Matrix Multiplication on Three Heterogeneous Processors

Brett A. Becker and Alexey Lastovetsky

Heterogeneous Computing Laboratory

School of Computer Science and Informatics

University College Dublin

PMAA'06 Rennes Sept. 8, 2006

Outline

- Motivation and Goals
- Background: ‘Square Corner’ Partitioning for 2 Processors
- Extension of ‘Square Corner’ Partitioning to 3 Processors
- Experiments / Results
- Conclusion

Motivation and Goals

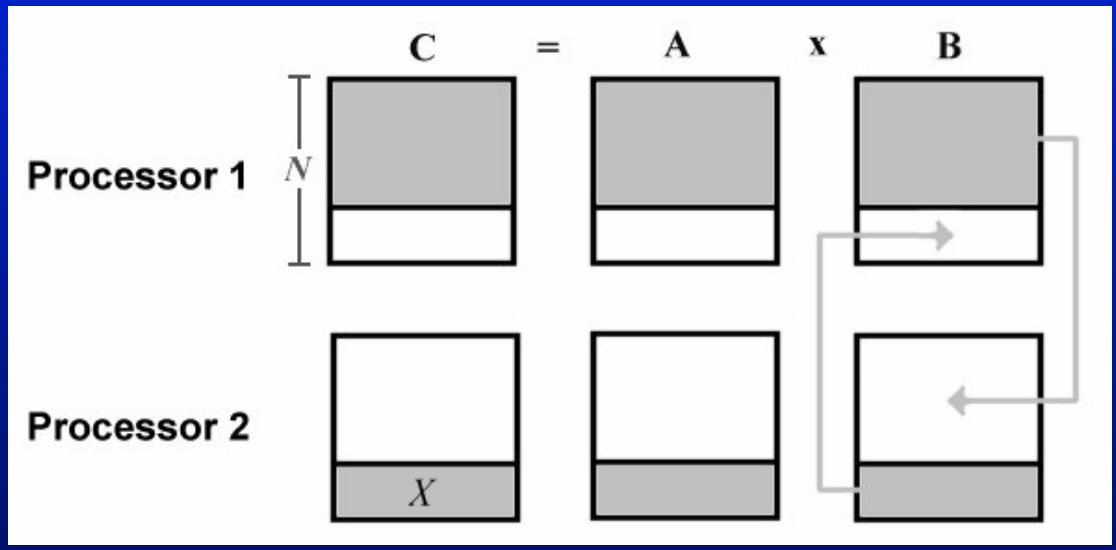
- Partitioning algorithms for MMM designed for *n* processors result in partitionings which are not always optimal on a small number of processors

- We previously investigated the problem of optimal partitioning for MMM on 2-Processors and developed a partitioning algorithm which substantially reduces communication volumes and execution times.

- Our goal is to extend this proven 2-Processor ‘Square-Corner’ algorithm to 3-Processor Networks

- Our ultimate interest is to determine if the Square-Corner partitioning is a viable technique for deployment on 3 interconnected Clusters, as is the case for 2 clusters.

