Sparse Matrix-Matrix Multiplication: Applications, Algorithms, and Implementations

Organizers: Grey Ballard and Alex Druinsky

SIAM Conference on Applied Linear Algebra

October 26, 2015

Part I: 10:15–12:15
Part II: 3:00–4:30
Techwood - Atlanta Conference Level

Terminology

Sparse matrix-matrix multiplication or sparse matrix multiplication?

- sparse matrix multiplication can be confused with sparse matrix times dense vector (SpMV)
- sparse matrix-matrix multiplication is a mouthful

SpMM or SpGEMM?

- SpMM can be confused with sparse matrix times dense matrix (typically sparse matrix times multiple dense vectors)

In any case, we're talking about sparse matrix times sparse matrix in this mini, and I'll use SpGEMM unless there are (violent) objections

Schedule of Talks: Morning Session (MS5)

10:15 Hypergraph Partitioning for SpGEMM

- *Grey Ballard, Sandia National Labs*

10:45 Exploiting Sparsity in Parallel SpGEMM

- *Cevdet Aykanat, Bilkent University*

11:15 SpGEMM and Its Use in Parallel Graph Algorithms

- *Ariful Azad, Lawrence Berkeley National Lab*

11:45 The Input/Output Complexity of SpGEMM

- *Morten Stöckel, IT University of Copenhagen*

Schedule of Talks: Afternoon Session (MS12)

3:00 Analyzing SpGEMM on GPU Architectures

- *Steven Dalton, NVIDIA*

3:30 A Framework for SpGEMM on GPUs and Heterogeneous Processors

- *Weifeng Liu, University of Copenhagen*

4:00 The Distributed Block-Compressed Sparse Row Library: Large Scale and GPU Accelerated SpGEMM

- *Alfio Lazzaro, ETH Zürich*

4:30 ~~Strong Scaling and Stability: SpAMM Acceleration for the Matrix Square Root Inverse and the Heavyside Function~~

- *Matt Challecombe, Los Alamos National Lab*

Hypergraph Partitioning for Parallel Sparse Matrix-Matrix Multiplication

Grey Ballard, Alex Druinsky, Nicholas Knight, Oded Schwartz

SIAM Conference on Applied Linear Algebra

October 26, 2015

Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY

National Nuclear Security Administration

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.

(SAND2015-5153 C)

Summary

- Parallel **SpGEMM** is an **irregular computation** whose performance is **communication bound**
- We have a useful **classification** of parallel SpGEMM algorithms based on a geometric interpretation
- **Hypergraph partitioning** can relate parallel algorithms to their communication costs
- Using hypergraphs, we obtain theoretical communication **lower bounds** and practical **algorithmic insight** for parallel SpGEMM

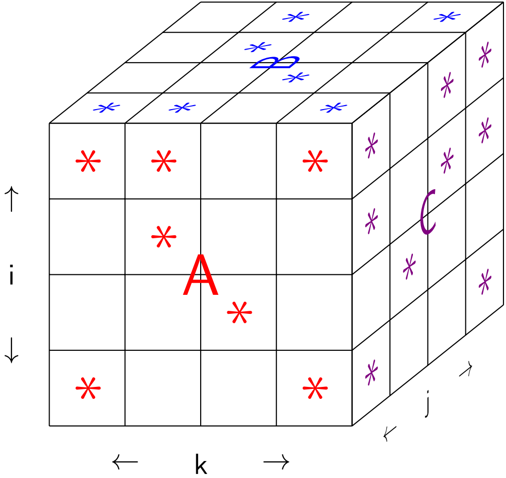
Sparse matrix-matrix multiplication (SpGEMM)

$$\begin{array}{|c|c|c|c|}\hline * & & * & * \\ \hline * & & * & * \\ \hline & * & & \\ \hline * & & & * \\ \hline \end{array} \quad = \quad \begin{array}{|c|c|c|c|}\hline * & * & & * \\ \hline & * & & \\ \hline & & * & \\ \hline * & & & * \\ \hline \end{array} \quad \bullet \quad \begin{array}{|c|c|c|c|}\hline * & & & \\ \hline * & & * & * \\ \hline & * & & \\ \hline * & & & * \\ \hline \end{array}$$

