

Lecture Notes in Computer Science

589

Edited by G. Goos and J. Hartmanis

Advisory Board: W. Brauer D. Gries J. Stoer



U. Banerjee D. Gelernter A. Nicolau
D. Padua (Eds.)

Languages and Compilers for Parallel Computing

Fourth International Workshop
Santa Clara, California, USA, August 7-9, 1991
Proceedings

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
W-7500 Karlsruhe, FRG

Juris Hartmanis
Department of Computer Science
Cornell University
5148 Upson Hall
Ithaca, NY 14853, USA

Volume Editors

Utpal Banerjee
Intel Corporation
2801 Northwestern Parkway
Santa Clara, CA 95052-8122, USA

David Gelernter
Dept. of Computer Science, Yale University
51 Prospect St., New Haven, CT 06520, USA

Alex Nicolau
Dept. of Information & Computer Science
University of California
444 Computer Science Bldg., Irvine, CA 92717, USA

David Padua
Center for Supercomputing Research and Development
University of Illinois
104 S. Wright St., Urbana, IL 61801, USA

CR Subject Classification (1991): F.1.2, D.1.3, D.3.1, B.2.1, I.3.1

ISBN 3-540-55422-X Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-55422-X 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 1992
Printed in Germany

Typesetting: Camera ready by author
Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
45/3140-543210 - Printed on acid-free paper

Foreword

This book contains the papers presented at the Fourth Workshop on Languages and Compilers for Parallel Computing held during August 7-9, 1991 in Santa Clara, California. The workshop was sponsored this year by the Intel Corporation. The previous workshops in this series were held in Irvine, California (1990), Urbana, Illinois (1989), and Ithaca, New York (1988).

The papers in this book cover several important topics including: (1) languages and structures to represent programs internally in the compiler, (2) techniques to analyze and manipulate sequential loops in order to generate a parallel version, (3) techniques to detect and extract fine-grain parallelism, (4) scheduling and memory-management issues in automatically-generated parallel programs, (5) Parallel programming language designs, and (6) compilation of explicitly parallel programs.

We are very pleased with the breadth and depth of the work presented in these papers. Taken together, these papers are an accurate reflection of the state of research in languages and compilers for parallel computing in 1991. We hope this book will be as interesting to the reader as it was for us to compile.

January 1992

Utpal Banerjee
David Gelernter
Alex Nicolau
David Padua

Contents

I. EXPLICITLY PARALLEL LANGUAGES

Distributed Execution of Actor Programs	1
G. Agha, C. Houck, and R. Panwar <i>University of Illinois at Urbana-Champaign</i>	
An Overview of the Fortran D Programming System.	18
S. Hiranandani, K. Kennedy, C. Koebel, U. Kremer, and C.-W. Tseng <i>Rice University, Houston, Texas</i>	
The Interaction of the Formal and the Practical in Parallel Programming . . .	35
Environment Development: CODE J. Werth, J. Browne, S. Sobek, T. Lee, P. Newton, and R. Jain <i>University of Texas at Austin</i>	
Hierarchical Concurrency in Jade	50
D. Scales, M. Rinard, M. Lam, and J. Anderson <i>Stanford University, Stanford, California</i>	

II. EXPERIMENTATION WITH PARALLEL PROGRAMMING

Experience in the Automatic Parallelization of Four Perfect-Benchmark . . .	65
Programs R. Eigenmann, J. Hoeflinger, Z. Li, and D. Padua <i>University of Illinois at Urbana-Champaign</i>	
Programming SIMPLE for Parallel Portability	84
J. Lee, C. Lin, and L. Snyder <i>University of Washington at Seattle</i>	

III. INTERNAL REPRESENTATION

Compilation of Id	99
Z. Ariola, <i>Harvard University, Cambridge, Massachusetts</i> Arvind, <i>Massachusetts Institute of Technology, Cambridge, Massachusetts</i>	
An Executable Representation of Distance and Direction	122
R. Johnson, W. Li, and K. Pingali <i>Cornell University, Ithaca, New York</i>	

