

Lecture Notes in Computer Science

797

Wolfgang Gentzsch Uwe Harms (Eds.)

High-Performance Computing and Networking

International Conference and Exhibition

Munich, Germany, April 1994

Proceedings, Volume II: Networking and Tools



Springer-Verlag

Google Exhibit 1010
Google v. Singular

Wolfgang Gentzsch Uwe Harms (Eds.)

High-Performance Computing and Networking

International Conference and Exhibition
Munich, Germany, April 18-20, 1994

Proceedings

Volume II: Networking and Tools

Springer-Verlag

Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Series Editors

Gerhard Goos
Universität Karlsruhe
Postfach 69 80
Vincenz-Priessnitz-Straße 1
D-76131 Karlsruhe, Germany

Juris Hartmanis
Cornell University
Department of Computer Science
4130 Upson Hall
Ithaca, NY 14853, USA

Volume Editors

Wolfgang Gentzsch
FH Regensburg and GENIAS Software GmbH
Erzgebirgstraße 2, D-93073 Neutraubling, Germany

Uwe Harms
Harms Supercomputing Consulting
Bunsenstraße 5, D-81735 München, Germany



CR Subject Classification (1991): C.2-4, D, F.2, G.1, H.2, J.1-2, J.6, K.6

ISBN 3-540-57981-8 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-57981-8 Springer-Verlag New York Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

© Springer-Verlag Berlin Heidelberg 1994
Printed in Germany

Typesetting: Camera-ready by author
SPIN: 10131015 45/3140-543210 - Printed on acid-free paper

Contents

13. Networking

High-Performance Computing and Networking for Numerical Weather Prediction (Invited Paper) <i>G.-R. Hoffmann, ECMWF, Reading</i>	1
High-Performance Computing and Networking as Base Element of the Environmental Information System UIS of Baden-Württemberg (Invited Paper) <i>F. Schmidt, IKE, Stuttgart</i>	5
High Speed Networking for the Research Community in Europe (Invited Paper) <i>K. Ullmann, DFN, Berlin</i>	13
The National Information Infrastructure Testbed (NIIT) <i>M. McKeown</i>	18
Parallel Knowledge Processing in Open Networks <i>P.M. Borst, H.-T. Goetz, P.S. Sapaty, W. Zorn</i>	24
Distributed File Caching in Parallel Architectures Utilizing High Speed Networks <i>A. Klauser, R. Posch</i>	30
Application Fundamentals of Distributed, High-Performance Computing and Networking Systems <i>H. Westphal, D. Popovic</i>	36
Massive Parallelism with Workstation Clusters - Challenge or Nonsense? <i>C.H. Cap</i>	42

14. Future European Cooperative Working Possibilities in Industry and Research, as Demonstrated by the RACE Project PAGEIN

Networking Perspectives for Collaborative Supercomputing in European Aerospace Research and Industry <i>J.P. Peltier, P. Christ</i>	53
Future Aerospace Working Scenarios Using High Speed Networks and Supercomputers Applied to Flow Simulation for Complete Aircraft <i>S. Rill, R. Grosso</i>	60
A Software Environment for Cooperative Simulation and Visualization in the Aerospace Field <i>U. Lang</i>	70
Implementation of Visualization Tools in a Distributed Environment <i>S. Causse, F. Juaneda, M. Grave</i>	77