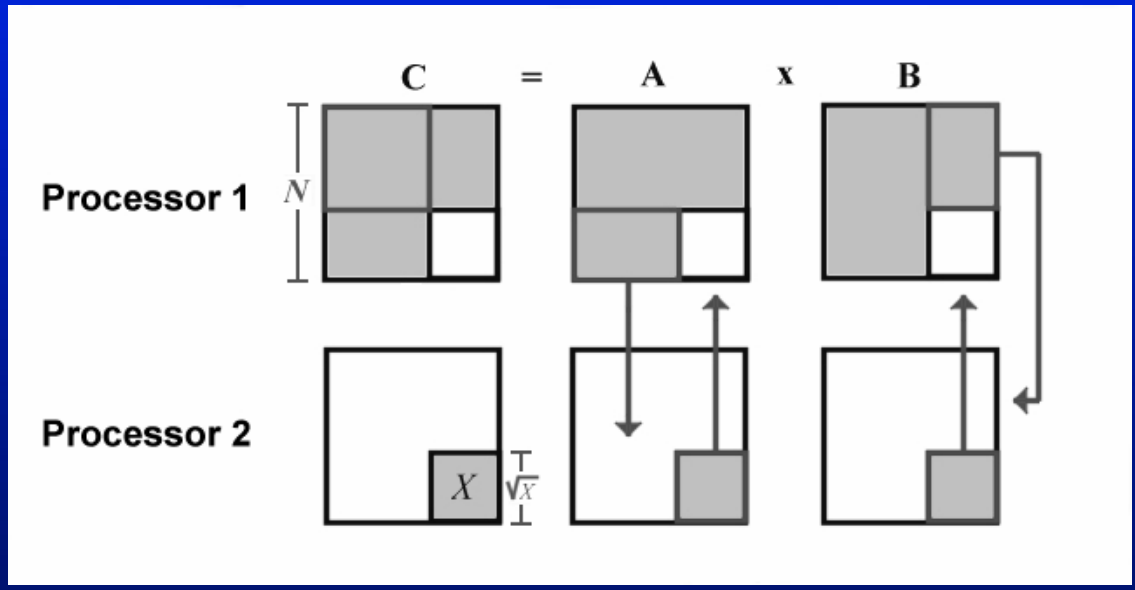
Background: Straight-Line Partitioning



Total Volume of Inter-Processor Communication (TVC) = N^2

$$\text{as } X \to 0, \text{TVC} \to N^2$$

Background: Square-Corner Partitioning



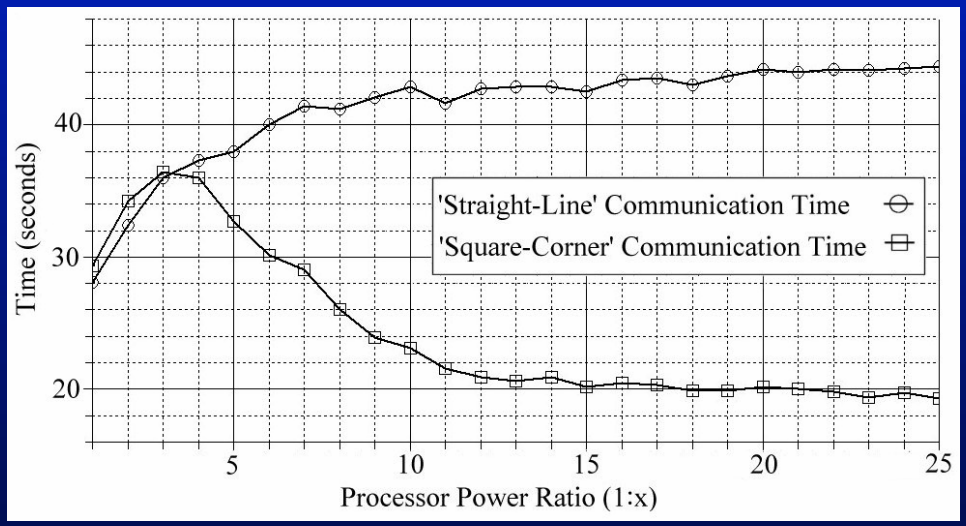
$$\text{TVC} = 2N\sqrt{X}$$

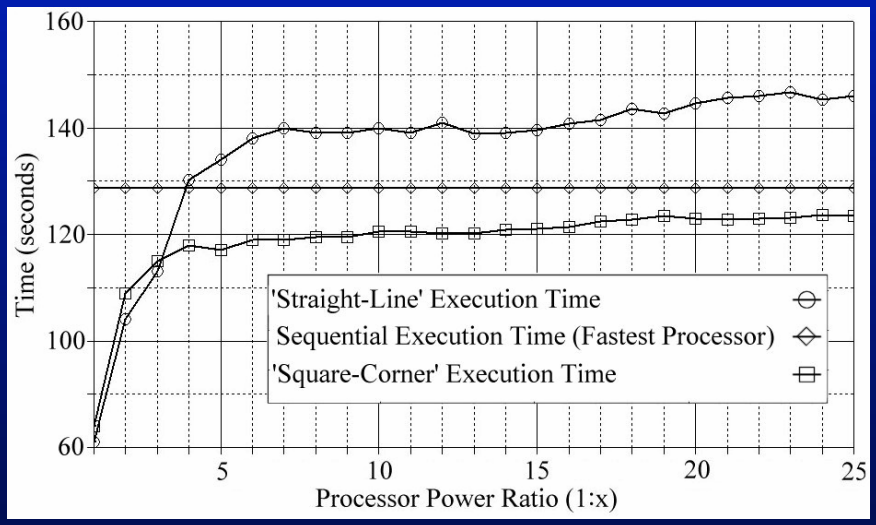
$$\text{as } X \to 0, \text{TVC} \to 0$$

