Optimal Data Partitioning Shape for Matrix Multiplication on Three Fully Connected Heterogeneous Processors

Ashley DeFlumere and Alexey Lastovetsky

School of Computer Science and Informatics
University College Dublin
Belfield, Dublin 4, Ireland

Abstract. Parallel Matrix Matrix Multiplication (MMM) is used in scientific codes across many disciplines. While it has been widely studied how to optimally divide MMM among homogenous compute nodes, the optimal solution for heterogeneous systems remains an open problem. Dividing MMM across multiple processors or clusters requires consideration of the performance characteristics of both the computation and the communication subsystems. The degree to which each of these affects execution time depends on the system and the algorithm used to divide, communicate, and compute the MMM data. Our previous work has determined the optimum shape must be, for all ratios of processing power, communication bandwidth and matrix size, one of six well-defined shapes for each of the five MMM algorithms studied. This paper further reduces the number of potentially optimal candidate shapes to three defined shapes known as Square Corner, Square Rectangle, and Block Rectangle. We then find, for each algorithm and all ratios of computational power among processors, ratios of overall computational power and communication speed, and problem size, the optimum shape. The Block Rectangle, a traditional 2D rectangular partition shape, is predictably optimal when using relatively homogeneous processors, and is also optimal for heterogeneous systems with a fast, medium and slow processor. However, the Square Corner shape is the optimum for heterogeneous environments with a powerful processor and two slower processors, and the Square Rectangle is optimal for heterogeneous environments composed of a two fast processors and a single less powerful processor. These theoretical results are confirmed using a series of experiments conducted on Grid'5000, which show both that the predicted optimum shape is indeed optimal, and that the remaining two partition shapes perform in their predicted order.

1 Introduction

The problem of partitioning Parallel Matrix Matrix Multiplication (MMM) optimally over an arbitrary number of processors has been the subject of extensive study. While this problem, when approached using homogeneous processors, presents a challenge, it is significantly more substantive when considering heterogeneous systems. High performance scientific computing platforms are increasingly heterogeneous, so it is necessary to find the optimum heterogeneous MMM data partition shape[1]. While a system may be heterogeneous in its computational power, its communication interconnect, or some combination of both, this paper will focus on heterogeneity in computational power.

The bulk of the previous study of MMM partitioning on heterogeneous platforms has been concerned with finding the optimal *rectangular* partitioning[2][3][4]. Even when restricting the optimality problem to only rectangular shapes, it is complex and NP-complete for an arbitrary number of heterogeneous processors[5]. The underlying assumption that the optimal shape should be rectangular has only recently been questioned.

Our previous work challenged this traditional assumption, and explored both rectangular and non-rectangular data partition shapes[6][7]. These papers, encompassing work with both two and three processor systems, show optimal, and potentially optimal, partition shapes that have both expected and unexpected shapes. The two processor case, for instance, has an optimal data partition shape which is non-rectangular for highly heterogeneous systems, *i.e.*, when the ratio of computational power between the two processors is greater than three.

The complexity of the optimal shape problem necessitates beginning with a small number of processors in order to establish an extensible method for identifying potentially optimal partition shapes. This novel method, called the Push Technique, incrementally improves a partition shape by decreasing its volume of communication. The Push Technique has previously been applied to the case of three heterogeneous processors, and identified six potentially optimal partition shapes, called candidates. These are seen in Fig. 1.

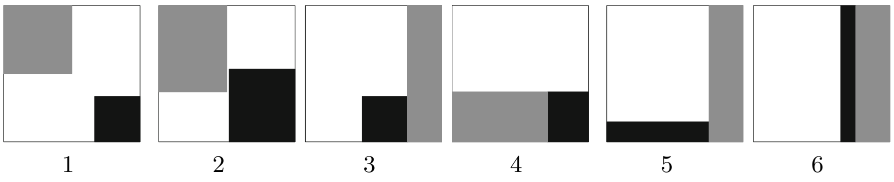


Fig. 1. The candidate partition shapes previously identified as potentially optimal three processor shapes. Processors P, R, and S are in white, grey, and black, respectively. (1) Square Corner (2) Rectangle Corner (3) Square Rectangle (4) Block 2D Rectangular (5) L Rectangular (6) Traditional 1D Rectangular.

