

SAND2015-5153C

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



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)

*	*		*
	*		
		*	
*			*

A

.

*			
*		*	*
	*		
*			*

B

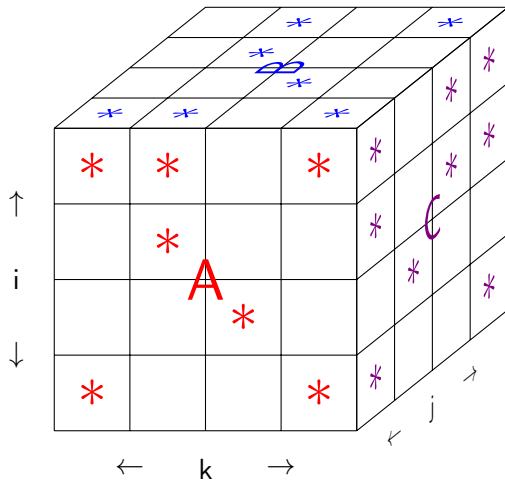
=

*		*	*
*		*	*
	*		
*			*

C

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

Geometric view of the computation



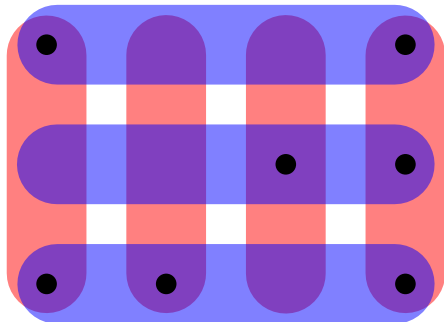
Communication hypergraphs and hypergraph partitioning

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



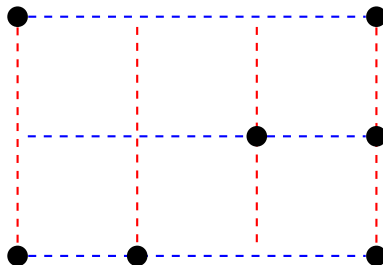
Communication hypergraphs and hypergraph partitioning

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



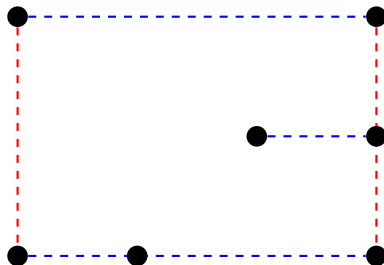
Communication hypergraphs and hypergraph partitioning

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



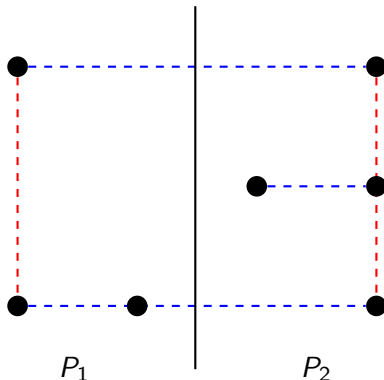
Communication hypergraphs and hypergraph partitioning

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

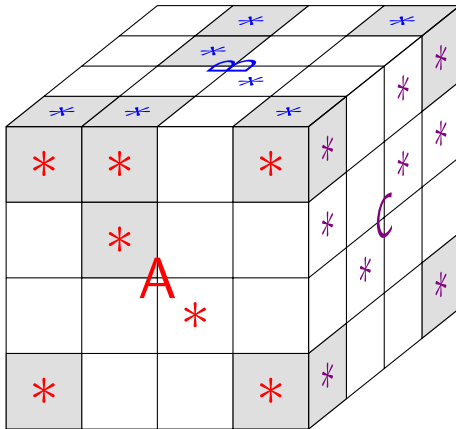
- for undirected graphs, nets are sets of exactly two vertices

For our purposes:

- vertices correspond to computation
- nets correspond to data

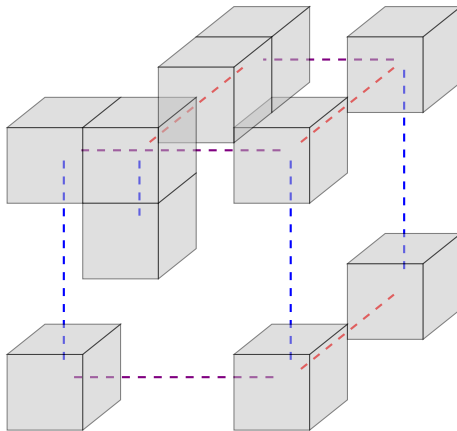


SpGEMM's hypergraph



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

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

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)

A diagram illustrating the matrix equation $A_c = P^T A_f P$. On the left, a small square box contains the label A_c . To its right is an equals sign. Further right, three boxes are arranged horizontally: a horizontal rectangle containing P^T , a large square containing A_f , and a vertical rectangle containing P .

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)

N	p	$A_f \cdot P$		$P^T \cdot (A_f P)$	
		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

- 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