$$2N\sqrt{X} < N^2, \text{provided processor power ratio} > 3:1$$

Background: Square-Corner Partitioning

Matrix-Matrix Multiplication, N=6500, Bandwidth = 80Mb/s

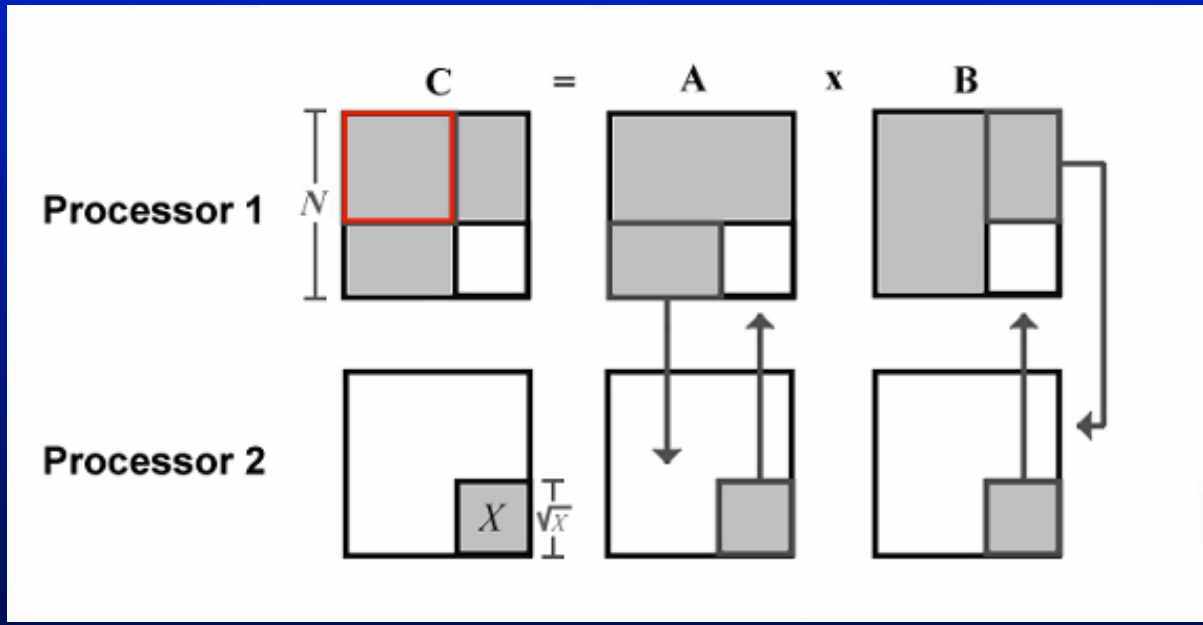




Lower TVC ⇨ Lower Communication Time ⇨ Lower Execution Time

Background: Square-Corner Partitioning

Overlapping Communication and Computation



A sub-partition of Processor 1's *C* Partition is Immediately Calculable