These cases, with small numbers of processors, are also practically significant. Consider a GPU-CPU hybrid system. The concept of abstract processors may be used to model this type of system[8]. Each logical processor represents an independent group of tightly coupled devices such as cores on the same socket, or a GPU and its host core. In this way, a modern hybrid compute node is modelled by a small number of abstract heterogeneous processors.

This paper proves that the optimal candidates may be further reduced to just three optimal partition shapes, the Square Corner, the Square Rectangle, and the

Block Rectangle. For each MMM algorithm, each of these shapes is optimal for a subset of the possible ranges of computational power ratios and communication bandwidths. Together, they describe the optimal shape for all possible ranges of these values. These theoretical results are further verified using experiments on GRID'5000.

2 Problem Description

Throughout, we will make several assumptions, as follows:

1. Matrices A, B and C are square, of size $N \times N$, and identically partitioned among Processors P, R, and S, represented in figures as white, grey and black, respectively.
2. Processor P computes faster than Processors R and S by ratio, $P_r : R_r : S_r$, where $S_r = 1$.
3. All Processors may communicate with all other Processors, with no constraints on network topology.

For all algorithms, we use the Hockney Model[9] of communication $T_{comm} = \alpha \times \beta M$. For simplicity, we will set $\alpha = 0$. The total volume of communication is calculated as $M = \sum_{i=1}^{N} N(p_i - 1) + \sum_{j=1}^{N} N(p_j - 1)$, where p_i is the number of processors assigned elements in row i, and p_j is the number of processors assigned elements in column j. The method of computation in all algorithms is assumed to be SUMMA[10].

3 Theoretical Results

3.1 Methodology

Partition shapes have defined metrics that are used to determine the optimality of a given shape in a particular problem space. Some of these metrics quantify the volume of communication of a particular shape. The volume of communication is, in turn, used to create the model of communication time, T_{comm}, within the constraints of the MMM algorithm. The volume of elements assigned to each processor for computation, and the relative computational power of each processor, is used to create the model of computation time, T_{comp}. These two fundamental parts of the MMM, T_{comm} and T_{comp}, are combined according to the MMM algorithm to create a total execution time, T_{exe}, for the particular partition shape.

The partition shape which minimises the execution time for a specific MMM algorithm is said to the be the optimum shape. However, no single shape is the global optimum for an entire MMM algorithm. Each shape has unique characteristics which allow for increased performance under certain conditions, such as varied processor computational ratios, and the ratio between overall computation and communication speeds.

The sections below describe the process of forming the T_{exe} model for each shape using each MMM algorithm and analysing those models to find the minimum, and thereby the optimum. The proofs for all theorems found throughout this paper may be examined in [11].

3.2 Pruning Candidates

Upon further inspection of the six potentially optimal candidate shapes found in [7], it is possible to analytically reduce this to three candidate shapes.

Theorem 1 (Three Candidates). *The three partition shapes known as Rectangle Corner, L Rectangle and Traditional Rectangle, have a higher theoretical volume of communication than the Block Rectangle shape. The optimal shape must be among the remaining three candidate shapes, Block Rectangle, Square Rectangle and Square Corner.*

From here, we will analyse only the remaining three candidate partition shapes: Square Corner, Square Rectangle, and Block Rectangle.

3.3 Serial Communication with Barrier (SCB)

Serial Communication with Barrier (SCB) is a simple MMM algorithm in which all data is sent by each processor *serially*, and only once communication completes among all processors does the computation proceed in *parallel* on each processor.

The execution time is given by,

$$T_{exe} = V\beta + \max(c_P, c_R, c_S)$$

where V is the volume of communication, β is the bandwidth of the communication links and c_X is the time taken to compute the assigned portion of the matrix on Processor X.

Each processor is assigned data in proportion to the computational power. Processors P, R and S, with ratios $P_r : R_r : 1$ will be assigned $\frac{P_r N^2}{T}, \frac{R_r N^2}{T}$ and $\frac{N^2}{T}$ elements to compute, respectively. For all shapes, the computation time is identical for barrier algorithms, so communication time is the focus.

