Partitioning for Parallel Matrix-Matrix Multiplication with Heterogeneous Processors: The Optimal Solution

Ashley DeFlumere, Alexey Lastovetsky, Brett A. Becker

Heterogeneous Computing Laboratory, University College Dublin

May 21, 2012

Outline

Introduction

- Motivation and goals
- Definitions and MMM algorithms

Analytical Methods:

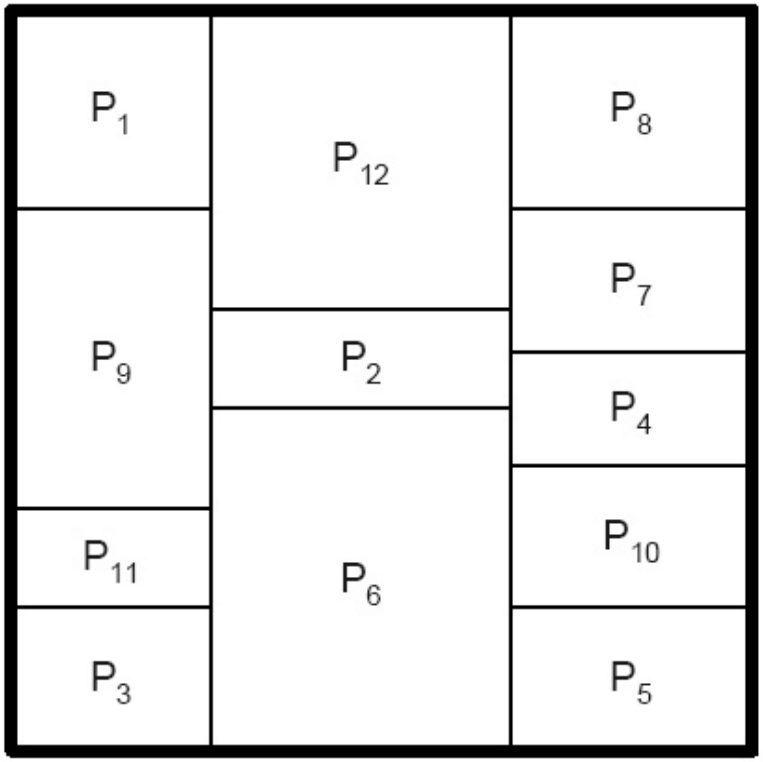
- Optimal Partition Shapes and Sizes
- Comparing Partition Shapes
- Finding Candidate Shapes

Results

- Theoretical Results
- Experimental Results
- Extension to 3 or More Processors

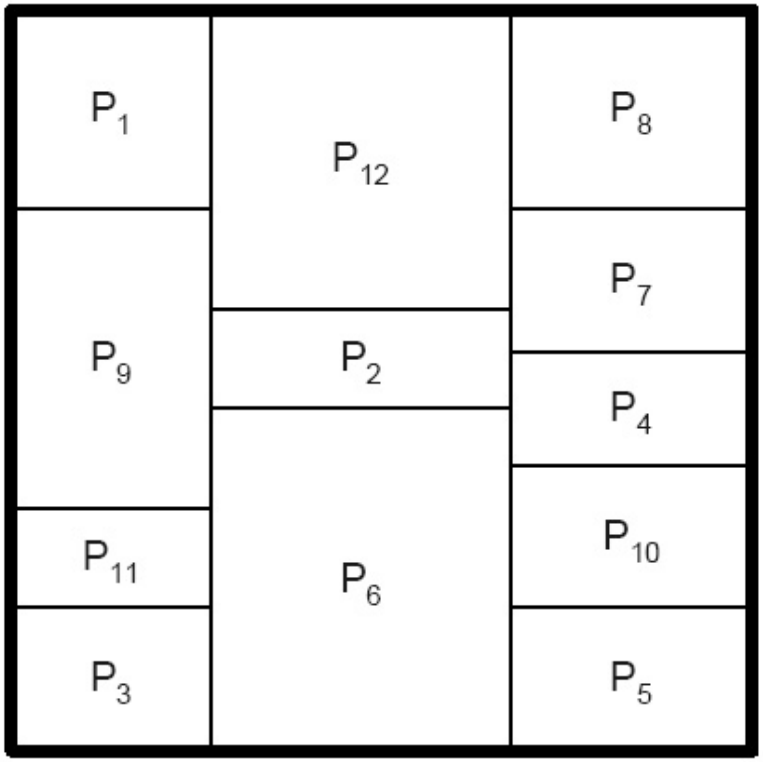
Motivation and Goals

- Traditionally, all processors assigned rectangles to compute
- Finding the optimal rectangular partition is difficult
- Is the rectangular shape optimal?



Motivation and Goals

- Traditionally, all processors assigned rectangles to compute
- Finding the optimal rectangular partition is difficult
- Is the rectangular shape optimal?
- Case of 2 heterogeneous processors



Methods

5 different MMM Algorithms:

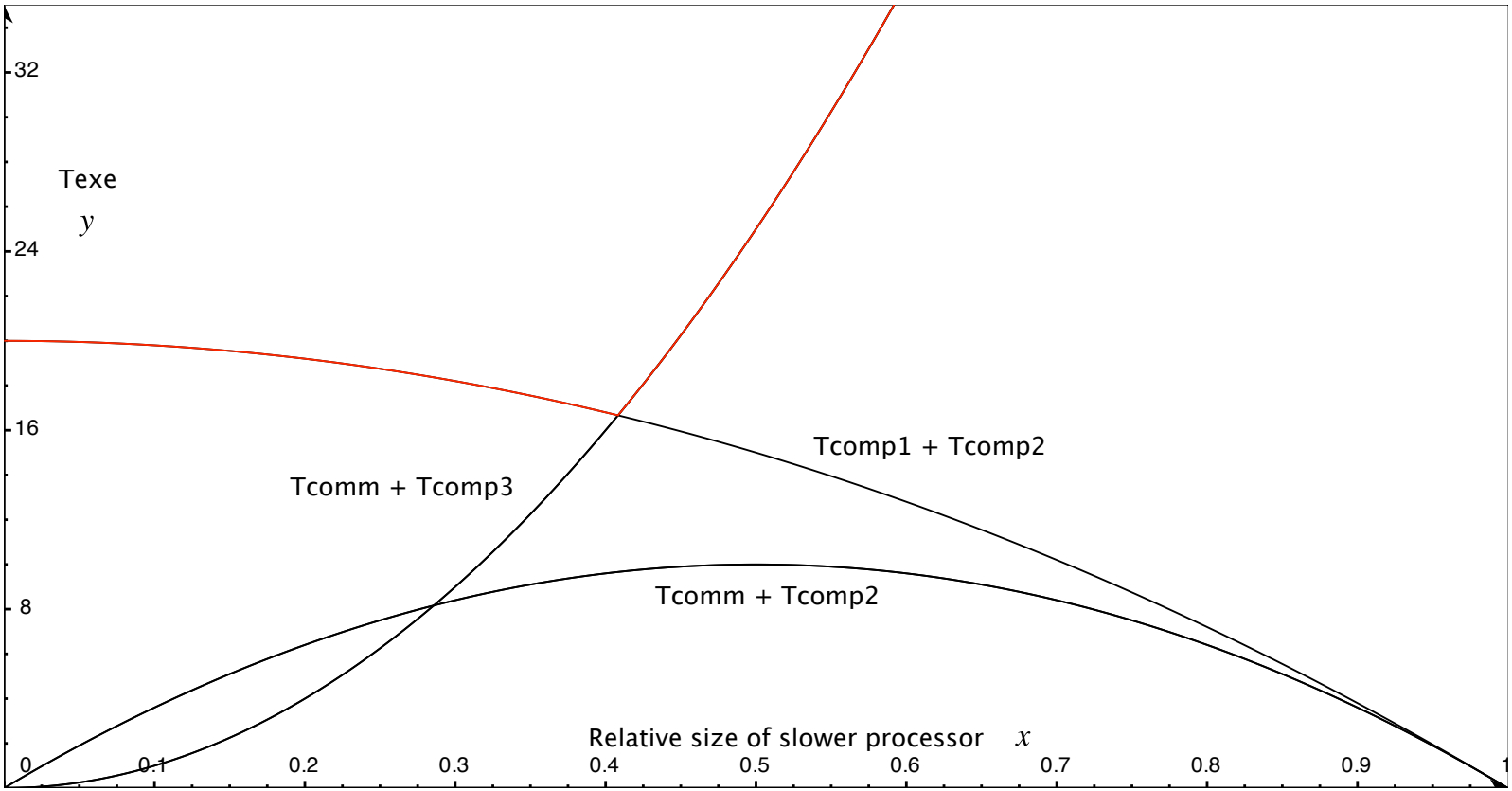
- 2 Barrier, communication then computation
- 2 Overlap, some immediate computation
- 1 Parallel, k-steps overlap all communication and computation