Background: Square-Corner Partitioning

Overlapping Communication and Computation

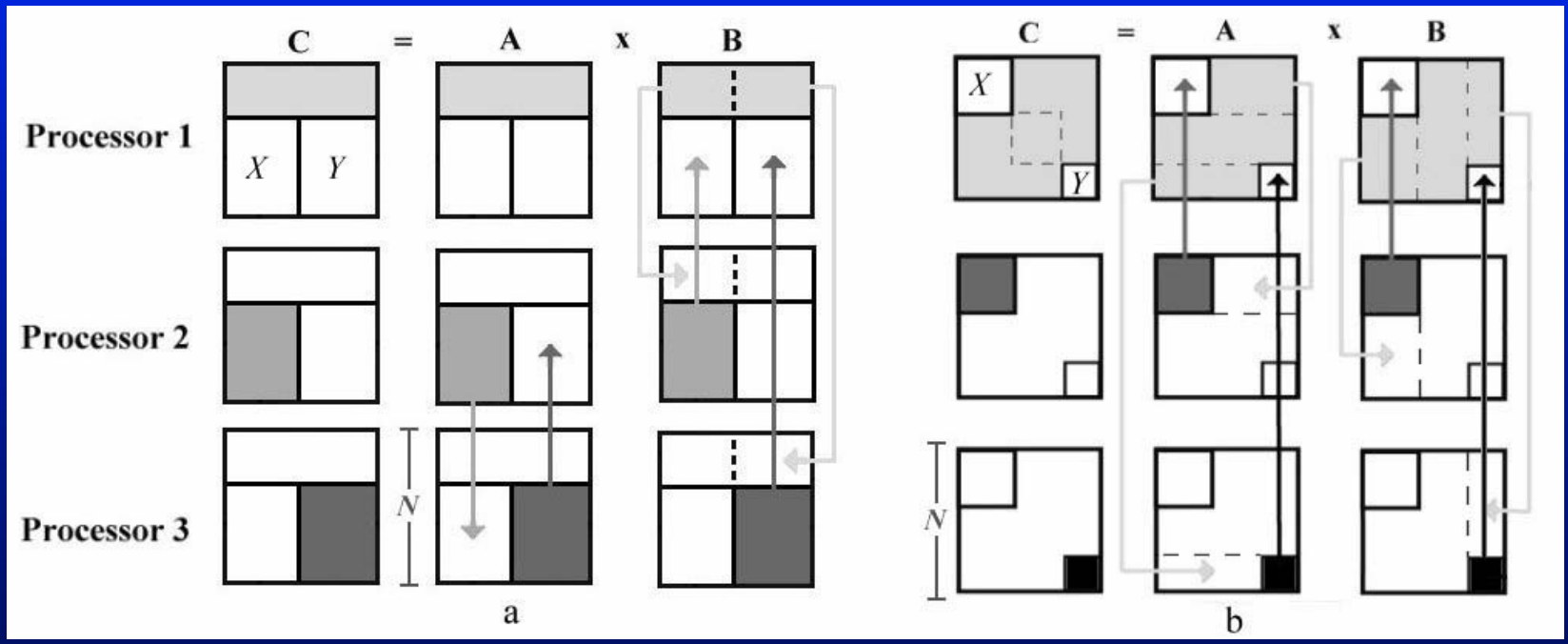
MM Multiplication, N=4500, Bandwidth=100Mb/s, Ratio=5:1,

Algorithm	Execution Time	Speedup
Straight-Line	83s	0.94
Square-Corner (No Overlapping)	69s	1.13
Square-Corner (Overlapping)	51s	1.53
Sequential	78s	N/A

Overlapping more than doubled advantage of Square-Corner algorithm.

