Heterogeneity Considered Harmful to Algorithm Designers

O. Beaumont* V. Boudet* A. Legrand* F. Rastello* Y. Robert*

Abstract

In this paper, we deal with algorithmic issues on heterogeneous platforms. We show that static scheduling and load-balancing strategies are absolutely needed to achieve good performances, in contrast to situation for homogeneous parallel machines where dynamic schemes often turn out to be very satisfactory. However, we also show that static strategies targeted to heterogeneous platforms are difficult to design and implement: intuitively, data distribution must obey a much more refined model than standard block-cyclic distributions to equally balance the load between processors of different speeds. Technically, we state several NP-completeness results that demonstrate the intrinsic difficulty of static load-balancing on heterogeneous platforms.

1 Matrix Product on 2D Homogeneous Grids

We shortly describe the parallel matrix multiplication algorithm on a 2D homogeneous grid which is used in Scalapack [4]. The A, B and C matrices are identically partitioned into $p \times q$ rectangles. There is a one-to-one mapping between these rectangles and the processors. Each processor is responsible for updating its C rectangle: more precisely, at step i, the i-th column of A is horizontally broadcasted, the i-th row of B is vertically broadcasted, and each processor updates its C rectangle with the product of (fragments of) these column and row as soon as received. As depicted in Figure 1, the *total* volume of data exchanged is proportional to the *sum* of the half perimeters of the $p \times q$ rectangles if the underlying communication network does not allow to perform communications in parallel. The work is perfectly balanced: at each step, each processor processes the same amount of data at the same speed as the others.

*LIP, UMR CNRS–ENS Lyon–INRIA 5668, Ecole Normale Superieure de Lyon, 46, Allée d'Italie, 69364 Lyon Cedex 07, France. E-mail: Firstname.Lastname@ens-lyon.fr

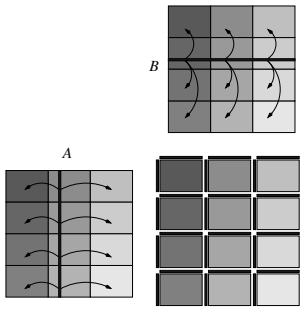

Figure 1. 4×3 homogeneous grid

2 Matrix Product on Heterogeneous Platforms

How to modify the previous algorithm for a heterogeneous platform? The idea is to keep the same framework: at each step, one pivot column and one pivot row are communicated to all processors, and independent updates take place. However, with different-speed processors, we cannot distribute same size rectangles from the C matrix to the processors. Intuitively, we want to balance the computing load so that each processor receives an amount of work in accordance to its computing power. Because all C blocks require the same amount of arithmetic operations, each processor executes an amount of work which is proportional to the number of blocks that are allocated to it, hence proportional to the area of its rectangle. Thus, parallelizing the matrix-matrix product $C = AB$ on a heterogeneous platforms turns out to be equivalent to partition the unit square into rectangles of prescribed area while minimizing a cost function based on half-perimeters of these rectangles.

3 Rectangle Partitioning

Different partition types are summarized in Figure 2. We present complexity results on problems equivalent to matrix distribution on heterogeneous platforms with network like Ethernet ($\sum$) or Myrinet ($\max$) (see Table 1). It is well-known that 1D partitions are not scalable since the volume of communication grows proportionally to the number of machines. Adapting this kind of distribution to heterogeneous platforms is fairly easy but still leads to non-scalable distributions. Nevertheless designing heterogeneous 2D distributions turns out to be more

Proceedings of the IEEE International Conference on Cluster Computing (CLUSTER'00)
0-7695-0896-0/00 $ 17.00 © 2000 IEEE