Square Corner. The Square Corner shape is composed of a matrix partitioned into two small squares for Processors R and S, while Processor P is assigned the non-rectangular remainder of the matrix. This shape type is only valid for computational power ratios such that non-overlapping squares for Processors R and S may be formed, which is possible when $P_r \geq 2\sqrt{R_r}$.

$$T_{comm(SC)} = 2N\left(\sqrt{\frac{R_r N^2}{T}} + \sqrt{\frac{N^2}{T}}\right) \times \beta \quad (1)$$

Square Rectangle. The Square Rectangle shape is composed of an N height rectangle, R, and a square, S, while Processor P is assigned the non-rectangular remainder of the matrix. The communication time is given by,

$$T_{comm(SR)} = \left(N^2 + 2N\sqrt{\frac{N^2}{T}}\right) \times \beta \tag{2}$$

Block Rectangle. The Block Rectangle partition shape is composed of two h height rectangles of combined width N. Processor P is assigned the rectangular remainder of the matrix.

$$T_{comm(BR)} = \left(2N^2 - \frac{P_r N^2}{T}\right) \times \beta \tag{3}$$

Optimum SCB Shape. The optimum data partitioning shape minimises T_{comm}. A graphical representation of these three functions can be seen in Fig. 2.

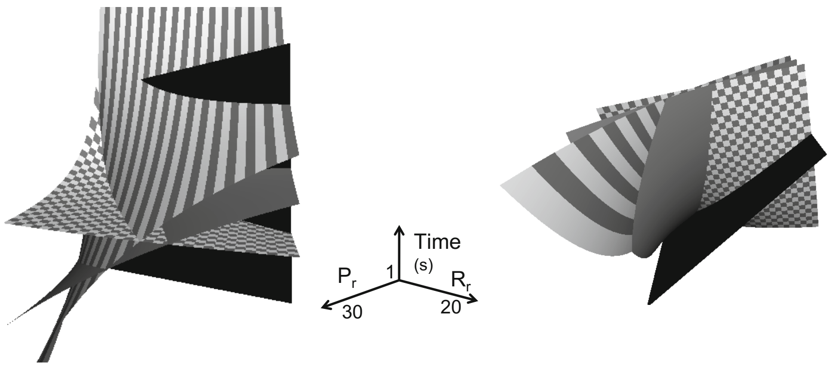


Fig. 2. The SCB T_{comm} functions for the three candidate shapes, Square Corner (white and grey stripes), Block Rectangle (solid grey), and Square Rectangle (white and grey checkerboard). The x-axis is the relative computational power of P, P_r, from 1 to 30. The y-axis is the relative computational power of R, R_r, from 1 to 20. The z-axis is the communication time in seconds. The vertical black surface is the equation $x = y$, and represents the problem constraint $P_r \geq R_r$. On the left, viewed from the front, on the right, viewed from underneath (the lowest function is optimal).

Theorem 2 (SCB Square Corner). *The Square Corner partition shape minimises execution time, i.e. is the optimum, using the SCB MMM algorithm for all processor computational power ratios such that* $P_r < 2T - 2\sqrt{R_r T} - 2\sqrt{T}$.

Theorem 3 (SCB Square Rectangle). *The Square Rectangle partition shape minimises execution time, i.e. is the optimum, using the SCB MMM algorithm for all processor computational power ratios such that* $P_r < T - 2\sqrt{T}$.

Corollary 4 (SCB Block Rectangle) *The Block Rectangle partition shape minimises execution time, i.e. is the optimum, for all processor computational power ratios except those specified in Theorems 2 and 3.*

3.4 Parallel Communication with Barrier (PCB)

In the Parallel Communication with Barrier (PCB) algorithm, all data is sent among processors in *parallel*, and only once communication completes does the computation processed in *parallel* on each processor. The execution time of this algorithm is given by,

$$T_{exe} = \max(v_P, v_R, v_S) \times \beta + \max(c_P, c_R, c_S)$$

where v_X is the volume of data elements which must be sent by Processor X. As with SCB, the focus in this algorithm is on communication time because computation time is not dependent on the data partition shape.

Communication Time Functions. The communication times of partition shapes Square Corner (SC), Square Rectangle (SR), and Block Rectangle (BR) are given by,

$$T_{comm(SC)} = 2N^2\beta \times \max\left(\sqrt{\frac{R_r}{T}} - \frac{R_r}{T} + \sqrt{\frac{1}{T}} - \frac{1}{T}, \frac{R_r}{T}, \frac{1}{T}\right) \quad (4)$$

$$T_{comm(SR)} = N^2\beta \times \max\left(1 + \frac{2}{\sqrt{T}} - \frac{R_r}{T} - \frac{R_r}{T\sqrt{T}} - \frac{3}{T}, \frac{R_r}{T} + \frac{R_r}{T\sqrt{T}}, \frac{3}{T}\right) \quad (5)$$

$$T_{comm(BR)} = N^2\beta \times \max\left(\frac{P_r}{T}, \frac{2R_r}{T}, \frac{2}{T}\right) \quad (6)$$

PCB Optimal Shape. The optimum partition shape minimises T_{comm}. The graph of these three functions is found in Fig. 3.