Models:

- Constant Performance Model
- Hockney Model of Communication

Finding the Partition Size - PCO

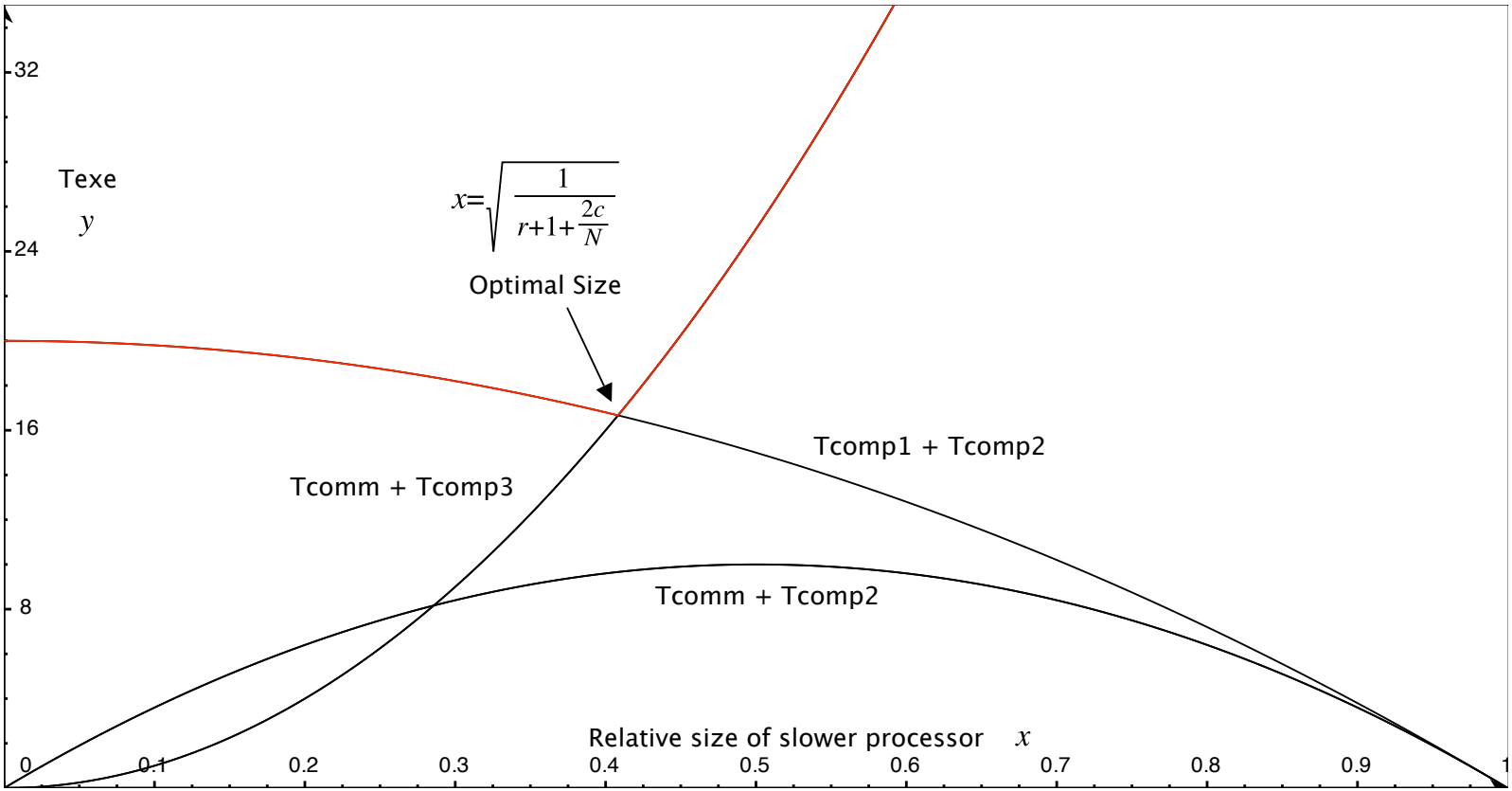
$$Texe = \max(\max(\max(Tcomm), Tcomp_1) + Tcomp_2,\ \max(Tcomm) + Tcomp_3)$$



Finding the Partition Size - PCO

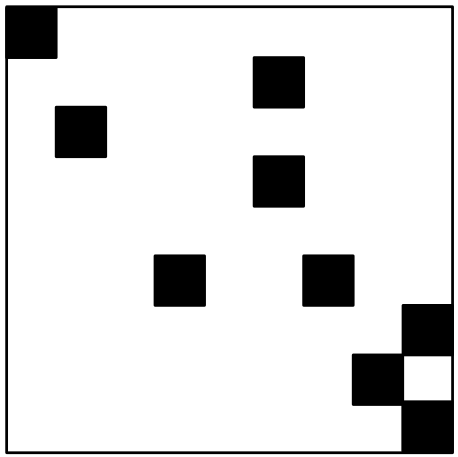
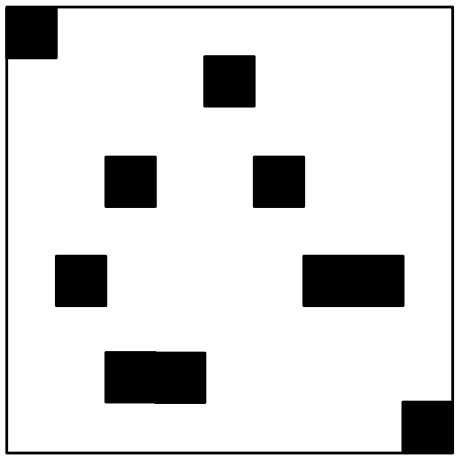
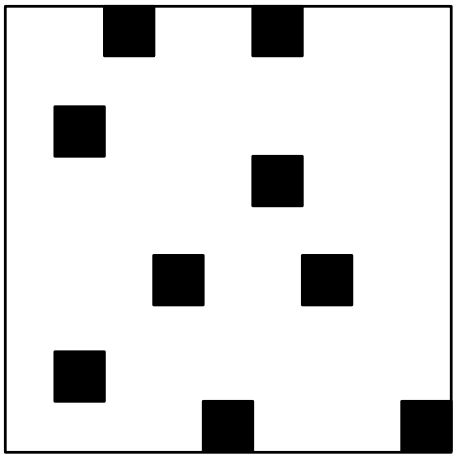
$$Texe = \max(\max(\max(Tcomm), Tcomp_1) + Tcomp_2,\ \max(Tcomm) + Tcomp_3)$$