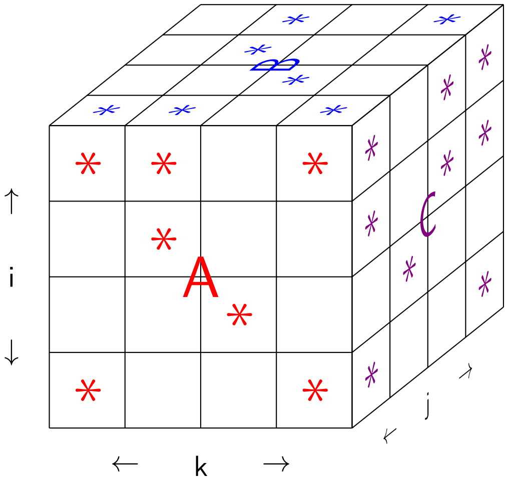
C A B

$$C_{ij} = \sum_k A_{ik} \cdot B_{kj}$$

Geometric view of the computation



Geometric view of the computation



Parallel algorithms partition the nonzero multiplies across processors

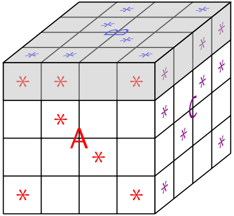
Classification of Algorithms

- **1D Algorithms**: parallelization over only **1** dimension of cube
  - Only 3 types: row-wise, column-wise, or outer-product
- **2D Algorithms**: parallelization over **2** dimensions of cube
  - include Sparse SUMMA and Sparse Cannon
  - can be classified into 3 subclasses
- **3D Algorithms**: parallelization over all **3** dimensions of cube
  - most general/flexible class

1D Algorithms

Row-Wise Algorithm: $A(i,:) \cdot B = C(i,:)$

$$\begin{bmatrix} * & * & & * \\ & * & & \\ & & * & \\ * & & & * \end{bmatrix} \bullet \begin{bmatrix} * & & & \\ * & & * & * \\ & * & & \\ * & & & * \end{bmatrix} = \begin{bmatrix} * & & * & * \\ * & & * & * \\ & * & & \\ * & & & * \end{bmatrix}$$

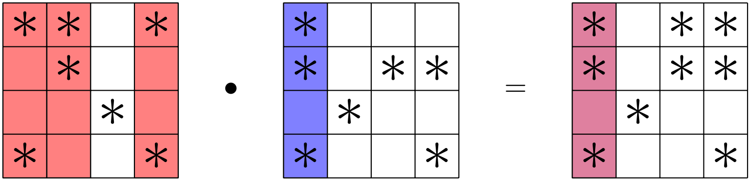


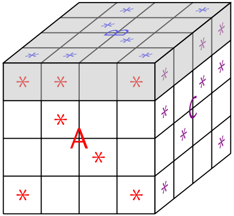
Row-Wise

1D Algorithms

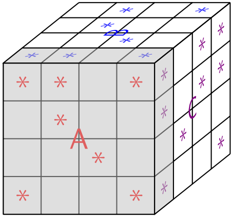
Column-Wise Algorithm:

