Improved Data Partitioning by Nested Dissection with Applications to Information Retrieval

SIAM CSC09

10/29/2009

***Michael Wolf*, Erik Boman, Cédric Chevalier**

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.

Sparse Matrix Partitioning Motivation

- Sparse matrix-vector multiplication (SpMV) is common kernel in many numerical computations
  - Iterative methods for solving linear systems
  - PageRank computation
  - …
- Need to make parallel SpMV kernel as fast as possible

Parallel Sparse Matrix-Vector Multiplication

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

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

- Partition matrix nonzeros
- Partition vectors

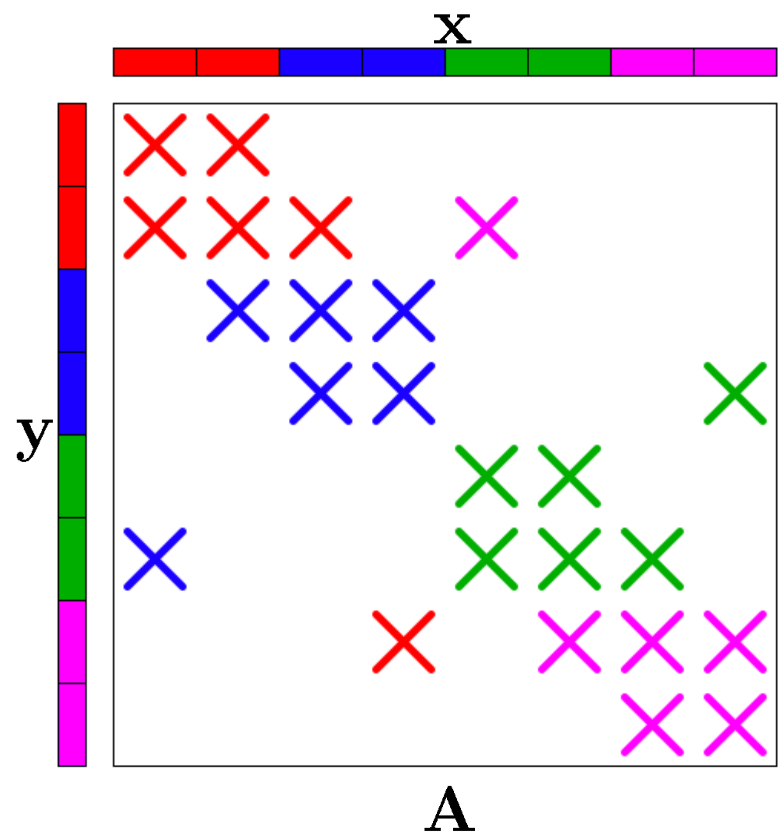
Objective

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

Parallel Matrix-Vector Multiplication

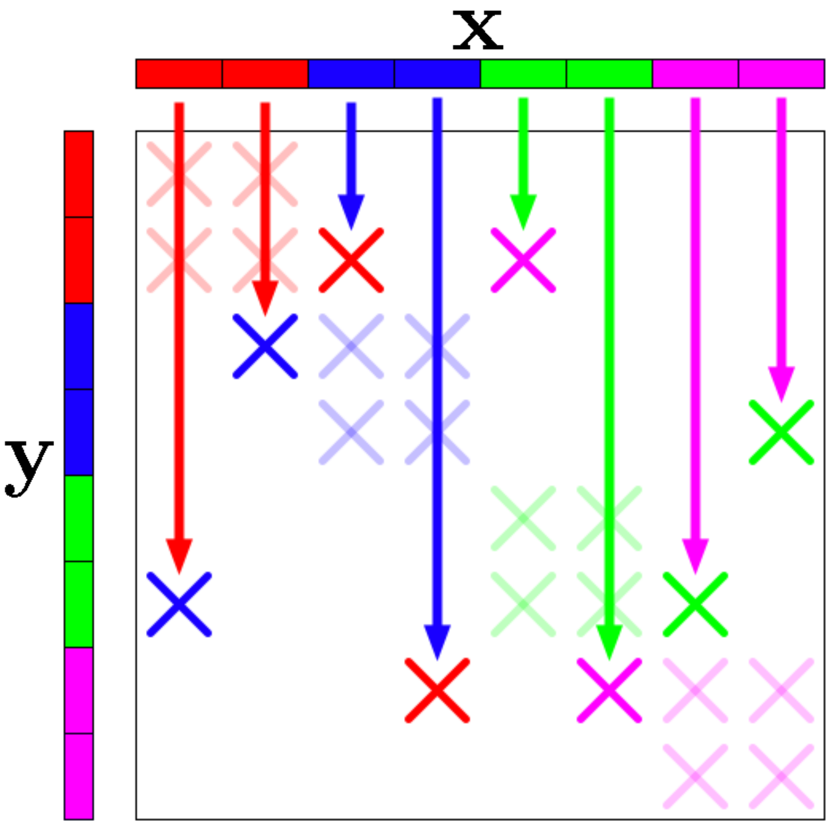
$$
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix}
=
\begin{bmatrix}
1 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & 1 & 9 & 0 & 5 & 0 & 0 & 0 \\
0 & 8 & 1 & 7 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 1 & 0 & 0 & 0 & 7 \\
0 & 0 & 0 & 0 & 1 & 8 & 0 & 0 \\
4 & 0 & 0 & 0 & 3 & 1 & 3 & 0 \\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

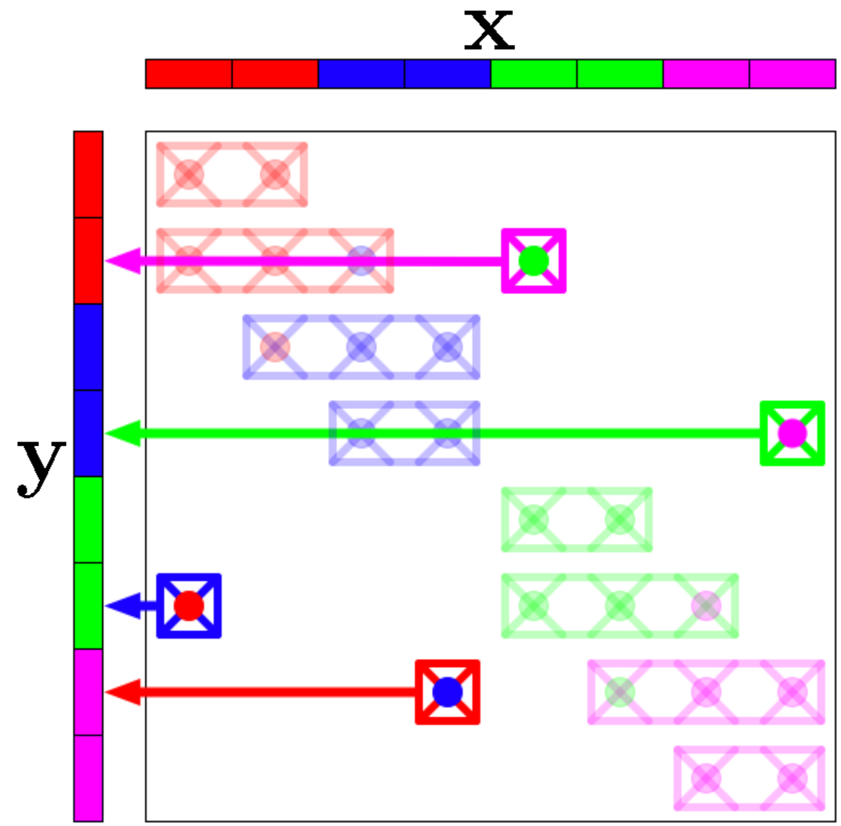
$$\mathbf{y} = \mathbf{A}\mathbf{x}$$



- Alternative way of visualizing partitioning

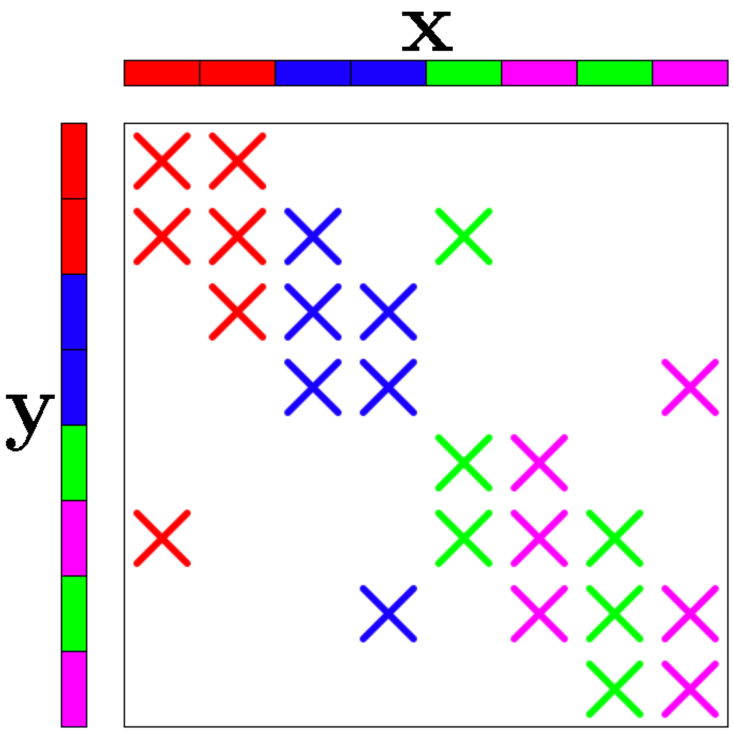
Parallel SpMV Communication



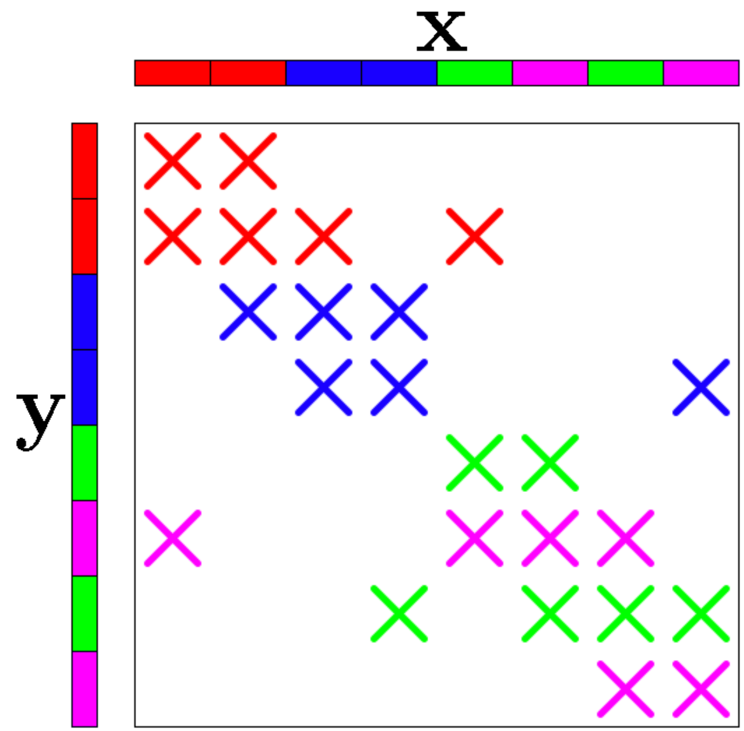


- x_j sent to remote processes that have nonzeros in column j
- Partial inner-products sent to process that owns vector element y_i