Networking Issues in PAGEIN: The "N" of "HPCN" <i>P. Haas, P. Christ</i>	86
Wavelet Interactive In-Line Analysis with Pégase <i>F.X. Roux, J. Ryan, B. Troff, T.H. Lê, K. Dang-Tran</i>	94
 15. HPCN Computer Centers Aspects	
European HPC Competitiveness Centers for Industry (Invited Paper) <i>A. Scheidegger, CSCS Switzerland</i>	103
An Open High-Performance Computing Infrastructure in Industry <i>R. Iffert, U. Harms</i>	109
A Catalog of Classifying Characteristics for Massively Parallel Computers <i>T. Bönniger, R. Esser, D. Krekel</i>	114
Building a Parallel Computing Environment from Heterogeneous Resources <i>K. Koski</i>	120
A Distributed Computing Center Software for the Efficient Use of Parallel Computer Systems <i>F. Ramme, T. Römke, K. Kremer</i>	129
InfoMall: A Scalable Organisation for the Development of High-Performance Computing and Communications – Software and Systems <i>G.C. Fox, E. Bogucz, D.A. Jones, K. Mills, M. Podgorny, K.A. Hawick</i>	137
Cluster Management Software for Open Object-Oriented Systems <i>A. Uszok, A. Król, K. Zielinski</i>	140
 16. Performance Evaluation and Benchmarking	
Transmission Rates and Performance of a Network of Computers <i>K. Boryczko, M. Bubak, M. Gajeczki, J. Kitowski, J. Moscinski, M. Pogoda</i>	142
Measurement of Communication Rates on the Cray T3D Interprocessor Network <i>R.W. Numrich, P.L. Springer, J.C. Peterson</i>	150
A Evaluation of the Meiko CS-2 Using the GENESIS Benchmark Suite <i>M. Baker, R. Cloke</i>	158
A Very High-Performance Algorithm for NAS EP Benchmark <i>R.C. Agarwal, F.G. Gustavson, M. Zubair</i>	164
Evaluation of Parallel Operating Systems <i>C. Lazou, G.R. Nudd</i>	170
JNNIE: Evaluation of Scalable Parallel Processing Architectures <i>S. Hotovy, T. Sterling</i>	172

17. Numerical Algorithms for Engineering

Recent Developments in Hybrid CG Methods (Invited Paper) <i>H.A. van der Vorst, Utrecht</i>	174
The Preconditioned Conjugate Gradient Method on Distributed Memory Systems <i>L.G.C. Crone</i>	184
Communication Cost Reduction for Krylov Methods on Parallel Computers <i>E. de Sturler, H.A. van der Vorst</i>	190
Parallel Arnoldi Method for the Construction of a Krylov Subspace Basis: an Application in Magnetohydrodynamics <i>J.G.L. Booten, P.M. Meijer, H.J.J. te Riele, H.A. van der Vorst</i>	196
A Scalable Paradigm for Effectively-Dense Matrix Formulated Applications <i>G. Cheng, G.C. Fox, K.A. Hawick</i>	202
Automatic Differentiation on Distributed Memory MIMD Systems <i>L. De Luca, P. Fiorino</i>	211
Optimisations for the Memory Hierarchy of a Singular Value Decomposition Algorithm Implemented on the MIMD Architecture <i>A. Czezowski, P. Strazdins</i>	215
Parallelization and Data Locality of a Multigrid Solver on the KSR1 <i>Ulrich Schwardmann</i>	217
On the Parallel Evaluations of Recurrences <i>A. Kiper</i>	219

18. Domain Decomposition in Engineering

Implementation of a Parallel Euler Solver with GRIDS <i>U. Geuder, M. Härdtnr, B. Wörner, R. Zink</i>	221
Distributed Mapping of SPMD Programs with a Generalized Kernighan-Lin Heuristic <i>J. de Keyser, D. Roose</i>	227
Mesh Decomposition and Communication Procedures for Finite Element Applications on the Connection Machine CM-5 System <i>Z. Johan, K.K. Mathur, S.L. Johnsson, T.J.R. Hughes</i>	233
CAPTtools - Semiautomatic Parallelization of Mesh-Based Computational Mechanics Codes <i>M. Cross, C.S. Ierotheou, S.P. Johnson, P.F. Leggett</i>	241