Theorem 5 (PCB Square Corner). *The Square Corner partitioning shape minimizes execution time, i.e. is the optimum shape, when using the PCB MMM algorithm and the computational power ratios are such that* $P_r > 2(\sqrt{R_r T} - R_r + \sqrt{T} - 1)$.

Theorem 6 (PCB Square Rectangle). *The Square Rectangle partitioning shape minimizes execution time, i.e. is the optimum shape, when using the PCB MMM algorithm and the computational power ratios are such that* $P_r < 2R_r + \frac{R_r}{\sqrt{T}} - 2\sqrt{T} - 1$ *and* $P_r > 5 + \frac{R_r - 2}{\sqrt{T}}$.

Corollary 7 (PCB Block Rectangle) *The Block Rectangle partition shape minimises execution time, i.e. is the optimum, for all processor computational power ratios except those specified in Theorems 5 and 6.*

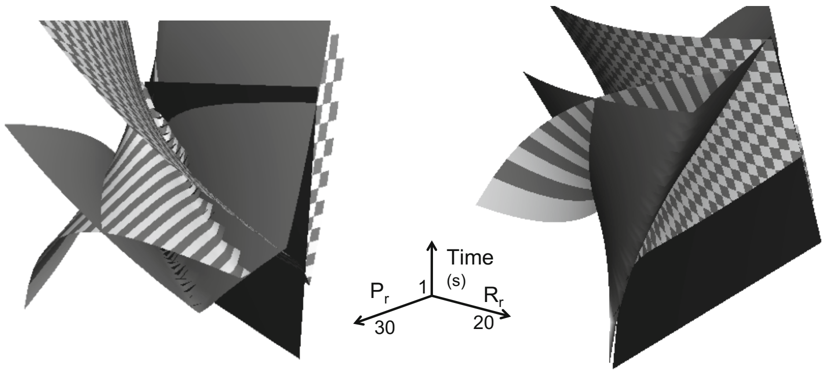


Fig. 3. The PCB T_{comm} functions for the three candidate shapes, Square Corner (white and grey stripes), Block Rectangle (solid grey), and Square Rectangle (white and grey checkerboard). The vertical black surface is the equation $x = y$, and represents the problem constraint $P_r \geq R_r$. On the left, viewed from the front, on the right, view from underneath (the lowest function is optimal).

3.5 Serial Communication with Bulk Overlap (SCO)

In the Serial Communication with Bulk Overlap (SCO) algorithm, all data is sent by each processor *serially*, while in *parallel* any elements that can be computed without communication are computed. Only once both communication and overlapped computation are complete does the remainder of the computation begin. The execution time is given by,

$$T_{exe} = \max\Big(\max(T_{comm}, o_P) + c_P, \max(T_{comm}, o_R) + c_R, \max(T_{comm}, o_S) + c_S\Big)$$

where T_{comm} is the same as that of the SCB algorithm, o_X is the number of seconds taken by Processor X to compute any elements not requiring communication, and c_X is the number of seconds taken to compute the remainder of the elements assigned to Processor X.

Square Corner Of the three candidate partitions, only the Square Corner has an o_X term which is not equal to zero, *i.e.* it contains elements which may be computed without any communication amongst processors. The overlap-able area may be seen Fig. 4. The addition of the non-zero o_P term implies that c_P will no longer be equal to c_R and c_S if we continue to naively assign the volume of elements as $\frac{N^2 P_r}{T}$. As Processor P should be assigned a larger portion of the matrix to compute than suggested by P_r.

To determine this optimal size, we first assume that the volumes (and thereby the size of the squares) assigned to Processors R and S should decrease in proportion to each other, so their computation times remain equal ($c_R = c_S$). The size of a side of the square R, r, and a side of the square S, s, is set at $s = \sqrt{\frac{r^2}{R_r}}$.

