Programming Environments for Massively Parallel Distributed Systems

Working Conference of the **IFIP WG** 10.3, April 25-29, 1994

edited by Karsten M. Decker and René M. Rehmann

BIRKHÄUSER VERLAG AG, Basel, 1994, 420 p.

ISBN 3-7643-5090-3

Luminita Todor was born in Romania in 1963. She graduated in Technological Physics from the University of Bucharest in 1986, her major being in "Measurements, Testing and Control". Since 1989 she has worked with Information Systems for Quality Assurance Laboratory, the Research Institute for Informatics, Bucharest. She is preparing a Ph D thesis on Numerical Techniques and Parallelism in Simulation in Physics.

On editing the Proceedings, Dr. K.M.Decker and Dr. R.M.Rehmann from the Swiss Scientific Computing Center Zurich, Switzerland, pledged to giving the most profitable topical arrangement of the 42 papers presented at IFIP WG 10.3, and succeeded in putting into bold relief the state-of-the-art in massively parallel systems.

The strategic importance of Massively Parallel Systems for the advancement of high performant computing is well-recognised in the national and international information technology programs in Europe, USA and Japan. The programming of massively parallel systems is a complex task, and so, alleviating this software problem is sometimes referred to as the big challenge of the 1990's.

The papers gave evidence on R&D carried out for pushing the frontier of massively parallel systems application in industry. For the time being, such insolubilities should be admitted:

- an insufficient level of abstraction on programming massively parallel systems, and particularly the lack of high-level programming methods;
- the syntax required in the programming process is still too far convoluted to be easily mastered by application users;
- the portability issue needs more attention.

Subjects do not recur from one paper to another, and so, no headings introduce them. A few words will be said on each chapter, so that readers be spared browsing, and go directly to the subject they are interested in.

In **"The Cray Research MPP FORTRAN programming Model"**, T. MacDonald and Z. Sekera, from Cray Research, Switzerland, present the features of a FORTRAN based programming model that allows programmers to specify both explicit and implicit communication in a parallel program.

"Resource Optimization via Structured Parallel Programming", by B. Bacci, M.Danelutto and S. Pelagatti, from the Department of Computer Science , University of Pisa, Italy, try to convince that, once adopted a structured style of programming (the P3L language) and a set of template-based compiling tools, most of the burden in writing massively parallel applications will be relieved at the expense of the compiler design phase.

V.A.Serebiakov, A.N.Bezdushny and C.G.Belov , from the Computing Center of the Russian Academy of Sciences, propose and present a performant language **"SYNAPS/3 - An Extension of C for Scientific Computations"**.

In the paper **"The Pyramid Programming System"**, Z. Lin, S. Zhou and W. Li , USA, discuss a parallel programming system that supports a parallel programming methodology decoupling parallel control from sequential codes. This is done by embedding computation into a special kind of directed-arc graphs, the

fork-join graphs, in which many interesting applications are shown to be representable.

A very interesting and intelligent assisting-tool in writing programs that are to be run on distributed memory multicomputers - Alfer -, is presented by S. N. McIntosh-Smith, B. M. Brown and S. Hurley, from the University of Wales at Cardiff, UK, in their paper **"Intelligent Algorithm Decomposition for Parallelism with Alfer"**.

In his paper **"Symbolic Array Data Flow Analysis and Pattern Recognition in Numerical Codes"**, C. W. Keßler, from Graduiertenkolleg Informatik Universität des Saarlandes, Germany, focused on the symbolic array dataflow analysis techniques of PARAMAT's pattern recognition tool.

M. R. Gilder's, M. S. Krishnamoorthy's and J. R. Punin's (Rensselaer Polytechnic Institute, USA) paper **"A GUI for the Parallel Code Generation"** describes the Interactive Visualisation Tool(IVT) developed for the HICOR interactive parallelling compiler.

"Formal Techniques Based on Nets, Object Orientation and Reusability for Rapid Prototyping of Complex Systems", by F. Kordon from Institute Blaise Pascal, deals with a technique currently implemented under a software environment.

The next paper **"ADAPTOR - A Transformation Tool for HPF Programs"** is authored by T. Brandes and F. Zimmermann, from German National Center for Mathematics and Computer Science. ADAPTOR (Automatic Data Parallelism Translator) is a compilation tool for transforming programs written in data parallel FORTRAN style into programs with explicit message passing constructs.

Aiming to solve large two and three-dimensional CFD problems using unstructured meshes, a general framework has been formulated to enable parallel and sequential execution of a single source FORTRAN 77 code; this is achieved via the straightforward insertion of OPlus library routine calls (Oxford Parallel Library for Unstructured Solvers). It has been dedicated the paper titled **"Parallel Framework for Unstructured Grid Solves"** by D. A. Burgess, P. I. Crumpton and M. B. Giles, from Oxford University Computing Laboratory, UK.