- No Overlapping → 17% faster than Straight-Line algorithm.
- Overlapping → 39% faster than Straight-Line algorithm.

Extension to 3 Processors



a. Straight-Line Partitioning:

$$\text{TVC} = N^2 + X + Y$$

as $X \to 0$ and $Y \to 0, \text{TVC} \to N^2$

b. Square-Corner Partitioning:

$$\text{TVC} = 2N(\sqrt{X} + \sqrt{Y})$$

as $X \to 0$ and $Y \to 0, \text{TVC} \to 0$

Extension to 3 Processors

(Fully Connected Network)

Theorem: The Square-Corner Algorithm is Optimal on a fully connected network,

$$\text{provided } \sqrt{S2} + \sqrt{S3} < 1 - \frac{S1}{2}$$

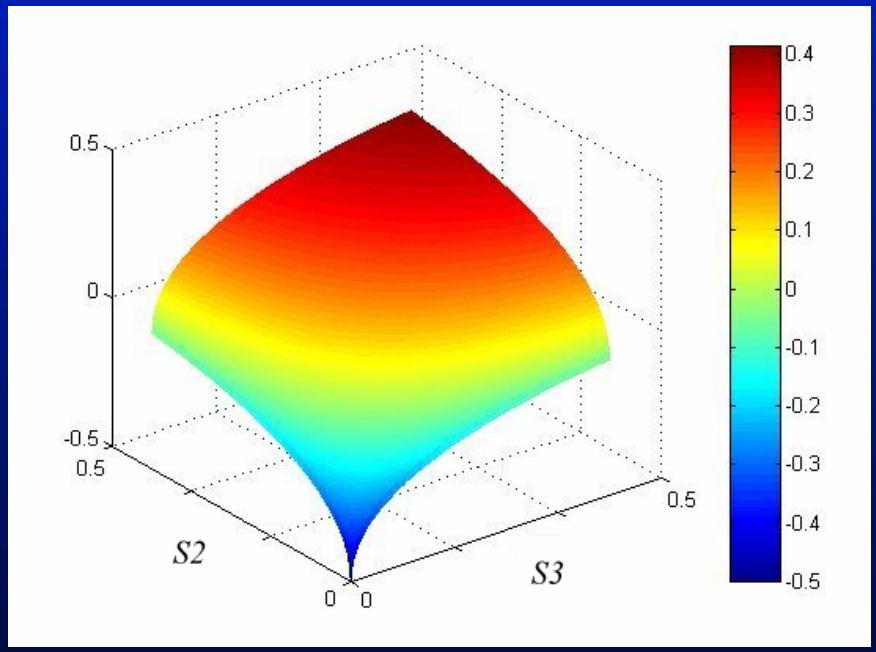
where $S1 : S2 : S3$ is the processor power ratio,

such that $S1 \geq (S2 + S3)$,

and $S1 + S2 + S3 = 1$

Extension to 3 Processors

(Fully Connected Network)