$$A \cdot B(:, j) = C(:, j)$$





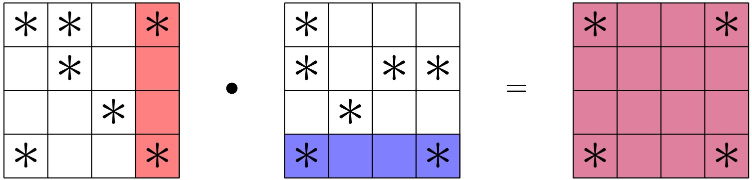
Row-Wise

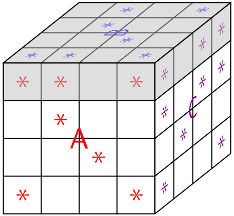


Column-Wise

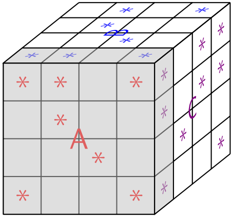
1D Algorithms

Outer-Product Algorithm: $A(:,k) \cdot B(k,:) = C^{(k)}$

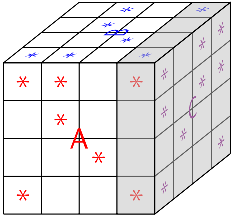




Row-Wise



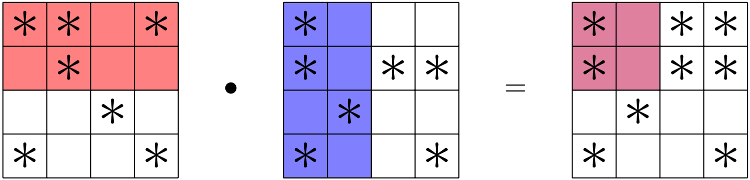
Column-Wise

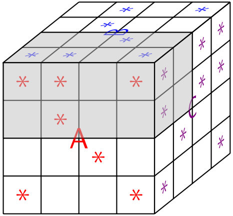


Outer-Product

2D Algorithms

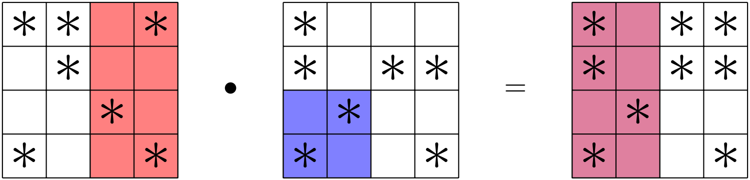
Monochrome-C Algorithm: $A(I,:) \cdot B(:,J) = C(I,J)$

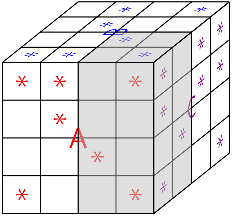




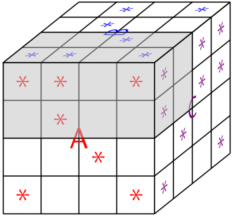
Monochrome-C

Monochrome-B Algorithm: $A(:,K) \cdot B(K,J) = C(:,J)$





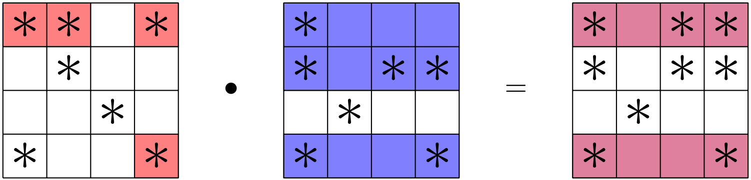
Monochrome-B

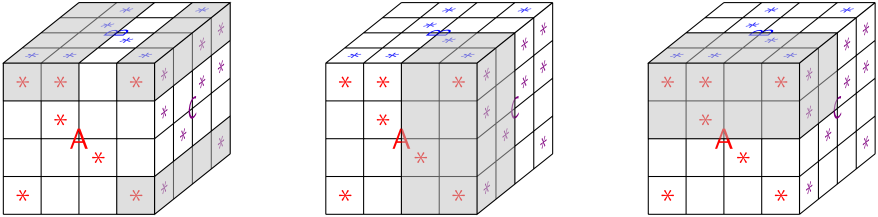


Monochrome-C

2D Algorithms

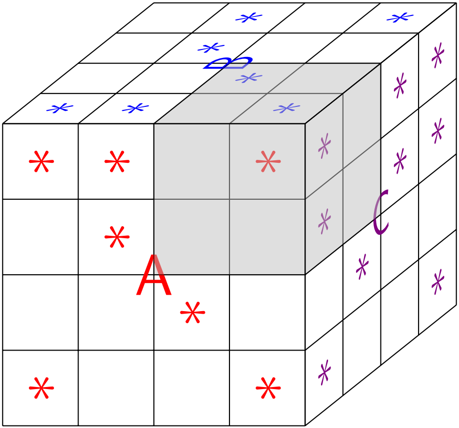
Monochrome-A Algorithm: $A(I,K) \cdot B(K,:) = C(I,:)$

