Max-Plus Algebra and Discrete Event Simulation on Parallel Hierarchical Heterogeneous Platforms

Brett A. Becker and Alexey Lastovetsky,

School of Computer Science and Informatics, University College Dublin,
Belfield, Dublin 4, Ireland
{brett.becker, alexey.lastovetsky}@ucd.ie

Abstract. In this paper we explore computing max-plus algebra operations and discrete event simulations on parallel hierarchal heterogeneous platforms. When performing such tasks on heterogeneous platforms parameters such as the total volume of communication and the top-level data partitioning strategy must be carefully taken into account. Choice of the partitioning strategy is shown to greatly affect the overall performance of these applications due to different volumes of inter-partition communication that various strategies impart on these operations. One partitioning strategy in particular is shown to reduce the execution times of these operations more than other, more traditional strategies. The main goal of this paper is to present benefits waiting to be exploited by the use of max-plus algebra operations on these platforms and thus speeding up more complex and quite common computational topic areas such as discrete event simulation.

Keywords: Data Partitioning, Heterogeneous Computing, Parallel Computing, Tropical Algebra, Max-Plus algebra, Discrete Event Simulation, Hierarchal Algorithms, Square-Corner Partitioning

1 Introduction

Max-plus algebra is a relatively new field of mathematics which grew from the advent of tropical geometry in the early 1980s and has since been shown to have many diverse application areas. MPA is (along with min-plus algebra) a sub-category of tropical algebra. MPA obeys most laws of basic algebra with the operations of addition $(a + b)$ and multiplication $(c \times d)$ replaced by the operations $max(a, b)$ and addition $(c + d)$ respectively. Min-plus algebra is similar, but with the maximum operation replaced with a minimum function.

Discrete event simulation is an extremely expansive area of continuing and intense research which may broadly be characterised as a collection of techniques and methods which when applied to the study of discrete-event dynamical systems generate sequences which characterize system behaviour. This includes modelling concepts for abstracting essential features of a system into a set of precedence and mathematical relationships, which can be used to describe the system and more importantly for system design, to predict behaviour, performance, and drawbacks/bottlenecks. DES is used to design and model a vast number of systems

including travel timetables, operating systems, communication networks, autonomous guided vehicles, operating systems, CPUs and other complex systems. There are many approaches to designing DES including Petri nets, alphabet based approaches, perturbation methods, control theoretic techniques and expert systems design. Recently MPA and other techniques involving both logical and algebraic components have shown to be capable of simplifying simulations while maintaining the desired outputs [11]. One such method is explored later in this paper.

The square-corner partitioning (SCP) is a top-level partitioning method for parallel hierarchal heterogeneous computing which when applied to problems such as matrix-matrix multiplication (MMM) and all linear algebra kernels reducible to MMM, optimally reduces the total volume of communication (TVC) between computing entities (processors, clusters, etc.) when the power ratios between entities meet certain, yet numerous and very common ratios. This partitioning also has other benefits including simpler communication schedules and the possibility of overlapping communication and computation [2], [3]. As this paper demonstrates the SCP can extend these benefits to many application areas.

The rest of this paper is outlined as follows: In Section 2 we review and formally define the MPA, and introduce a specific approach for solving DES problems. We then outline the SCP and its application to these operations on heterogeneous parallel platforms. Section 3 presents results of MPI experiments applying the SCP to MPA operations and a DES example which uses a mixed algebraic/logical approach. Section 4 presents our conclusions and future work.

2 Background and Related Work

2.1 Max-Plus Algebra

Max-plus algebra is a relatively new field in mathematics, dating back approximately 30 years. It has since been shown to have several application areas such as discrete event simulation, dynamic programming, finite dimensional linear algebra, modelling communication networks, operating systems, combinatorial optimization, solving systems of linear equations, biological sequence comparisons and even problems such as crop rotation [4], [8], [9], [11], [13]. In many scientific and computational applications the structure of MPA matrix multiplication is an important aspect. Additionally, higher powers of MPA matrices are of significant interest and necessary in many application areas [5], [11].