1D Partitioning



1D Column

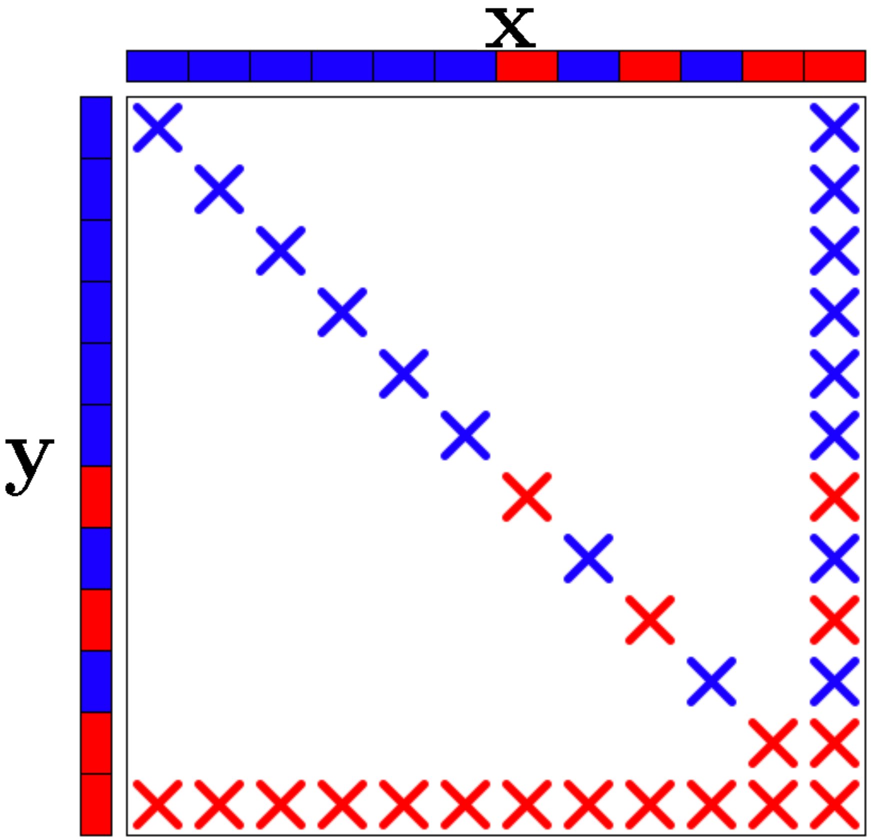


1D Row

- Each process assigned nonzeros for set of columns

- Each process assigned nonzeros for set of rows

When 1D Partitioning is Inadequate



"Arrowhead" matrix

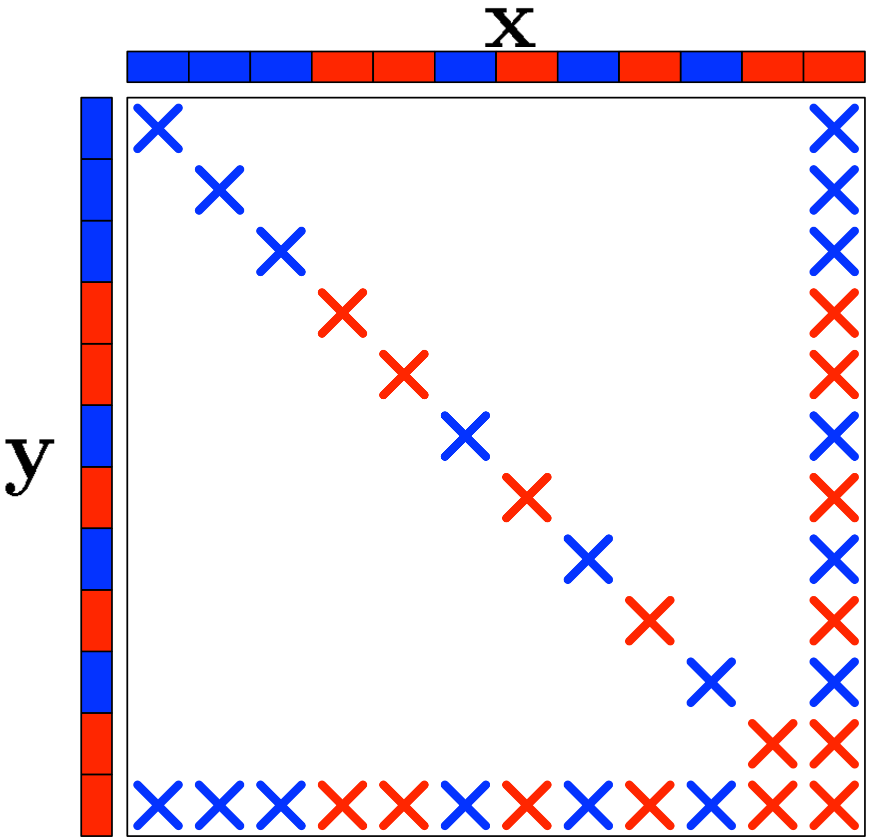
n=12

nnz=34 (18,16)

volume = 9

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

When 1D Partitioning is Inadequate



"Arrowhead" matrix

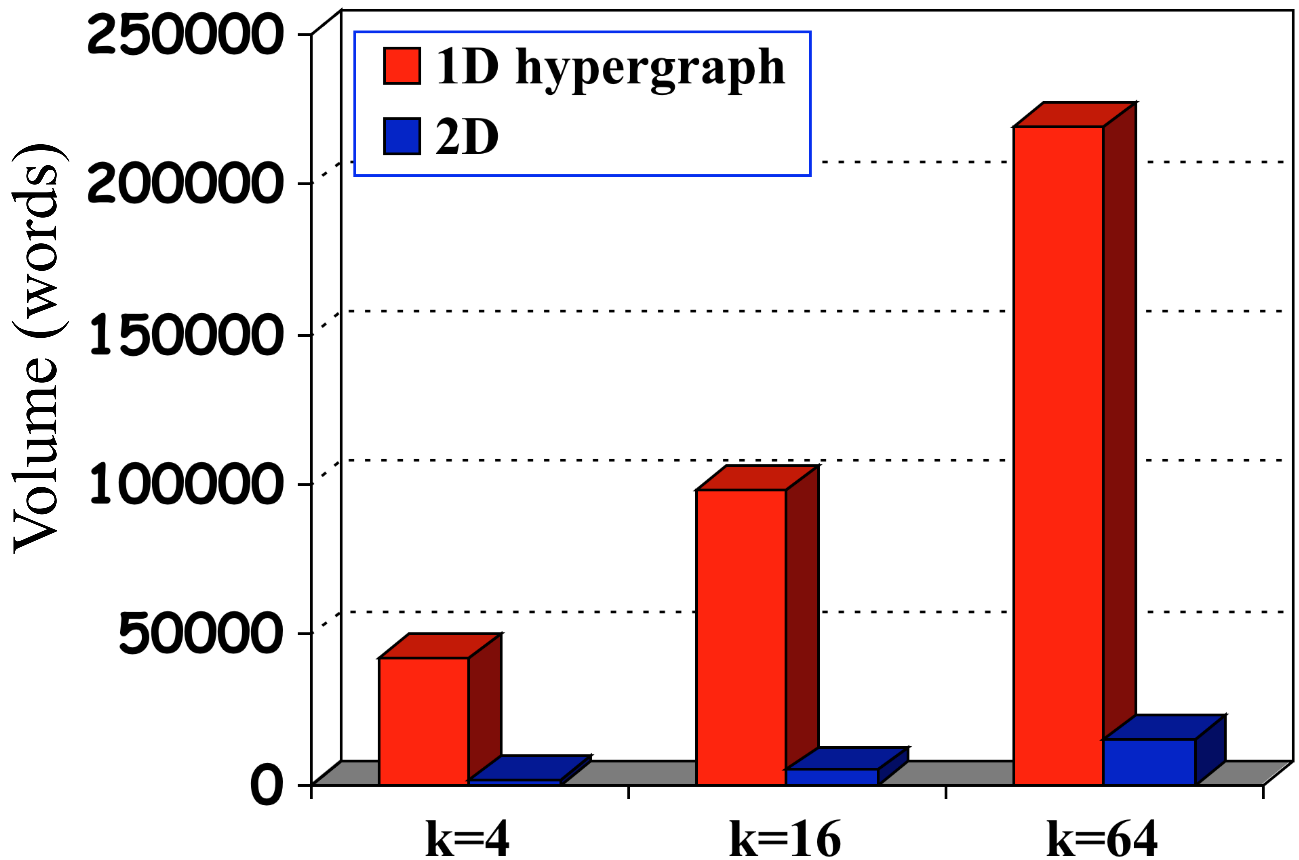
n=12

nnz=34 (16,18)

volume = 2

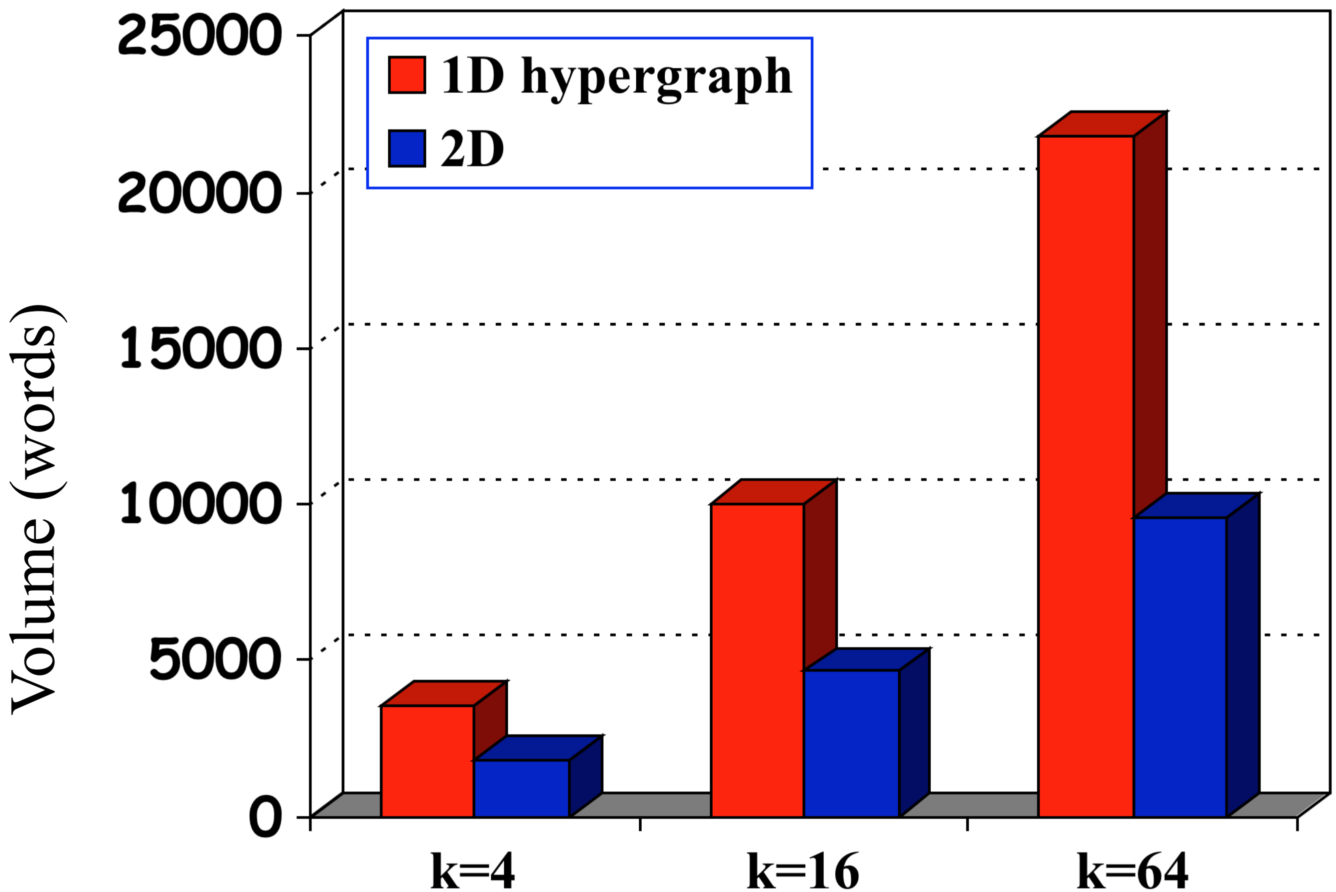
- 2D partitioning
- O(k) volume partitioning possible

1D is Inadequate



- c-73: nonlinear optimization (Schenk)
  - UF sparse matrix collection
  - n=169,422 nnz=1,279,274

1D is Inadequate



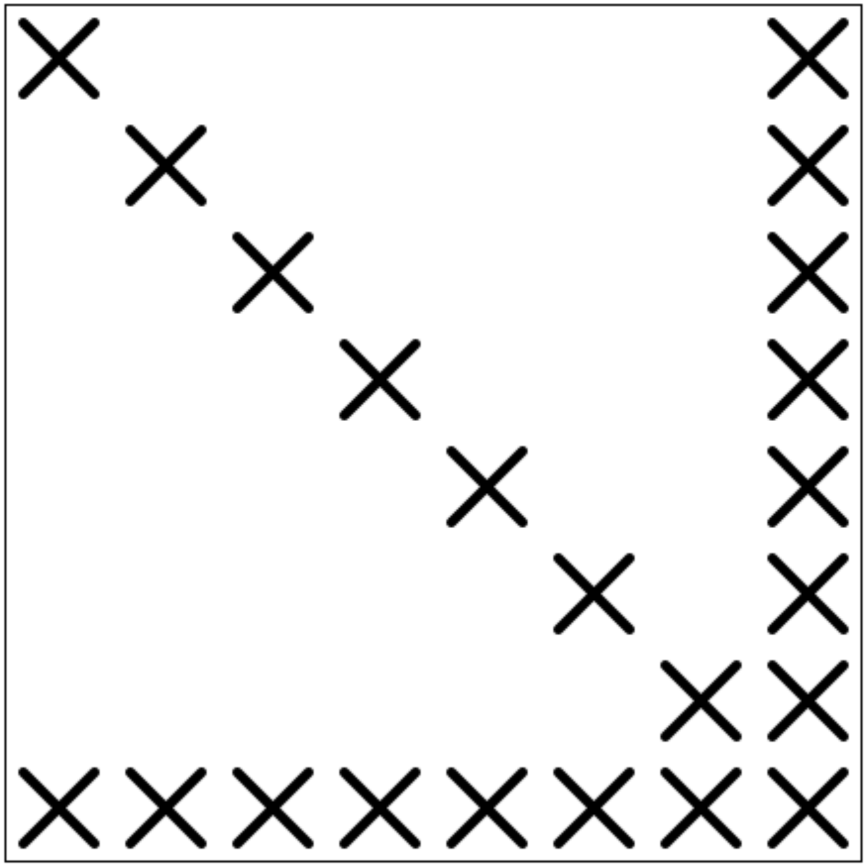
- asic680ks: Xyce circuit simulation (Sandia)
  - n=682,712 nnz=2,329,176

