A Hybrid 2D Method for Sparse Matrix Partitioning

Rob Bisseling, Tristan van Leeuwen
Utrecht University

Ümit Çatalyürek
Ohio State University

Support from BSIK-BRICKS/MSV and NCF
Outline

1. Introduction
   - Mondriaan 2D matrix partitioning
   - Fine-grain 2D partitioning

2. New: hybrid method for 2D partitioning
   - Combining the Mondriaan and fine-grain methods

3. Experimental results
   - PageRank matrices: Stanford-Berkeley subdomain
   - Other sparse matrices: term-by-document, linear programming, polymers

4. Conclusions and future work
Parallel sparse matrix–vector multiplication \( u := Av \)

A sparse \( m \times n \) matrix, \( u \) dense \( m \)-vector, \( v \) dense \( n \)-vector

\[
u_i := \sum_{j=0}^{n-1} a_{ij} v_j
\]

4 phases: communicate, compute, communicate, compute
Hypergraph with 9 vertices and 6 hyperedges (nets), partitioned over 2 processors
1D matrix partitioning using hypergraphs

Column bipartitioning of $m \times n$ matrix

- Hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{N}) \Rightarrow$ exact communication volume in sparse matrix–vector multiplication.
- Columns $\equiv$ Vertices: $0, 1, 2, 3, 4, 5, 6$.
  - Rows $\equiv$ Hyperedges (nets, subsets of $\mathcal{V}$): 
    
    $$
    n_0 = \{1, 4, 6\}, \quad n_1 = \{0, 3, 6\}, \quad n_2 = \{4, 5, 6\}, \\
    n_3 = \{0, 2, 3\}, \quad n_4 = \{2, 3, 5\}, \quad n_5 = \{1, 4, 6\}.
    $$

Minimising communication volume

- **Broken nets**: \( n_1, n_2 \) cause one horizontal communication
- Use Kernighan–Lin/Fiduccia–Mattheyses for hypergraph bipartitioning
- Multilevel scheme: **merge** similar columns first, **refine** bipartitioning afterwards
- Used in PaToH (Çatalyürek and Aykanat 1999) for 1D matrix partitioning.
Mondriaan 2D matrix partitioning

- Block distribution (without row/column permutations) of $59 \times 59$ matrix `impcol_b` with 312 nonzeros, for $p = 4$

- Mondriaan package v1.0 (May 2002). Originally developed by Vastenhouw and Bisseling for partitioning term-by-document matrices for a parallel web search machine.
Recursively split the matrix into 2 parts.
Try splits in row and column directions, allowing permutations. Each time, choose the best direction.
Fine-grain 2D partitioning

- Assign each nonzero of $A$ individually to a part.
- Each nonzero becomes a vertex in the hypergraph.
- Each matrix row and column becomes a hyperedge.
- Hence $\text{nz}(A)$ vertices and $m + n$ hyperedges.
View the fine-grain hypergraph as an incidence matrix.

- \( m \times n \) matrix \( A \) with \( nz(A) \) nonzeros
- \( (m + n) \times nz(A) \) matrix \( F = F_A \) with \( 2 \cdot nz(A) \) nonzeros
- \( a_{ij} \) is \( k \)th nonzero of \( A \) \( \iff \) \( f_{ik}, f_{m+j,k} \) are nonzero in \( F \)
Communication for fine-grain 2D partitioning

- Broken net in first $m$ nets of hypergraph of $F$: nonzeros from row $a_{i*}$ are in different parts, hence horizontal communication in $A$.

