# Automatic Parallelization An Overview Of Fundamental Compiler Techniques Samuel P Midkiff

Advanced Parallel Processing Technologies
23rd International Workshop, LCPC 2010, Houston, TX, USA, October 7-9, 2010. Revised Selected Papers
Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2015
An Overview of Fundamental Compiler Techniques
Programmer-Assisted Automatic Parallelization
Automatic Parallelization of Non-uniform Loops
Network and Parallel Computing
Automatic Parallelization Via Loop Separation
An Overview of Fundamental Compiler Techniques
Automatic Parallelization: An Incremental, Optimistic, Practical Approach
Extending Automatic Parallelization to Optimize High-Level Abstractions for Multicore
Compiler Construction
Introduction to Scientific and Technical Computing
Input/Output Intensive Massively Parallel Computing
8th International Conference, ICA3PP 2008, Agia Napa, Cyprus, June 9-11, 2008, Proceedings
Scheduling and Automatic Parallelization
Scheduling and Automatic Parallelization
Foundations, HPF Realization, and Scientific Applications
Scheduling and Automatic Parallelization: Multidimensional Problems. 4. Systems of Uniform Recurrence Equations. 5. Parallelism Detection in Nested Loops
Final Report of Work Under Contract W36-034-ORD-7593 Between the Ordinance Department, Department of the Army and the University of Pennsylvania, Moore School of Electrical Engineering
Automatic Parallelization Techniques for Massively Parallel Machines

Symbolic Analysis Techniques for Effective Automatic Parallelization
New Approaches to Code Generation, Data Distribution, and Performance Prediction
Network and Parallel Computing
Automatic parallelization for distributed-memory multiprocessing systems
Languages and Compilers for Parallel Computing
9th International Symposium, APPT 2011, Shanghai, China, September 26-27, 2011, Proceedings
Automatic Parallelization of Loops with Data Dependent Control Flow and Array Access Patterns
Automatic Parallelization of Loop Programs for Distributed Memory Architectures
Multithreaded Computer Architecture: A Summary of the State of the ART
Automatic Performance Prediction of Parallel Programs
Introduction to Parallel Computing
Algorithms and Architectures for Parallel Processing
Handbook of Grammatical Evolution
The Data Parallel Programming Model
Semantic-Aware Automatic Parallelization of Modern Applications Using High-Level Abstractions
12th International Conference, CC 2003, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2003, Warsaw, Poland, April 7-11, 2003, Proceedings
A Functional Description of the Edvac [an Automatically-sequence Serial Binary Electronic Digital Computer
Automatic Parallelization



## KORBIN DANIELA

**Advanced Parallel Processing Technologies** Springer Science & Business Media
This edited book presents scientific results of the 16th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2015) which was held on June 1 – 3, 2015 in Takamatsu, Japan. The aim of this conference was to bring together researchers and scientists, businessmen and entrepreneurs, teachers, engineers, computer users, and students to discuss the numerous fields of computer science and to share their experiences and exchange new ideas and information in a meaningful way. Research results about all aspects (theory, applications and tools) of computer and information science, and

to discuss the practical challenges encountered along the way and the solutions adopted to solve them.

**23rd International Workshop, LCPC 2010, Houston, TX, USA, October 7-9, 2010. Revised Selected Papers** CRC Press
This monograph-like book assembles the thorougly revised and cross-reviewed lectures given at the School on Data Parallelism, held in Les Menuires, France, in May 1996. The book is a unique survey on the current status and future perspectives of the currently very promising and popular data parallel programming model. Much attention is paid to the style of writing and complementary coverage of the relevant issues throughout the 12 chapters. Thus these lecture notes are ideally suited for advanced courses or self-instruction on data parallel programming. Furthermore, the book is indispensable reading for anybody doing research in data parallel programming and related areas.

**Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2015** World Scientific
Created to help scientists and engineers write computer code, this practical book addresses the important tools and techniques that are necessary for scientific computing, but which are not yet commonplace in science and engineering curricula. This book contains chapters summarizing the most important topics that computational researchers need to know about. It leverages the viewpoints of passionate experts involved with scientific computing courses around the globe and aims to be a starting point for new computational scientists and a reference for the experienced. Each contributed chapter focuses on a specific tool or skill, providing the content needed to provide a working knowledge of the topic in about one day. While many individual books on specific computing topics exist, none is explicitly focused on getting technical professionals and students up and running immediately across a variety of computational areas.