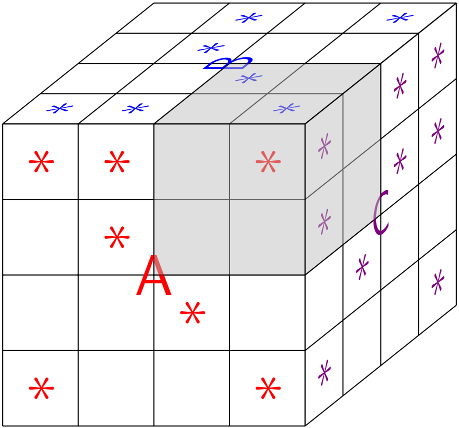




Monochrome-A **Monochrome-B** **Monochrome-C**

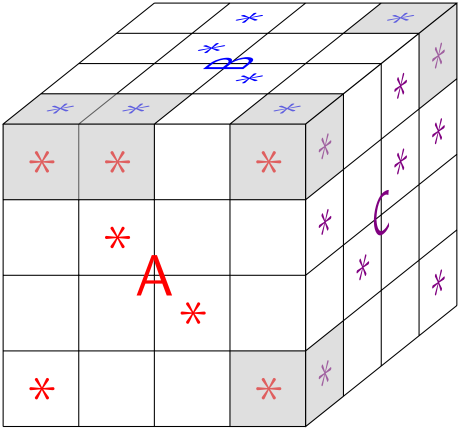
3D Algorithms





Sparsity oblivious: partition dense cube,
processors compute nonzero multiplies in their partition

3D Algorithms



Sparsity sensitive: partition nonzero multiplies,
enforce load balance directly

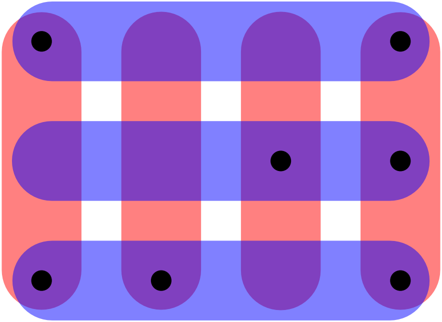
Communication hypergraphs and hypergraph partitioning

Hypergraphs consist of vertices and nets, or sets of vertices (of any size)

- for undirected graphs, nets are sets of exactly two vertices

For our purposes:

- vertices correspond to computation
- nets correspond to data



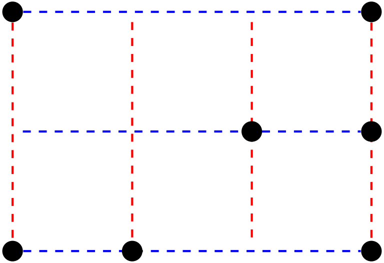
Communication hypergraphs and hypergraph partitioning

Hypergraphs consist of vertices and nets, or sets of vertices (of any size)

- for undirected graphs, nets are sets of exactly two vertices

For our purposes:

- vertices correspond to computation
- nets correspond to data



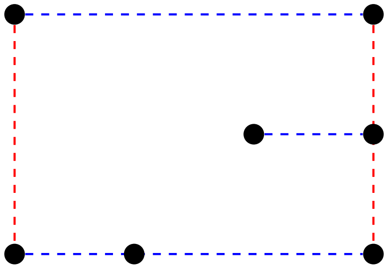
Communication hypergraphs and hypergraph partitioning

Hypergraphs consist of vertices and nets, or sets of vertices (of any size)

- for undirected graphs, nets are sets of exactly two vertices

For our purposes:

- vertices correspond to computation
- nets correspond to data



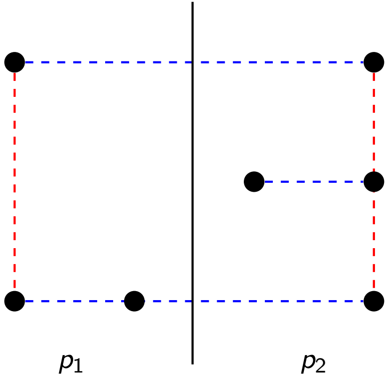
Communication hypergraphs and hypergraph partitioning

Hypergraphs consist of vertices and nets, or sets of vertices (of any size)

