Measuring computer performance

A practitioner's guide

Measuring computer performance sets out the fundamental techniques used in analyzing and understanding the performance of computer systems. Throughout the book, the emphasis is on practical methods of measurement, simulation and analytical modeling.

The author discusses performance metrics and provides detailed coverage of the strategies used in benchmark programs. He gives intuitive explanations of the key statistical tools needed to interpret measured performance data. He also describes the general 'design of experiments' technique, and shows how the maximum amount of information can be obtained for the minimum effort. The book closes with a chapter on the technique of queueing analysis.

Appendices listing common probability distributions and statistical tables are included, along with a glossary of important technical terms. This practically oriented book will be of great interest to anyone who wants a detailed, yet intuitive, understanding of computer systems performance analysis.

Measuring computer performance

A practitioner's guide

David J. Lilja

Department of Electrical and Computer Engineering
University of Minnesota, Minneapolis

CAMBRIDGE UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK http://www.cup.cam.ac.uk
40 West 20th Street, New York, NY 10011-4211, USA http://www.cup.org
10 Stamford Road, Oakleigh, Melbourne, 3166, Australia
Ruiz de Alarcón 13, 28014 Madrid, Spain

First published 2000

Printed in the United Kingdom at the University Press, Cambridge

Typeface Times 10.5/14pt by Keyword Typesetting Services Ltd, Wallington, Surrey

A catalogue record for this book is available from the British Library

Library of Congress Cataloguing in Publication data

Lilja, David J.
Measuring computer performance: a practitioner's guide / David J. Lilja.
p. cm
ISBN 0 521 64105 5
1. Electronic digital computers – Evaluation – Congresses. I. Title.
QA76.9.E94 L54 2000
004.2′4–dc21 99-057225 CIP

ISBN 0 521 64105 5 hardback

To Sarah
for her unwavering love, support,
and encouragement; and to Andrew,
for helping me to remember what's really important.

Contents

Preface

"Education is not to reform students or amuse them or to make them expert technicians. It is to unsettle their minds, widen their horizons, inflame their intellects, teach them to think straight, if possible."

Robert M. Hutchins

Goals

Most fields of science and engineering have well-defined tools and techniques for measuring and comparing phenomena of interest and for precisely communicating results. In the field of computer science and engineering, however, there is surprisingly little agreement on how to measure something as fundamental as the performance of a computer system. For example, the speed of an automobile can be readily measured in some standard units, such as meters traveled per second. The use of these standard units then allows the direct comparison of the speed of the automobile with that of an airplane, for instance. Comparing the performance of different computer systems has proven to be not so straightforward, however.

The problems begin with a lack of agreement in the field on even the seemingly simplest of ideas, such as the most appropriate metric to use to measure performance. Should this metric be MIPS, MFLOPS, QUIPS, or seconds, for instance? The problems then continue with many researchers obtaining and reporting results using questionable and even, in many cases, incorrect methodologies. Part of this lack of rigor in measuring and reporting performance results is due to the fact that tremendous advances have been made in the performance of computers in the past several decades using an *ad hoc* 'seat-of-the-pants' approach. Thus, there was little incentive for researchers to report results in a scientifically defensible way. Consequently, these researchers never taught their students sound scientific methodologies to use when conducting their own experiments.

One of the primary goals of this text is to teach the fundamental concepts behind the tools and techniques used in computer-systems performance analysis.

While specific programs or tools are sometimes mentioned as example implementations of the concepts presented, the goal is to teach the basic ideas, not the details of specific implementations. It is purposefully not a goal to make you an expert in using a specific software tool or system instrumentation package. My belief is that, if you understand the fundamental ideas behind these tools and techniques, including the basic assumptions inherent in the tool and in any statistical methods used to interpret the measured data, you can easily figure out how to use a specific tool or technique. True proficiency with individual tools can come only when you deeply understand the concepts on the basis of which they are developed.

So, if you want to learn how to use the latest and greatest version of the hottest simulation package, for instance, go buy its user's manual. If you want to develop a deeper understanding of the field of performance analysis, however, keep reading. When you then decide to use that exciting new software package, you will understand how to interpret its results in a meaningful way. Also you will, I hope, have developed some insight into the potential benefits and hazards that it may present.

Philosophy

The question of which topics to include in a text, and, perhaps more importantly, which topics to omit, has as many different answers as there are potential authors. So does the related question of how to treat each topic. Some authors would prefer to take a more mathematically rigorous approach to the topics than I have used in this text, perhaps, or they may choose to emphasize different aspects of the selected topics. Many, perhaps even most, texts on computer-systems performance analysis tend to focus on analytical modeling techniques from a rather mathematical point of view. While this is an important topic with an extensive body of literature, I chose in this text to instead emphasize practical tools and techniques that I have found to be useful in analyzing and evaluating the performance of a variety of different types of computer systems.