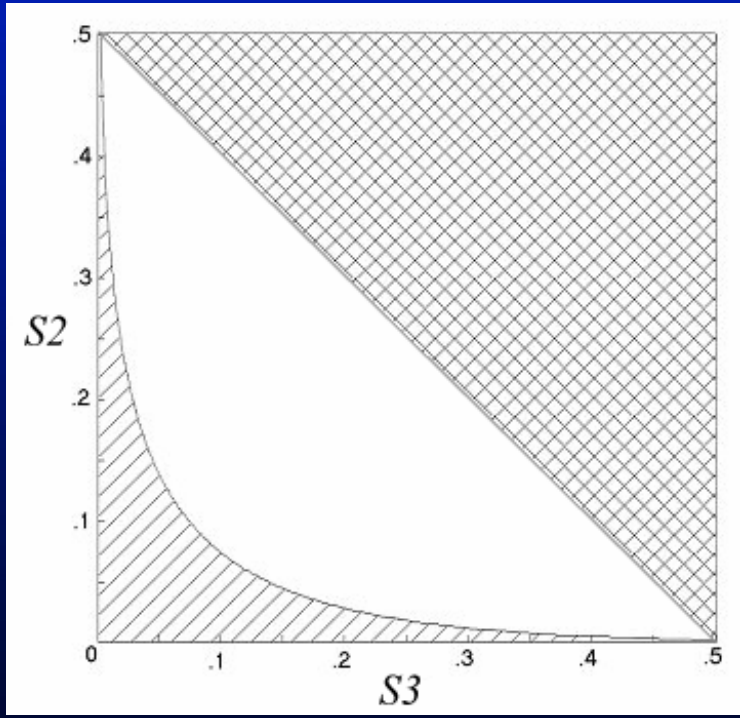
Ratio: S1:S2:S3

Plot of $\sqrt{S2} + \sqrt{S3} - \left(1 - \frac{S1}{2}\right)$

(Square-Corner TVC – Straight-Line TVC)

Extension to 3 Processors

(Fully Connected Network)



Contour plot of

$$\sqrt{S2}+\sqrt{S3}-\left(1-\frac{S1}{2}\right)$$

at $z = 0$

⇨ Violates $S1 \geq (S2+S3),\ S1+S2+S3 = 1$

⇨ Square-Corner TVC < Straight-Line TVC

Extension to 3 Processors: Results

(Fully Connected Network)

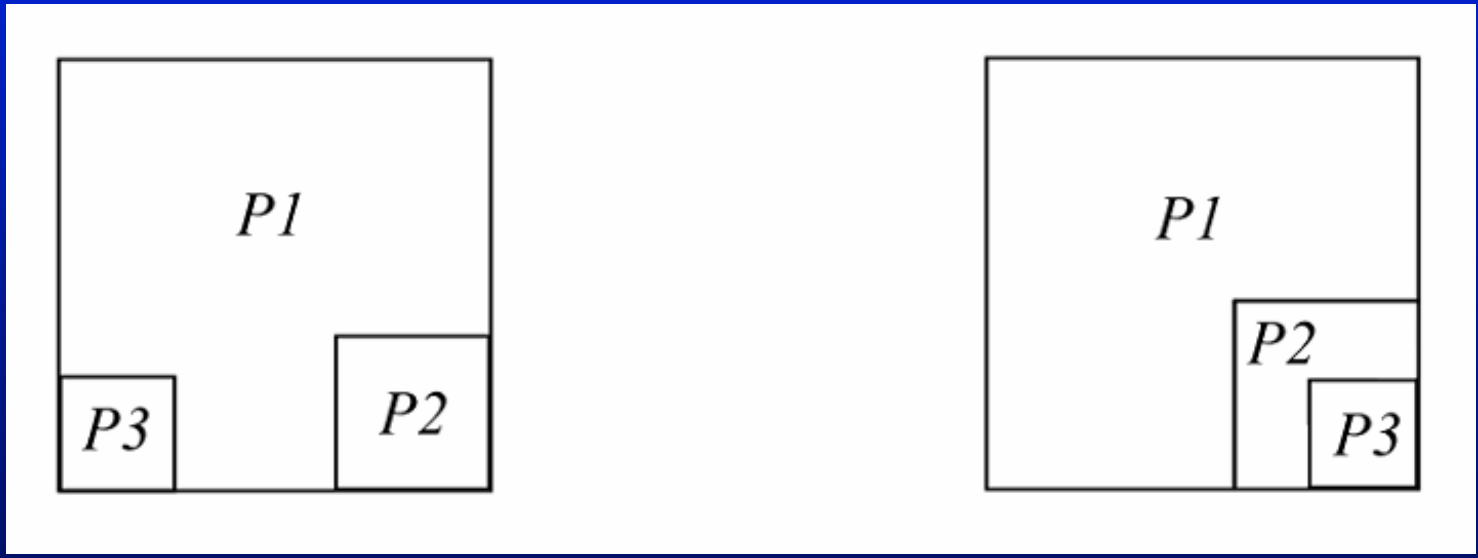
MM Multiplication, 14.5 : 1 : 1 ratio, Bandwidth = 80MB/s, N=5500