19. Parallel Programming Environments

The MPI Standard for Message Passing (Invited Paper) <i>R. Hempel, GMD, St. Augustin</i>	247
A Comparison of the Iserver-Occam, Parix, Express, and PVM Programming Environments on a Parsytec GCel <i>P.M.A. Sloot, A.G. Hoekstra, L.O. Hertzberger</i>	253
Message-Passing-Systems on Workstation Clusters and Parallel Computers - The Impact of Software- and Network-Architectures on Applications <i>M. Resch, A. Geiger, J. Zikeli</i>	260
EASYPVM - An Enhanced Subroutine Library for PVM <i>S. Saarinen</i>	267
DynamicPVM - Dynamic Load Balancing on Parallel Systems <i>L. Dikken, F. van der Linden, J. Vasseur, P.M.A. Sloot</i>	273
A Large-Scale Metacomputer Approach for Distributed Parallel Computing <i>V. Strumpen</i>	278
Developing Applications for Multicomputer Systems on Workstation Clusters <i>G. Stellner, A. Bode, S. Lamberts, Th. Ludwig</i>	286
System Support for Distributed Computing <i>S. Maffeis</i>	293
Assembly Environment for Development of Application Parallel Program <i>V.E. Malyshkin</i>	302
MaX - Investigating Shared Virtual Memory <i>R.G. Hackenberg</i>	308
Experiences with the Mether-NFS Virtual Shared Memory System <i>J.-Th. Pfenning</i>	316
Adapting PICL and ParaGraph Tools to Transputer-Based Multicomputers <i>V. Blanco, F.F. Rivera, O. Plata</i>	324
SCIDDLE: A Tool for Large Scale Cooperative Computing <i>P. Arbenz, H.P. Lüthi, Ch. Sprenger, S. Vogel</i>	326
TRAPPER: A Graphical Programming Environment for Parallel Systems <i>L. Schäfers, C. Scheidler, O. Krämer-Fuhrmann</i>	328

20. Load Balancing and Performance Optimization

Load Leveling on the Paragon Multicomputer <i>S. Tritscher, R. Zajcew, M. Barnett</i>	330
A Data-Parallel View of the Load Balancing - Experimental Results on MasPar MP-1 <i>C. Fonlupt, P. Marquet, J.-L. Dekeyser</i>	338

Distributed Scheduling for Multicomputers <i>L. Philippe, G.-R. Perrin</i>	344
A Tool for Optimizing Programs on Massively Parallel Computer Architectures <i>O. Hansen</i>	350
A Case Study in the Parallel Implementation of Functional Languages Using Skeletons <i>J. Schwarz, F.A. Rabhi</i>	357
A Dynamic Task Allocation Scheme in Benes-Type Nonblocking Networks <i>T. Feng, Y. Kim</i>	363
A Methodology for Dimensioning SPMD Distributed Systems Applied to Industrial Applications <i>A. Pinti, J.-C. Grossetie</i>	365
 21. Monitoring, Debugging, and Fault Tolerance	
Do-Loop-Surface: An Abstract Performance Data Visualization <i>O. Naim, A.J.G. Hey</i>	367
A Graphic Distributed Monitoring System for Unix-Based Architectures <i>G. De Pietro, A. Esposito, G. Reitano</i>	373
Parallel Application Design: The Simulation Approach with HASTE <i>P. Pouzet, J. Paris, V. Jorrand</i>	379
On-line Distributed Debugging on Scaleable Multicomputer Architectures <i>T. Bemmerl, R. Wismüller</i>	394
The FTMPs-Project: Design and Implementation of Fault-Tolerance Techniques for Massively Parallel Systems <i>J. Vounckx, G. Deconinck, R. Lauwereins, G. Viehöver, R. Wagner, H. Madeira, J.G. Silva, F. Balbach, J. Altmann, B. Bieker, H. Willeke</i>	401
 22. Programming Languages in HPC	
High-Performance Fortran Languages: Advanced Applications and Their Implementation (Invited Paper) <i>B. Chapman, H. Zima, Vienna, P. Mehrotra, Hampton</i>	407
Evaluation of High-Performance Fortran on Some Real Applications <i>T. Brandes</i>	417
Experiments with HPF Compilation for a Network of Workstations <i>F. Coelho</i>	423
High-Performance Fortran <i>G.M. van Waveren</i>	429