We may safely ignore the third term (Processor S) of the SCO max function, as it will always be equal to the second term (Processor R). Execution time is given by,

$$\frac{T_{exe}}{N^3\beta} = \max\left(\max\left(\frac{2}{N}\Big(\sqrt{\frac{R_r}{T}}+\sqrt{\frac{1}{T}}\Big), \frac{1-\frac{r}{\sqrt{R_r}}-2r+\frac{r^2}{R_r}+\frac{2r^2}{\sqrt{R_r}}+r^2}{c}\right)\right.$$
$$\left. +\frac{2}{c}\Big(r-r^2-\frac{r^2}{\sqrt{R_r}}+\frac{r}{\sqrt{R_r}}-\frac{r^2}{R_r}\Big), \quad \frac{2}{N}\Big(\sqrt{\frac{R_r}{T}}+\sqrt{\frac{1}{T}}\Big)+\frac{r^2P_r}{cR_r}\right)$$

In order to make the execution time equations easier to analyse, the constant factor $N^3\beta$ has been removed. This introduces a new variable, a ratio between computation and communication speeds, $c = Sp\beta$, where $\frac{Sp}{N}$ is the number of elements computed per second by Processor P. The size of N and r have been normalised, so that $\frac{r}{N}$ becomes r, and r is understood to be $0 \leq r < 1$.

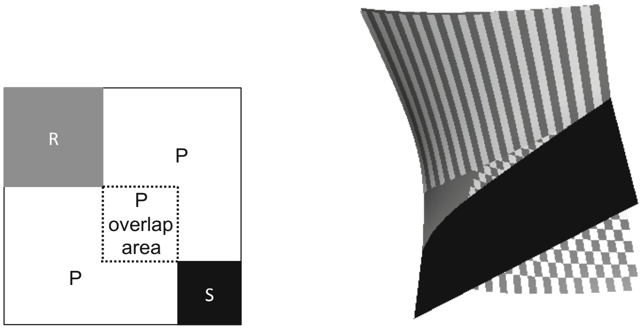


Fig. 4. On the left, the area of Processor P which does not require communication in the Square Corner partition shape is enclosed in dotted lines. On the right, the graph of execution time functions for the SCO algorithm. Axes as are in previous graphs, and $N = 3000$ and $c = 50$.

Optimal Size of *R* and *S*. The optimal size of r is given by,

$$r = \frac{\sqrt{-(\frac{P_r}{R_r}+1+\frac{1}{R_r})(\frac{2c}{N}\sqrt{\frac{R_r}{T}}+\frac{2c}{N}\sqrt{\frac{1}{T}}-1)}}{(\frac{P_r}{R_r}+1+\frac{1}{R_r})} \tag{7}$$

The full derivation of this value may be found in [11].

Square Rectangle and Block Rectangle. The computation of no portion of matrix C may be overlapped with communication. The execution time function is equivalent to that for the SCB algorithm. Total execution time is given by,

$$\frac{T_{exe(SR)}}{N^3\beta} = \frac{1}{N} + \frac{2}{N}\sqrt{\frac{1}{T}} + \max\left(\frac{P_r}{Tc}, \frac{P_r}{Tc}, \frac{P_r}{Tc}\right)$$
$$\frac{T_{exe(BR)}}{N^3\beta} = \frac{2}{N} - \frac{P_r}{TN} + \max\left(\frac{P_r}{Tc}, \frac{P_r}{Tc}, \frac{P_r}{Tc}\right)$$

SCO Optimal Shape

Theorem 8 (SCO Square Corner). *The Square Corner partition shape minimizes execution time, i.e. is the optimum shape, when using the SCO MMM algorithm for computational ratios such that* $P_r > \frac{\frac{2}{N}(\sqrt{\frac{R_r}{T}}+\sqrt{\frac{1}{T}})+\frac{2}{c}(r-r^2-\frac{r^2}{\sqrt{R_r}}+\frac{r}{\sqrt{R_r}}-\frac{r^2}{R_r})-\frac{2}{N}}{\frac{1}{Tc}-\frac{1}{TN}}$ *and* $P_r > \frac{2c}{N}(\sqrt{R_r T} + \sqrt{T}) + 2T(r - r^2 - \frac{r^2}{\sqrt{R_r}} + \frac{r}{\sqrt{R_r}} - \frac{r^2}{R_r}) - \frac{Tc}{N} - \frac{2c}{N}\sqrt{T}$*, where r is the optimal size of the square R, given in (7).*