- Broken net in last $n$ nets of hypergraph of $F$: vertical communication in $A$. 
Fine-grain 2D partitioning

- Recursively split the matrix into 2 parts
- Assign individual nonzeros to parts
- For visualisation: move mixed rows to middle, red up, blue down. Same for columns.
Hybrid 2D partitioning

- Recursively split the matrix into 2 parts
- Try splits in row and column directions, and fine-grain
- Each time, choose the best of 3
Recursive, adaptive bipartitioning algorithm

MatrixPartition\((A, p, \epsilon)\)

**input:** \(\epsilon = \) allowed load imbalance, \(\epsilon > 0\).

**output:** \(p\)-way partitioning of \(A\) with imbalance \(\leq \epsilon\).

*if* \(p > 1\) *then*

\[
q := \log_2 p;
\]

\[
(A^r_0, A^r_1) := h(A, \text{row}, \epsilon/q); \quad \text{hypergraph splitting}
\]

\[
(A^c_0, A^c_1) := h(A, \text{col}, \epsilon/q);
\]

\[
(A^f_0, A^f_1) := h(A, \text{fine}, \epsilon/q);
\]

\[
(A_0, A_1) := \text{best of } (A^r_0, A^r_1), (A^c_0, A^c_1), (A^f_0, A^f_1);
\]

\[
\maxnz := \frac{nz(A)}{p} (1 + \epsilon);
\]

\[
\epsilon_0 := \frac{\maxnz}{nz(A_0)} \cdot \frac{p}{2} - 1; \quad \text{MatrixPartition}(A_0, p/2, \epsilon_0);
\]

\[
\epsilon_1 := \frac{\maxnz}{nz(A_1)} \cdot \frac{p}{2} - 1; \quad \text{MatrixPartition}(A_1, p/2, \epsilon_1);
\]

*else* output \(A\);
Non-power-of 2 algorithm

MatrixPartition$(A, p, \epsilon)$

*input:* $\epsilon =$ allowed load imbalance, $\epsilon > 0$.

*output:* $p$-way partitioning of $A$ with imbalance $\leq \epsilon$.

1. **if** $p > 1$ **then**
   
   $q := \lceil \log_2 p \rceil$;
   
   $(A_r^0, A_r^1) := h(A, \text{row}, \epsilon/q)$;
   
   $(A_c^0, A_c^1) := h(A, \text{col}, \epsilon/q)$;
   
   $(A_f^0, A_f^1) := h(A, \text{fine}, \epsilon/q)$;
   
   $(A_0, A_1) := \text{best of } (A_r^0, A_r^1), (A_c^0, A_c^1), (A_f^0, A_f^1)$;

2. **Choose** $p_0, p_1 \geq 1$ with $p = p_0 + p_1$;

3. **maxnz** := $\frac{\text{nz}(A)}{p} (1 + \epsilon)$;

4. $\epsilon_0 := \frac{\text{maxnz}}{\text{nz}(A_0)} \cdot p_0 - 1$; **MatrixPartition**$(A_0, p_0, \epsilon_0)$;

5. $\epsilon_1 := \frac{\text{maxnz}}{\text{nz}(A_1)} \cdot p_1 - 1$; **MatrixPartition**$(A_1, p_1, \epsilon_1)$;

**else** output $A$;
Similarity metric for column merging (coarsening)

Column-scaled inner product:

\[ M(u, v) = \frac{1}{\omega_{uv}} \sum_{i=0}^{m-1} u_i v_i \]

- \( \omega_{uv} = 1 \) measures overlap
- \( \omega_{uv} = \sqrt{d_u d_v} \) measures cosine of angle
- \( \omega_{uv} = \min\{d_u, d_v\} \) measures relative overlap
- \( \omega_{uv} = \max\{d_u, d_v\} \)
- \( \omega_{uv} = d_{u \cup v} \), Jaccard metric from information retrieval

Here, \( d_u \) is the number of nonzeros of column \( u \).
Speeding up the fine-grain method

- \( \text{ip} \) = standard inner product matching
- \( \text{ip1} \) = inner product matching using an upper bound on the overlap, e.g. \( d_u \) to stop searching early. For fine-grain method, bound is sharper: 1 at first level.
- \( \text{ip2} \) = alternate between matching with overlap in top and bottom rows.
- \( \text{rnd} \) = choose a random match with overlap \( \geq 1 \)
Web searching: which page ranks first?
The link matrix $A$

- Given $n$ web pages with links between them. We can define the sparse $n \times n$ link matrix $A$ by

$$a_{ij} = \begin{cases} 1 & \text{if there is a link from page } j \text{ to page } i \\ 0 & \text{otherwise.} \end{cases}$$

- Let $e = (1, 1, \ldots, 1)^T$, representing an initial uniform importance (rank) of all web pages. Then

$$(Ae)_i = \sum_j a_{ij}e_j = \sum_j a_{ij}$$

is the total number of links pointing to page $i$.