MPA is based on replacing the "normal" algebraic addition operation with a binary max function, and the "normal" multiplication operation with addition. Formally, if we define $\varepsilon \overset{\text{def}}{=} -\infty$ and $e \overset{\text{def}}{=} 0$ then denote $\mathbb{R}_{max}$ to be the set $\mathbb{R} \cup \{\varepsilon\}$ then for elements $a, b \in \mathbb{R}_{max}$,the operations $\oplus$ and $\otimes$ are defined respectively by the following.

$$a \oplus b \overset{def}{=} max(a, b) \text{ and } a \otimes b \overset{def}{=} a + b \quad (1)$$

Therefore, $a \oplus \varepsilon = max(\varepsilon, a) = a$ and $a \otimes \varepsilon = \varepsilon + a = \varepsilon$. We can now formally define max-plus algebra as $\mathfrak{R}_{max} = (\mathbb{R}_{max}, \oplus, \otimes, \varepsilon, e)$. Finally, the $\otimes$ operation has priority over the $\oplus$ operation.

MPA matrices are denoted $\mathbb{R}_{max}^{n \times m}$, where n and m are the matrix dimensions. For the MPA matrices $A \in \mathbb{R}_{max}^{n \times m}$ and, $B \in \mathbb{R}_{max}^{m \times q}$ the matrix product $A \otimes B$ is the same as in normal linear algebra, but following the operation substitutions in (1). From this, matrix powers are straight-forward, and represented $A^{\otimes k}$ for the k^{th} power of A. As max-plus matrix multiplication and max-plus matrix powers are integral parts of many applications of MPA we further discuss this in Section 3.1.

2.2 Discrete Event Simulation

Discrete event simulation is a very broad and well-studied field and therefore the purpose of this Section is to acquaint the reader with the specific technique utilized in this paper. Briefly, DES is a collection of techniques and methods which when applied to the study of a discrete-event dynamical system generates sequences which characterize the system behaviour. This includes modelling concepts for abstracting essential features of the system into a set of precedence and mathematical relationships, which can be used to describe the system and more importantly for design, and to predict its behaviour, performance, and drawbacks/bottlenecks. For more see any good DES text such as [7].

As most DES algorithms are computationally intensive, efforts to parallelize them are numerous. The complexity of most practical DES algorithms however poses numerous obstacles in effective and efficient parallelization. Amongst these are synchronization and timing inconsistencies, synchronous vs. asynchronous simulation, deadlock avoidance and detection, conservative vs. optimistic simulation, recovery strategies, and memory management to name a few [6].

In Section 3.2 we present results of the parallelization of a DES modelling technique which although as presented in [13] is sequential, lends itself to parallelization due to a computationally intensive algorithmic core which can be efficiently ported to hierarchal heterogeneous parallel platforms. This core is very similar to a max-plus matrix operation but using logical and/or operations instead of max-plus operations. We employ this technique – called the Matrix Discrete Event Model (MDEM) – using MPI and utilizing the SCP [2], [3], for the core routine.

The Matrix Discrete Event Model

The authors of [13] note that the design, simulation, and analysis of large-scale, complex systems using existing DES techniques such as Petri nets, alphabet-based approaches, perturbation methods, control theoretic techniques, and expert systems design are often difficult to implement and are very labour and time intensive. The MDEM is a hybrid system with logical and algebraic components that seeks to make these processes more efficient. Although the examples in [13] focus on manufacturing systems, the formulation is also applicable to many DES situations such as travel timetables, operating systems, communication networks, autonomous guided vehicles, operating systems, and many others. Clearly the number of degrees of freedom, state

possibilities, and general complexity of such systems often result in simulations with several thousands (or more) event components.

The MDEM approach is a rule-based model described by four equations: the model state equation, start equation, resource release equation, and the product output equation. Each of these equations are *logical*, only using *or, and,* and *negation* operations. Additionally, all vectors and matrices in these equations are binary – only composed of 0's and 1's. For instance, the vector which is the output of the start equation contains a '1' for each job which is to be started at the given state of the simulation, and a '0' otherwise.