2D Partitioning

- More flexibility in partitioning
- No particular part for given row or column
- More general sets of nonzeros assigned parts
- Several methods of 2D partitioning
  - Fine-grain hypergraph, Mondriaan, …

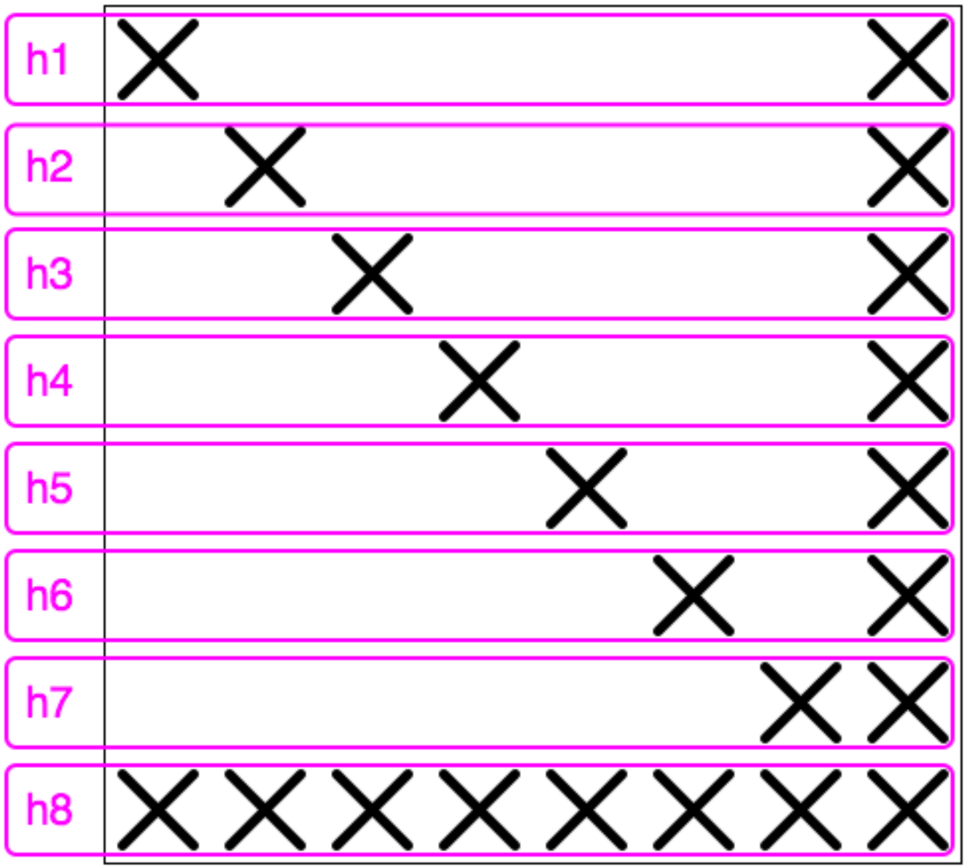
- Fine-grain hypergraph
- Graph model for symmetric 2D partitioning
- Nested dissection symmetric partitioning method

Fine-Grain (FG) Hypergraph Model



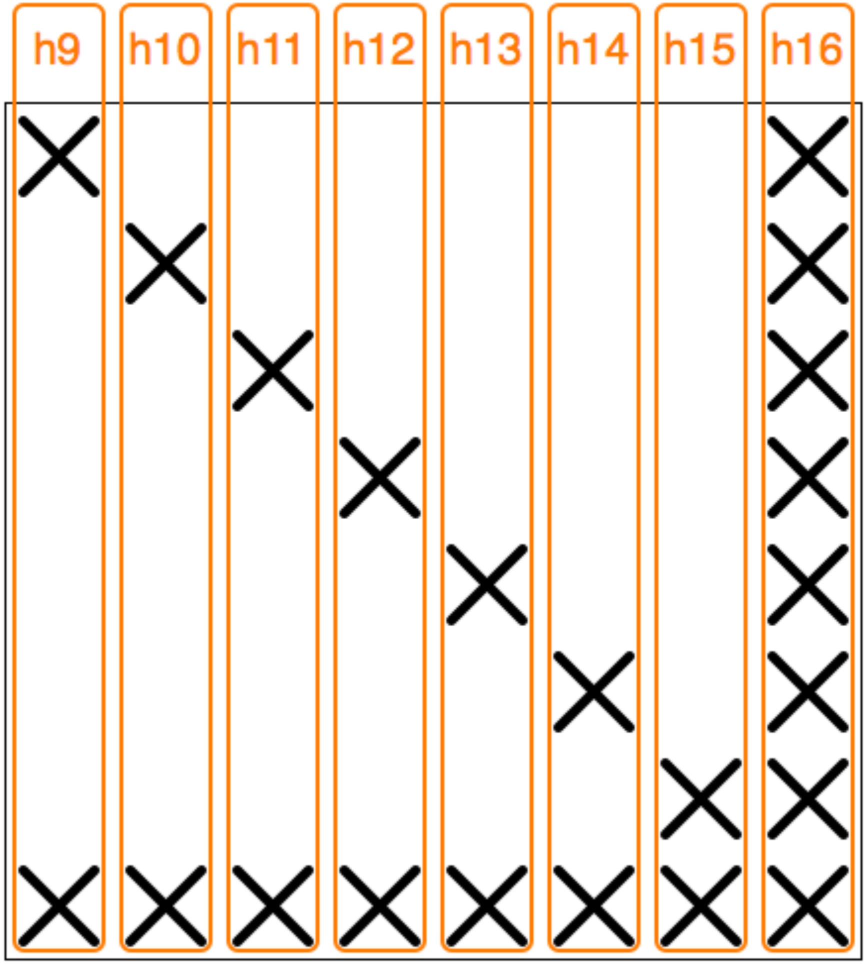
- Catalyurek and Aykanat (2001)
- Nonzeros represented by vertices in hypergraph

Fine-Grain Hypergraph Model



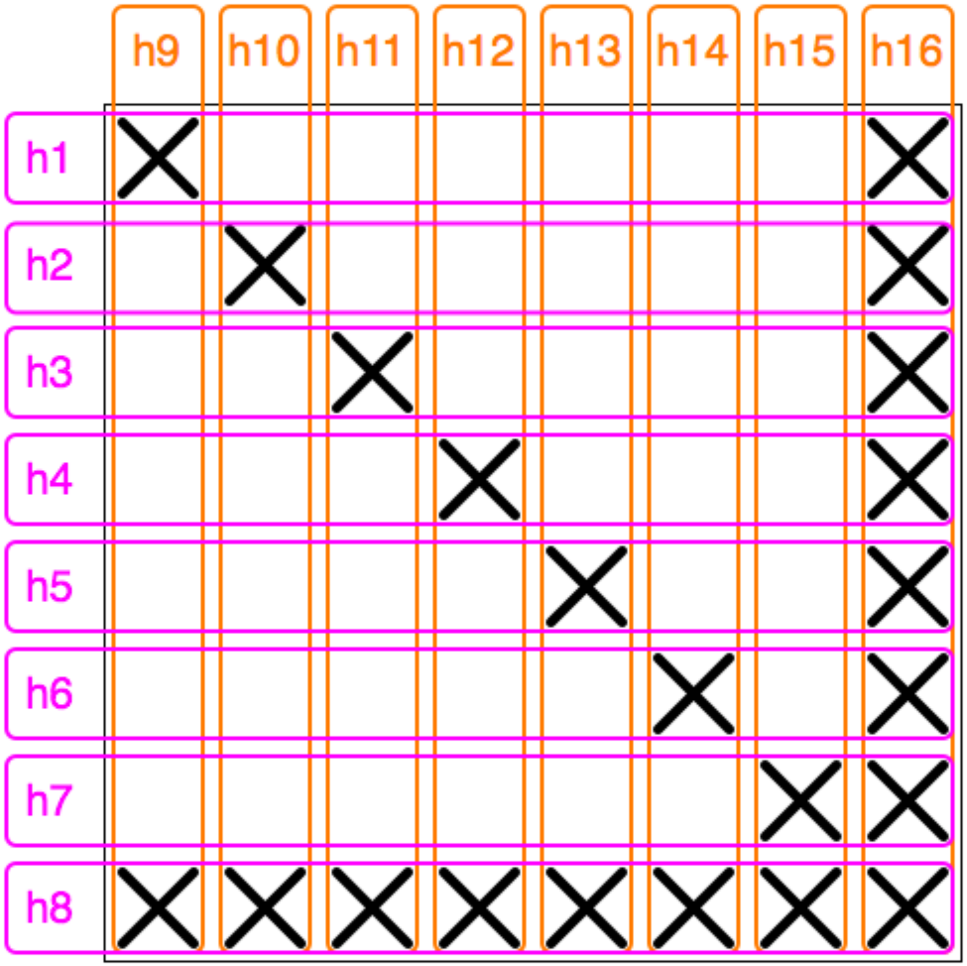
- Rows represented by hyperedges
- Hyperedge - set of one or more vertices