- The vector $Ae$ represents the importance of the pages; $A^2e$ takes the importance of the pointing pages into account as well; and so on.
The Google matrix

- A web surfer chooses each of the outgoing $N_j$ links from page $j$ with equal probability. Define the $n \times n$ diagonal matrix $D$ with $d_{jj} = 1/N_j$.

- Let $\alpha$ be the probability that a surfer follows an outlink of the current page. Typically $\alpha = 0.85$. The surfer jumps to a random page with probability $1 - \alpha$.

- The Google matrix is defined by (Brin and Page 1998)

$$G = \alpha AD + (1 - \alpha)ee^T/n.$$ 

- The PageRank of a set of web pages is obtained by repeated multiplication by $G$, involving sparse matrix–vector multiplication by $A$, and some vector operations.
Comparing 1D, 2D fine-grain, and 2D Mondriaan

- The following 1D and 2D fine-grain communication volumes for PageRank matrices are published results from the parallel program Par\textsc{k}way v2.1 (Bradley, de Jager, Knottenbelt, Trifunović 2005).

- The 2D Mondriaan volumes are results with all our improvements (incorporated in v2.0), but using only row/column partitioning, not the fine-grain option.
Communication volume: Stanford_Berkeley

$n = 683,446$, $nz(A) = 8,262,087$ nonzeros.

- Represents the Stanford and Berkeley subdomains, obtained by a web crawl in Dec. 2002 by Sep Kamvar.
Meaning of results

- Both 2D methods *save an order of magnitude* in communication volume compared to 1D.
- Parkway fine-grain is *slightly better* than Mondriaan, in terms of partitioning quality. This may be due to a better implementation, or due to the fine-grain method itself. Further investigation is needed.
- 2D Mondriaan is *much faster* than fine-grain, since the hypergraphs involved are much smaller: \(7 \times 10^5\) vs. \(8 \times 10^6\) vertices for Stanford_Berkeley.
Transition matrix cage6 of Markov model

- Reduced transition matrix cage6 with $n = 93$, $\text{nz}(A) = 785$ for polymer length $L = 6$.

- Larger matrix cage10 is included in our test set of 18 matrices representing various applications: 3 linear programming matrices, 2 information retrieval, 2 chemical engineering, 2 circuit simulation, 1 polymer simulation, . . .
Test set of 18 matrices (smaller than PageRank matrices).

Volume relative to original Mondriaan program, v1.02

Implementation: Mondriaan’s own hypergraph partitioner

Fine-grained method has more freedom to find a good partitioning, but shows no gains on average
Average communication volume for 3 methods

- Test set of 18 matrices.
- Volume relative to original Mondriaan program, v1.02
- Implementation: PaToH hypergraph partitioner. Highly optimised, and it shows.
- Hybrid method shows a little gain over 2D Mondriaan
Conclusions and . . .

- We have presented a hybrid method which combines two different 2D matrix partitioning methods: Mondriaan and fine-grain. The hybrid improves upon both.

- With a highly optimised hypergraph partitioner such as PaToH as the partitioning engine, the Mondriaan 2D method achieves almost the same quality as the hybrid method, but much faster.

- PageRank is a prime application of 2D matrix partitioning.
Mondriaan and PaToH are sequential. Parallel hypergraph partitioner has been released in Zoltan by Sandia National Laboratories.

New release of Mondriaan, v2.0, is scheduled July 10. Currently final testing. Features:
- Improved vector distribution, often optimal
- Much faster
- 10% lower communication volume, on average
- Many new partitioning strategies, including hybrid

- Visualisation through Matlab interface
- Cut-net metric