The simulation itself is carried out by first calculating initial conditions from the description of the system. The core of the simulation is carried out by the successive calculation of 'firing vectors' which carry the simulation to the next state. This amounts to the repeated calculation of an equation which has the form of a matrix-matrix multiplication except that since the approach of the MDEM technique is hybrid – having both algebraic and logical components – the algebraic multiplication and addition operations are replaced with logical 'or' and 'and' operations respectively. It is this step that constitutes the bulk of the calculation time for the MDEM technique as all other calculations only need to be carried out once.

2.3 The Square-Corner Partitioning

The square-corner partitioning is a partitioning method for parallel hierarchal heterogeneous computing which when applied to problems such as matrix-matrix multiplication and all linear algebra kernels reducible to MMM reduces the total volume of communication (TVC) between clusters optimally when the power ratios between clusters is greater than 3:1.[1] This partitioning also has other benefits such as simplified communication schedules and the possibility of overlapping communication and computation. A defining feature of the SCP is that it removes the restriction that all partitions be rectangular, which at first may seem unintuitive [12].

An existing state-of-the-art heterogeneous partitioning scheme (referred to here as the straight line partitioning or SLP) which does carry such a restriction is introduced in [1] which presents a column based partitioning based on that of [10]. The SLP balances the workload between processors of different speeds in an attempt to minimize the TVC between processors. First the matrix is partitioned into rectangles proportional in area to the speed of each processor. These rectangles are then arranged into columns in a defined manner. The TVC is proportional to the sum of the half-perimeters s of each rectangle, given by (2), where p is the number of processors and h_i and w_i are the height and width of the rectangle assigned to processor i, respectively.

$$s = \sum_{i=1}^{p}(h_i + w_i) \quad (2)$$

Since the perimeter of any rectangle enclosing a given area is minimized when that rectangle is a square, there is a natural lower bound l of (2), shown by (3), where a_i is the area of the partition belonging to processor i.

[1] In this Section the words processor and cluster are used more or less interchangeably as some papers simulate individual clusters with processors for simplicity of modelling/verification purposes.

$$l = 2 \times \sum_{i=1}^{p} \sqrt{a_i} \quad \textbf{(3)}$$

In considering the case of two clusters, we can inspect the case with relative speeds such that cluster 1 receives a rectangle of area $a_1 = 1 - \varepsilon$, and cluster 2 receives a rectangle of area $a_2 = \varepsilon$, where $\varepsilon > 0$ is an arbitrarily small number. In order to partition the unit matrix into two rectangles using the straight line partitioning, a line of length 1 must divide the matrix. Using (2) this results in a sum of half-perimeters equal to 3, regardless of the value of ε, but (3) shows that the lower bound can get arbitrarily close to 2, (as $\varepsilon \to 0$).

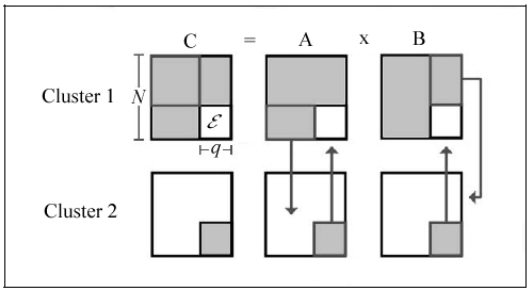

Fig. 2.1. The square-corner partitioning (for two partitions) and the necessary communication steps. Shaded areas belong to the respective clusters. Clearly if $\varepsilon = 0$, no communication is necessary at all.

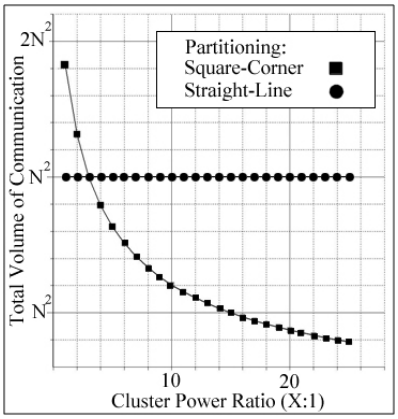

Fig. 2.2. Comparison of the total volume of communication between two clusters for the square-corner and straight-line partitionings.

