Parallelism in Computational Algorithms and the Physical World

Björn Engquist

Department of Mathematics, UCLA Los Angeles and NADA, KTH Stockholm

1 Introduction

Most processes in the real world are local and contain a high degree of parallelism. A simple example is weather prediction. The weather at any location depends on the weather at earlier times in the neighborhood. Causality is, however, another important physical principle preventing parallelism. The weather at one instance must be known before the weather at later times can be computed. For problems of the size of weather prediction the required recursiveness in the algorithm does not, in practice, prohibit a high degree of parallelism. The parallel computation at each time instance saturates the computer.

The computational algorithms should capture the parallelism in these processes and map them efficiently onto the current architecture. Quite often the original physical process is approximated in such a way that the local dependence is lost. This happens, for example, when steady state is assumed. Furthermore, many modern computational methods are hierarchical and contain some global interconnection even if the underlying process is local. The overall efficiency depends on how well this connectivity is supported by the architecture. Different classes of modern methods in scientific computing and their parallel implementation will be discussed.

2 Explicit Time Dependent Algorithms

In this type of algorithm the unknown quantities are first given as initial values in a real array. Time is discretized and all these unknowns are updated at each discrete time level. In the update the new values in the array are computed as functions of a finite, and usually small, number of real values from the old array. This is the typical form of a finite difference or lumped finite element method approximating a partial differential equation, [4]. The partial differential equation is, by far, the most common mathematical model describing physical processes in many areas as, for example, weather prediction, electro-magnetic and elastic wave propagation, combustion and other chemical processes.

The update mentioned above is given by a fixed function of the old values in the array. These values are all physically close in the one, two or three dimensional space for the independent spatial variables of the simulation. This results directly from differentiation being a local operator.

A natural technique in modern coarse grain MIMD computers, as the IBM SP-1 and Cray T3D, is to decompose the the array following a decomposition

of physical space. In this domain decomposition, [3], where each domain corresponds to one processor and its local memory, most operations are internal to the nodes. The amount of data in the communication between the domains is much smaller than the number of floating point operations. This corresponds to the boundary of the domains being of one dimension lower than the interior. The parallelization is thus quite trivial in theory and the optimization amounts to load balancing between the sizes of the domains and the grouping of data in space and time for the message passing depending on bandwidth and latency.

For earlier massively parallel computers with smaller local memory and slower communication it was much more important for the interconnection architecture to mimic the connectivity in the approximation of the partial differential algorithm and thus in physical space. Examples are the nearest neighbor mesh in the early ILLIAC IV and the ICL DAP. The hypercubes and the CM200 are also closely related to the nearest neighbor mesh architecture. See [1] for an informative discussion along these lines.

3 Nonlocal Algorithms

The nonlocal algorithms are a challenge also for the modern coarse grain parallel architectures with fast communication. Each value in the update of the array of unknowns is now a function of a potentially larger number of values scattered over the physical space and thus over the array. This situation occurs for implicit computational methods approximating time dependent problems. In each time step a system of algebraic equations, linking all unknown values must be solved. It also occurs for processes which are approximated to be steady or instantaneous in time. The elliptic equations in structural mechanics and the divergence free condition in incompressible flow are examples of these approximations.

The algebraic equations mentioned above are either directly linear or are reduced to linear systems by some nonlinear equation solver. If standard direct solvers are used the parallelization becomes a problem in the linear algebra of sparse systems. For large problems iterative techniques are becoming more important. Modern hierarchical iterative methods, as for example multigrid and domain decomposition, can be seen as compromises between the explicit local algorithms and the direct more interconnected techniques.

The efficiency of these hierarchical iterative methods are based on the fact that there are some local nature even in the operators describing steady state or instantaneous processes. In domain decomposition a sequence of localized problems will give an approximation to the global problem. In multigrid a sequence of explicit methods with grids of differnt coarseness are utilized, [1], [3]. The overall efficiency is now a more delicate balance between the algorithmic efficiency on a scalar basis and the scalability of the parallel distribution of data and computations. In [2] a multigrid method to be used on coarse grain MIMD computers is described. The paper also contains convergence analysis.

References

1. Chan, T.: The Physics of Parallel Machines, Opportunities and Constraints of Parallel Computing. Sanz, J. L. C. ed., Springer Verlag, pp. 15–20, 1989.
2. Eliasson, P., Engquist, B.: Multigrid Methods for Hyperbolic Differential Equations, to appear.
3. Gropp, W. D.: Parallel Computing and Domain Decomposition, *Domain Decomposition Methods*, Chan, T., Keyes, D. E., Meurant, G., Scroggs, J. S., Voigt, R. G. eds, SIAM, 1992.
4. Rizzi, A., Engquist B.: Selected Topics in the Theory and Practice of Computational Fluid Dynamics. J. Comp. Phys. **72** (1987) 1–69