$$Texe = \max\left(\max\left(\max\left(\frac{2x}{N} - \frac{2x^2}{N}, \frac{2x^2}{N}\right), \frac{(1-x)^2}{c}\right) + 2\frac{(x-x^2)}{c}, \max\left(\frac{2x}{N} - \frac{2x^2}{N}, \frac{2x^2}{N}\right) + \frac{x^2}{\frac{c}{a}}\right)$$



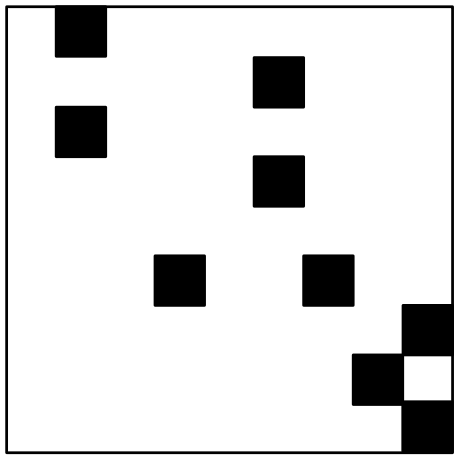
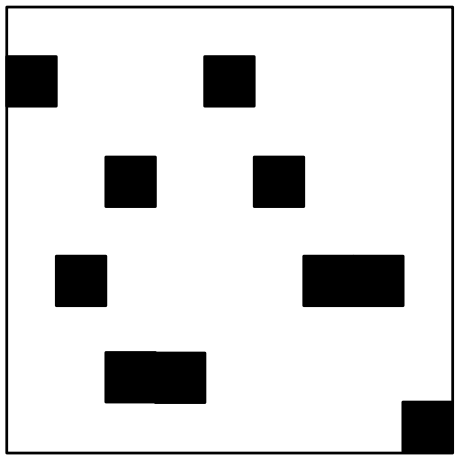
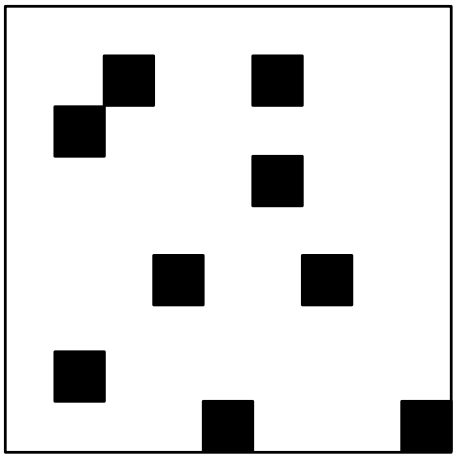
Finding Partition Shapes

- Consider all possible shapes, reduce using Push Technique

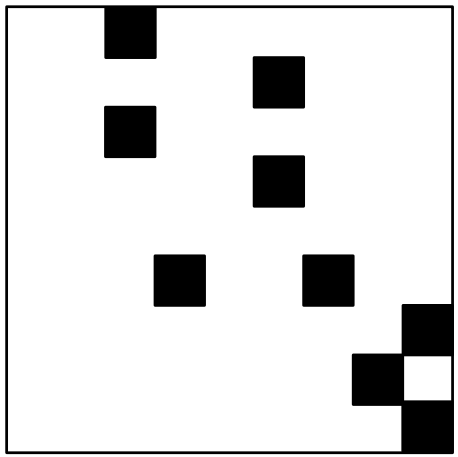
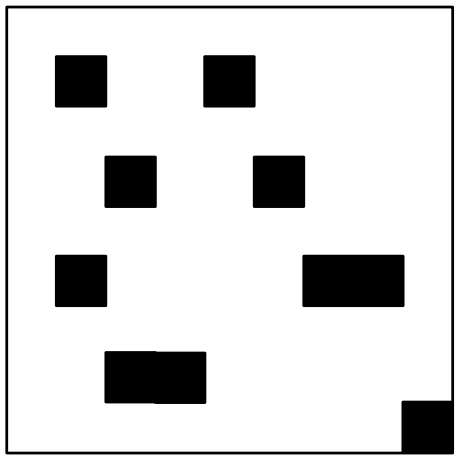
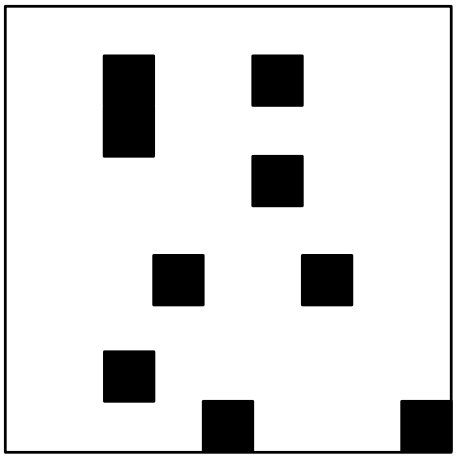
Finding Partition Shapes

- Consider all possible shapes, reduce using Push Technique

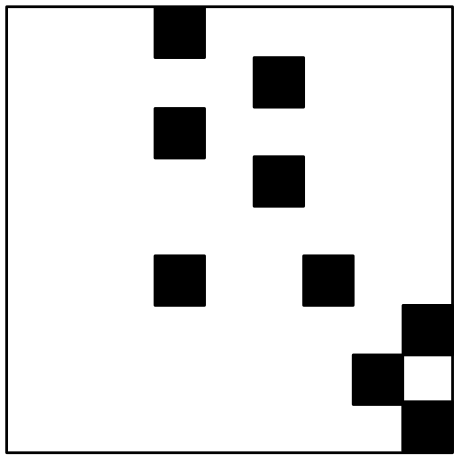
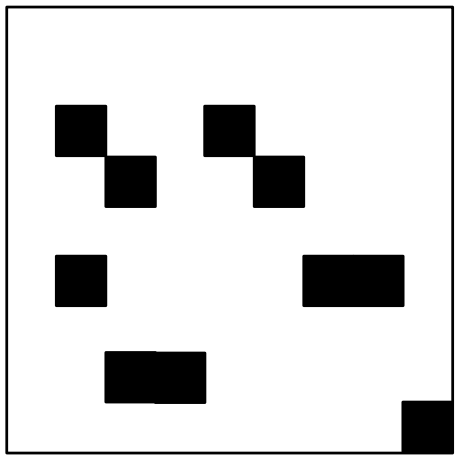
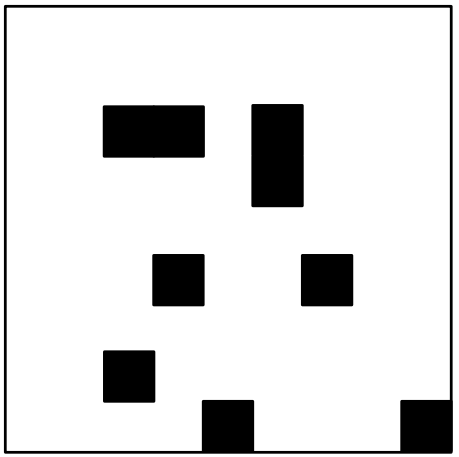
Finding Partition Shapes