Theorem 9 (SCO Square Rectangle). *The Square Rectangle partition shape minimizes execution time, i.e. is the optimum shape, when using the SCO MMM algorithm for computational ratios such that* $P_r < T - 2\sqrt{T}$ *and* $P_r < \frac{2c}{N}(\sqrt{R_r T} + \sqrt{T}) + 2T(r - r^2 - \frac{r^2}{\sqrt{R_r}} + \frac{r}{\sqrt{R_r}} - \frac{r^2}{R_r}) - \frac{Tc}{N} - \frac{2c}{N}\sqrt{T}$

Corollary 10 (SCO Block Rectangle) *The Block Rectangle partition shape minimizes execution time, i.e. is the optimum shape, for all processor computational power ratios except those specified in Theorems 8 and 9.*

3.6 Parallel Communication with Bulk Overlap (PCO)

In the Parallel Communication with Bulk Overlap (PCO) algorithm all data is sent among processors in *parallel*, while in *parallel* any elements that can be computed without communication are computed. Once both communication and overlapped computation are complete, the remainder of the computation begins. The execution time for this algorithm is given by,

$$T_{exe} = \max\Big(\max(T_{comm}, o_P) + c_P, \max(T_{comm}, o_R) + c_R, \max(T_{comm}, o_S) + c_S\Big)$$

where T_{comm} is the same as that of the PCB algorithm. As with the SCO algorithm, we simplify the equations by removing constant $N^3\beta$, normalising N, and making the size of s dependent on the size of r. The optimal size of r is derived in [11].

Square Corner

$$\frac{T_{exe}}{N^3\beta} = \max\left(\max\left(\frac{2}{N}\max\left(r - r^2 + \frac{r}{\sqrt{R_r}} - \frac{r^2}{R_r}, r^2\right), \frac{1 - \frac{r}{\sqrt{R_r}} - 2r + \frac{r^2}{R_r} + \frac{2r^2}{\sqrt{R_r}} + r^2}{c} \right.\right.$$
$$\left.\left. +2\frac{r - r^2 + \frac{r}{\sqrt{R_r}} - \frac{r^2}{\sqrt{R_r}} - \frac{r^2}{R_r}}{c}, \qquad \frac{2}{N}\max\left(r - r^2 + \frac{r}{\sqrt{R_r}} - \frac{r^2}{R_r}, r^2\right) + \frac{r^2 P_r}{cR_r}\right)\right) \tag{8}$$

Square Rectangle and Block Rectangle. As with the SCO algorithm, the Square Rectangle and Block Rectangle shapes do not have a portion which may be overlapped with communication. The time of execution, as with PCB model, is given by,

$$\frac{T_{exe(SR)}}{N^3\beta} = \max\left(\frac{1}{N} + \frac{2}{N\sqrt{T}} - \frac{R_r}{NT} - \frac{R_r}{NT\sqrt{T}} - \frac{3}{NT}, \frac{R_r}{NT} + \frac{R_r}{NT\sqrt{T}}, \frac{3}{NT}\right) + \frac{P_r}{Tc} \tag{9}$$

$$\frac{T_{exe(BR)}}{N^3\beta} = \max\left(\frac{P_r}{NT}, \frac{2R_r}{NT}, \frac{2}{NT}\right) + \frac{P_r}{Tc} \tag{10}$$

PCO Optimal Shape. As with the PCB algorithm, the Block Rectangle shape is superior to the Square Rectangle shape when $P_r > 2R_r + \frac{R_r}{\sqrt{T}} - 2\sqrt{T} + 2$. When examining all three shapes to determine the optimal, we see that as c decreases, all three equations converge. However, for larger values of c, the clear winner for all computational power ratios is the Square Corner shape as seen in Fig. 5. The full proof of this is found in [11].

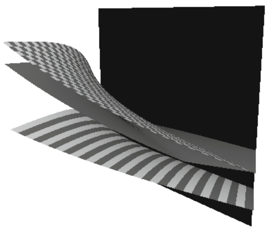


Fig. 5. The PCO execution time functions for Square Corner (white and grey stripes), Block Rectangle (solid grey), and Square Rectangle (white and grey checkerboard). The x-axis, P_r, is 1 to 30, and the y-axis, R_r displays values 1 to 20. The vertical black surface is $x = y$. The Square Corner shape is increasingly superior as c increases. Shown here is $N = 3000$ and $c = 300$.