Value-Based Distributions in Fortran D <i>R. v. Hanxleden, K. Kennedy, J. Saltz</i>	434
Basic Linear Operators in Fortran 90 <i>R. Hanson, T. Leite</i>	441
parLisp - Parallel Symbolic Processing with Lisp on the Distributed Memory Machine MANNA <i>A. Sodan, H. Bi</i>	445

23. Compilers and Data Parallel Structures

Parallelizing Compilers: What Can Be Achieved? (Invited Paper) <i>M. Dion, Y. Robert, Ecole Nat. Lyon, J.-L. Philippe, Société Archipel</i>	447
A Family of Data-Parallel Derivations <i>M. Clint, S. Fitzpatrick, T.J. Harmer, P.L. Kilpatrick, J.M. Boyle</i>	457
Automatic Data Distribution <i>P. Crooks, R.H. Perrott</i>	463
Some Synthesis Aspects for Data Parallel Programming <i>G.-R. Perrin</i>	469
Influence of Fortran 90 Features on Performance on Cray Vector Computer Systems <i>U. Küster, M. Zürn</i>	475
Mapping of Patterns - The Way of Insight into Multi-Dimensional Data Structures <i>W.Dzwinel</i>	484

24. Architectural Aspects

The Instruction Systolic Array - Implementation of a Low-Cost Parallel Architecture as Add-On Board for Personal Computers <i>M. Schimmler, H.-W. Lang, R. Maaß</i>	487
The BBN Butterfly Family: An Overview <i>J.S. Alherbish</i>	489
Architectural Principles of the RPM <i>Y.A. Mamatov, V.P. Yemelin</i>	491

25. Late Papers

Applications Performance on Intel Paragon XP/S-15 <i>S. Saini, H.D. Simon</i>	493
Status Report on ESPRIT Project P7519 Palace: Parallelization of GEANT <i>C. Casari, C. Catelli, M. Guanziroli, M. Mazzeo, G. Meola, S. Punzi, A.L. Scheinine, P. Stofella</i>	499

A Practical Experience in Parallelizing a Large CFD Code: The ENSOLV Flow Solver	508
<i>M. Amato, A. Matrone, P. Schiano</i>	
Author Index	515

The Instruction Systolic Array - Implementation of a Low-Cost Parallel Architecture as Add-On Board for Personal Computers

Manfred Schimmler, Hans-Werner Lang, Rüdiger Maaß

ISATEC Soft- und Hardware GmbH, Kiel

Abstract. In this paper the architecture of a 1024 processor instruction systolic array is described. Besides the presentation of the concept and the architectural details of this novel type of massively parallel system the paper focusses on the problems that had to be solved during the development phase.

1. Introduction

Instruction systolic arrays (ISA's) have been developed to combine the simplicity of hardwired systolic arrays with the flexible programmability of MIMD-type processor arrays. The concept of the ISA has been introduced in [La86]. It is based on a mesh connected processor array, where each processor has a data link to its four neighbors (north, south, east, west). Open links at the boundary are used for external data communication.

The ISA is controlled by two orthogonal streams of information which are pumped through the array: a stream of instructions along the columns and a stream of binary selectors (0's or 1's) along the rows. The two streams are combined such that an instruction is executed whenever it meets a 1 and turned into a *no operation*, otherwise.

Since any information exchange is local (between adjacent processors), the concept avoids long wires for data as well as for control information. In addition, the bit-serial structure of the processors allows to design them small enough such that 64 processors fit onto a single integrated circuit.

The relevance of the ISA concept has been demonstrated by extensive studies of applications in [La86, Schm86, Schi87, DiS88, Stra90], its performance in comparison to other models of parallel computers was evaluated in [KLSS88].

This paper describes the architecture and the implementation of a parallel computer based on an ISA. It has been developed as an add-on board for personal computers. Besides the parallel unit it contains a fast interface logic between the ISA and the host system. There are 16 ISA chips on the board, each comprising 64 processors. Together they form a 32 x 32 array.