- Consider all possible shapes, reduce using Push Technique

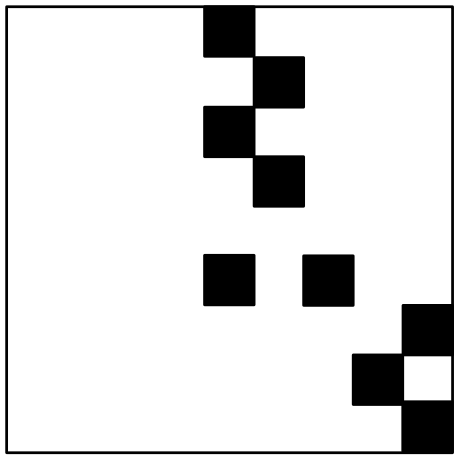
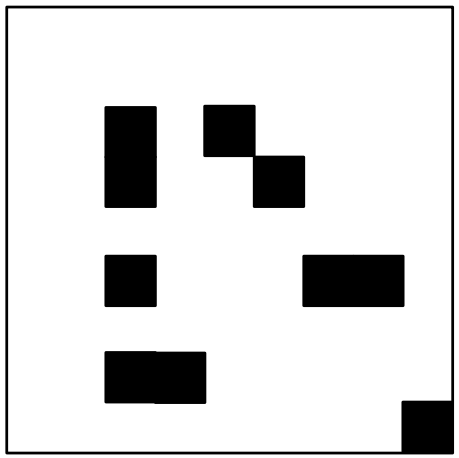
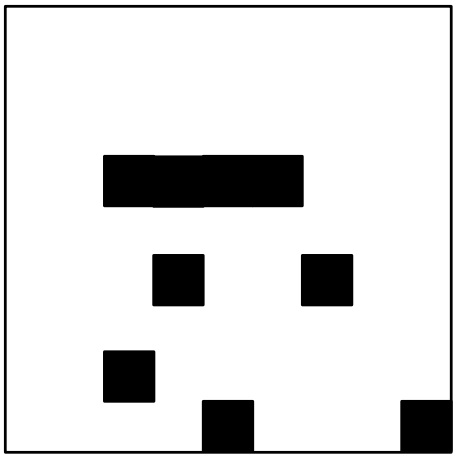
Finding Partition Shapes

- Consider all possible shapes, reduce using Push Technique

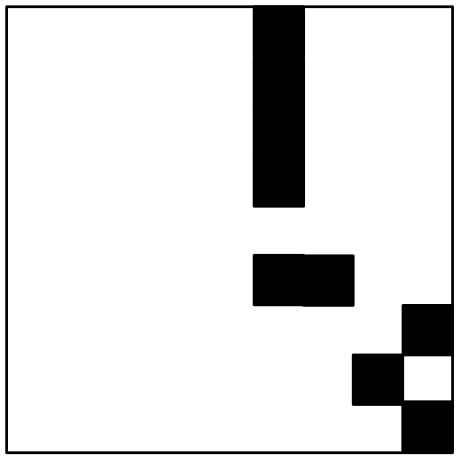
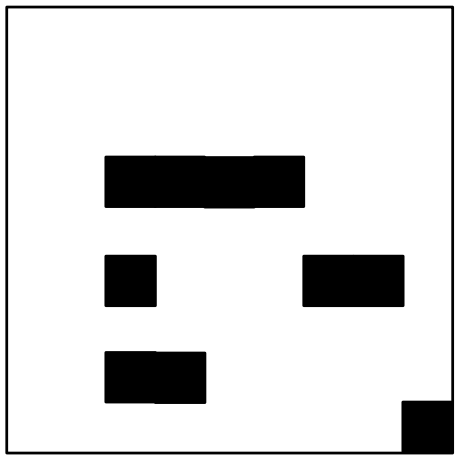
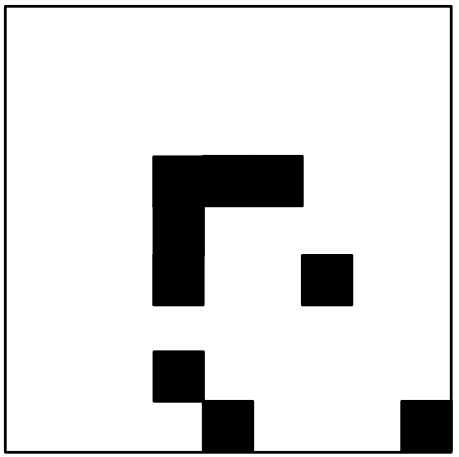
Finding Partition Shapes

- Consider all possible shapes, reduce using Push Technique

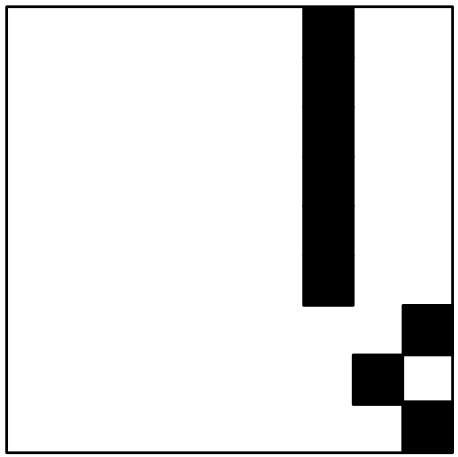
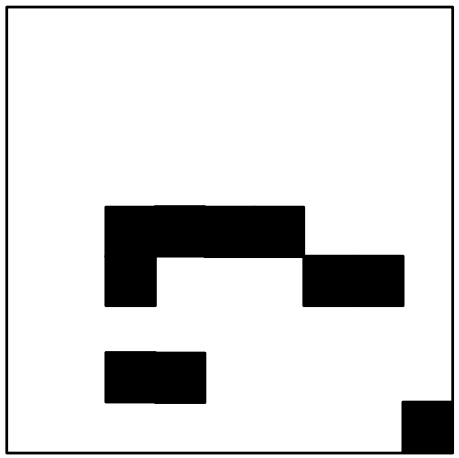
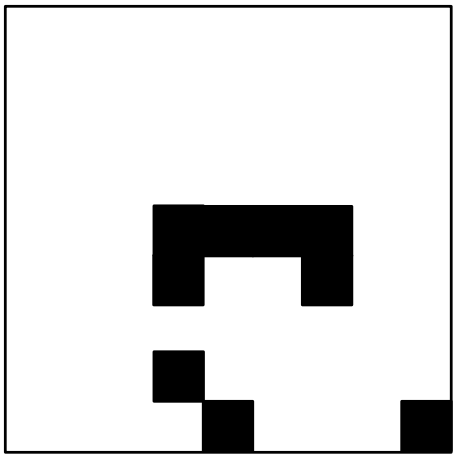
Finding Partition Shapes