A glance at Figure 2.1 illustrates that for the SCP (unlike the SLP), as $\varepsilon \to 0$, the sum of half-perimeters – and therefore the TVC – approaches 2, showing that the SCP is optimal. A more detailed discussion and proof are given in [2].

Figure 2.2 shows the TVC of the SCP compared to that of the SLP. It is clear that when the power ratio between clusters is 3:1, the TVC values are equal, and for ratios

above 3:1 the SCP TVC is less. By the time the ratio reaches 15:1, the SCP TVC is exactly half that of the SLP.

3 MPI Experiments

3.1 Max-Plus MMM Using the Square-Corner Partitioning

As outlined in Section 2.1 we experimented with performing a MPA MMM using C and MPI. We used a two cluster heterogeneous platform with power ratios between clusters ranging from 1:1 to 6:1. For all experiments we use double precision and N = 7,000. Local computations utilized BLAS. The local interconnect was 2Gb/s Infiniband and the inter-cluster interconnect was 1Gb/s Ethernet. Figure 3.1 shows the communication times for both the SCP and SLP partitionings. Firstly, it can be seen that as expected the SCP does not show improvement in communication time until the power ratio is 3:1, as this is when the SCP results in a lower TVC as shown in [2]. After this (as the system becomes more heterogeneous), the gap between the two communication times widens, and would be expected to widen.

Figure 3.1 shows the resulting difference in execution times between the SCP and SLP. As expected we also see the crossover around ratio 3:1, and note that the lower TVC that the SCP brings also results in lower execution times for ratios above 3:1. Again this gap would be expected to widen.

It is worth noting that since carrying out a matrix power operation A^n amounts to nothing more than n repeated matrix multiplications, carrying out matrix power operations would also benefit from the above.

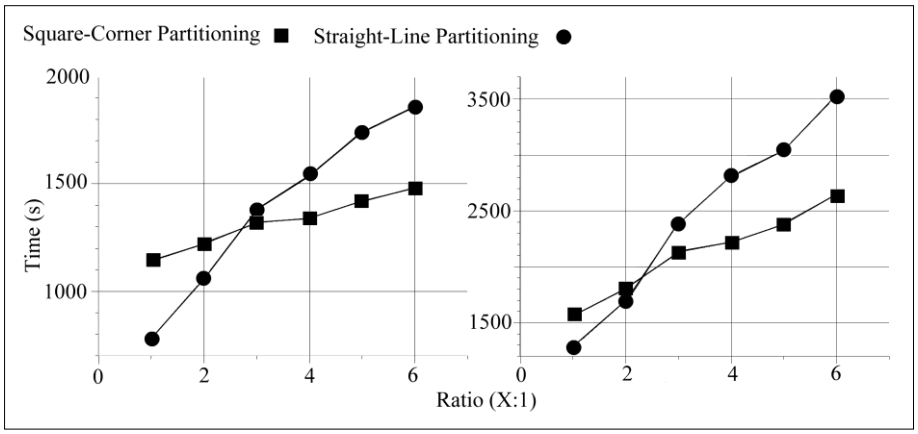

Fig. 3.1. Communication times (left) and execution times (right), Max-Plus MMM, N = 7000.

3.2 The Square-Corner Partitioning for Discrete Event Simulation

In Section 2.2 we outlined the MDEM model for discrete event simulations. We use the same experimental platform as in Section 3.1 to demonstrate results on a parallel, heterogeneous platform of the MDEM model. We utilize both the SLP and the SCP for the core routine which is a matrix "and/or" multiplication. We generate the initial conditions so that the core routine involves a large system (N = 5000). All initial calculations and cleanup are carried out on a single processor as these calculations are carried out only once and make up a very small percentage of the overall execution time.

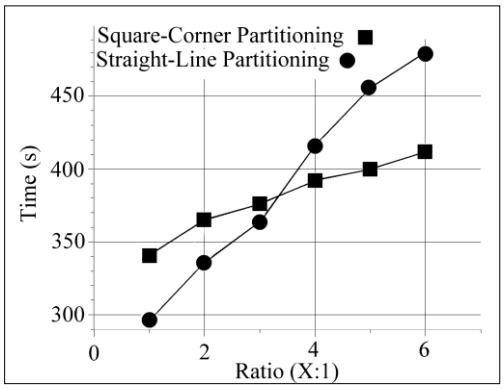

Fig. 3.2. Total execution times, MDEM DES model, N = 5000.