3.7 Parallel Interleaving Overlap (PIO)

The Parallel Interleaving Overlap (PIO) algorithm, unlike the previous algorithms described, does not use bulk communication. At each step data is sent, a row and a column (or k rows and columns) at a time, by the relevant processor(s) to all processor(s) requiring those elements, while, in *parallel*, all processors compute using the data sent in the previous step. The execution time for this algorithm is given by,

$$T_{exe} = \text{Send } k + (N-1) \times \max\Big(\beta(V_k), \max\big(k_P, k_R, k_S\big)\Big) + \text{ Compute } (k+1)$$

where V_k is the number of elements sent at step k, and k_X is the number of seconds to compute step k on Processor X.

In the case of the PIO algorithm, the processors compute at the same time, meaning the optimal distribution will be in proportion to their computational power. The optimal size of the r and s is therefore $\sqrt{\frac{R_r N^2}{T}}$ and $\sqrt{\frac{N^2}{T}}$, respectively. In order to analyse the equations, we remove the constant factor $N^4\beta$ and focus on the dominant middle term which is multiplied by $(N-1)$.

Execution Time. The execution time for each partition shape, Square Corner (SC), Square Rectangle (SR), and Block Rectangle (BR), is given by,

$$\frac{T_{exe(SC)}}{N^4\beta} = \max\left(\frac{2}{N^2}\left(\sqrt{\frac{R_r}{T}} + \sqrt{\frac{1}{T}}\right), \frac{P_r}{Tc}\right) \tag{11}$$

$$\frac{T_{exe(SR)}}{N^4\beta} = \max\left(\frac{2}{N^2}\left(1 + 2\sqrt{\frac{1}{T}}\right), \frac{P_r}{Tc}\right) \tag{12}$$

$$\frac{T_{exe(BR)}}{N^4\beta} = \max\left(\frac{P_r}{N^2 T}, \frac{P_r}{Tc}\right) \tag{13}$$

PIO Optimal Shape. When computation time dominates, all three partition shapes are equivalent. However, when communication time dominates, they differ.

Theorem 11 (PIO Block Rectangle). *The Block Rectangle partition shape minimises execution time when using the PIO algorithm for computational power ratios such that $P_r < 4\sqrt{T}$.*

Corollary 12 (PIO Square Corner) *The Square Corner partition shape minimises execution time, i.e. is the optimum shape, for all processor computational power ratios except those specified in Theorem 11 when using the PIO algorithm.*

4 Experimental Results

To validate the theoretical results of this paper we present experiments undertaken on Grid'5000 in France using the Edel cluster at the Grenoble site. Each algorithm was tested using three nodes, comprised of 2 Intel Xeon E5520 2.2 GHz CPUs per node, with 4 cores per CPU. The communication interconnect is MPI over gigabit ethernet, and the computations use ATLAS. Heterogeneity in processing power was achieved using the cpulimit program, an open source code that limits the number of cycles a process may be active on the CPU to a percentage of the total. For space considerations, we present only results from SCB and PCB here.

4.1 Serial Communication with Barrier

The experimental results, for communication time, with the SCB algorithm can be found in Fig. 6. Note it is not possible to form a Square Corner shape at ratio 1 : 1 : 1. These experiments show that the theoretical optimum does indeed outperform the other possible shapes, which also perform in the expected order. We did find, that while the Square Corner and Square Rectangle shapes are theoretically identical at the 14 : 5 : 1 ratio, the Square Rectangle performed slightly better experimentally.

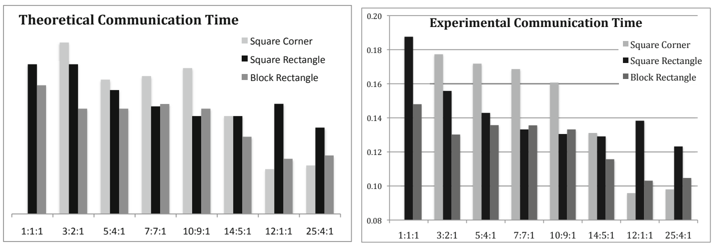
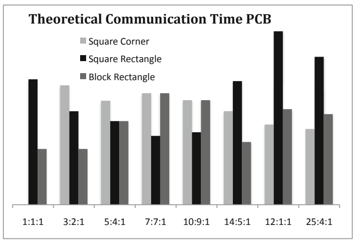


Fig. 6. On the left is the theoretical relative communication time for Square Corner, Square Rectangle and Block Rectangle when using the SCB algorithm. On the right is the experimental communication time (in seconds) for given ratios of $P_r : R_r : 1$. The value of N is 5000.