*An Overview of Fundamental Compiler Techniques* Springer
The automatic generation of parallel code from high level sequential description is of key importance to the wide spread use of high performance machine architectures. This text considers (in detail) the theory and practical realization of automatic mapping of algorithms generated from systems of uniform recurrence equations (do-lccps) onto fixed size architectures with defined communication primitives. Experimental results of the mapping scheme and its implementation are given. Contents:Survey and Analysis of Partitioning and MappingImprovements to Some Existing Partitioning and Mapping MethodsA Methodology of Partitioning and Mapping for Given (N-1)-D Regular ArraysOptimal Mapping onto Lower-Dimensional Regular ArraysThe Structure of Parallel ProgramsParallel Code GenerationExperimental Results and DiscussionsConclusionBibliography Readership: Postgraduate students and researchers in academics & industry, compiler writers, computer scientists and electrical engineers. keywords:

Programmer-Assisted Automatic Parallelization Springer Science & Business Media
Compiling for parallelism is a longstanding topic of compiler research. This book describes the fundamental principles of compiling "regular" numerical programs for parallelism. We begin with an explanation of analyses that allow a compiler to understand the interaction of data reads and writes in different

statements and loop iterations during program execution. These analyses include dependence analysis, use-def analysis and pointer analysis. Next, we describe how the results of these analyses are used to enable transformations that make loops more amenable to parallelization, and discuss transformations that expose parallelism to target shared memory multicore and vector processors. We then discuss some problems that arise when parallelizing programs for execution on distributed memory machines. Finally, we conclude with an overview of solving Diophantine equations and suggestions for further readings in the topics of this book to enable the interested reader to delve deeper into the field. Table of Contents: Introduction and overview / Dependence analysis, dependence graphs and alias analysis / Program parallelization / Transformations to modify and eliminate dependences / Transformation of iterative and recursive constructs / Compiling for distributed memory machines / Solving Diophantine equations / A guide to further reading

Automatic Parallelization of Non-uniform Loops Springer

A complete source of information on almost all aspects of parallel computing from introduction, to architectures, to programming paradigms, to algorithms, to programming standards. It covers traditional Computer Science algorithms, scientific computing algorithms and data intensive algorithms.

Network and Parallel Computing Springer

With the era of increasing clock speeds coming to an end, parallel computing architectures have now become main-stream. Due to the wide range of architectures available today that can be used to exploit parallelism, ranging from multicore CPUs, to GPGPUs, to distributed memory machines; adapting applications for efficient execution on all these architectures poses a significant challenge.

**Automatic Parallelization Via Loop Separation** Morgan & Claypool Publishers

This book constitutes the proceedings of the 14th IFIP WG 10.3 International Conference on Network and Parallel Computing, NPC 2017, held in Hefei, China, in October 2017.The 9 full papers and 10 short papers presented in this book were carefully reviewed and selected from 88 submissions. The papers cover traditional areas of network and parallel computing including parallel applications, distributed algorithms, software environments, and distributed tools.

Springer Science & Business Media

The automatic generation of parallel code from high level sequential description is of key importance to the wide spread use of high performance machine architectures. This text considers (in detail) the theory and practical realization of automatic mapping of algorithms generated from systems of uniform recurrence equations (do-lccps) onto fixed size architectures with defined communication primitives. Experimental results of the mapping scheme and its implementation are given.

*An Overview of Fundamental Compiler Techniques* Springer

This book constitutes the thoroughly refereed post-proceedings of the 23rd International Workshop on Languages and Compilers for Parallel Computing, LCPC 2010, held in Houston, TX, USA, in October 2010. The 18 revised full papers presented were carefully reviewed and selected from 47 submissions. The scope of the workshop spans foundational results and practical experience, and targets all classes of parallel platforms including

concurrent, multithreaded, multicore, accelerated, multiprocessor, and cluster systems

*Automatic Parallelization: An Incremental, Optimistic, Practical Approach* Springer Science & Business Media