Integrating Scalar Optimization and Parallelization	137
S. Tjiang, M. Wolf, M. Lam, K. Pieper, and J. Hennessy <i>Stanford University, Stanford, California</i>	
Optimization of Data/Control Conditions in Task Graphs	152
M. Girkar and C. Polychronopoulos <i>University of Illinois at Urbana-Champaign</i>	
IV. LOOP PARALLELISM	
Recognizing and Parallelizing Bounded Recurrences	169
D. Callahan <i>Tera Computer Company, Seattle, Washington</i>	
Communication-Free Hyperplane Partitioning of Nested Loops	186
C.-H. Huang and P. Sadayappan <i>The Ohio State University, Columbus, Ohio</i>	
Parallelizing Loops with Indirect Array References or Pointers.	201
L.-C. Lu and M. Chen <i>Yale University, New Haven, Connecticut</i>	
V. FINE GRAIN PARALLELISM	
Register Allocation, Renaming and Their Impact on Fine-Grain Parallelism	218
A. Nicolau, R. Potasman, and H. Wang <i>University of California at Irvine</i>	
Data Flow and Dependence Analysis for Instruction Level Parallelism	236
B.R. Rau <i>Hewlett-Packard Laboratories, Palo Alto, California</i>	
VI. ANALYSIS TECHNIQUES	
Extending Conventional Flow Analysis to Deal with Array References	251
A. Kallis and D. Klappholz <i>Stevens Institute of Technology, Hoboken, New Jersey</i>	
VII. COMPILERS & SCHEDULING	
Run-Time Management of Lisp Parallelism and the Hierarchical Task	266
Graph Program Representation M. Furnari, <i>Institute of Cybernetics, Arco Felice (Na), Italy</i> C. Polychronopoulos, <i>University of Illinois at Urbana-Champaign</i>	

A Multi-Grain Parallelizing Compilation Scheme for OSCAR (Optimally Scheduled Advanced Multiprocessor)	283
H. Kasahara, H. Honda, A. Mogi, A. Ogura, K. Fujiwara, and S. Narita <i>Waseda University, Tokyo, Japan</i>	

Balanced Loop Partitioning Using GTS	298
J. Labarta, E. Ayguade, J. Torres, M. Valero, and J.M. Llaberia <i>Polytechnic University of Catalonia, Barcelona, Spain</i>	

VIII. CACHE MEMORY ISSUES

An Iteration Partition Approach for Cache or Local Memory Thrashing on Parallel Processing	313
J. Fang, <i>Hewlett-Packard Laboratories, Palo Alto, California</i> M. Lu, <i>Texas A&M University, College Station, Texas</i>	

On Estimating and Enhancing Cache Effectiveness	328
J. Ferrante, <i>IBM T.J. Watson Research Center, Yorktown Heights, New York</i> V. Sarkar, <i>IBM Palo Alto Scientific Center, Palo Alto, California</i> W. Thrash, <i>University of Washington at Seattle</i>	

Reduction of Cache Coherence Overhead by CompilerData Layout and Loop Transformation	344
Y.-J. Ju and H. Dietz <i>Purdue University, Lafayette, Indiana</i>	

IX. COMPILERS FOR DATAFLOW MACHINES

Loop Storage Optimization for Dataflow Machines	359
G. Gao and Q. Ning <i>McGill University, Montreal, Quebec, Canada</i>	

Optimal Partitioning of Programs for Data Flow Machines.	374
R. Hardon and S. Pinter <i>Technion - Israel Institute of Technology, Technion City, Haifa</i>	

X. ANALYSIS OF EXPLICITLY PARALLEL PROGRAMS

A Foundation for Advanced Compile-time Analysis of Linda Programs	389
N. Carriero and D. Gelernter <i>Yale University, New Haven, Connecticut</i>	

Analyzing Programs with Explicit Parallelism	405
H. Srinivasan and M. Wolfe <i>Oregon Graduate Institute, Beaverton, Oregon</i>	