Fine-Grain Hypergraph Model



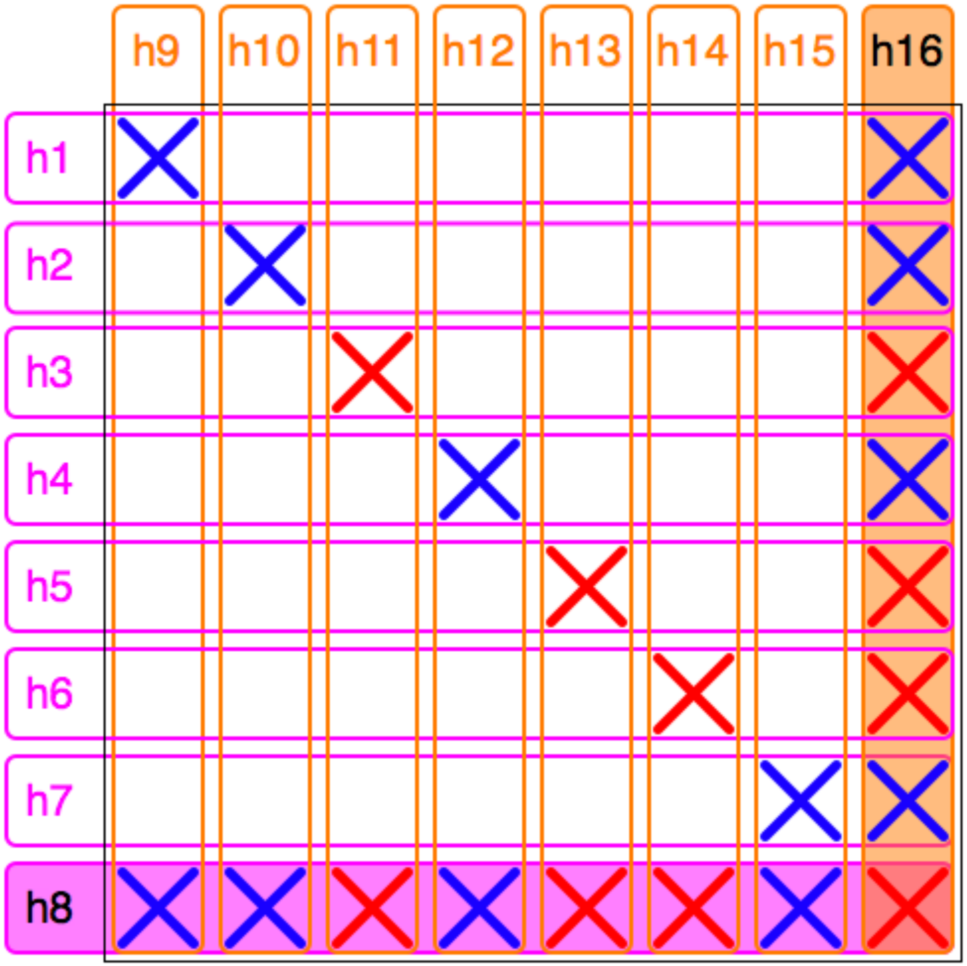
- Columns represented by hyperedges

Fine-Grain Hypergraph Model



- 2n hyperedges

Fine-Grain Hypergraph Model



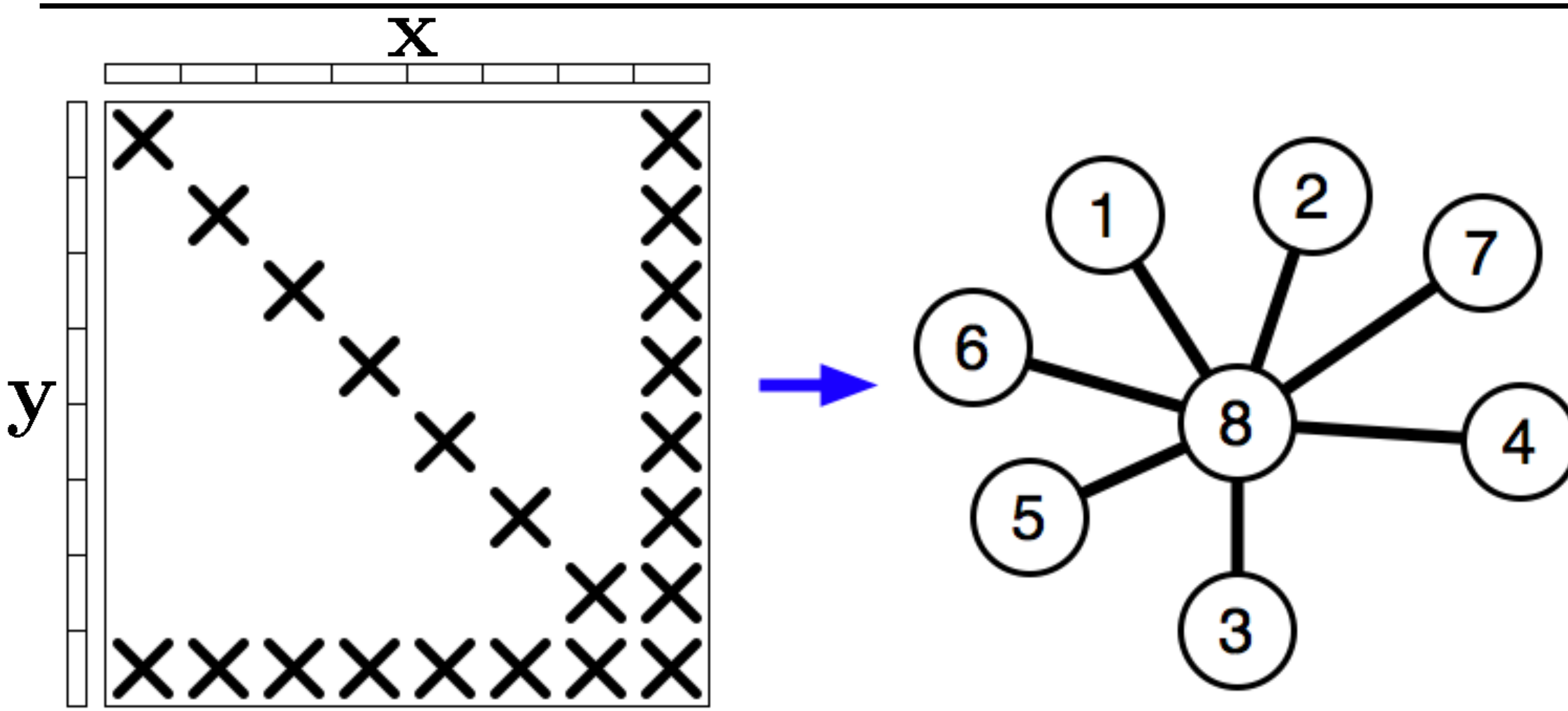
k=2, volume = cut = 2

- 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

Graph Model for Symmetric 2D Partitioning

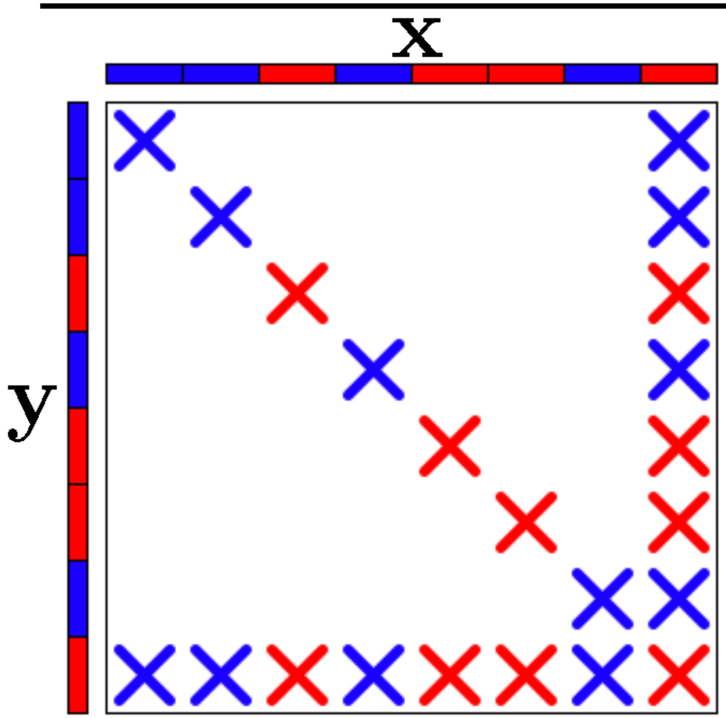
- Exact model of communication for symmetric 2D partitioning
- Given matrix A with symmetric nz structure
- Symmetric partition
  - a(i,j) and a(j,i) assigned same part
  - Input and output vectors have same distribution
- Corresponding graph G(V,E)
  - Vertices correspond to vector elements
  - Edges correspond to off-diagonal nonzeros

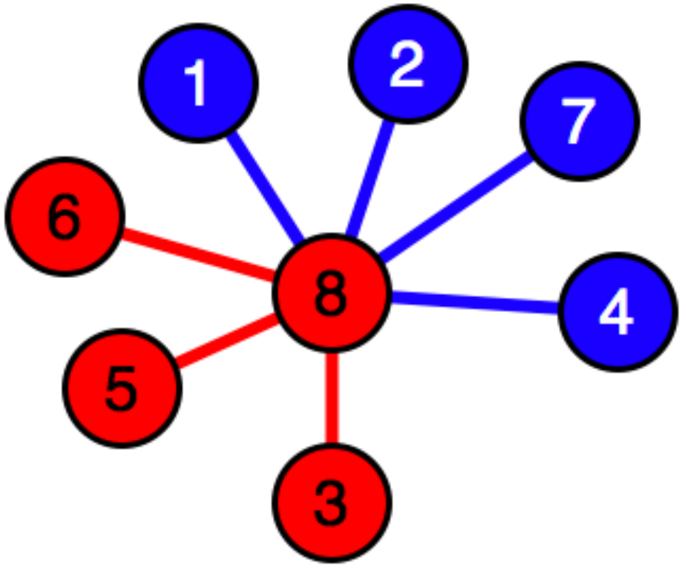
Graph Model for Symmetric 2D Partitioning



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

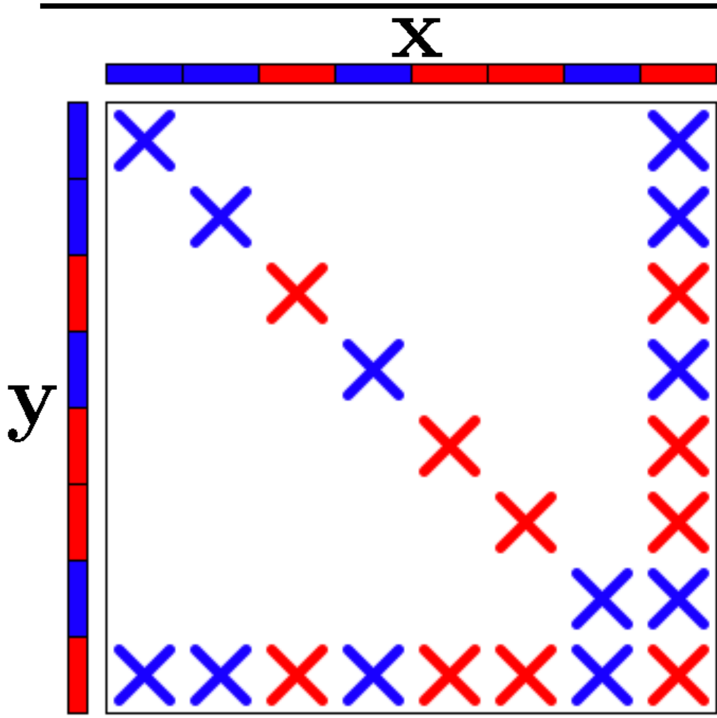
Graph Model for Symmetric 2D Partitioning

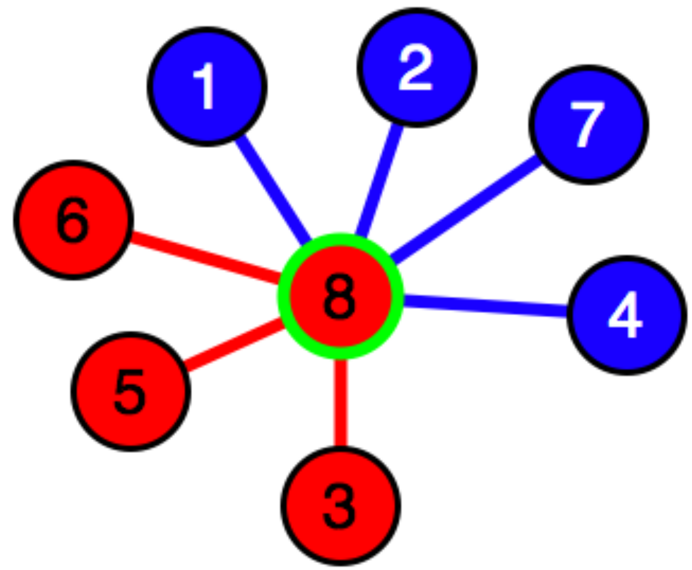




- Symmetric 2D partitioning
  - Partition both V and E
  - Gives partitioning of both matrix and vectors

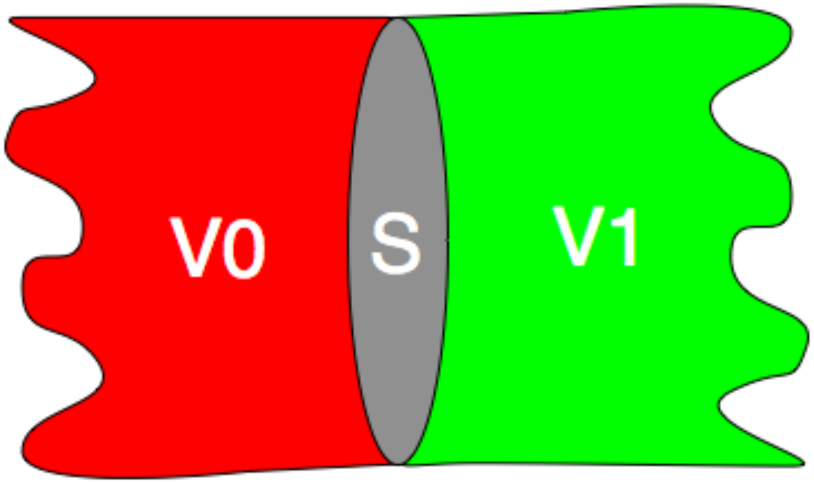
Communication in Graph Model





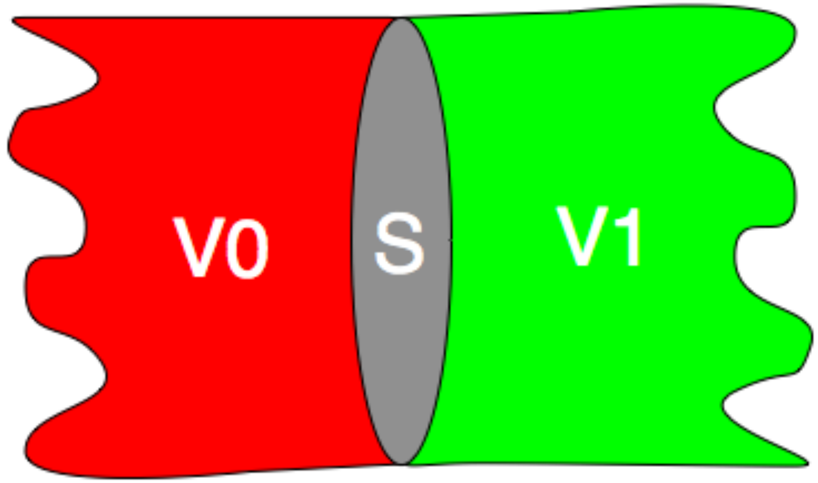
- Communication is assigned to vertices
- Vertex incurs communication iff incident edge is in different part
- Want small vertex separator -- $S=\{V_8\}$
- For bisection, volume = 2 |S|

