Brief Announcement: Hypergraph Partitioning for Parallel Sparse Matrix-Matrix Multiplication

Grey Ballard, Alex Druinsky, Nicholas Knight, Oded Schwartz

Symposium on Parallelism in Algorithms and Architectures

June 13, 2015

Sandia National Laboratories

Summary

- Parallel **sparse matrix-matrix multiplication** is an **irregular computation** whose performance is **communication bound**

- **Hypergraph partitioning** can relate parallelization schemes to communication costs

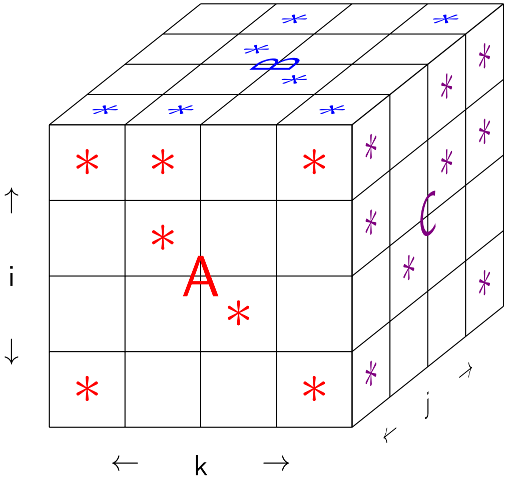
- Using hypergraphs, we obtain theoretical communication **lower bounds** and practical **algorithmic insight** for parallel sparse matrix-matrix multiplication

Sparse matrix-matrix multiplication (SpGEMM)

$$\underset{C}{\begin{array}{|c|c|c|c|}\hline \ast & & \ast & \ast \\ \hline \ast & & \ast & \ast \\ \hline & \ast & & \\ \hline \ast & & & \ast \\ \hline \end{array}} = \underset{A}{\begin{array}{|c|c|c|c|}\hline \ast & \ast & & \ast \\ \hline & \ast & & \\ \hline & & \ast & \\ \hline \ast & & & \ast \\ \hline \end{array}} \bullet \underset{B}{\begin{array}{|c|c|c|c|}\hline \ast & & & \\ \hline \ast & & \ast & \ast \\ \hline & \ast & & \\ \hline \ast & & & \ast \\ \hline \end{array}}$$

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

Geometric view of the computation



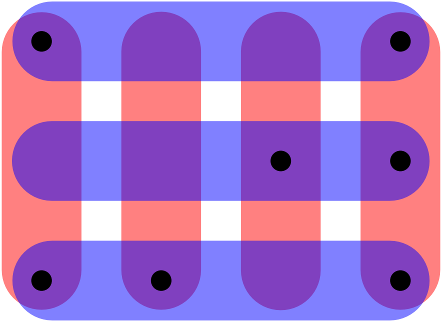
Communication hypergraphs and hypergraph partitioning

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



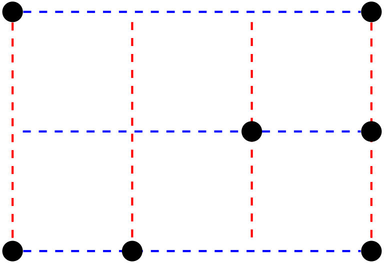
Communication hypergraphs and hypergraph partitioning

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



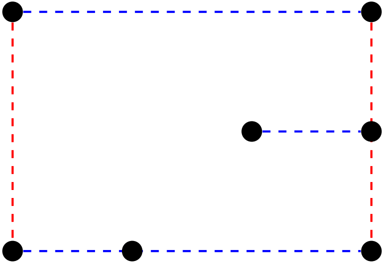
Communication hypergraphs and hypergraph partitioning

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



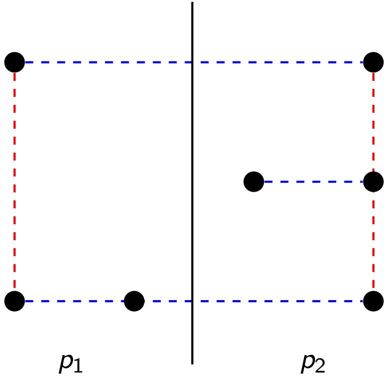
Communication hypergraphs and hypergraph partitioning

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

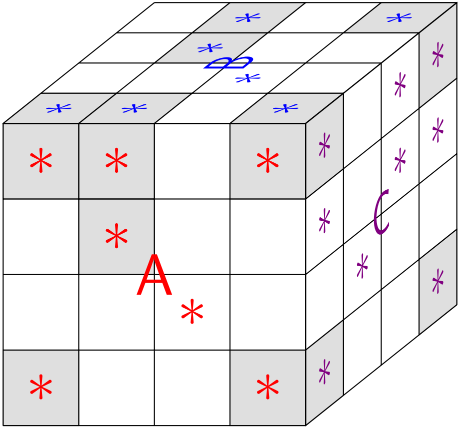
- for undirected graphs, nets are sets of exactly two vertices

For our purposes:

- vertices correspond to computation
- nets correspond to data

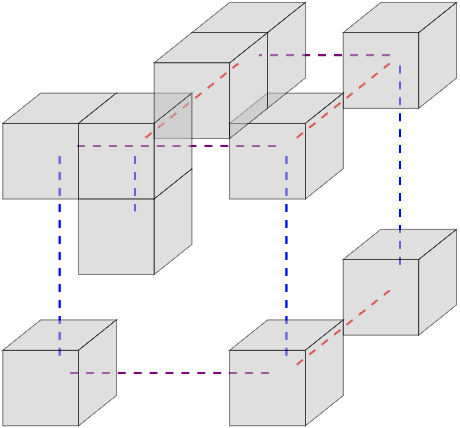


SpGEMM's hypergraph



Vertices correspond to computation (nonzero multiplication)

SpGEMM's hypergraph



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

Main theoretical result

Theorem

The communication cost of SpGEMM using p processors is at least

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

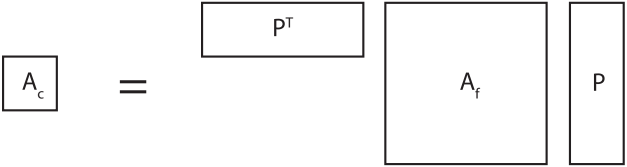
where $\mathcal{P}$ is the set of all sufficiently load-balanced partitions.

Proof.

The hypergraph models communication perfectly. □

Main practical result

- 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)



Main practical result

- 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	standard	hypergraph	standard	hypergraph
19,683	27	5,528	4,649	10,712	964
91,125	125	5,528	5,823	10,712	1,324
250,047	343	5,528	6,160	10,712	1,444
531,441	729	5,528	6,914	10,712	1,491
970,299	1,331	5,528	6,679	10,712	1,548

Table: Comparison of standard algorithm with best hypergraph partition

Summary

- Parallel **sparse matrix-matrix multiplication** is an **irregular computation** whose performance is **communication bound**
- **Hypergraph partitioning** can relate parallelization schemes to communication costs
- Using hypergraphs, we obtain theoretical communication **lower bounds** and practical **algorithmic insight** for parallel sparse matrix-matrix multiplication

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.

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

Grey Ballard, Jonathan Hu, and Christopher Siefert.
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.

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

References II

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.