- Consider all possible shapes, reduce using Push Technique

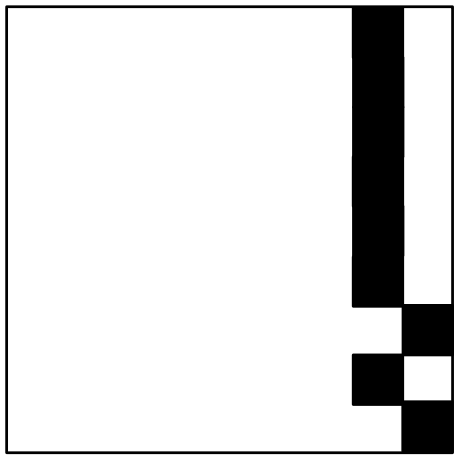
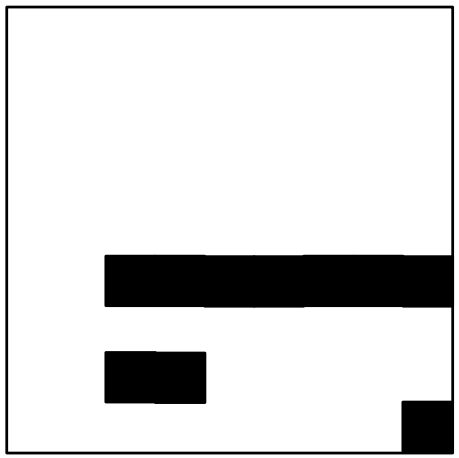
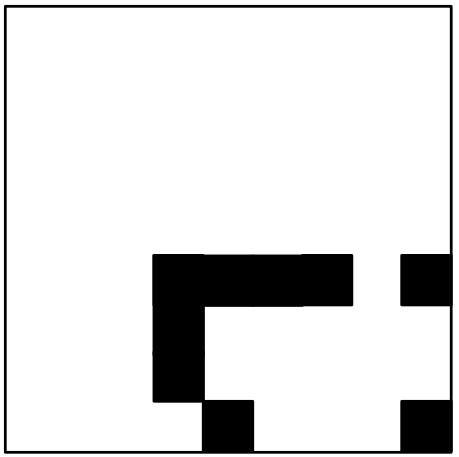
Finding Partition Shapes

- Consider all possible shapes, reduce using Push Technique

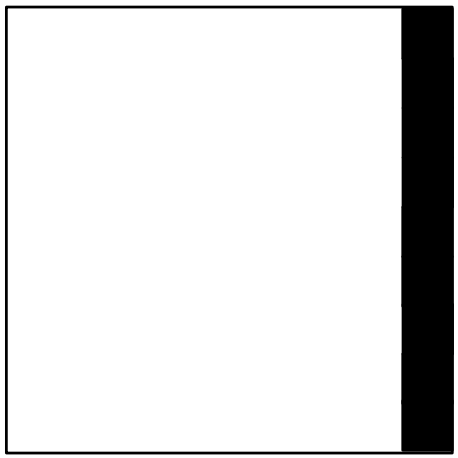
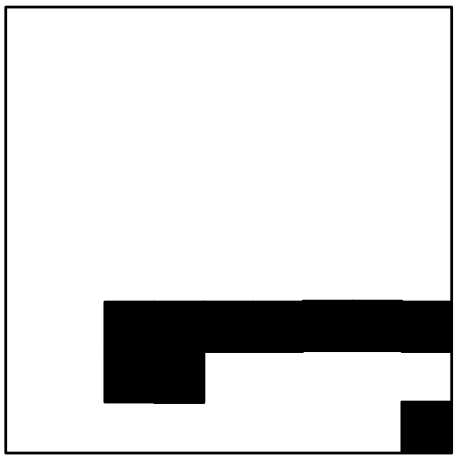
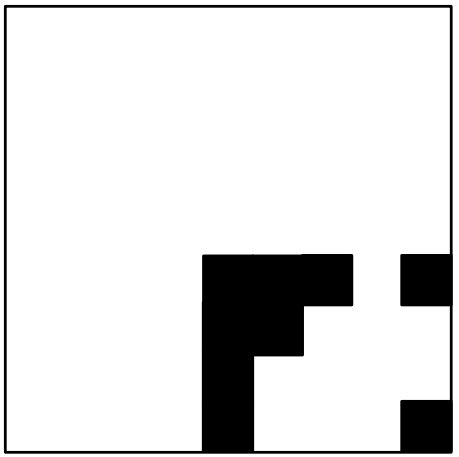
Finding Partition Shapes

- Consider all possible shapes, reduce using Push Technique

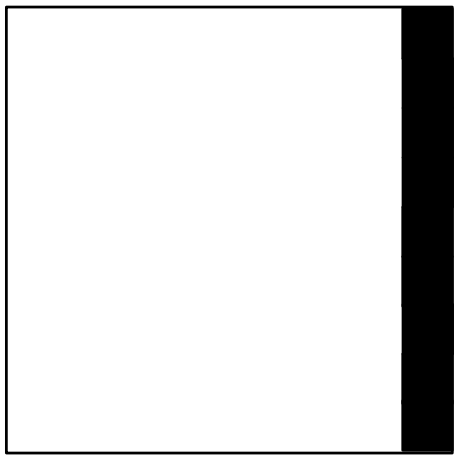
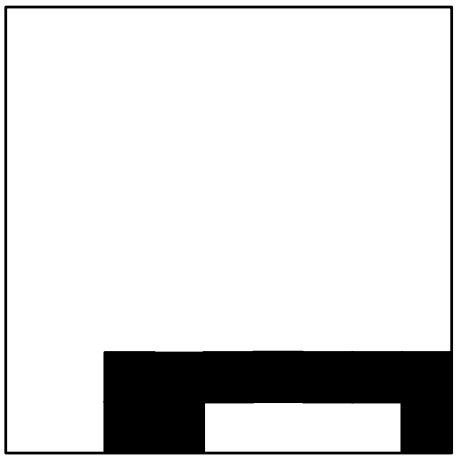
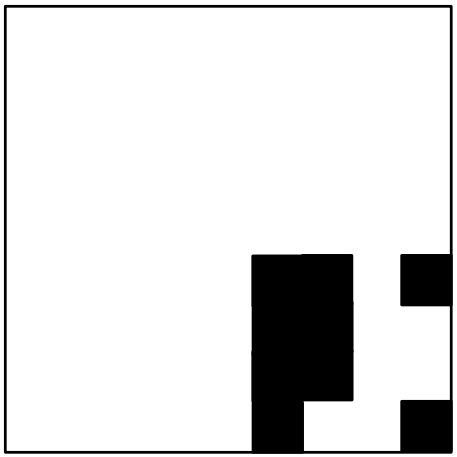
Finding Partition Shapes