Algorithm	Communication Time	Execution Time
Straight-Line	31s	97s
Square-Corner	26s	93s
Sequential	N/A	74s

16% reduction In Communication Time, 4% reduction in Execution Time compared to Straight-Line Partitioning

Extension to 3 Processors

Some Other Strategies Considered



‘Adjacent Corners’:

TVC = Square-Corner TVC

More Communication Steps Necessary

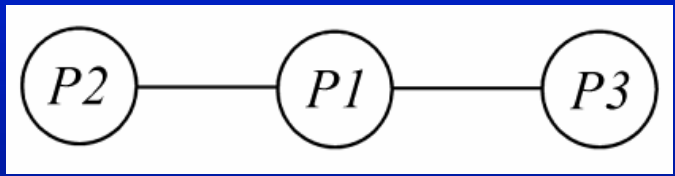
‘Nested’:

TVC > Square-Corner TVC

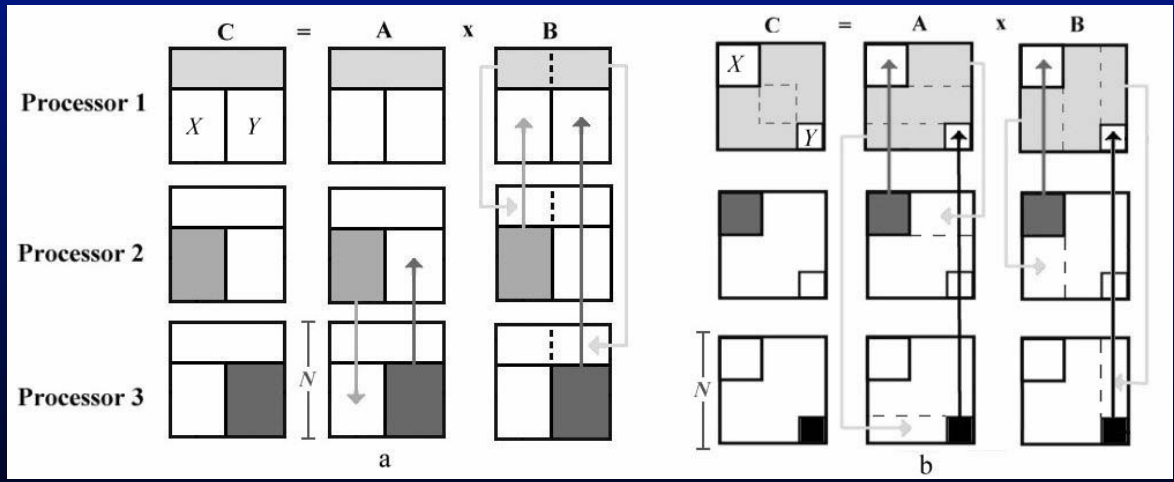
More Communication Steps Necessary

Extension to 3 Processors

(Linear Array)



$$S1 \geq S2 \geq S3$$



Extension to 3 Processors

(Linear Array)

Theorem: The Square-Corner Algorithm is Optimal on a Linear Array, provided the processor with speed $S1$ is the centre node.