M. Parashar, S. Hariri, T. Haupt and G. Fox, from Northeast Parallel Architecture Center at Syracuse University, USA, worked out **"A Study of Software Development for High Performance Computing"**, where the stages typically encountered in this process and the support required at each stage, are highlighted. The modelling of stock option pricing is taken as an example all over the study.

The concept of Parallel Computational Frames (PCF) for the combination of the parallelization techniques and related solutions, is introduced by M. Fruscione, P. Flocchini, E. Giudici, S.Punzi, P.Stofella from ACS (Advanced Computing Systems), Milan, Italy,in their paper **"Parallel Computational Frames: An Approach to Parallel Application Development Based on Message Passing Systems"**.

"A Knowledge-Based Scientific Parallel Programming Environment" is presented by K. M. Decker, J. J. Dvorak, R. M. Rehmann from CSCS-ETH Zurich, Switzerland. The environment features a program - oriented specification formalism and is based on a skeleton - oriented programming methodology.

"Parallel Distributed Algorithm Design Through Specification Transformation: The Asynchronous Vision System", by D. Buchs, D. Monteiro, F. Mourlin and D. Brunet, from Swiss Federal Institute of Technology, is a case study on a new class of general approaches to vision processing - the use of formal techniques to model complex concurrent systems having abstract specification.

Skeleton-oriented programming is a new technique for re-usability of software components in massively parallel systems. H.Burkhart and S.Gutzwiller from Informatics Department, University of Basel, Switzerland, describe it in their paper **"Steps Toward Reusability and Portability in Parallel Programming"**. They dwell on two aspects connected with the use of a library of algorithmic skeletons: first the BACS (Basel Algorithm Classification Scheme), and a

sample algorithm are summarized and presented; next TINA-the skeleton generator is introduced.

"An Environment for Portable Distributed Memory Parallel Programming" shows a first prototype currently installed of an integrated tool environment consisting of an extended High Performance Fortran (HPF) compiler, a parallel performance monitor and analyzer, and a parallel debugger for distributed memory parallel processors. The authors ,C.Clémencon, A.Endo, J.Fritscher, A.Müller, R.Rühl, have done the work as part of a Joint CSCS-ETH/NEC Collaboration in Parallel Processing.

"Reuse, Portability and Parallel Libraries" is a synthesis paper written by a group of scientists from Edinburgh Parallel Computing Centre, UK. The Common High-Level Interface for Message Passing (CHIMP) project is referred. The extent to which the project requirements are met in the new standard Message Passing Interface (MPI), and how the selection of the parallel libraries in Parallel Utility Library project, could help, are accounted for.

In his paper **"Assessing the Usability of Parallel Programming Systems: The Cowichan Problems"**, G. V. Wilson, Edinburgh Parallel Computing Centre, UK, describes a problem suite based on which usability, rather than performance, of parallel programming systems may be assessed.

D. Szafron and J. Schaeffer from the Department of Computing Science, University of Alberta, Canada, devote their paper **"Experimentally Assessing the Usability of Parallel Programming Systems"** to an experiment for comparing the usability of two parallel programming systems.

The European Centre for Medium Range Weather Forecasts ' efforts on parallel programming and the on-going PPPE work are emphasized by the paper **"Experiences with Parallel Programming Tools"**.

"An Efficient Implementation of MPI" is proposed by H. Franke, P. Hochschild, P. Pattnaik and M. Snir from IBM T. J. Watson Research Center, USA. MPI-F is a high performance implementation of MPI on the IBM SP1.

M.Aguilar and B.Hirsbrunner from IIUF-Institute of Informatics, University of Fribourgh, Germany, wrote a paper titled **"Post: A New Postal Delivery Model"**, on a new communication technique based on a postal delivery model, providing transparent mail managing and routing facilities for the implementation of load balancing strategies.

The resolution algorithm of an extension of the Prolog language to backtrackable communications, is presented in the paper **"Asynchronous Backtrackable Communications in the SLOOP Object-Oriented Language"**, written by a group of scientists from the Research Institute for Informatics in Toulouse, France.

PIOUS is a parallel file system architecture for providing process groups access to permanent storage within a heterogeneous network computing environment. The paper **"A Parallel I/O System for High-Performance Distributed Computing"**, by S. A. Moyer and V. S. Sunderam from the Department of Mathematics and Computer Science, Emory University, USA, discuss the PIOUS architecture and programming model. PIOUS supports parallel applications' development by providing co-ordinated access to parafile file objects with guaranteed consistency semantics and a dynamical-selectable fault tolerance level.

