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
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
  University of Illinois at Urbana-Champaign

An Overview of the Fortran D Programming System .............. 18
  S. Hiranandani, K. Kennedy, C. Koelbel, U. Kremer, and C.-W. Tseng
  Rice University, Houston, Texas

The Interaction of the Formal and the Practical in Parallel Programming .... 35
  J. Werth, J. Browne, S. Sobek, T. Lee, P. Newton, and R. Jain
  University of Texas at Austin

Hierarchical Concurrency in Jade ................................. 50
  D. Scales, M. Rinard, M. Lam, and J. Anderson
  Stanford University, Stanford, California

II. EXPERIMENTATION WITH PARALLEL PROGRAMMING

Experience in the Automatic Parallelization of Four Perfect-Benchmark Programs .... 65
  R. Eigenmann, J. Hoeflinger, Z. Li, and D. Padua
  University of Illinois at Urbana-Champaign

Programming SIMPLE for Parallel Portability ..................... 84
  J. Lee, C. Lin, and L. Snyder
  University of Washington at Seattle

III. INTERNAL REPRESENTATION

Compilation of Id ............................................. 99
  Z. Ariola, Harvard University, Cambridge, Massachusetts
  Arvind, Massachusetts Institute of Technology, Cambridge, Massachusetts

An Executable Representation of Distance and Direction ............ 122
  R. Johnson, W. Li, and K. Pingali
  Cornell University, Ithaca, New York
Integrating Scalar Optimization and Parallelization .......... 137
S. Tjiang, M. Wolf, M. Lam, K. Pieper, and J. Hennessy
Stanford University, Stanford, California

Optimization of Data/Control Conditions in Task Graphs .......... 152
M. Girkar and C. Polychronopoulos
University of Illinois at Urbana-Champaign

IV. LOOP PARALLELISM

Recognizing and Parallelizing Bounded Recurrences .......... 169
D. Callahan
Tera Computer Company, Seattle, Washington

Communication-Free Hyperplane Partitioning of Nested Loops .......... 186
C.-H. Huang and P. Sadayappan
The Ohio State University, Columbus, Ohio

Parallelizing Loops with Indirect Array References or Pointers .......... 201
L.-C. Lu and M. Chen
Yale University, New Haven, Connecticut

V. FINE GRAIN PARALLELISM

Register Allocation, Renaming and Their Impact on Fine-Grain Parallelism .......... 218
A. Nicolau, R. Potasman, and H. Wang
University of California at Irvine

Data Flow and Dependence Analysis for Instruction Level Parallelism .......... 236
B.R. Rau
Hewlett-Packard Laboratories, Palo Alto, California

VI. ANALYSIS TECHNIQUES

Extending Conventional Flow Analysis to Deal with Array References .......... 251
A. Kallis and D. Klappholz
Stevens Institute of Technology, Hoboken, New Jersey

VII. COMPILERS & SCHEDULING

Run-Time Management of Lisp Parallelism and the Hierarchical Task .......... 266
Graph Program Representation
M. Furnari, Institute of Cybernetics, Arco Felice (Na), Italy
C. Polychronopoulos, University of Illinois at Urbana-Champaign
A Multi-Grain Parallelizing Compilation Scheme for OSCAR

(Optimally Scheduled Advanced Multiprocessor)

Waseda University, Tokyo, Japan

Balanced Loop Partitioning Using GTS

J. Labarta, E. Ayguade, J. Torres, M. Valero, and J.M. Llaberia
Polytechnic University of Catalonia, Barcelona, Spain

VIII. CACHE MEMORY ISSUES

An Iteration Partition Approach for Cache or Local Memory Thrashing on Parallel Processing

J. Fang, Hewlett-Packard Laboratories, Palo Alto, California
M. Lu, Texas A&M University, College Station, Texas

On Estimating and Enhancing Cache Effectiveness

J. Ferrante, IBM T.J. Watson Research Center, Yorktown Heights, New York
V. Sarkar, IBM Palo Alto Scientific Center, Palo Alto, California
W. Thrash, University of Washington at Seattle

Reduction of Cache Coherence Overhead by CompilerData Layout and Loop Transformation

Y.-J. Ju and H. Dietz
Purdue University, Lafayette, Indiana

IX. COMPILERS FOR DATAFLOW MACHINES

Loop Storage Optimization for Dataflow Machines

G. Gao and Q. Ning
McGill University, Montreal, Quebec, Canada

Optimal Partitioning of Programs for Data Flow Machines

R. Hardon and S. Pinter
Technion - Israel Institute of Technology, Technion City, Haifa

X. ANALYSIS OF EXPLICITLY PARALLEL PROGRAMS

A Foundation for Advanced Compile-time Analysis of Linda Programs

N. Carriero and D. Gelernter
Yale University, New Haven, Connecticut

Analyzing Programs with Explicit Parallelism

H. Srinivasan and M. Wolfe
Oregon Graduate Institute, Beaverton, Oregon