



Preface

About This Book

This book is a guide to getting the best performance out of computers running the Solaris operating system. The target audience is developers and software architects who are interested in using the tools that are available, as well as those who are interested in squeezing the last drop of performance out of the system.

The book caters to those who are new to performance analysis and optimization, as well as those who are experienced in the area. To do this, the book starts with an overview of processor fundamentals, before introducing the tools and getting into the details.

One of the things that distinguishes this book from others is that it is a practical guide. There are often two problems to overcome when doing development work. The first problem is knowing the tools that are available. This book is written to cover the breadth of tools available today and to introduce the common uses for them. The second problem is interpreting the output from the tools. This book includes many examples of tool use and explains their output.

One trap this book aims to avoid is that of explaining how to manually do the optimizations that the compiler performs automatically. The book's focus is on identifying the problems using appropriate tools and solving the problems using the easiest approach. Sometimes, the solution is to use different compiler flags so that a particular hot spot in the application is optimized away. Other times, the solution is to change the code because the compiler is unable to perform the optimization; I explain this with insight into why the compiler is unable to transform the code.

Goals and Assumptions

The goals of this book are as follows.

- Provide a comprehensive introduction to the components that influence processor performance.
- Introduce the tools that you can use for performance analysis and improvement, both those that ship with the operating system and those that ship with the compiler.
- Introduce the compiler and explain the optimizations that it supports to enable improved performance.
- Discuss the features of the SPARC and x64 families of processors and demonstrate how you can use these features to improve application performance.
- Talk about the possibilities of using multiple processors or threads to enable better performance, or more efficient use of computer resources.

The book assumes that the reader is comfortable with the C programming language. This language is used for most of the examples in the book. The book also assumes a willingness to learn some of the lower-level details about the processors and the instruction sets that the processors use. The book does not attempt to go into the details of processor internals, but it does introduce some of the features of modern processors that will have an effect on application performance.

The book assumes that the reader has access to the Sun Studio compiler and tools. These tools are available as free downloads. Most of the examples come from using Sun Studio 12, but any recent compiler version should yield similar results. The compiler is typically installed in `/opt/SUNWspro/bin/` and it is assumed that the compiler does appear on the reader's path.

The book focuses on Solaris 10. Many of the tools discussed are also available in prior versions. I note in the text when a tool has been introduced in a relatively recent version of Solaris.

Chapter Overview

Part I—Overview of the Processor

- Chapter 1—The Generic Processor
- Chapter 2—The SPARC Family
- Chapter 3—The x64 Family of Processors

Part II—Developer Tools

- Chapter 4—Informational Tools
- Chapter 5—Using the Compiler
- Chapter 6—Floating-Point Optimization
- Chapter 7—Libraries and Linking
- Chapter 8—Performance Profiling Tools
- Chapter 9—Correctness and Debug

Part III—Optimization

- Chapter 10—Performance Counter Metrics
- Chapter 11—Source Code Optimizations

Part IV—Threading and Throughput

- Chapter 12—Multicore, Multiprocess, Multithread

Part V—Concluding Remarks

- Chapter 13—Performance Analysis

Acknowledgments

A number of people contributed to the writing of this book. Ross Towle provided an early outline for the chapter on multithreaded programming and provided comments on the final version of that text. Joel Williamson read the early drafts a number of times and each time provided detailed comments and improvements. My colleagues Boris Ivanovski, Karsten Gutheridge, John Henning, Miriam Blatt, Linda Hsi, Peter Farkas, Greg Price, and Geetha Vallabhenini also read the drafts at various stages and suggested refinements to the text. A particular debt of thanks is due to John Henning, who provided many detailed improvements to the text.

I'm particularly grateful to domain experts who took the time to read various chapters and provide helpful feedback, including Rod Evans for his input on the linker, Chris Quenelle for his assistance with the debugger, Brian Whitney for contributing comments and the names of some useful tools for the section on tools, Brendan Gregg for his comments, Jian-Zhong Wang for reviewing the materials on compilers and source code optimizations, Alex Liu for providing detailed comments on the chapter on floating-point optimization, Marty Izkowitz for comments on the performance profiling and multithreading chapters, Yuan Lin, Ruud van der Pas,

Alfred Huang, and Nawal Copty for also providing comments on the chapter on multithreading, Josh Simmons for commenting on MPI, David Weaver for insights into the history of the SPARC processor, Richard Smith for reviewing the chapter on x64 processors, and Richard Friedman for comments throughout the text.

A number of people made a huge difference to the process of getting this book published, including Yvonne Prefontaine, Ahmed Zandi, and Ken Tracton. I'm particularly grateful for the help of Richard McDougall in guiding the project through the final stages.

Special thanks are due to the Prentice Hall staff, including editor Greg Doench and full-service production manager Julie Nahil. Thanks also to production project manager Dmitri Korzh from Techne Group.

Most importantly, I would like to thank my family for their support and encouragement. Jenny, whose calm influence and practical suggestions have helped me with the trickier aspects of the text; Aaron, whose great capacity for imaginatively solving even the most mundane of problems has inspired me along the way; Timothy, whose enthusiastic sharing of the enjoyment of life is always uplifting; and Emma, whose arrival as I completed this text has been a most wonderful gift.