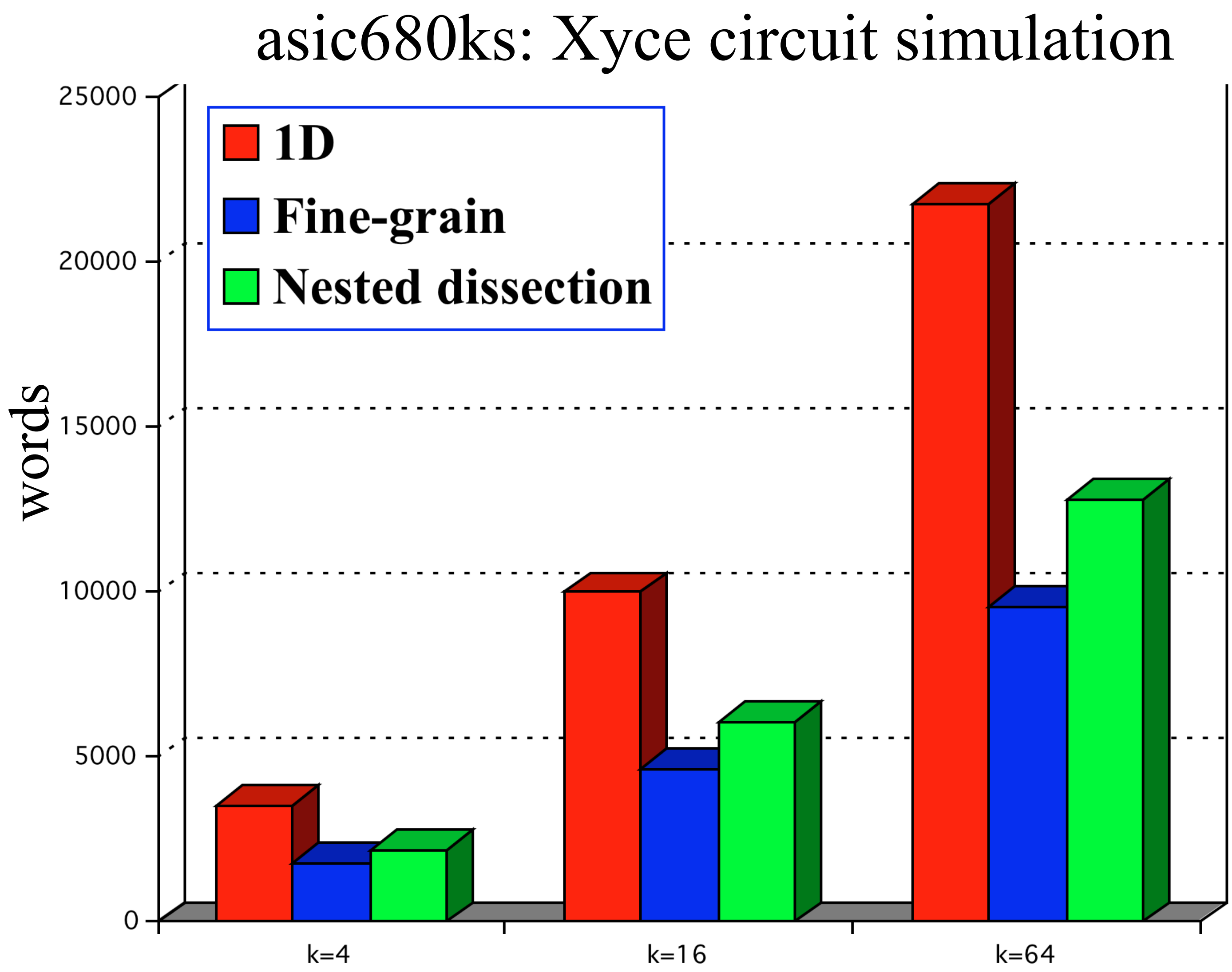
Communication Volume: 1D is Inadequate



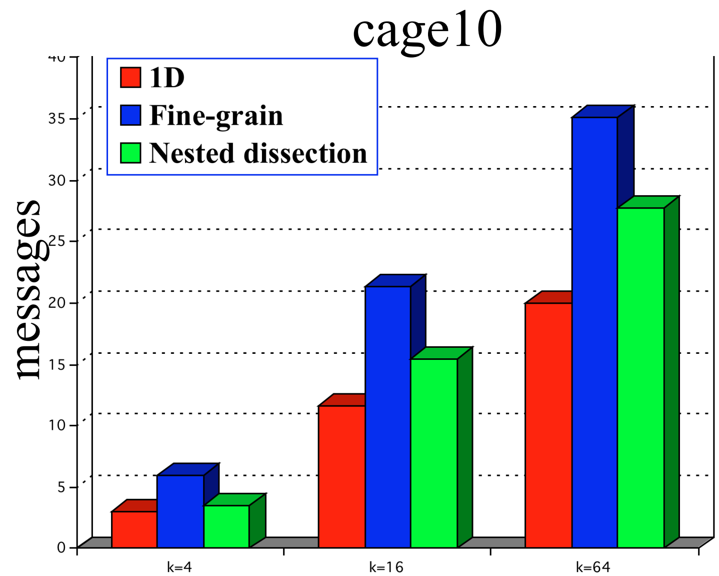
Communication Volume: 1D is Inadequate

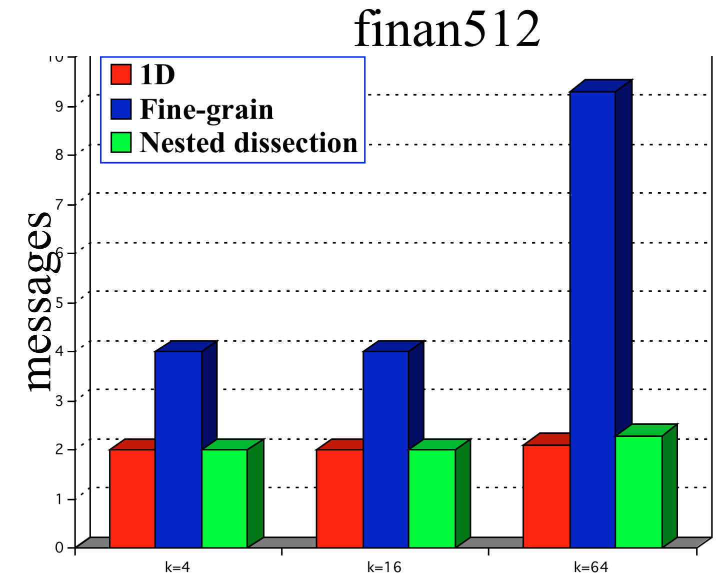


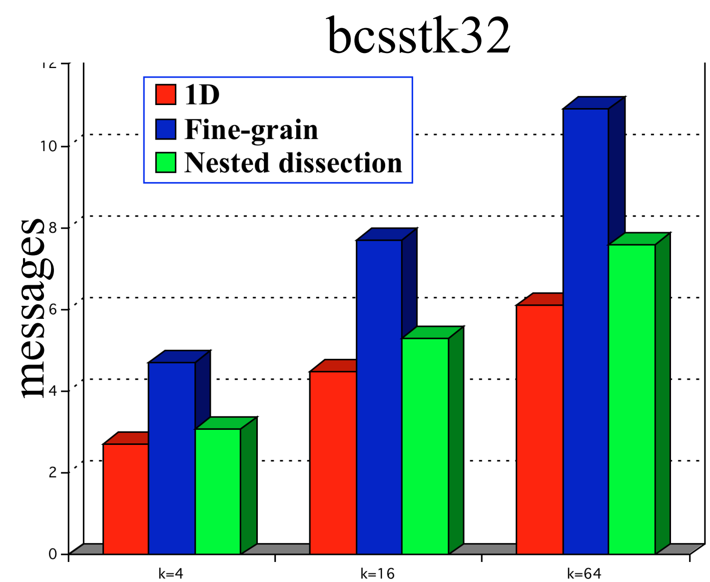
Communication Volume: 1D is Inadequate

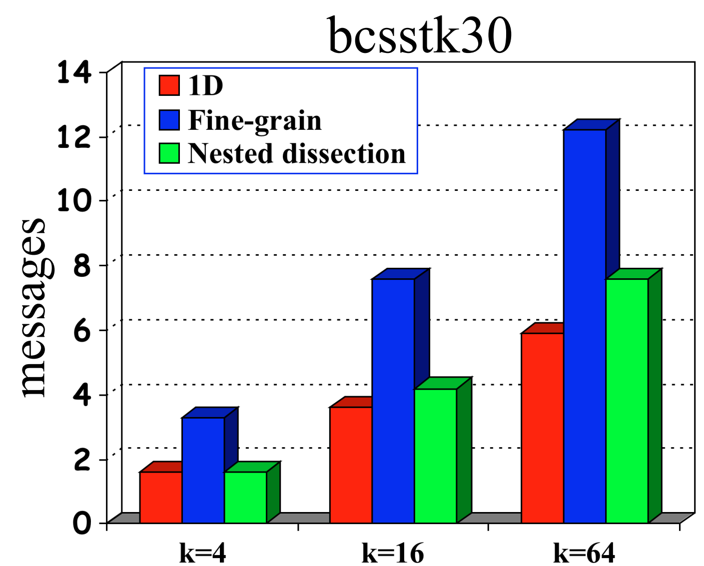


Another Important Metric: Messages Sent/Received

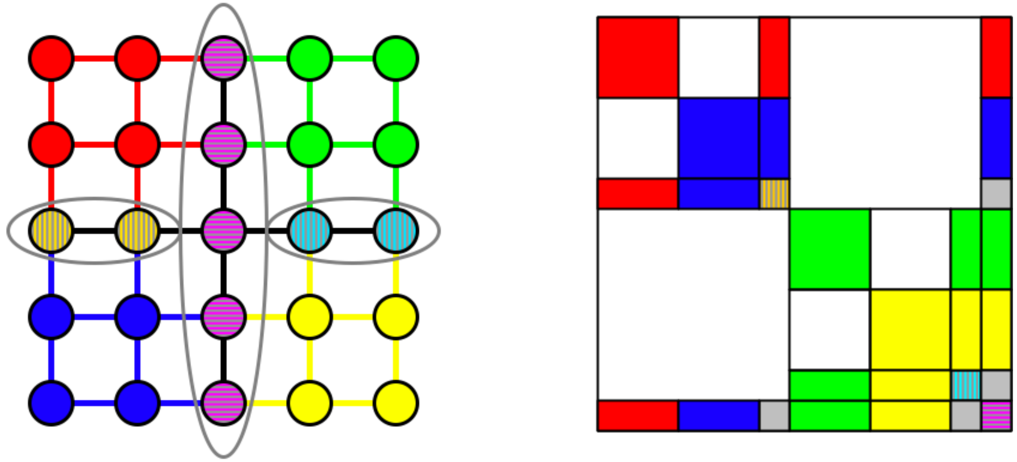