- Consider all possible shapes, reduce using Push Technique

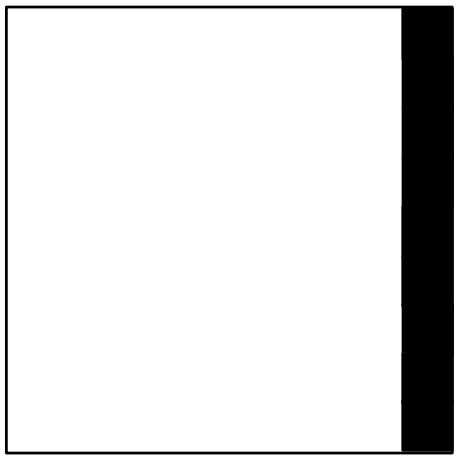
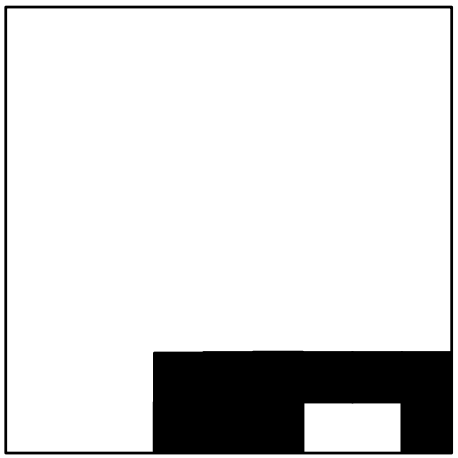
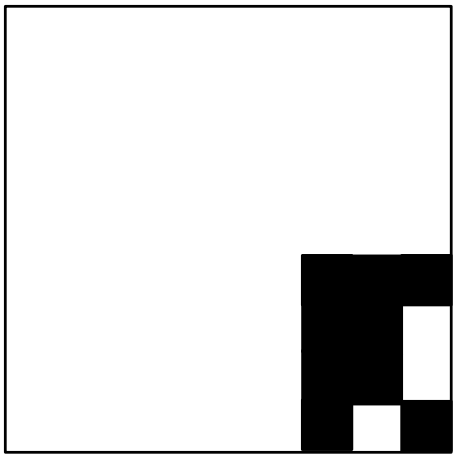
Finding Partition Shapes

- Consider all possible shapes, reduce using Push Technique

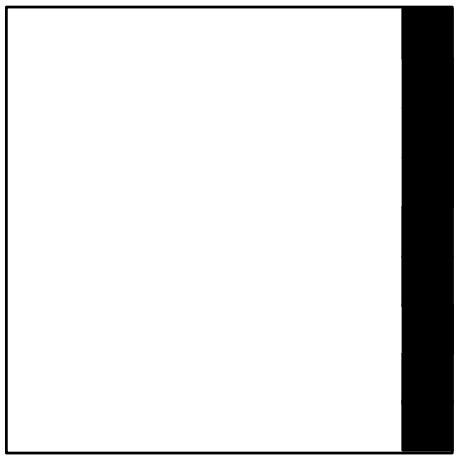
Finding Partition Shapes

- Consider all possible shapes, reduce using Push Technique

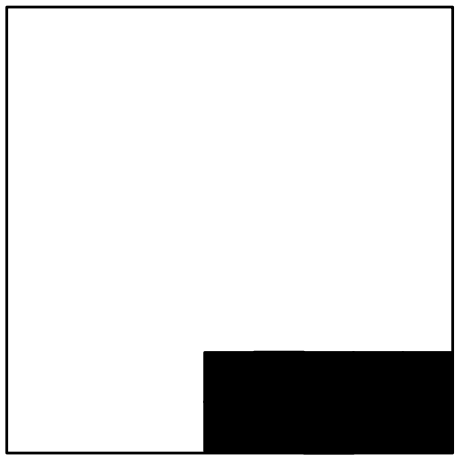




Finding Partition Shapes

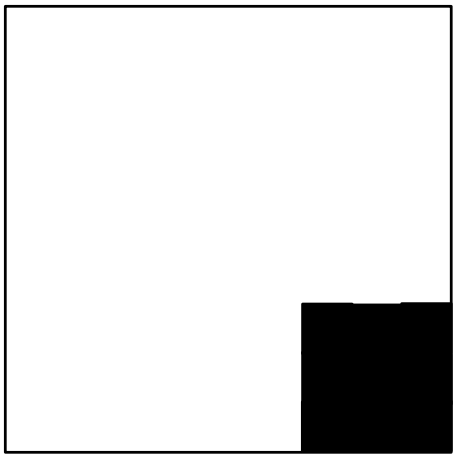
- Consider all possible shapes, reduce using Push Technique