2. Single Instruction Systolic Arrays

The SISA (single instruction systolic array) [La87] is a quadratic array of identical processors. Every processor has a register set of fixed size including one designated *communication register* C which can be read by its four direct neighbors (north, south, east, west). Every processor can execute instructions from a small (RISC-like) instruction set I. The array is synchronized by a global clock and the execution of every instruction requires one clock cycle. This is achieved by breaking up complicated operations like division or floating point operations in sequences of machine instructions.

Input and output occur at the boundary of the array. In particular, a data item is input whenever a processor on the boundary of the array executes an instruction that attempts to read from the communication register of a nonexistent neighbor.

The instruction stream is reduced to one instruction per clock cycle, input into the top left processor. From there it is propagated in horizontal and vertical direction. In this way, the instructions are pumped through the array along the diagonals. In addition to the horizontal row selector stream, a vertical column selector stream is introduced. An instruction is executed in its processor if and only if it meets a row selector 1 and a column selector 1. Otherwise, it is interpreted as *no operation*.

Although the SISA is a special case of the ISA, it has been shown that most applications can be implemented on an SISA without any loss in performance [La87, Stra90].

3. Processor Architecture and Board Architecture

The ISA processor has a local memory of 32 words of 16 bits each. As active units it comprises an integer ALU, a bit serial multiplier, a unit for conditional instructions, the communication register, and several units for bit serial floating point arithmetic, e.g. a shifter, a tag manager, a flag manager, and an address modifier. The instruction decoder controls and synchronizes the different activities within the processor.

The processor operates in a bit serial way, i.e. execution of one instruction takes 16 internal clock cycles. Depending on the instruction, one or two operands are chosen from the local memory, the communication register, the communication registers of the neighbors, or the constants "0" or "1". The result of each operation is produced with the least significant bit first. It can be stored in one of the local registers, the communication register or in both.

The processor is clocked with 64 MHz for the internal bit-clock. Since every instruction takes 16 bit clocks, the global system clock is 4 MHz.

The parallel unit is laid out as a 4x4 square of ISA chips comprising an 8x8 array of processors each. Interchip communication is again done across short wires between physically adjacent pins, thus keeping to the restriction of local communication only.

The computing power of the ISA can only be exploited if there is a fast interface between the parallel unit and the host system, i.e. the PC. Therefore, a sophisticated memory structure has been implemented on the board to guarantee fast data transfer between ISA and PC.

4. Conclusion

The 1024 processor ISA board is a massively parallel unit which provides a computing power of up to 300 MFlops for an arbitrary PC. Its processor architecture and its memory hierarchy allows an efficient implementation of a large variety of applications. The processor chips have been optimized in terms of power efficiency in order to meet the requirements caused by the PC as a host system.

References

- [DiS88] Dittrich, A., Schmeck, H.: Given's Rotation on the Instruction Systolic Array. In: Wolf, G., Legendi, T., Schendel, U. (eds.): PARCELLA '88, Mathematical Research, Bd. 48, Akademie Verlag, Berlin, 340-346 (1988)
- [KLSS88] Kunde, M. et al., The Instruction Systolic Array and its Relation to Other Models of Parallel Computers, *Parallel Computing* 7, 25-39, (1988)
- [La86] Lang, H.-W., The Instruction Systolic Array, a parallel architecture for VLSI, *Integration, the VLSI Journal* 4, 65-74, (1986)
- [La87] Lang, H.-W., ISA and SISA, Two Variants of a General Purpose Systolic Array Architecture, 2nd Int. Conf on Supercomputing, 460-465, (1988)
- [Schi87] Schimmler, M.: Fast Sorting on the Instruction Systolic Array, Technical Report No. 8709, Christian-Albrechts-University, Kiel, (1987)
- [Schm86] Schmeck, H., A Comparison-Based Instruction Systolic Array, In: Robert, Y. et al. (eds): *Parallel Algorithms and Architectures*, North Holland (1986)
- [Stra90] Strazdins, P.E.: Control Structures for Mesh-Connected Networks, PhD Thesis, The Australian National University, Canberra, Australia, 1990.