Nested Dissection Partitioning - Bisection



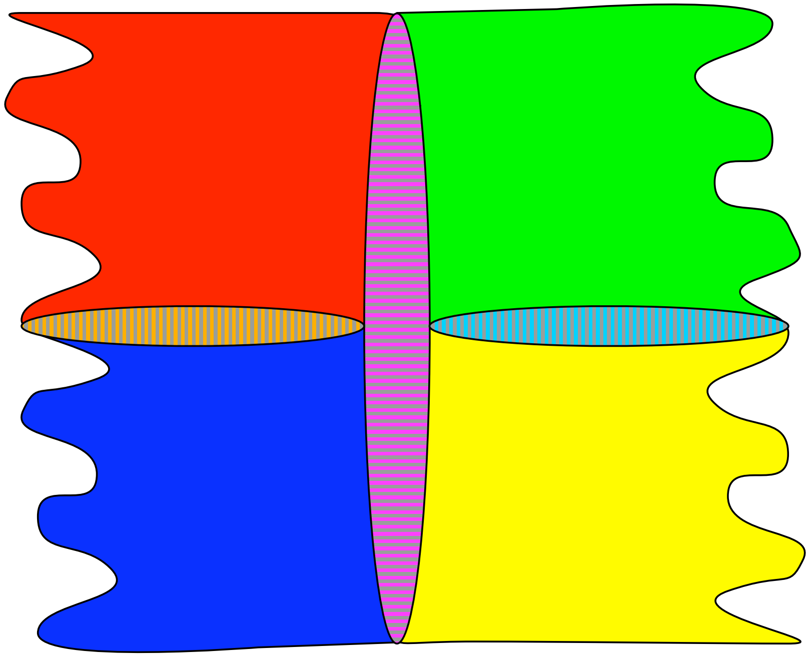
- Suppose A is structurally symmetric
- Let G(V,E) be graph of A
- Find small, balanced separator S
  - Yields vertex partitioning V = (**V0**,**V1**,S)
- Partition the edges such that
  - **E0** = {edges incident to a vertex in **V0**}
  - **E1** = {edges incident to a vertex in **V1**}

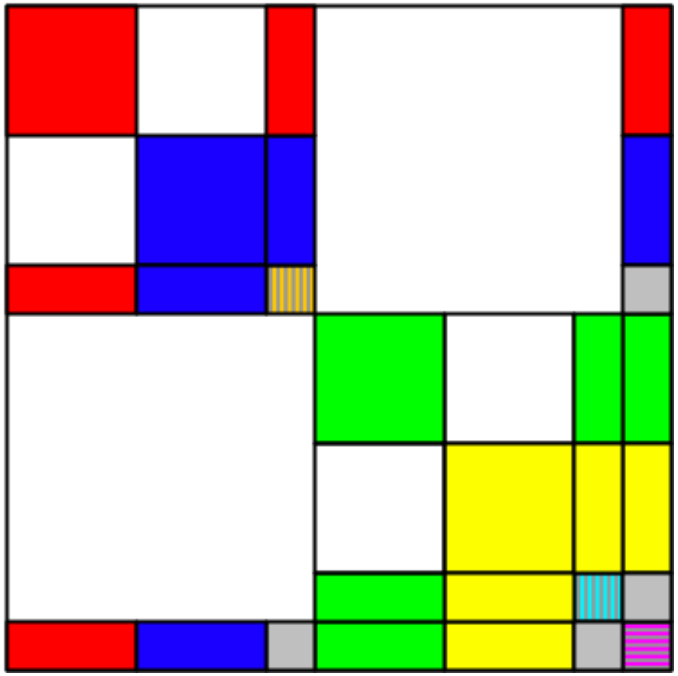
Nested Dissection Partitioning - Bisection



- 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!

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

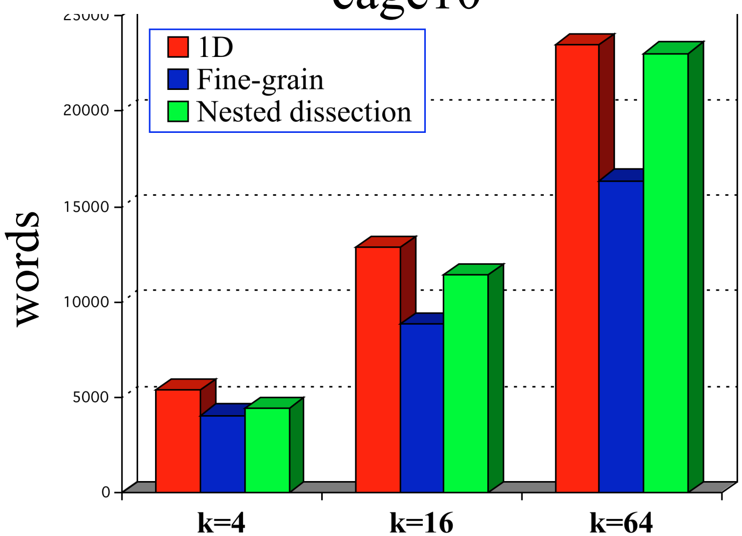
$$A' = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$$

Initial Numerical Experiments

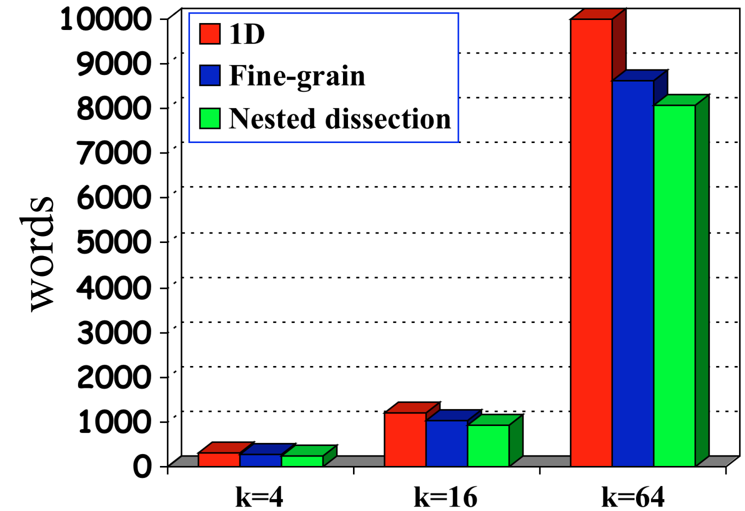
- Structurally symmetric matrices
- k = 4, 16, 64 parts using
  - 1D hypergraph partitioning
  - Fine-grain hypergraph partitioning (2D)
    - Good quality partitions but slow
  - Nested dissection partitioning (2D)
- Hypergraph partitioning for all methods
  - Zoltan (Sandia) with PaToH (Catalyurek)
  - 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

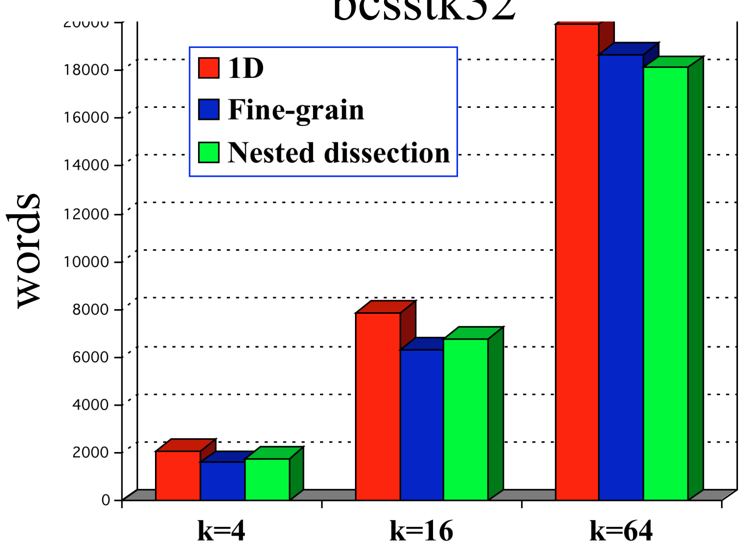
cage10



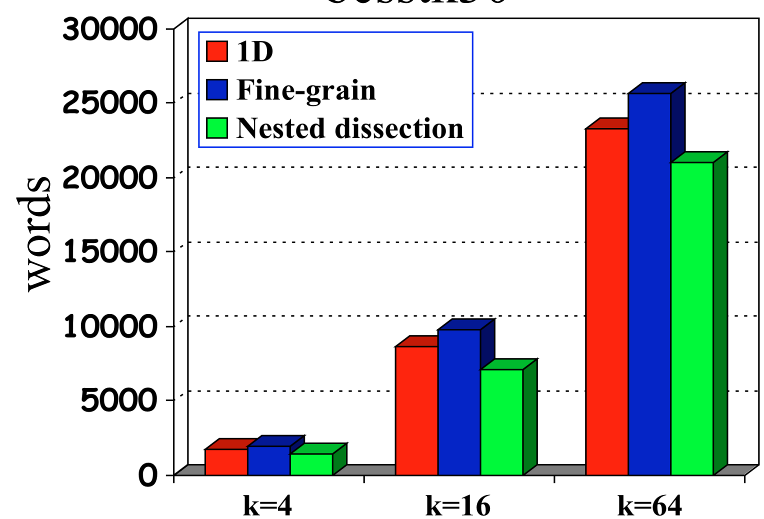
finan512



bcsstk32

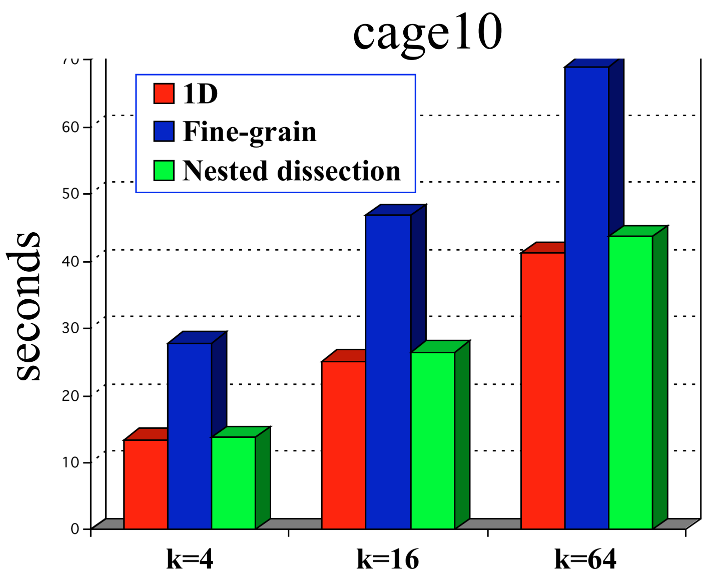


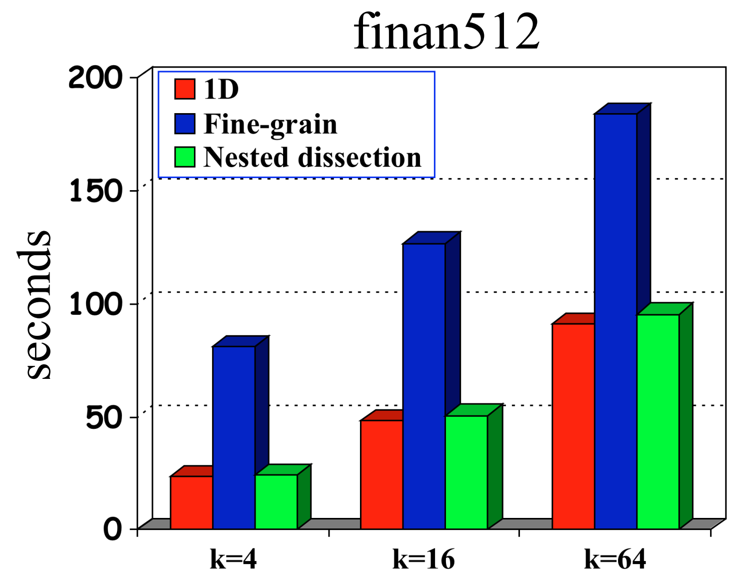
bcsstk30

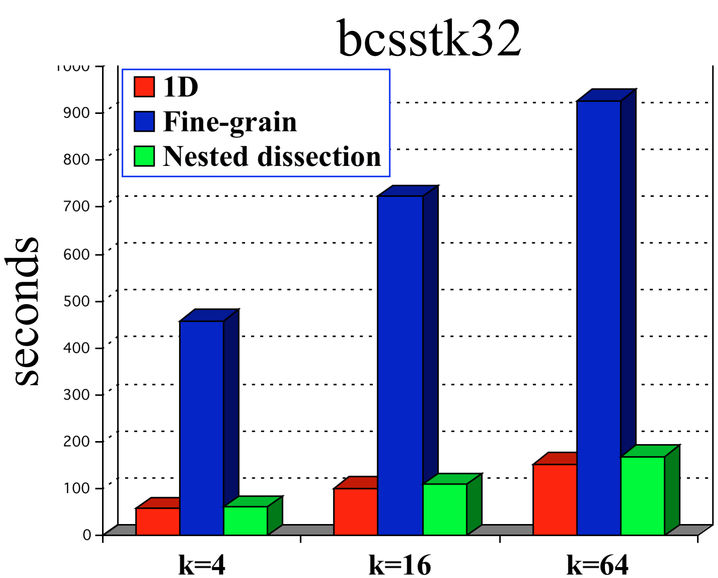


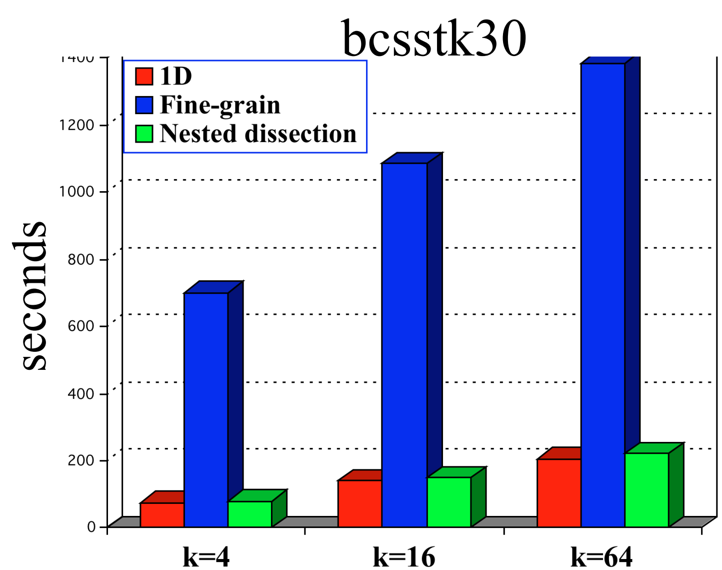
Test matrices from Rob Bisseling (Utrecht)