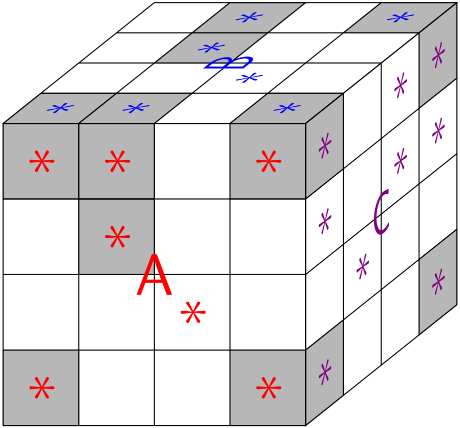
- for undirected graphs, nets are sets of exactly two vertices

For our purposes:

- vertices correspond to computation
- nets correspond to data

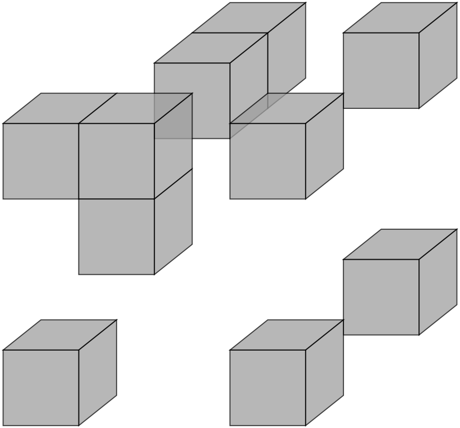


SpGEMM's "fine-grained" hypergraph



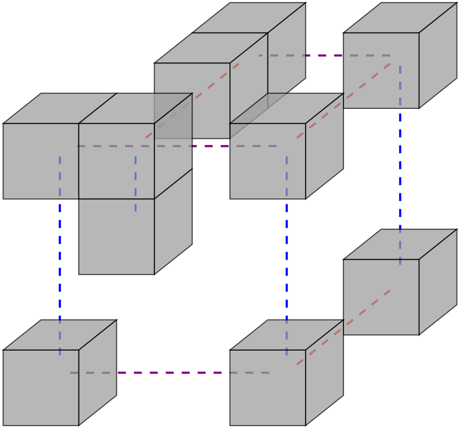
Vertices correspond to computation (nonzero multiplication)

SpGEMM's "fine-grained" hypergraph



Vertices correspond to computation (nonzero multiplication)

SpGEMM's "fine-grained" hypergraph



Vertices correspond to computation (nonzero multiplication)
Nets correspond to data (nonzero entries)

Theoretical result

Theorem ([BDKS15])

The communication cost of SpGEMM using p processors is at least

$$\min_{\{\mathcal{V}_1,\ldots,\mathcal{V}_p\}\in\mathcal{P}} \max_{i\in[p]} \left\{\#\ \textit{cut nets with vertices in } \mathcal{V}_i\right\},$$

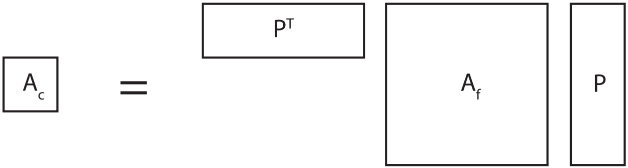
where $\mathcal{P}$ is the set of all sufficiently load-balanced partitions.

Proof.

The hypergraph models communication perfectly. □

Practical result for application-specific algorithm selection

- Hypergraph partitioning software can *estimate* lower bound
- Key application of SpGEMM: algebraic multigrid triple product
  - compute $A_c = P^T A_f P$ using two calls to SpGEMM
  - we analyze a model problem (off-line)



Practical result for application-specific algorithm selection

- Hypergraph partitioning software can *estimate* lower bound
- Key application of SpGEMM: algebraic multigrid triple product
  - compute $A_c = P^T A_f P$ using two calls to SpGEMM
  - we analyze a model problem (off-line)

		$A_f \cdot P$		$P^T \cdot (A_f P)$		
N	p	row-wise	fine-grained	row-wise	outer	fine-grained
19,683	27	5,528	4,649	10,712	2,072	964
91,125	125	5,528	5,823	10,712	2,072	1,324
250,047	343	5,528	6,160	10,712	2,072	1,444
531,441	729	5,528	6,914	10,712	2,072	1,491
970,299	1,331	5,528	6,679	10,712	2,072	1,548

