

# Preface

The diversity of today’s architectures has forced programmers to spend additional efforts to port and tune their application source code across different platforms. In this process, compilers need additional tuning to generate better code. Recent compilers offer a vast number of multilayered optimizations, capable of targeting different code segments of an application. Choosing the right set of optimizations can significantly impact the performance of the code. This is a challenge made more complicated by the need to find the best order in which they should be applied given an application. Finding the best ordering is a long-standing problem in compilation research called the phase-ordering problem. The traditional approach for constructing compiler heuristics to solve it simply cannot cope with the enormous complexity of choosing the right ordering of optimizations for each code segment in an application. The current research focuses on exploring, studying, and developing an innovative approach to the problem of identifying the compiler optimizations that maximize the performance of a target application.

## Overview of this Book

This book addresses two fundamental problems involved in compilation research: the problem of *selecting* the best compiler optimizations and the *phase-ordering* problem. Statistical analyses were extensively used to relate the performance of an application to the applied optimizations. More precisely, machine learning models were adapted to predict an outcome. Here, an outcome is described either in terms of performance metrics, or the use of a certain compiler optimization. Similar to other machine learning approaches, we use a set of training applications to learn the statistical relationships between application features and compiler optimizations. For instance, Bayesian networks are used to learn the statistical model to which an

application can be represented with. We call these representations an application feature. Thus, given a new application not included in the training set, its features are fed to the Bayesian network as evidence. This evidence generates a bias on the distribution, as compiler optimizations are correlated with software features. The obtained probability distribution is application-specific and effectively focuses on the prediction of the most promising compiler optimizations. This will be discussed in detail in Chap. 3.

## Who is this Book for?

This is a textbook that aims to showcase the very recent developments of research approaches in the compilation research, specifically autotuning. Therefore, all researchers in the compiler community, computer architecture, parallel computing, and machine learning can benefit from reading it. Additionally, given the potential industrial impact of the provided approaches, it is recommended to read to other technical professionals as well.

## Summary and Organization

This book tackles the major problems of compiler autotuning. We use machine learning, DSE, and meta-heuristic techniques to construct efficient and accurate models to induce prediction models.

It is organized as follows: First, we provide an extensive review of the state of the art in Chap. 1. We survey more than hundred recent papers of the past twenty-five years since when the applications of machine learning have been introduced to compiler optimization field. Following the literature review, in Chap. 2, we provide a co-exploration approach using design space exploration technique for an embedded domain, namely VLIW. We show that this technique can speed up the performance of an application by using certain optimizations pass over our proposed VLIW micro-architecture. In Chap. 3, we present a novel machine learning approach to selecting the most promising compiler optimizations using Bayesian networks. This technique significantly improves application's performance against using fixed optimization available at GCC where our Bayesian network selects the most promising compiler passes. Chapters 4 and 5 are presenting our novel machine learning predictive models on how to tackle the phase-ordering problem. The former presents an intermediate approach, and the latter showcases a complete sequence speedup predictor on the very problem.

Finally, we present some concluding remarks and future works. Note that in this book, the bibliography is chapter-wise.

We hope this book brings the latest research done to a wide range of readers and promotes the use of machine learning on the field of compilation.

Toronto, ON, Canada

Milan, Italy

Newark, DE, USA

Milan, Italy

Amir H. Ashouri

Gianluca Palermo

John Cavazos

Cristina Silvano



<http://www.springer.com/978-3-319-71488-2>

Automatic Tuning of Compilers Using Machine Learning

Ashouri, A.H.; Palermo, G.; Cavazos, J.; Silvano, C.

2018, XVII, 118 p. 23 illus., 6 illus. in color., Softcover

ISBN: 978-3-319-71488-2