An approach to the problem of performing parallel input-output in data parallel computations, is made by R. Bordawekar and A. Choudhary from Syracuse University, USA, in their paper **"Language and Compiler Support for Parallel I/O"**.

A trade-off between realism and simplicity must exist in a computational model for parallel computing, which effective tools for design parallel algorithms are based on. P. Thanisch, M.G. Norman, C. Boeres, S. Pelagatti from Department of Computer Science, University of Edinburgh, UK, discuss it in their paper **"Locality in Scheduling Models of Parallel Computation"**.

The paper **"A Load Balancing Algorithm for Massively Parallel Systems"** by M. Cannataro, G. Spezzano and D. Talia from CRAI, Italy, presents a new load balancing algorithm for massively parallel computers, an algorithm called Probabilistic Strategy with Neighbourhood Synchronization.

A group of Japanese scientists from NEC addresses in the paper **"Static Performance Prediction in PCASE: A Programming Environment for Parallel Supercomputers"**, a performance estimator prototype implemented in PCASE.

The ParaMap tool design and implementation are described by R.B.Irvin and B.P.Miller in the paper **"A Performance Tool for High-Level Parallel Programming Languages"**.

PIMSY (for Parallel Implementation of a Monitoring System) - for analysing a large amount of data with a scalable approach, is a tool presented in the paper **"Implementation of a Scalable Trace Analysis Tool"**, written by X. Vigouroux, from École Normale Supérieure de Lyon, France.

Initial steps in **"The Design of a Tool for Parallel Program Performance Analysis and Tuning"** are guided by A.Hondroudakis and R.Procter from Edinburgh University, UK.

A very interesting tool helping the users tune the performance of their Cray T3D applications, is revealed by W. Williams, T. Hoel and D. Pase from the Cray Research, USA, in the paper **"The MPP Apprentice Performance Tool: Delivering the Performance of the Cray T3D"**.

A. Fagot's and J. C. de Kergommeaux's (LGI-IMAG, France) **"Optimized Record - Replay Mechanism for RPC-Based Parallel Programming"**, proposes a mechanism for record-replay of parallel programs written in a remote procedure call based parallel programming model.

In order to present a debugging approach, T. Kunz 's and J. Black's paper on **"Abstract Debugging of Distributed Applications"** use a faulty implementation of a distributed simulation. Using various abstract visualisations of an execution, the likely cause of an error is identified iteratively.

F.Guidec and J. M. Jézéquel demonstrate in **"Design of a Parallel Object-Oriented Linear Algebra Library"**, that a purely sequential object-oriented language can be used to build parallel libraries that permit an efficient and transparent use of Distributed Memory Parallel Computers.

The LOCCS (Low Overhead Communication and Computation Subroutines) library potential is presented in the paper **"A Library for Coarse Grain Macro-Pipelining in Distributed Memory Architectures"**, by F. Desprez, from University of Tennessee, USA.

F. Bréant and J. F. Pradat - Peyre present a technique for implementing massively parallel applications specified by means of coloured Petri nets, in **"An Improved Massively Parallel Implementation of Colored Petri-Net Specifications"**.

At University of Pisa, Italy, researches are carried out for developing **"A Tool for Parallel System Configuration and Program Mapping Based on Genetic Algorithms"**. The tool is driven by two genetic algorithms, one being involved in the fitness evaluation of the other.

Due to high computational costs of this strategy, a parallel implementation is proposed and its first results are shown.

In their paper **"Emulating a Paragon XP/S on a Network of Workstations"**, G. Stellner, A. Bode, S.Lamberts and T.Ludwig, from Technische Universität München, Germany, describe the NXLib environment which emulates a Paragon system on a network of workstations.

B. Bacci, M. Danelutto and S. Pelagatti, from the Department of Computer Science , University of Pisa, Italy, make a second and interesting contribution to the volume, this time on **"Evaluating VLIW-in-the-large"**. Starting from the VLIW-in-the-large computing model, which allows both coarse grain and fine grain parallelism to be exploited in the execution of programs onto the coMP architecture, experimental results come to validate both the computing model and the design choices relative

to the coMP, massively parallel computing architecture.

In **"Implementing a N-Mixed-Memory Model on a Distributed Memory System"**, V. Cholvi-Juan and J. B. Bernabéu-Aubán from the Computer Science Department of the Polytechnic University of Valencia, Spain, propose their model as a generalization of the strict and sufficient memory models. With this model, the strict memory model, if loose, still maintains its simplicity.

"Programming Environments for Massively Parallel Distributed Systems" - Proceedings of the Working Conference of the IFIP WG 10.3, April 25-29, 1994, Switzerland, is a collection of high-level articles. It offers an image of the state-of-the-art in the field of parallel programming by providing potential users with tools and techniques for easing the programming effort and for increasing performance.

Luminita Todor