Figure 3.2 shows the execution times for the MDEM DES using both partitioning techniques. All times are averaged over five runs. It is seen that the use of the SCP for the core kernel of the MDEM DES algorithm significantly reduces the execution time for ratios above 3:1. Again the expected crossover occurs near the ratio of 3:1. The overall shapes of the curves are similar to those of Section 3.1 as the "and/or" MMM in the MDEM involves a similar computational cost as the max-plus MMM.

4 Conclusion and Future Work

In this paper we explored computing max-plus algebra matrix operations and a MDEM discrete event simulation on parallel hierarchal heterogeneous platforms. We found that the initial top-level data partitioning – particularly the use of the square-corner partitioning - significantly affects overall execution time due to the total volume of inter-cluster communication involved. Notably the square-corner partitioning outperformed the straight-line partitioning in all cases. Future work involves applying similar strategies to speed up more complex routines on parallel hierarchal heterogeneous platforms and experimenting on more complex networks.

Acknowledgments

This work was partially funded by the University College Dublin School of Computer Science and Informatics (see http://csiweb.ucd.ie).

This work was supported by Science Foundation Ireland

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).

The authors would like to thank Dr. Mark Dukes of the University of Iceland for useful suggestions.

References

1. Beaumont, O., et. al..: Partitioning a Square into Rectangles: NP-Completeness and Approximation Algorithms. Algorithmica. vol.34, no.3, pp.217-239 (2002)
2. Becker, B.A., Lastovetsky, A.: Data Partitioning For Matrix Multiplication on Two Interconnected Processors: Proceedings of the 8th IEEE International Conference on Cluster Computing (Cluster 2006): IEEE Computer Society, New York (2006)
3. Becker, B.A., Lastovetsky, A.: Towards Data Partitioning for Parallel Computing on Three Interconnected Clusters: Proceedings of the 6th International Symposium on Parallel and Distributed Computing (ISPDC 2007): IEEE Computer Society, New York (2007)
4. Comet, J.P.: Application of max-plus algebra to Biological Sequence Comparisons. Theoretical Computer Science 293, 189-217 (2003)
5. De Schutter, B., and De Moor, B.: On the Sequence of Consecutive Matrix Powers of Boolean Matrices in the max-plus algebra in Theory and Practice of Control and Systems. In: Tornamb, A., Conte, G., Perdon, A.M.: World Scientific, pp. 672-677, Singapore (1999)
6. Fersha, A.,: Parallel and Distributed Simulation of Discrete Event Systems. In: Handbook of Parallel and Distributed Computing, McGraw Hill (1995)
7. Fishman, G. S.: Discrete-Event Simulation: Modeling, Programming, and Analysis. Springer-Verlag, New York (2001)
8. Gaubert, S. and Plus, M.: Methods and applications of (max,+) linear algebra. In: R. Reischuk and M. Morvan, (eds.). STACS2007.LNCS, vol. 3088. Springer. Lübeck (2007)
9. Heidergott, B., Jan Olsder, G., van der Woude, J: Max Plus at Work. Princeton University Press, Princeton (2006)
10. Kalinov, A., Lastovetsky, A.: Heterogeneous Distribution of Computations While Solving linear algebra Problems on Networks of Heterogeneous Computers: Proceedings of the 7th International Conference on High Performance Computing and Networking Europe (HPCN`99) (1999)
11. Kirov, M. V.: The Transfer-Matrix and max-plus algebra Method for Global Combinatorial Optimization: Application to Cyclic and Polyhedral Water Clusters. Physica A. 388, 1432-445 (2009)
12. Lastovetsky, L. and Dongarra, J.: High Performance Heterogeneous Computing. Wiley-Blackwell, Hoboken (2009)
13. Tacconi, D., Lewis, F.L.: A New Matrix Model for Discrete Event Systems. Application to Simulation: IEEE Control Systems 97, 6-71 (1997)