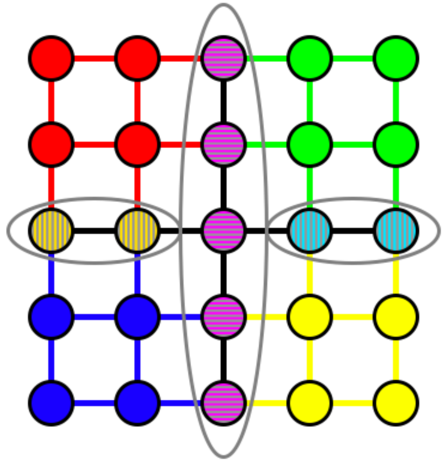


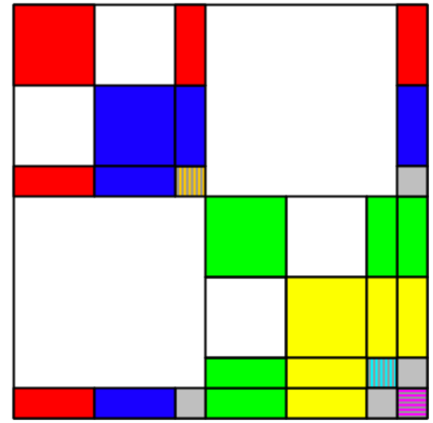
Summary



- New 2D matrix partitioning algorithm
  - Nested dissection used in new context
  - Good trade off between communication volume and partitioning time
    - Communication volume (comparable to fine-grain)
    - Partitioning time (comparable to 1D)
    - Also, fewer messages than fine-grain
- ND method partitioning effective for some matrices

Future Work





- Integrate ND partitioning algorithm into paralllel numerical software framework (e.g., Trilinos)
  - Boman, et al. (SNL)
  - Isorropia, Zoltan2 packages
- Analysis of runtimes of SpMV using ND partitioning method
- Partitioning of scale-free networks with ND method
  - 2D methods are important for these problems
  - Finding balanced separator challenging