The historic focus of Automatic Parallelization efforts has been limited in two ways. First, parallelization has generally been attempted only on codes which can be proven to be parallelizeable. Unfortunately, the requisite dependence analysis is undecidable, and today's applications demonstrate that this restriction is more than just theoretical. Second, parallel program generation has generally been geared to custom multi-processing hardware. Although a network of workstations (NOW) could in principle be harnessed to serve as a multiprocessing platform, the NOW has characteristics which are at odds with elective utilization. This thesis shows that by restricting our attention to the important domain of "embarrassingly parallel" applications, leveraging existing scalable and efficient network services, and carefully orchestrating a synergy between compile-time transformations and a small runtime system, we can achieve a parallelization that not only works in the face of inconclusive program analysis, but is also efficient for the NOW. We optimistically parallelize loops whose memory access behavior is unknown, relying on the runtime system to provide efficient detection and recovery in the case of an overly optimistic transformation. Unlike previous work in speculative parallelization, we provide a methodology which is not tied to the Fortran language, making it feasible as a generally useful approach. Our runtime system implements Two-Phase Idempotent Eager Scheduling (TIES) for efficient network execution, providing an Automatic Parallelization platform with performance scalability for the NOW.

Extending Automatic Parallelization to Optimize High-Level Abstractions for Multicore World Scientific

This book constitutes the refereed proceedings of the 16th International Symposium on Static Analysis, SAS 2009, held in Los Angeles, CA, USA in August 2009 - co-located with LICS 2009, the 24th IEEE Symposium on Logic in Computer Science. The 21 revised full papers presented together with two invited lectures were carefully reviewed and selected from 52 submissions. The papers address all aspects of static analysis including abstract domains, abstract interpretation, abstract testing, compiler optimizations, control flow analysis, data flow analysis, model checking, program specialization, security analysis, theoretical analysis frameworks, type based analysis, and verification systems.

**Compiler Construction** Morgan & Claypool Publishers

Automatic Performance Prediction of Parallel Programs presents a unified approach to the problem of automatically estimating the performance of parallel computer programs. The author focuses primarily on distributed memory multiprocessor systems, although large portions of the analysis can be applied to shared memory architectures as well. The author introduces a novel and very practical approach for predicting some of the most important performance parameters of parallel programs, including work distribution, number of transfers, amount of data transferred, network contention, transfer time, computation time and number of cache misses. This approach is based on advanced compiler analysis that carefully examines loop iteration

spaces, procedure calls, array subscript expressions, communication patterns, data distributions and optimizing code transformations at the program level; and the most important machine specific parameters including cache characteristics, communication network indices, and benchmark data for computational operations at the machine level. The material has been fully implemented as part of P3T, which is an integrated automatic performance estimator of the Vienna Fortran Compilation System (VFCS), a state-of-the-art parallelizing compiler for Fortran77, Vienna Fortran and a subset of High Performance Fortran (HPF) programs. A large number of experiments using realistic HPF and Vienna Fortran code examples demonstrate highly accurate performance estimates, and the ability of the described performance prediction approach to successfully guide both programmer and compiler in parallelizing and optimizing parallel programs. A graphical user interface is described and displayed that visualizes each program source line together with the corresponding parameter values. P3T uses color-coded performance visualization to immediately identify hot spots in the parallel program. Performance data can be filtered and displayed at various levels of detail. Colors displayed by the graphical user interface are visualized in greyscale. Automatic Performance Prediction of Parallel Programs also includes coverage of fundamental problems of automatic parallelization for distributed memory multicomputers, a description of the basic parallelization strategy and a large variety of optimizing code transformations as included under VFCS.

*Introduction to Scientific and Technical Computing* Morgan & Claypool Publishers

Welcome to the proceedings of the 8th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2008). ICA3PP 2008 consist of two keynote addresses, seven technical sessions, and one tutorial. Included in these proceedings are papers whose authors are from Australia, Brazil, Canada, China, Cyprus, France, India, Iran, Israel, Italy, Japan, Korea, Germany, Greece, Mexico, Poland, Portugal, Romania, Spain, Switzerland, Taiwan, Tunisia, UAE, UK, and USA. Each paper was rigorously reviewed by at least three Program Committee members and/or external revi- ers, and the acceptance ratio is 35%. These papers were presented over seven technical sessions. Based on the paper review results, three papers were selected as the best papers. We would like to thank the many people who helped make this conference a successful event. We thank all authors who submitted their work to ICA3PP 2008, and all Program Committee members and additional reviewers for their diligent work in the paper review process ensuring a collection of high-quality papers. We are grateful to Hong Shen University of Adelaide, Australia and Kleanthis Psarris University of Texas at San Antonio, United States, for their willingness to be the keynote speakers. Our thanks go to Hai Jin and George Papapodoulos, the conference General Co-chairs, and Andrzej Goscinski, W- lei Zhou and Yi Pan, the conference Steering Committee Co-chairs for help in many aspects of organizing this conference. Finally, we thank all the conference participants for traveling to Cyprus.

