ELSEVIER

Future Generation Computer Systems 12 (1996) 263–264

FGCS
FUTURE GENERATION COMPUTER SYSTEMS

Guest editorial
Third Utrecht Computational Science Symposium

Rob Bisseling

Department of Mathematics, Utrecht University, Budapestlaan 6, PO Box 80 010, 3508 TA Utrecht, Netherlands

On 18 November 1994, the Utrecht Computational Science Centre organised the symposium *Parallel Computing Applications: A Path Towards the Future.* Central theme was the evolutionary path from the current situation of rather limited use of parallel computers in real-world applications to an envisioned future where parallel computing is a widespread and natural activity. The two main questions posed to the speakers of the symposium were: Which parallel programming tools are needed for applications? and which methodologies and practical tools are available?

Three speakers expanded their answers to these questions and wrote an article on the basis of their talk. The resulting papers represent the three corners of what may be termed the parallel computing triangle–theory, applications, and tools. *Bill McColl*, director of Oxford Parallel at Oxford University, shows the benefits that can be derived from a theoretical model for parallel programming. *Martyn Guest*, head of the High Performance Computational Chemistry group at Pacific Northwest National Laboratory, and his coworkers demonstrate how they tackle a large-scale computational chemistry application. *Karsten Decker*, head of the Research and Development section at the Swiss Scientific Computing Centre, and his coworkers discuss generic tools for the parallel programmer.

The paper *Scalability, portability and predictability: The BSP approach to parallel programming* by McColl shows us how a simple model such as the bulk synchronous parallel (BSP) model helps hardware designers and software developers in focusing their efforts. McColl discerns an evolutionary trend of parallel architectures converging towards a common standard and he argues that parallel computing may become the standard method for computing. He illustrates the use of the BSP model by matrix multiplication and fast Fourier transform. Finally, he compares BSP implementations with PVM and MPI.

The paper *High-performance computing in Chemistry: NWChem* by Guest et al. describes the NWChem package for solving computational chemistry problems on parallel computers. To achieve portability and efficiency, the package is built on top of the global arrays (GA) toolkit. It is interesting to note that starting from a desired application program a generic tool was developed that can be used in other applications as well. The GA software is based on data distribution, instead of replication, and easy access to remotely stored data through one-sided communications, i.e. *put* and *get* operations. It is taken into account that local data access is faster than remote data access, so that the user is allowed to influence the data distribution. These features are common to both the GA toolkit and to current BSP implementations. Using GA leads to natural interleaving of computation and communication; in BSP terms such phases

PII: S0167-739X(95)00012-X

would be called supersteps. The performance of the NWChem package is tested by using density functional theory to solve a zeolite problem on four different parallel computers with up to 128 processors.

The paper *Matching user requirements in parallel programming* by Decker et al. discusses possible objectives in building a user-friendly parallel programming environment. Such an environment would lead to more wide-spread use of parallel computers in computational science. Areas that may contribute to building this environment are functional languages, artificial intelligence, and compiler technology. Two prototype projects are described, both targeted at distributed-memory parallel computers. The first project builds the high-level tool PDE (program development environment), which is intended to ease the entry of new users in the field of parallel computing. It encapsulates algorithmic skeletons such as fast Fourier transforms, which can be refined or adapted by the user. The second project constructs the low-level tool Annai, which is primarily aimed at expert users. It accepts input in various forms; for example, it extends high performance Fortran by allowing the use of irregular data distributions.

As a guest editor, I would like to thank the authors for contributing their insights in the path to follow towards massive application of parallel computing. I would also like to thank the referees for their careful reading of the manuscripts.

Rob Bisseling received his B.Sc. cum laude in 1977 and his M.Sc. cum laude in 1981, both in Mathematics, from the Catholic University of Nijmegen, the Netherlands. In 1987, he received a Ph.D degree in theoretical chemistry from the Hebrew University of Jerusalem, Israel. From 1987 to 1993, he worked as a research mathematician at the Mathematics and Computer Science Department of the Koninklijke/Shell-Laboratorium, Amsterdam. Since 1993, he is an Associate Professor in Computational Science at the Mathematics Department of Utrecht University, the Netherlands. His main fields of interest are sparse matrix computations, parallel algorithms for scientific computation, and applications of the bulk synchronous parallel model.