Runtimes of Partitioning Methods

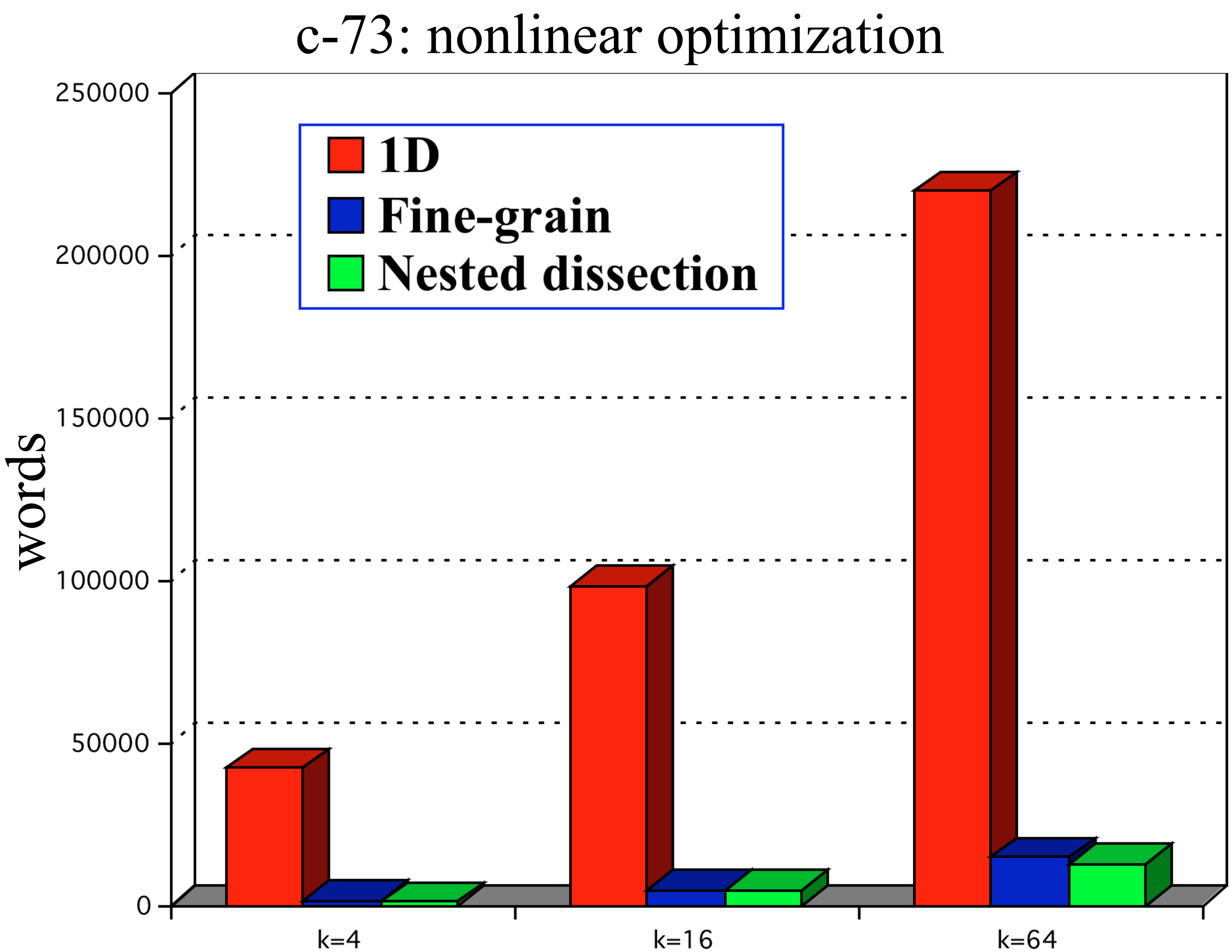




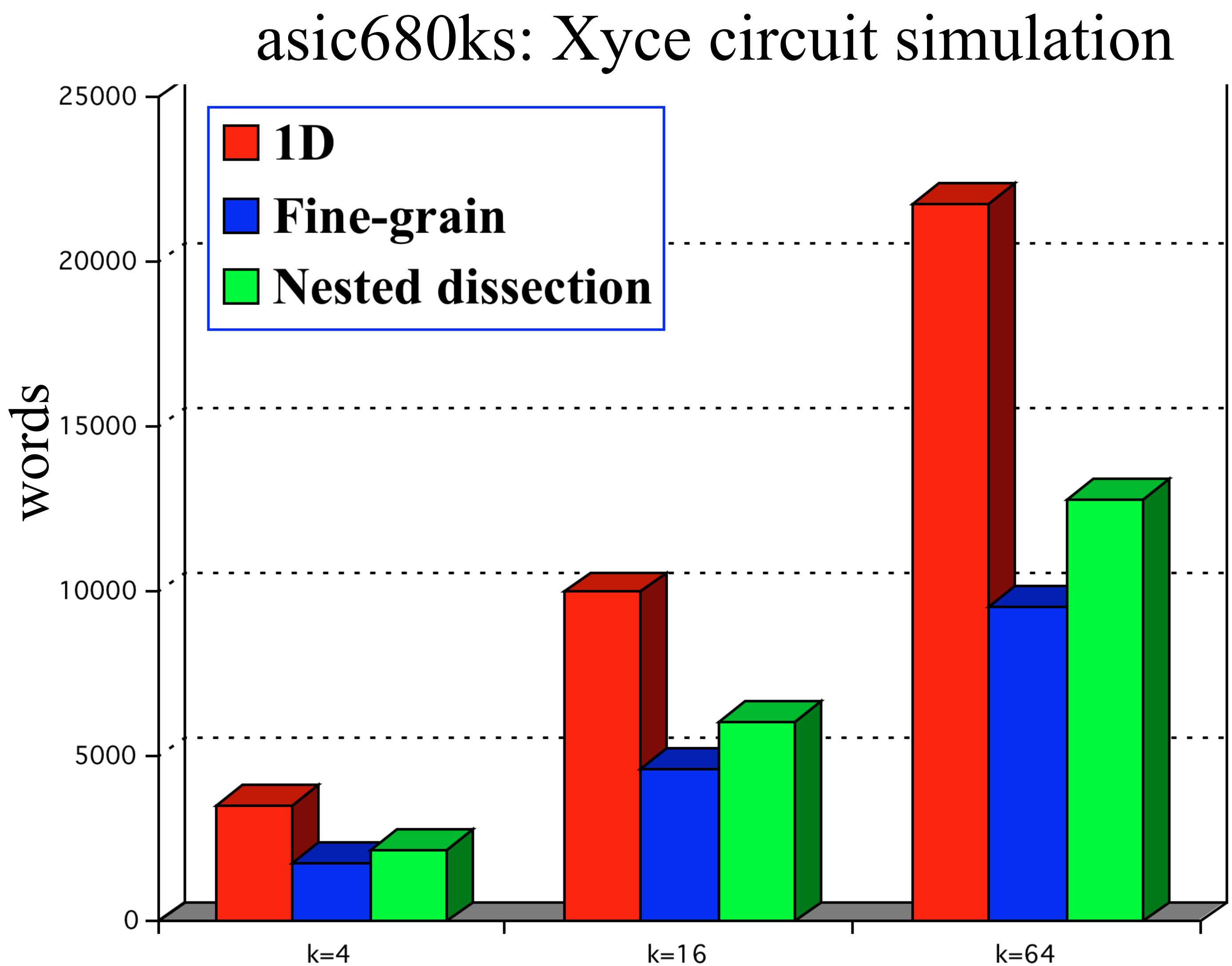




Communication Volume: 1D is Inadequate

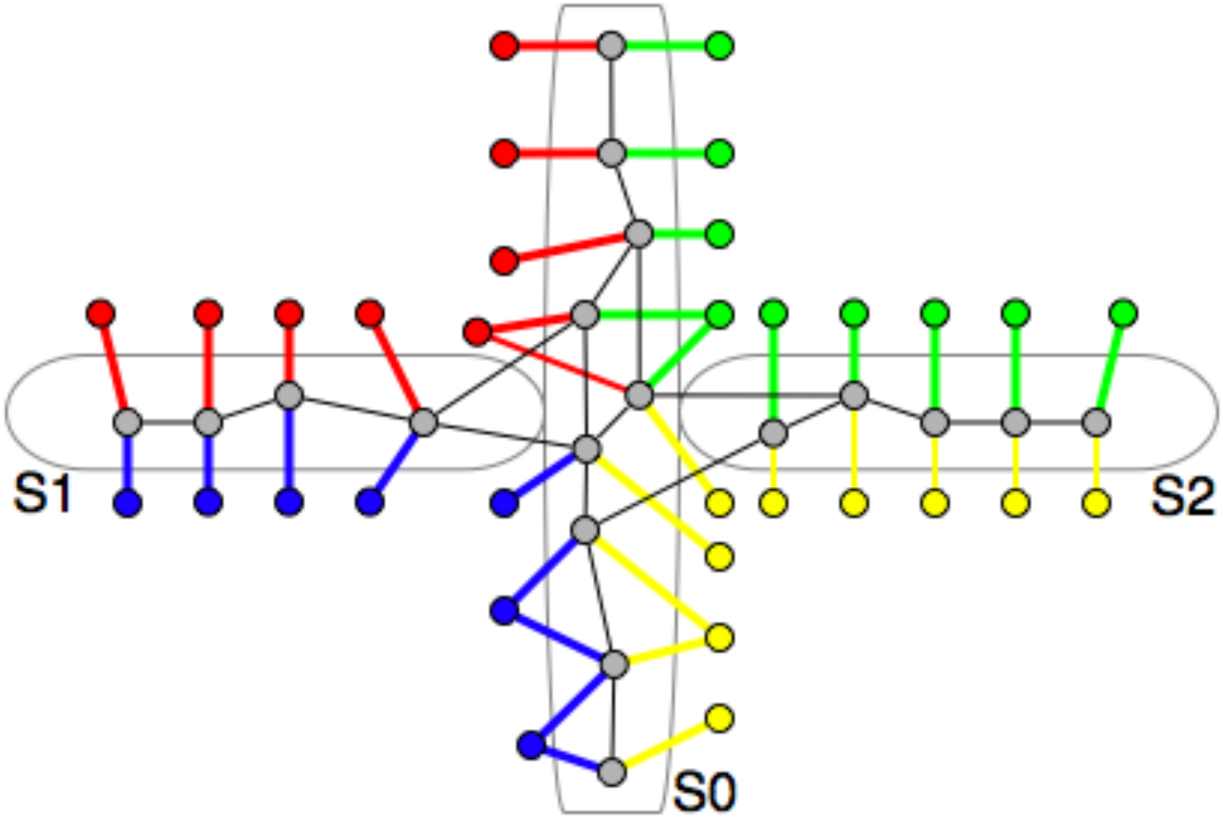


Communication Volume: 1D is Inadequate



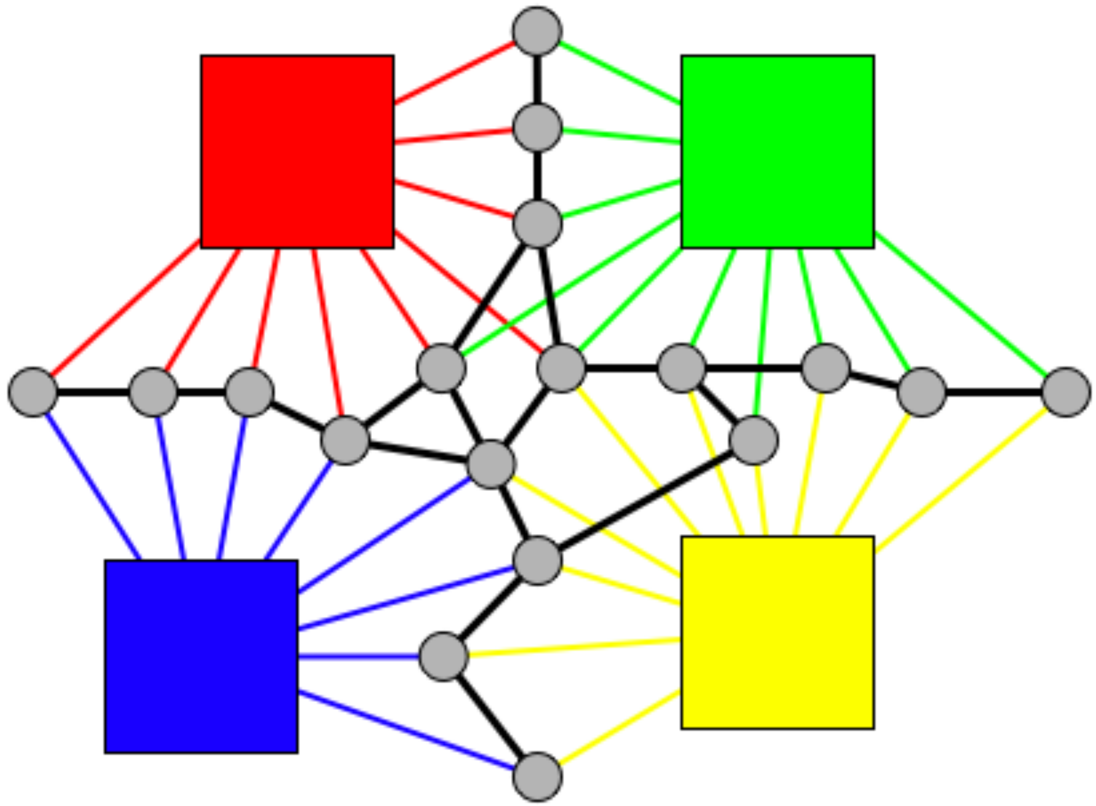
Sandia National Laboratories

Improving Separator Partitioning



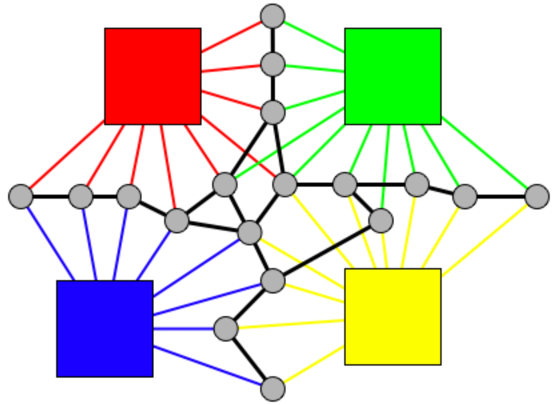
- Flexibility in how we partition separator vertices and separator-separator edges
- Original implementation used simple heuristic

Improved Separator Partitioning



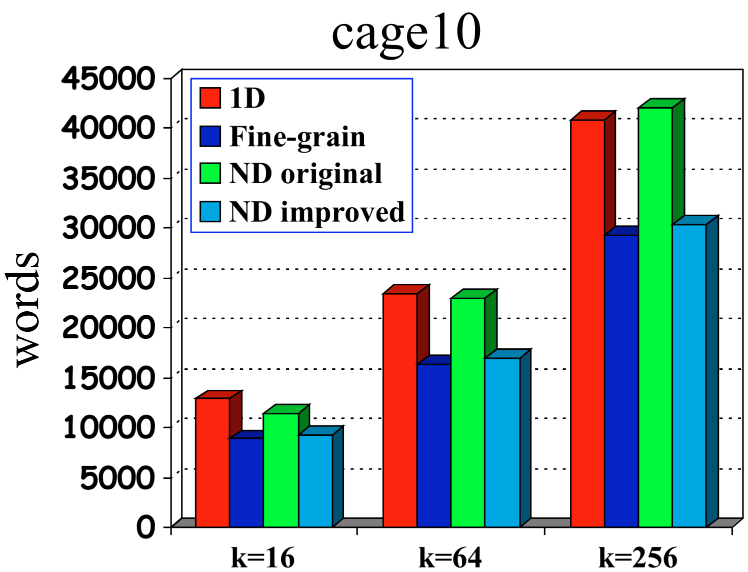
- Phase 2: partition separator vertices and edges
- Solve a second, much smaller partitioning problem
  - Fixed vertices/edges (1 vertex for each part)
  - Fine-grain hypergraph partitioning

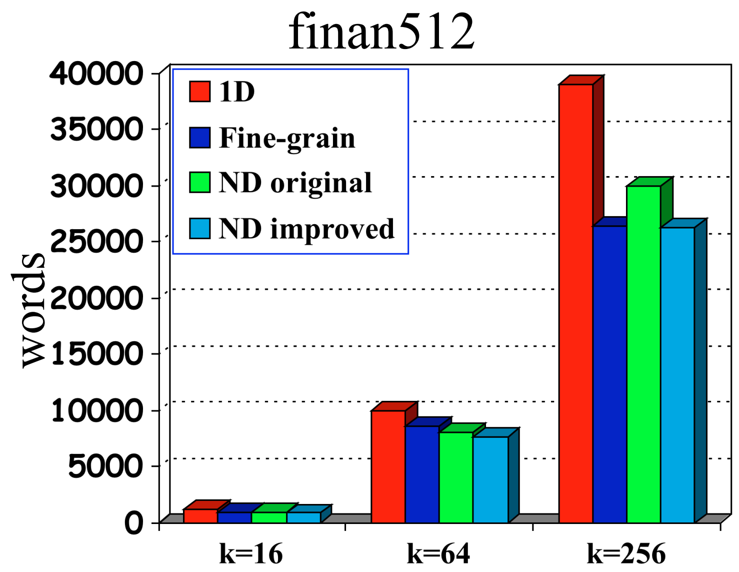
Summary of Improved (2-Phase) Method

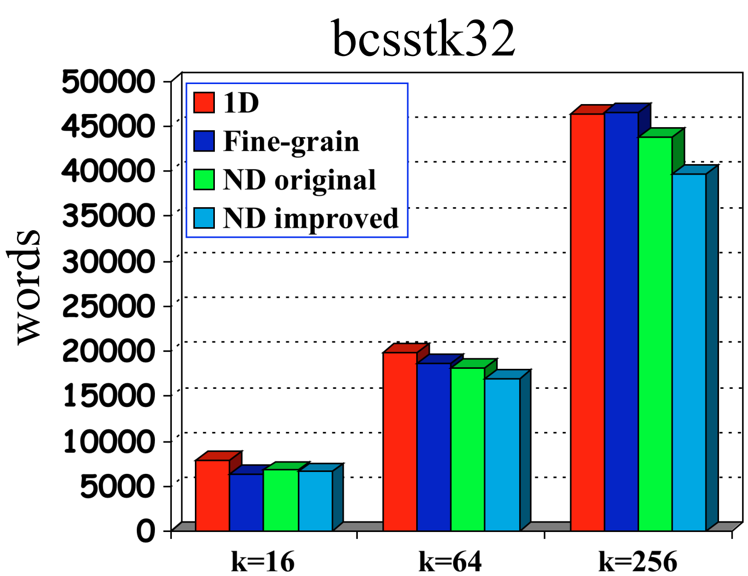


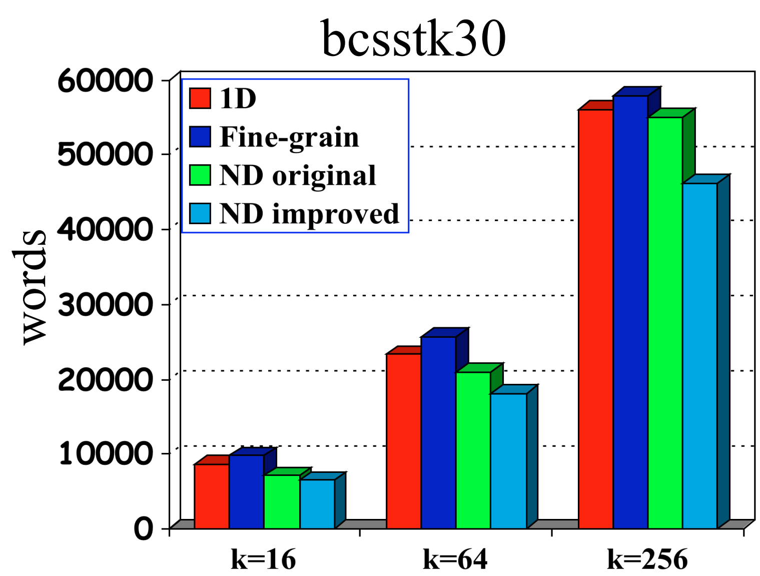
- Use heuristic to reduce partitioning problem (phase 1)