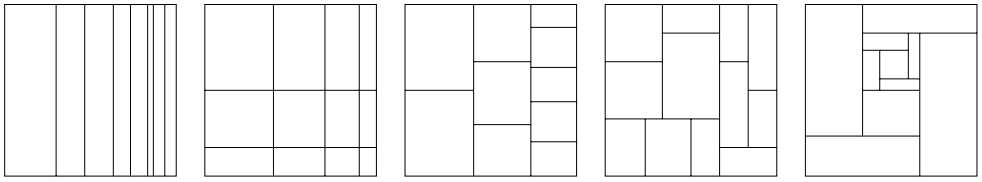

Figure 2. Partitioning the unit square: a taxonomy

	1D	2D	Column-based	Recursive	General
$\sum$	Polynomial	**NP-hard** [1]	Polynomial [2]	No known results so far	NP-hard. Guaranteed heuristic ($5/4$ from the optimum). [2]
max			**NP-hard** ([1]). **Guaranteed heuristic ($2/\sqrt{3}$ from the optimum)**.	No known results so far	NP-hard. Guaranteed heuristic ($2/\sqrt{3}$ from the optimum). [3]

Table 1. Various complexity results for rectangle partitionning

difficult:

Heterogeneous 2D Grid : Given p^2 real positive numbers $s_1, \ldots, s_p$ s.t. $\sum_{i=1}^{p^2} s_i = 1$ is there a heterogeneous grid partition of the unit square into p^2 rectangles R_i of size $r_i \times v_i$ s.t. $\sum_{i=1}^{p} r_i = \sum_{j=1}^{p} c_j = 1$, and a one-to-one mapping f from $[\![1,p]\!] \times [\![1,p]\!]$ to $[\![1,p^2]\!]$ minimizing $\max_{(i,j)\in[\![1,p]\!]\times[\![1,p]\!]} \left(\frac{r_i c_j}{s_{f(i,j)}}\right)$. This optimisation problem is NP-hard (see [1] for more details).

The communication pattern is simpler for 2D grids than for more general processor arrangements, but they do not allow for a perfectly balanced workload in general. Column-based partitions (with a different number of processors in each column) are always perfectly balanced, often close to the optimum solution and generally lead to a rather simple communication pattern.

Distributing a matrix columnwise on a heterogeneous platform made of different processors linked by a homogeneous network (like for example Myrinet) turns out to be equivalent to solving the following problem, where the s_i are the processor speeds:

Col-Peri-Max : Given p real positive numbers $s_1, \ldots, s_p$ s.t. $\sum_{i=1}^{p} s_i = 1$, find a column-based partition of the unit square into p rectangles R_i of area s_i and of size $h_i \times v_i$, so that $\max_{i=1}^{p}(h_i + v_i)$ is minimized. This optimisation problem is NP-hard, but we have provided a guaranteed heuristic (see [1] for more details).

4 Conclusion

We have stated different complexity results on the static load-balancing problem on heterogeneous platforms. The practical applicability of our theoretical results has been demonstrated through a series of experiments on one HNOW with Fast Ethernet and on another one with Myrinet/Bip as communication network. Our current work is focused on the design and the implementation of efficient on-the-fly redistribution to cope with potential variations of the machin loads.

References

[1] O. Beaumont, V. Boudet, A. Legrand, F. Rastello, and Y. Robert. Heterogeneity considered harmful to algorithm designers. Technical Report RR-2000-24, LIP, ENS Lyon, June 2000. Available at `www.ens-lyon.fr/LIP/`.

[2] O. Beaumont, V. Boudet, F. Rastello, and Y. Robert. Matrix-matrix multiplication on heterogeneous platforms. Technical Report RR-2000-02, LIP, ENS Lyon, Jan. 2000. Short version appears in the proceedings of ICPP'2000.

[3] O. Beaumont, V. Boudet, F. Rastello, and Y. Robert. Partitioning a square into rectangles: Np-completeness and approximation algorithms. Technical Report RR-2000-10, LIP, ENS Lyon, Feb. 2000.

[4] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley. *ScaLAPACK Users' Guide*. SIAM, 1997.