Furthermore, standard statistical tools, such as the analysis-of-variance (ANOVA) tests and the design-of-experiments concepts, have not found their way into the fabric of the computer-systems research community. This lack of statistical rigor has been a major weakness in the field of computer-systems performance-analysis. Consequently, I have chosen to present many practical statistical tools and techniques that are not commonly taught in a performance analysis course. While many of these ideas are presented in introductory probability and statistics courses, I have found that a surprisingly large number of computer engineering and computer science students never really learned these tools and techniques. Many, in fact, have never even been exposed to the ideas.

Instead of providing detailed mathematical derivations of these tools and techniques, though, I prefer to provide a more intuitive sense of why we might choose a certain type of statistical analysis, for instance. I want you to learn how to apply the appropriate tools and techniques so that you will become more thorough and careful in the experiments and analyses you perform. There are many other textbooks available if you then want to continue your studies into the deeper mathematical underpinnings of these tools and techniques.

Organization

The first chapter begins with an introduction to the basic ideas of measurement, simulation, and analytical modeling. It also describes some of the common goals of computer-systems performance analysis. The problem of choosing an appropriate metric of performance is discussed in Chapter 2, along with some basic definitions of speedup and relative change.

The next three chapters provide an intuitive development of several important statistical tools and techniques. Chapter 3 presents standard methods for quantifying average performance and variability. It also introduces the controversy surrounding the problem of deciding which of several definitions of the mean value is most appropriate for summarizing a set of measured values. The model of measurement errors developed in Chapter 4 is used to motivate the need for statistical confidence intervals. The ideas of accuracy, precision, and resolution of measurement tools are also presented in this chapter. Techniques for comparing various system alternatives in a statistically valid way are described in Chapter 5. This presentation includes an introduction to the analysis of variance (ANOVA), which is one of the fundamental statistical analysis techniques used in subsequent chapters.

While Chapters 3–5 focus on the use and interpretation of measured data, the next two chapters emphasize tools and techniques for actually obtaining these data. Chapter 6 begins with a discussion of the concept of events. It also describes several different types of measurement tools and techniques, including interval timers, basic block counting, execution-time sampling, and indirect measurement. The underlying ideas behind the development of benchmark programs are presented in Chapter 7, along with a brief description of several standard benchmark suites.

Chapter 8 uses a discussion of linear regression modeling to introduce the idea of developing a mathematical model of a system from measured data. Chapter 9 presents techniques for designing experiments to maximize the amount of information obtained while minimizing the amount of effort required to obtain this information.

Up to this point, the text has focused on gathering and analyzing measurements of existing computer systems. The fundamental problems involved in simulating systems are discussed in Chapter 10. The standard simulation strategies are presented along with a description of how the sequences of random numbers that are often needed to drive simulations can be generated. Finally, Chapter 11 concludes the text with a presentation of the fundamental analytical modeling techniques derived from queuing theory.

A glossary of some of the more important terms used in the text is presented in Appendix A. Several common probability distributions that are frequently used in simulation modeling are described in Appendix B. Appendix C tabulates critical values used in many of the statistical tests described in the earlier chapters.

While suggestions for further reading are provided at the end of each chapter, they by no means comprise an exhaustive bibliography of the field of computer-systems performance analysis. Rather, they are intended to guide the curious reader towards additional information that I have found to be both useful and interesting in my own research, in teaching my performance-analysis course, and in preparing this text.

Some exercises are provided at the end of each chapter to help the reader focus on some interesting aspects of the chapter's topic that are not covered in detail within the text. They also should help provide instructors using this book within a course with some ideas for further homework and project assignments.

Suggestions for using this text

This book is intended to be used as the primary text in a one-semester course for advanced undergraduate and beginning graduate students in computer science and engineering who need to understand how to rigorously measure the performance of computer systems. It should also prove useful as a supplemental text for students in other computer science and engineering courses who need to understand performance. It would make a good supplement for a course on high-performance computer architecture or high-speed computer networking, for instance.

This text will also be useful as a reference text for professional engineers and scientists who use computers in their daily work, or who design systems that incorporate computers as their primary control elements. Application experts from any discipline should also find this book useful in helping to understand how to analyze the performance of their systems and applications.

Acknowledgements

"The reason universities have students is so they can teach the professors."

John Wheeler, "No Ordinary Genius," p. 44.

I have had the pleasure of teaching a course on computer-systems performance analysis many times over the past several years. One of the advantages of this teaching has been the opportunity to practice many of the concepts and ideas presented in this text on (mostly) willing subjects. With that in mind, I would like to thank all of the students who have enrolled in this course over the years. Their insightful comments and feedback, both direct and indirect, helped me test out many of the following explanations and derivations. I especially would like to thank the students who enrolled in EE/CS 5863 in Spring quarter 1999. They gamefully waded their way through an early draft of this text as they tried to keep up with the course itself. Their comments and feedback on what did and did not work in the text are very much appreciated. Any remaining lack of clarity and errors are entirely my fault, however.

I also would like to thank Philip Meyler of Cambridge University Press for his comments on an early draft and for his efforts in helping transform the manuscript into an actual textbook; Steven Holt for his thorough and careful copy-editing; and all of the other individuals at the Press for their efforts in shepherding this book through the entire publication process.