Table: Comparison of 1D algorithms using geometric partitions [BSH15] with best hypergraph partition found by PaToH [ÇA99]

Restricted hypergraph models

Fine-grained model for SpGEMM is large

- # of vertices: # of scalar multiplies
- # of nets: # of nonzeros in inputs and output
- much more expensive to partition than to perform SpGEMM
- likely effective only as offline tool for classes of algorithms

Restricted hypergraph models

Fine-grained model for SpGEMM is large

- # of vertices: # of scalar multiplies
- # of nets: # of nonzeros in inputs and output
- much more expensive to partition than to perform SpGEMM
- likely effective only as offline tool for classes of algorithms

We can restrict the valid hypergraph partitions to classes of algorithms, significantly reducing the size of the hypergraph

- 1D: row-wise, column-wise, or outer-product hypergraphs
  - # of vertices/nets depends on matrix dimensions, not nnz
- 2D: monochrome-A, -B, or -C hypergraphs
  - # nets depends on nnz(A), nnz(B), or nnz(C)

Summary

- Parallel **SpGEMM** is an **irregular computation** whose performance is **communication bound**
- We have a useful **classification** of parallel SpGEMM algorithms based on a geometric interpretation
- **Hypergraph partitioning** can relate parallel algorithms to their communication costs
- Using hypergraphs, we obtain theoretical communication **lower bounds** and practical **algorithmic insight** for parallel SpGEMM

References I

K. Akbudak and C. Aykanat.
Simultaneous input and output matrix partitioning for outer-product–parallel sparse matrix-matrix multiplication.
SISC, 36(5):C568–C590, 2014.

G. Ballard, A. Buluç, J. Demmel, L. Grigori, B. Lipshitz, O. Schwartz, and S. Toledo.
Communication optimal parallel multiplication of sparse random matrices.
In *SPAA '13*, pages 222–231. ACM, 2013.

E. Boman, K. Devine, L.A. Fisk, R. Heaphy, B. Hendrickson, C Vaughan, Ü. Çatalyürek, D. Bozdag, W. Mitchell, and J. Teresco.
Zoltan 3.0: parallel partitioning, boad-balancing, and data management services; user's guide.
Technical Report SAND2007-4748W, Sandia Natl. Labs., 2007.

Grey Ballard, Alex Druinsky, Nicholas Knight, and Oded Schwartz.
Brief announcement: Hypergraph partitioning for parallel sparse matrix-matrix multiplication.
In *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures*, SPAA '15, pages 86–88, New York, NY, USA, 2015. ACM.

A. Buluç and J. R. Gilbert.
Parallel sparse matrix-matrix multiplication and indexing: implementation and experiments.
SISC, 34(4):C170–C191, 2012.

Grey Ballard, Christopher Siefert, and Jonathan Hu.
Reducing communication costs for sparse matrix multiplication within algebraic multigrid.
Technical Report SAND2015-3275, Sandia Natl. Labs., 2015.

Ümit Çatalyürek and Cevdet Aykanat.
PaToH: a multilevel hypergraph partitioning tool, version 3.0.
Technical report, Dept. of Computer Engineering, Bilkent Univ., 1999.

References II

Ü. Çatalyürek and Cevdet Aykanat.
A fine-grain hypergraph model for 2D decomposition of sparse matrices.
In *IPDPS '01*, pages 118–123, 2001.

Ümit Çatalyürek, Cevdet Aykanat, and Bora Uçar.
On two-dimensional sparse matrix partitioning: models, methods, and a recipe.
SISC, 32(2):656–683, 2010.

M.W. Gee, C.M. Siefert, J.J. Hu, R.S. Tuminaro, and M.G. Sala.
ML 5.0 Smoothed Aggregation User's Guide.
Technical Report SAND2006-2649, Sandia Natl. Labs., 2006.

S. Krishnamoorthy, Ü. Çatalyürek, Jarek Nieplocha, A. Rountev, and P. Sadayappan.
Hypergraph partitioning for automatic memory hierarchy management.
In *SC '06*, pages 34–46, 2006.

Thomas Lengauer.
Combinatorial Algorithms for Integrated Circuit Layout.
Wiley, 1990.