Straight-Line

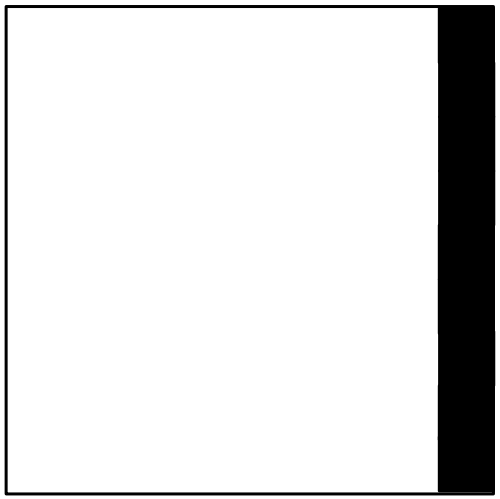


Rectangle-Corner

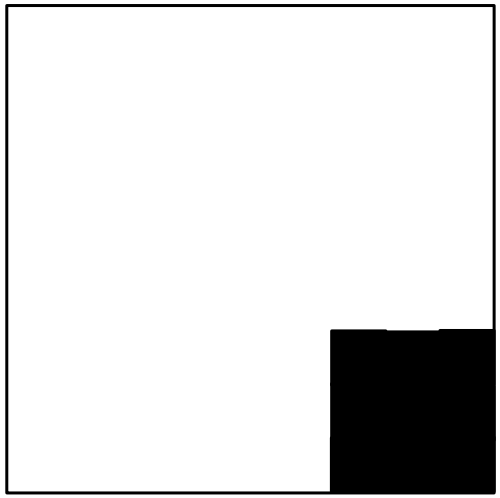


Square-Corner

Analyzing Partition Shapes



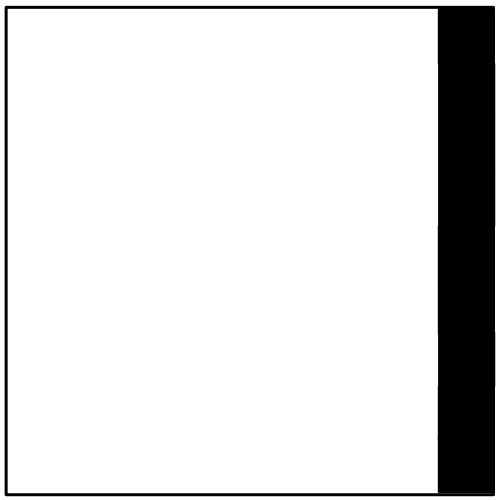
Straight-Line



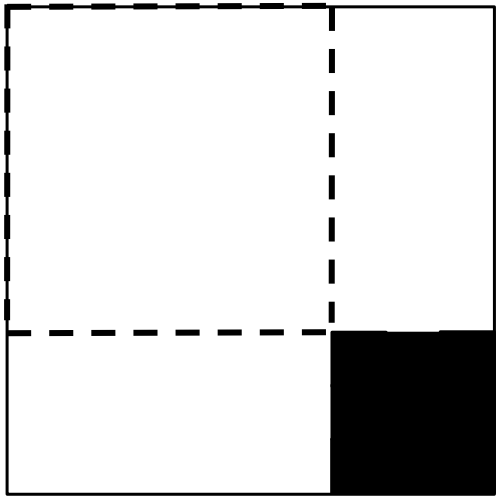
Square-Corner

Analyzing Partition Shapes

- Square-Corner optimal for all Overlap algorithms



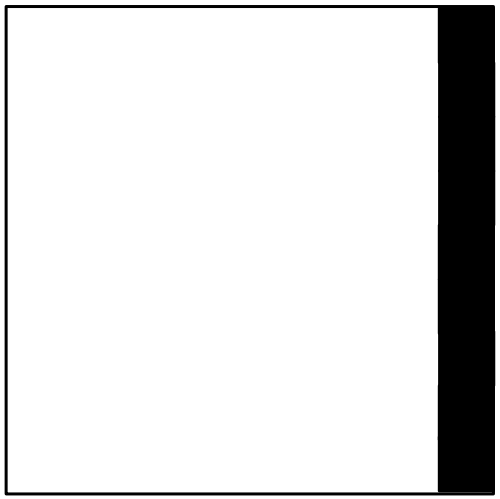
Straight-Line



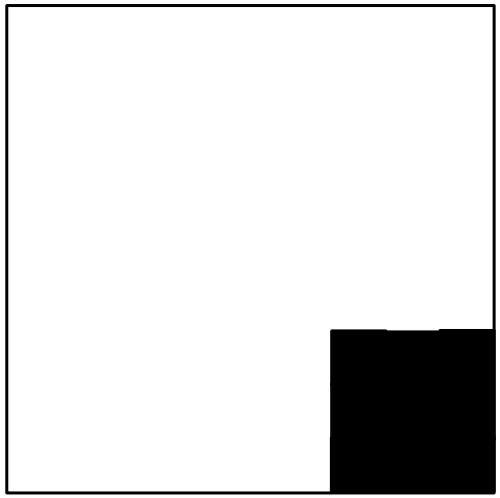
Square-Corner

Analyzing Partition Shapes

- Square-Corner optimal for all Overlap algorithms
- Square-Corner optimal for other algorithms when processor speed ratio > 3, Straight-Line optimal when speed ratio < 3