4.2 Parallel Communication with Barrier

The experimental results, for communication time, with the PCB algorithm can be found in Fig. 7. Note it is not possible to form a Square Corner shape at ratio 1 : 1 : 1. The results conform to the theoretical predictions with the optimum shape performing best, and the other two shapes performing in their predicted order.



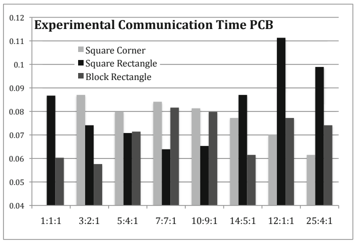


Fig. 7. On the left is the theoretical relative communication time for Square Corner, Square Rectangle and Block Rectangle partition shapes when using the PCB algorithm. On the right is the experimental communication time (in seconds) for given ratios of $P_r : R_r : 1$. The value of N is 5000.

5 Conclusions

On three fully connected heterogeneous processors, the optimal data partition shape depends on the relative computational power of each processor and the ratio between computational power and communication speed and is one of just three well-defined shapes. In general, the Square Corner shape is optimal for systems with a single fast processor, and two slower processors, the Square Rectangle shape is optimal for systems with two fast processors and a less powerful processor, and the Block Rectangle shape is optimal for relatively homogeneous systems and systems with a faster, medium and slower processor.

These results show that the optimal data partition is not exclusively rectangular. Of the three optimal shapes, two are non-rectangular. One of these, the Square Rectangle, has never before been considered. Without the Push technique, this non-symmetrical and unconventional shape would not be known to be the optimum.

Acknowledgement. Experiments presented in this paper were carried out using the Grid'5000 experimental testbed, being developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see https://www.grid5000.fr). This research was conducted with the financial support of Science Foundation Ireland under Grant Number 08/IN./I2054.

References

1. Dongarra, J.J., Meuer, H.W., Simon, H.D., Strohmaier, E.: Top500 supercomputer sites, `http://www.top500.org/`
2. Clarke, D., Lastovetsky, A., Rychkov, V.: Column-based matrix partitioning for parallel matrix multiplication on heterogeneous processors based on functional performance models. In: Alexander, M., et al. (eds.) Euro-Par 2011, Part I. LNCS, vol. 7155, pp. 450–459. Springer, Heidelberg (2012)

3. Dovolnov, E., Kalinov, A., Kilmov, S.: Natural bloc data decomposition for heterogeneous clusters. In: Proceedings of the 17th International Parallel and Distributed Processing Symposium, IPDPS 2003 (April 2003)
4. Kalinov, A., Lastovetsky, A.: Heterogeneous distribution of computations solving linear algebra problems on networks of heterogeneous computers. Journal of Parallel and Distributed Computing 61, 520–535 (2001)
5. Beaumont, O., Boudet, V., Rastello, F., Robert, Y.: Partitioning a square into rectangles: NP-completeness and approximation algorithms. Algorithmica 34, 217–239 (2002)
6. DeFlumere, A., Lastovetsky, A., Becker, B.A.: Partitioning for parallel matrix-matrix multiplication with heterogeneous processors: The optimal solution. In: Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 125–139 (2012)
7. DeFlumere, A., Lastovetsky, A.: Searching for the optimal data partitioning shape for parallel matrix matrix multiplication on 3 heterogeneous processors. In: Parallel and Distributed Processing Symposium Workshops, IPDPSW (2014)
8. Zhong, Z., Rychkov, V., Lastovetsky, A.: Data partitioning on heterogeneous multicore and multi-GPU systems using functional performance models of data-parallel applications. In: IEEE International Conference on Cluster Computing (CLUSTER), pp. 191–199. IEEE (2012)
9. Hockney, R.: The communication challenge for mpp: Intel paragon and meiko cs-2. Parallel Computing 20(3), 389–398 (1994)
10. Van De Geijn, R., Watts, J.: SUMMA: Scalable universal matrix multiplication algorithm. Concurrency-Practice and Experience 9(4), 255–274 (1997)
11. DeFlumere, A., Lastovetsky, A.: Theoretical results on optimal partitioning for matrix matrix multiplication on three fully connected heterogeneous processors. School of Computer Science and Informatics, University College Dublin, Tech. Rep. UCD-CSI-2014-01 (2014)