  - Heuristic = general ND partitioning algorithm
  - Heuristic is optimal for bisection
- Apply fine-grain hypergraph partitioning with fixed vertices to much smaller problem (phase 2)
  - One fixed vertex per part
- Smaller problem means fine-grain hypergraph will do excellent job of partitioning
  - Fast (relative to FG partitioning of original graph)
  - Better relative solution

Improved Method - Communication Volume









Sandia National Laboratories

Information Retrieval Matrices

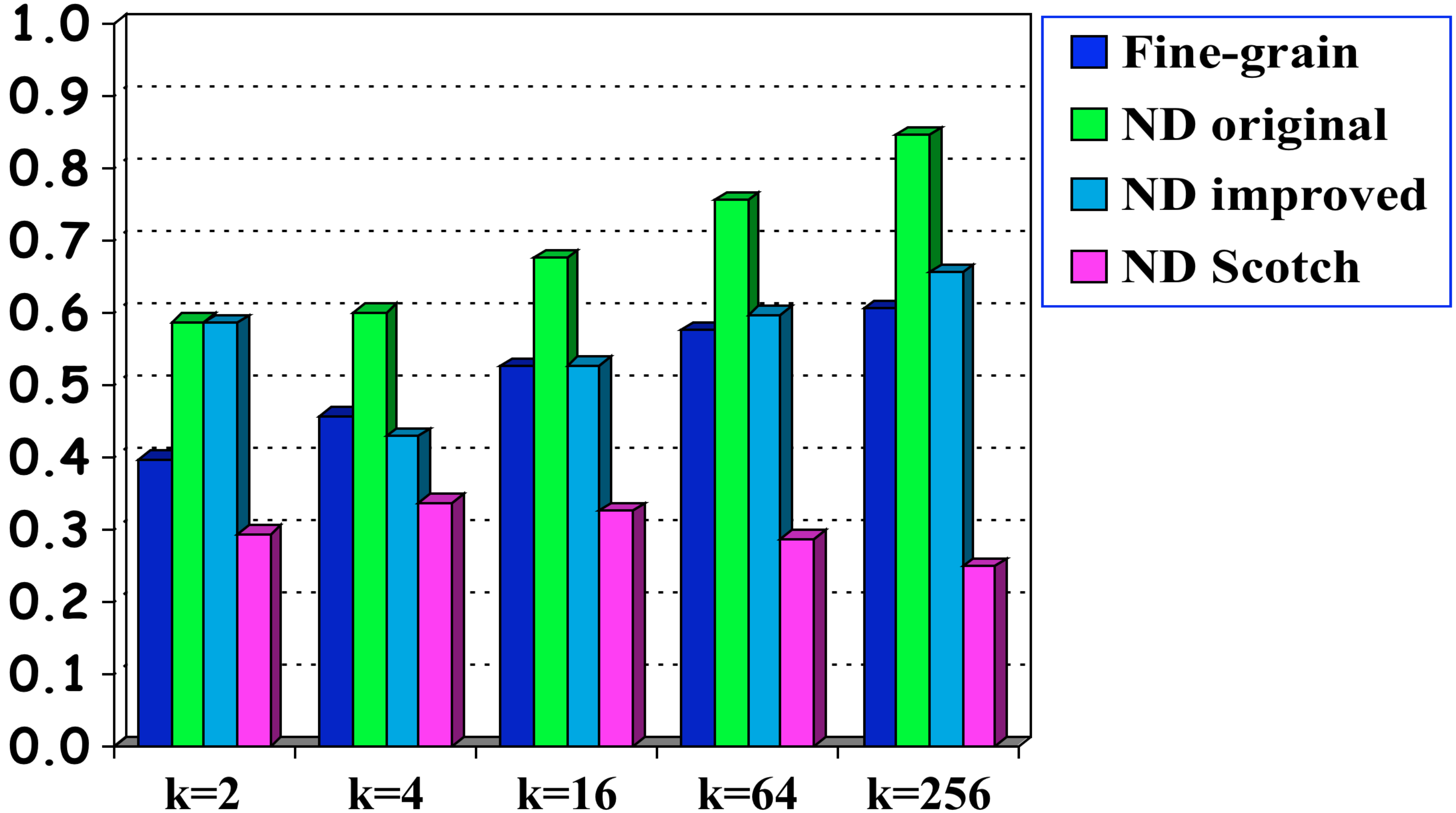
- Results for 2 types of matrices
  - Web-link matrices
    - R-MAT (Chakrabarti, et al.)
    - Stanford_Berkeley (Kamvar)
  - Term-by-term (Dunlavy, Sandia)
- 5 different partitioning methods
  - 1D hypergraph partitioning
  - Fine-grain hypergraph partitioning (2D)
  - Nested dissection partitioning (2D)
    - Original heuristic implementation
    - Improved implementation (2-phase method)
    - Improved implementation with Scotch (LaBRI, INRIA)