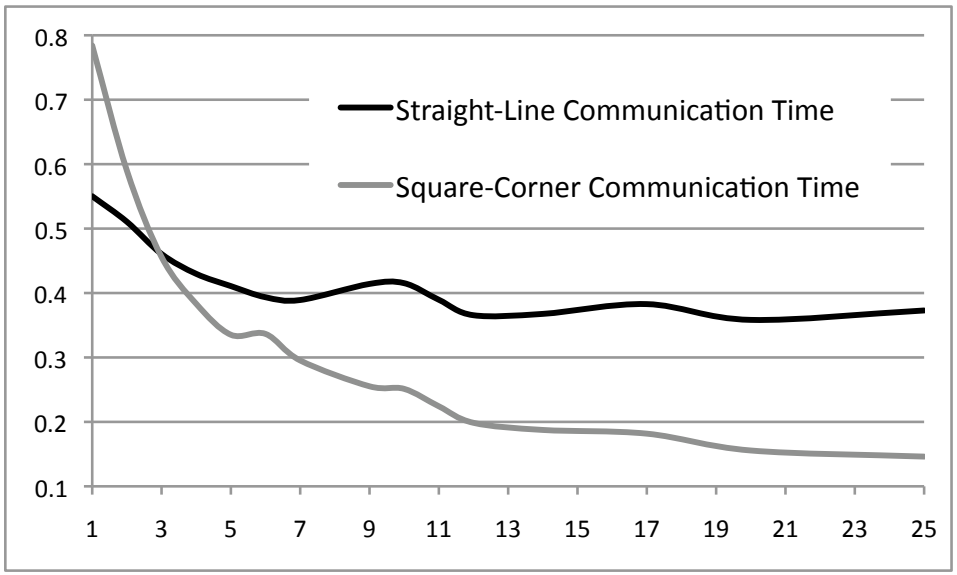
Straight-Line

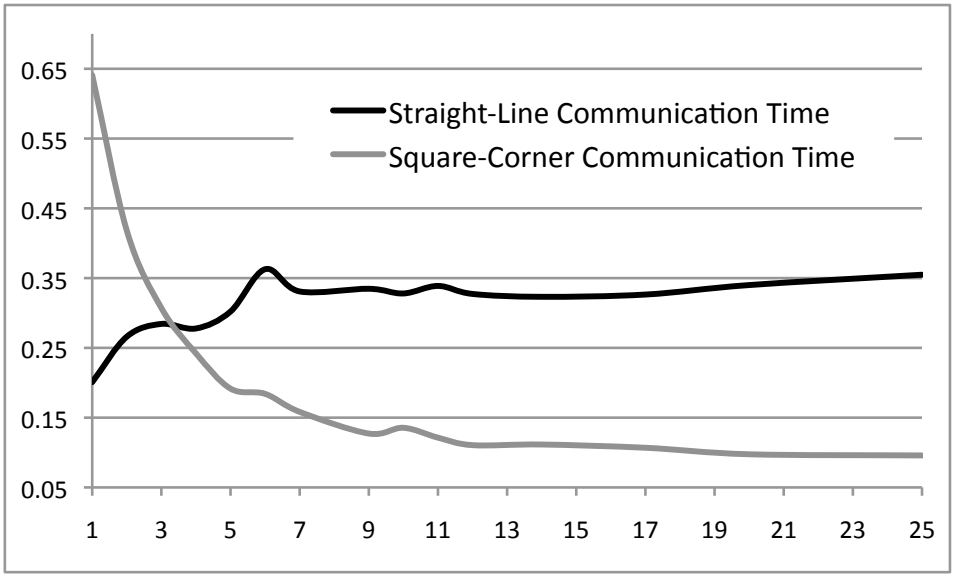


Square-Corner

Experimental Results

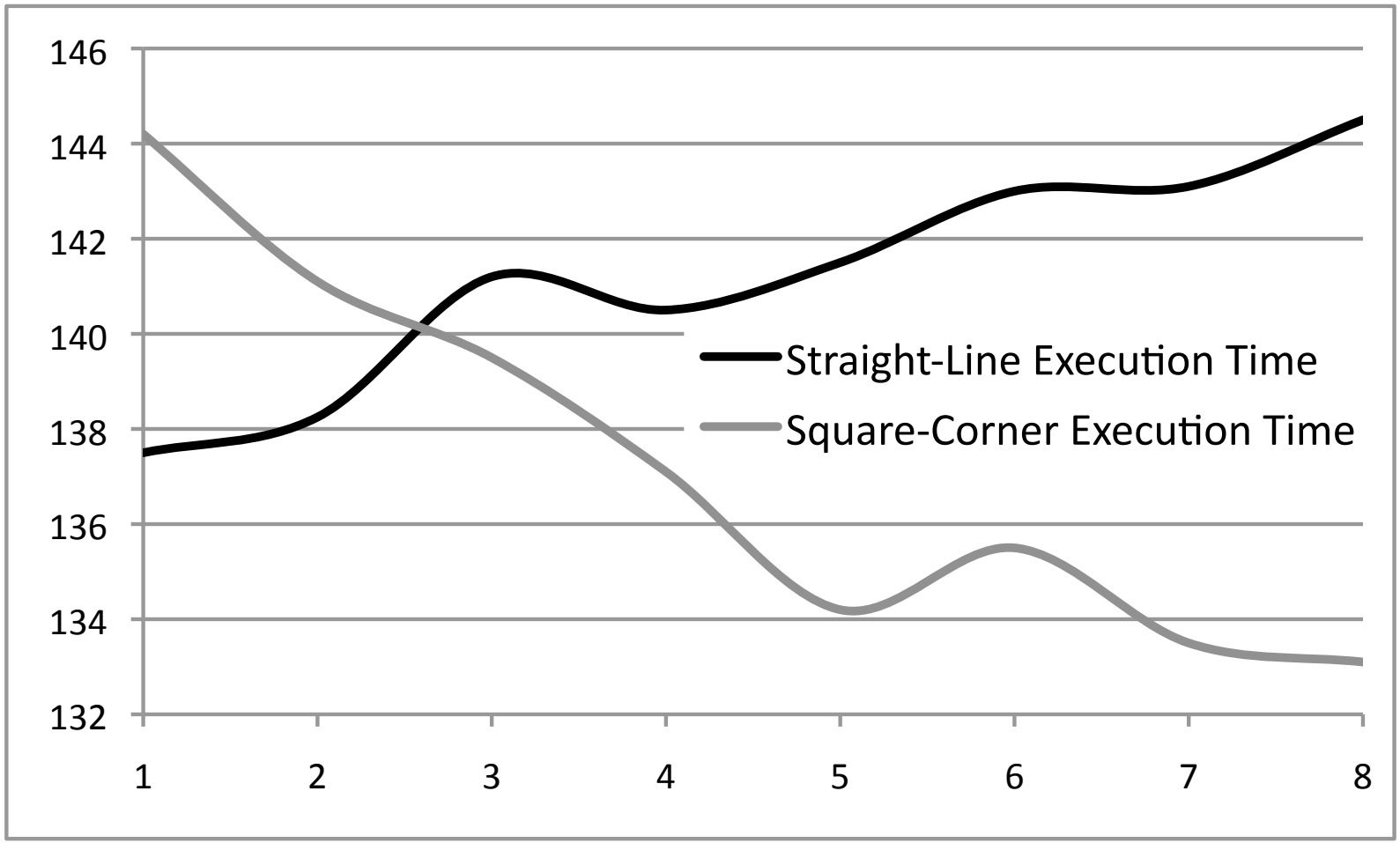
- Results on HCL Cluster - Serial Communication with Barrier (SCB) and Parallel Communication with Barrier (PCB) Algorithms
- Problem size, N = 3000





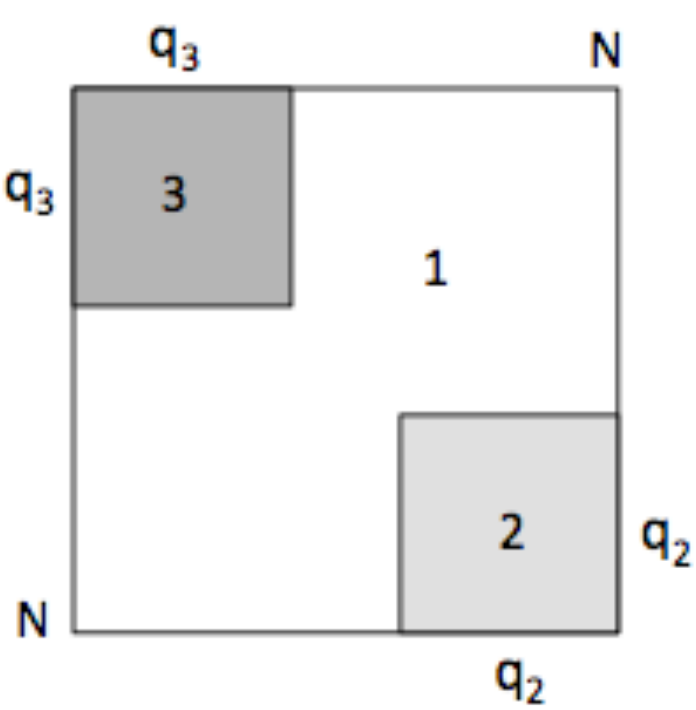
Experimental Results

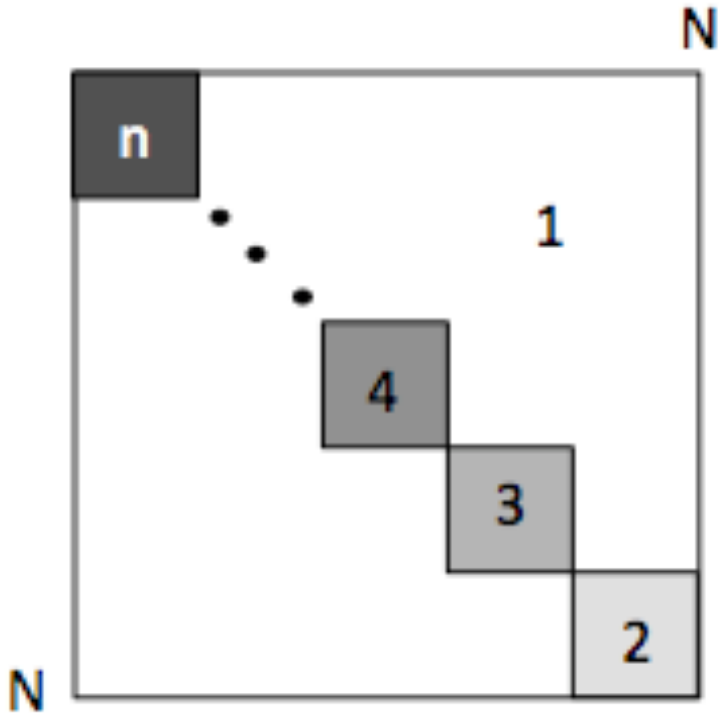
- Results on Grid'5000 - Serial Communication with Barrier (SCB)
- Problem size, N = 15,000. Network Bandwidth, 1 Gb/s



Using 3 or More Processors

- Know how to optimally partition Processors 1 and 2, or Processors 1 and 3, so combine the partitions without adding interaction between Processors 2 and 3.
- Same concept applies to an arbitrary number of processors, given processor speed ratios such that Processors 2 to n do not overlap





Conclusion

- The traditional rectangular approach to data partitioning is not universally optimal
- Non-rectangular partitions are particularly important for highly heterogeneous systems
- Even for more difficult problem of arbitrary number of processors, in some cases the non-rectangular solution is optimal

Thank You