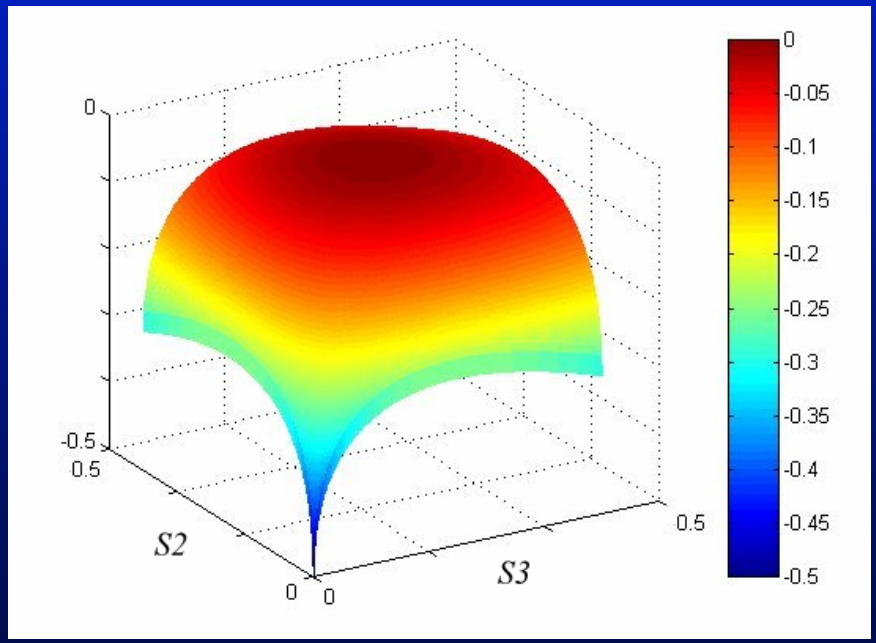
where $S1 : S2 : S3$ is the processor power ratio,

such that $S1 \geq (S2 + S3)$,

and $S1 + S2 + S3 = 1$

Extension to 3 Processors

(Linear Array)



Ratio: S1:S2:S3

Plot of $\sqrt{S2} + \sqrt{S3} - 1.5 + S1$

(Square-Corner TVC – Straight-Line TVC)

Extension to 3 Processors: Results

(Linear Array)

MM Multiplication, 5:3:2 ratio, Bandwidth = 80MB/s, N = 5500

Algorithm	Commuincation Time	Execution Time	Speedup
Straight-Line	81s	145s	0.88
Square-Corner	36s	102s	1.25
Sequential	N/A	128s	N/A

55% reduction In Communication Time, 30% reduction in Execution Time compared to Straight-Line Partitioning

Extension to 3 Processors: Results

(Linear Array)

MM Multiplication, 12:7:1 ratio, Bandwidth = 80MB/s, N=5500

Algorithm	Commuincation Time	Execution Time	Speedup
Straight-Line	53s	105s	1.02
Square-Corner	30s	82s	1.30
Sequential	N/A	107s	N/A

43% reduction In Communication Time, 22% reduction in Execution Time compared to Straight-Line partitioning

Extension to 3 Processors
Conclusion

- Fully Connected:

  In general, ratios where Square-Corner outperforms Straight-Line, parallel overhead is too great to yield useful speedup

- Linear Array:

  Square-Corner performs well enough to attempt implementation on 3 Clusters, provided P1 is the centre node [where S1>(S2+S3)]

Extension to 3 Processors
Future Work

- Overlapping Communication and Computation
- Deploy on Three Clusters

Acknowledgements

This work was supported by:

Heterogeneous
Computing
Laboratory

Extension to 3 Processors

(Linear Array)

Theorem: The Square-Corner Algorithm is Optimal on a Linear Array, provided the processor with speed *S*1 is the middle node

Proof Outline:

Start with Partitions that have an “inner” and “outer” space

Show that a disconnected partition has larger TVC than a single connected one.

Show that the TVC depends only on overall height and width of partition

Therefore a rectangle is no worse than any arbitrary shape of same area

Show that a square has lower TVC than a rectangle of the same area

Show that TVC of this partitioning is lower than Straight-Line partitionings

Extension to 3 Processors

Some examples of Suitable Linear Arrays

- Master – Slave Networks
- Star Networks (Below, Right)
- Heirarchial Networks (Below, Left)