Vertex Separator from SCOTCH

- Originally vertex separators obtained from edge separators
  - 1D hypergraph partitioning
  - Smaller separators perhaps possible using nested dissection algorithms

- SCOTCH (LaBRI, INRIA)
  - Multilevel graph/sparse matrix ordering algorithm
  - Attempts to find smallest balanced vertex separator
  - Used to reorder matrices to reduce fill
  - Used Zoltan interface to SCOTCH
- Pro: focus on finding small vertex separators
- Con: does not naturally balance nonzeros

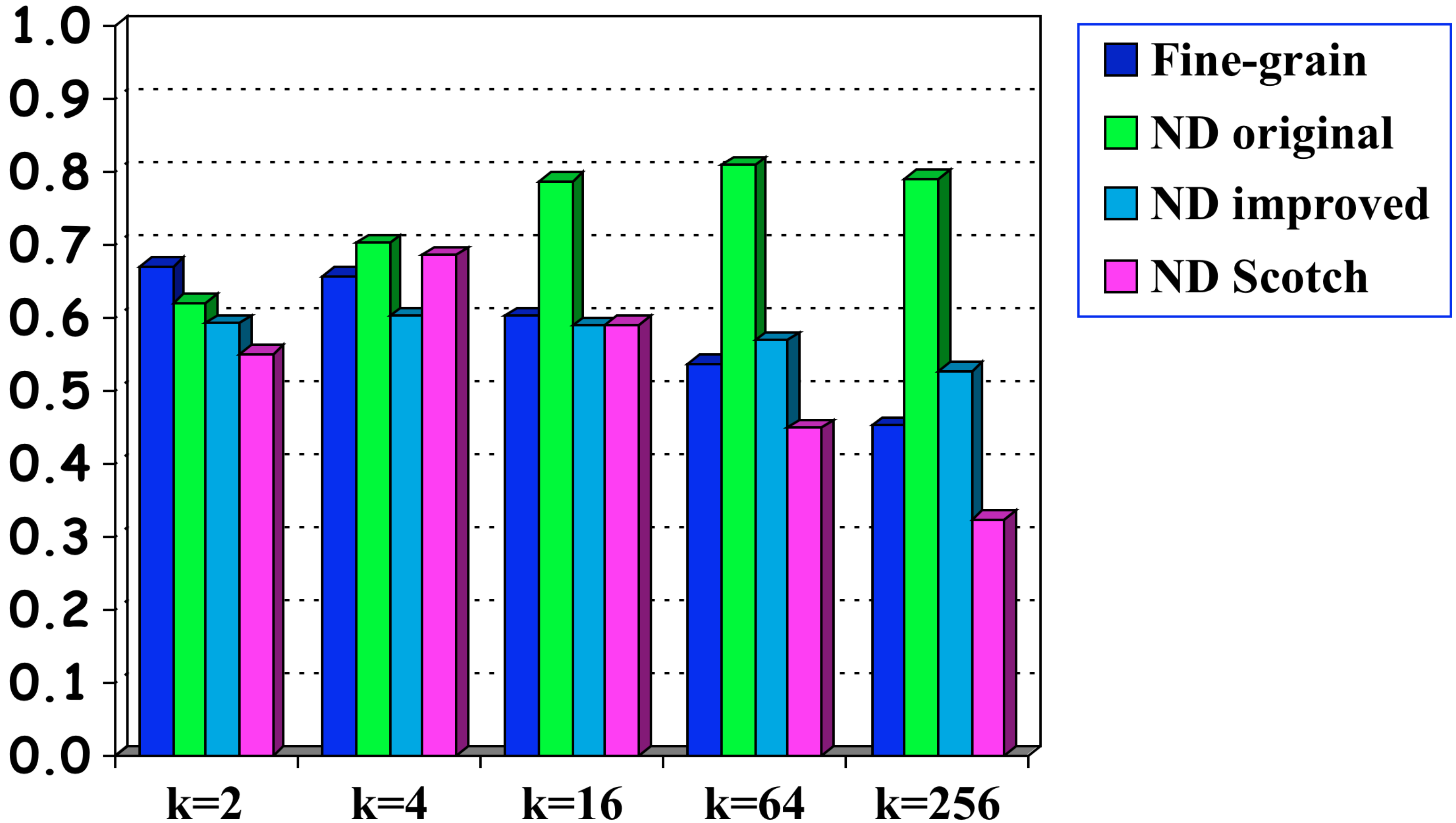
Web-link Results



- Communication volume relative to 1D* partitioning
  - Average for rmat18, rmat19, Stanford_Berkeley

** load imbalance for ND SCOTCH (k=256), FG failure to converge (RMAT19)

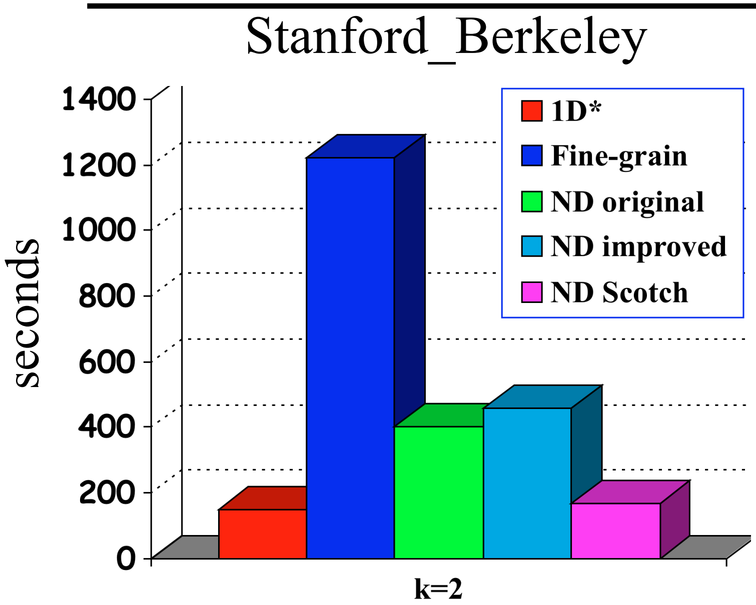
Term-by-Term Results

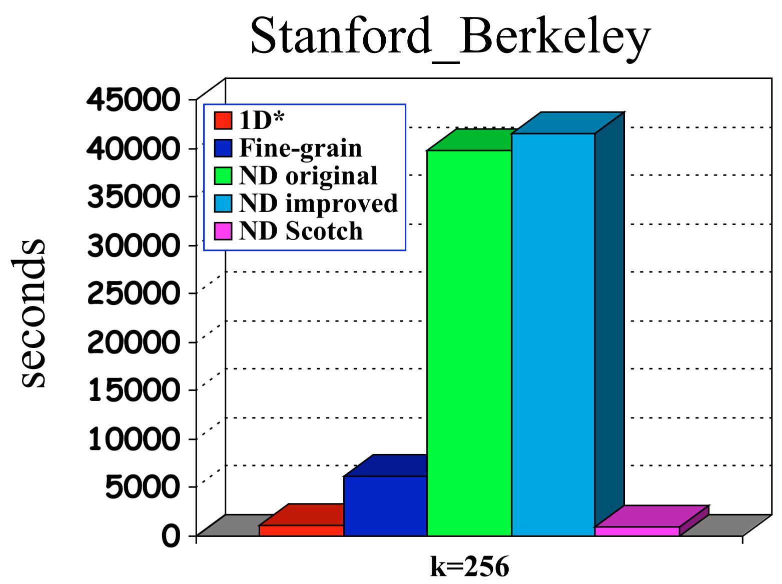


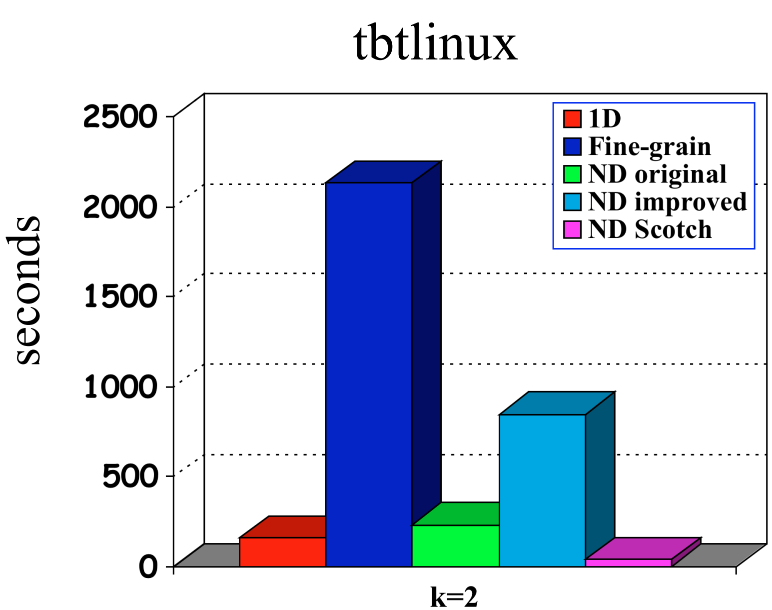
- Communication volume relative to 1D partitioning
  - Average for tbtlinux, tbtspock, tbtsandia2

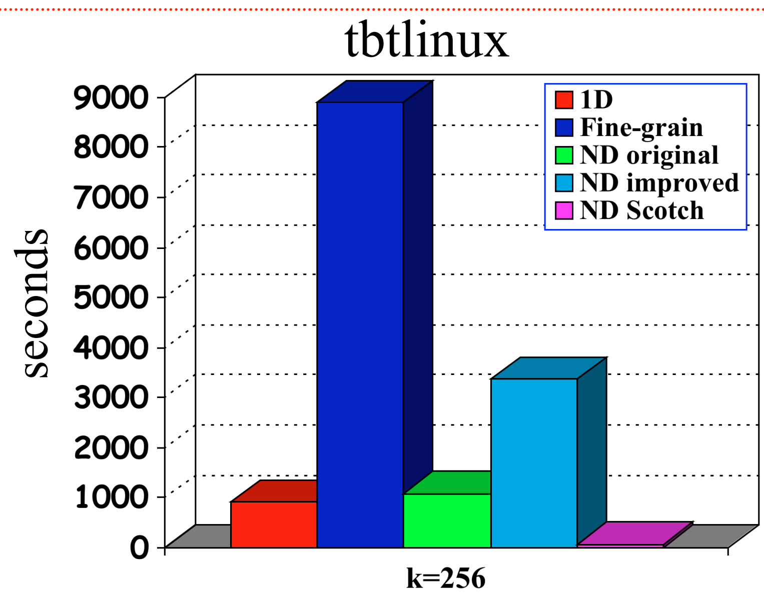
** load imbalance for ND SCOTCH (k=64, k=256)

Runtimes -- Select Matrices









Sandia National Laboratories

Summary

- New 2D matrix partitioning algorithm
- ND matrix partitioning algorithm
  - ND used in new context
  - Good trade off between communication volume and partitioning time
    - Communication volume (comparable to fine-grain)
    - Partitioning time (comparable to 1D)
  - Extensions for nonsymmetric matrices
  - Method shows promise for information retrieval
- Work with Erik Boman, et al. to implement 2D partitioning algorithms in Trilinos
  - Isorropia, package for CSC

Selection of Related Papers

Nested Dissection Partitioning:

E.G. Boman and M.M. Wolf, "A Nested Dissection Approach to Sparse Matrix Partitioning for Parallel Computations," SANDIA Technical Report 2008-5482J. (Submitted for publication)

M. Wolf, E. Boman, and C. Chevalier, "Improved Parallel Data Partitioning by Nested Dissection with Applications to Information Retrieval," SANDIA Technical Report 2008-7908J.

2D Partitioning:

U. Catalyurek and C. Aykanat, "A fine-grain hypergraph model for 2d decomposition of sparse matrices," In Proc. IPDPS 8th Int'l Workshop on Solving Irregularly Structured Problems in Parallel (Irregular 2001), April 2001.

U. Catalyurek, C. Aykanat, and B. Ucar. On two-dimensional sparse matrix partitioning: Models, methods, and a recipe. Technical Report BMI-TR-2008-n04, The Ohio State University, 2008.

B. Vastenhouw and R. H. Bisseling. A two-dimensional data distribution method for parallel sparse matrix-vector multiplication. SIAM Review, 47(1):67–95, 2005.

Sandia National Laboratories