*Input/Output Intensive Massively Parallel Computing* Springer Science & Business Media

Automatic introduction of OpenMP for sequential applications has attracted significant attention recently because of the proliferation of multicore processors and the simplicity of using OpenMP to express parallelism for shared-memory systems. However, most previous research has only focused on C and Fortran applications operating on primitive data types. Modern applications using high-level abstractions, such as C++ STL containers and complex user-defined class types, are largely ignored due to the lack of research compilers that are readily able to recognize high-level object-oriented abstractions and leverage their associated semantics. In this paper, we use a source-to-source compiler infrastructure, ROSE, to explore compiler techniques to recognize high-level abstractions and to exploit their semantics for automatic parallelization. Several representative parallelization candidate kernels are used to study semantic-aware parallelization strategies for high-level abstractions, combined with extended compiler analyses. Preliminary results have shown that semantics of abstractions can help extend the applicability of automatic parallelization to modern applications and expose more opportunities to take advantage of multicore processors.

**8th International Conference, ICA3PP 2008, Agia Napa, Cyprus, June 9-11, 2008, Proceedings** Springer Science & Business Media

Massively parallel processing is currently the most promising answer to the quest for increased computer performance. This has resulted in the development of new programming languages and programming environments and has stimulated the design and production of massively parallel supercomputers. The efficiency of concurrent computation and input/output essentially depends on the proper utilization of specific architectural features of the underlying hardware. This book focuses on development of runtime systems supporting execution of parallel code and on supercompilers automatically parallelizing code written in a sequential language. Fortran has been chosen for the presentation of the material because of its dominant role in high-performance programming for scientific and engineering applications.

**Scheduling and Automatic Parallelization** Springer

Automatic introduction of OpenMP for sequential applications has attracted significant attention recently because of the proliferation of multicore processors and the simplicity of using OpenMP to express parallelism for shared-memory systems. However, most previous research has only focused on C and Fortran applications operating on primitive data types. C++ applications using high-level abstractions, such as STL containers and complex user-defined types, are largely ignored due to the lack of research compilers that are readily able to recognize high-level object-oriented abstractions and leverage their associated semantics. In this paper, we automatically parallelize C++ applications using ROSE, a multiple-language source-to-source compiler infrastructure which preserves the high-level abstractions and gives us access to their semantics. Several representative parallelization candidate kernels are used to explore semantic-aware parallelization strategies for high-level abstractions, combined with extended compiler analyses. Those kernels include an array-base computation loop, a loop with task-level parallelism, and a domain-specific tree traversal. Our work

extends the applicability of automatic parallelization to modern applications using high-level abstractions and exposes more opportunities to take advantage of multicore processors.

**Scheduling and Automatic Parallelization** Springer Science & Business Media

I Unidimensional Problems.- 1 Scheduling DAGs without Communications.- 2 Scheduling DAGs with Communications.- 3 Cyclic Scheduling.- II Multidimensional Problems.- 4 Systems of Uniform Recurrence Equations.- 5 Parallelism Detection in Nested Loops.

**Foundations, HPF Realization, and Scientific Applications** Morgan & Claypool Publishers

This book constitutes the refereed proceedings of the 9th International Symposium on Advanced Parallel Processing Technologies, APPT 2011, held in Shanghai, China, in September 2011. The 13 revised full papers presented were carefully reviewed and selected from 40 submissions. The papers are organized in topical sections on parallel distributed system architectures, architecture, parallel application and software, distributed and cloud computing.

Scheduling and Automatic Parallelization: Multidimensional Problems. 4. Systems of Uniform Recurrence Equations. 5. Parallelism Detection in Nested Loops Pearson Education

The emerging three-dimensional (3D) chip architectures, with their intrinsic capability of reducing the wire length, promise attractive solutions to reduce the delay of interconnects in future microprocessors. 3D memory stacking enables much higher memory bandwidth for future chip-multiprocessor design, mitigating the "memory wall" problem. In addition, heterogenous integration enabled by 3D technology can also result in innovative designs for future microprocessors. This book first provides a brief introduction to this emerging technology, and then presents a variety of approaches to designing future 3D microprocessor systems, by leveraging the benefits of low latency, high bandwidth, and heterogeneous integration capability which are offered by 3D technology.

Related with Automatic Parallelization An Overview Of Fundamental Compiler Techniques Samuel P Midkiff:
• Racoon In Spanish Language : click here