

# **Computer Systems Performance Evaluation**

**DOMENICO FERRARI**

# Computer Systems Performance Evaluation

**DOMENICO FERRARI**

*Computer Science Division  
Department of Electrical Engineering  
and Computer Sciences  
University of California, Berkeley*

PRENTICE-HALL, INC., Englewood Cliffs, New Jersey 07632

*Library of Congress Cataloging in Publication Data*

Ferrari, Domenico (date)

Computer systems performance evaluation.

Bibliography: p.

Includes index.

1. Electronic digital computers—Evaluation.

I. Title.

QA76.9.E94F47 621.3819'58 77-15096

ISBN 0-13-165126-9

Computer Systems  
Performance Evaluation

*To Alessandra and Giuliarachele  
for their love, patience and support*

© 1978 by PRENTICE-HALL, INC.  
Englewood Cliffs, New Jersey 07632

All rights reserved. No part of this book  
may be reproduced in any form or  
by any means without permission in writing  
from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

PRENTICE-HALL INTERNATIONAL, INC., *London*  
PRENTICE-HALL OF AUSTRALIA PTY. LIMITED, *Sydney*  
PRENTICE-HALL OF CANADA, LTD., *Toronto*  
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*  
PRENTICE-HALL OF JAPAN, INC., *Tokyo*  
PRENTICE-HALL OF SOUTHEAST ASIA PTE. LTD., *Singapore*  
WHITEHALL BOOKS LIMITED, *Wellington, New Zealand*

# Preface

Computers are man-made machines. Thus, they belong to the realm of engineering. They are designed by engineers, according to certain cost-performance specifications, to satisfy the information-processing needs of individuals and organizations. The programs they execute are also designed by people who, when engaged in this endeavor, act as engineers.

An essential aspect of any engineering activity is the evaluation of the systems this activity is concerned with. Engineering systems are evaluated by their designers, manufacturers, buyers, managers, and users. One of the dimensions, and a very important one, along which systems are evaluated is the dimension of performance. This book deals with the evaluation of computer systems performance. Its subject is, or should be, of interest to the great majority of computer professionals. System designers, installation directors and staff members, data processing and corporate managers at all levels, systems analysts, program designers, and computer users all have daily to cope with problems whose solution may be made substantially easier and more satisfactory by some knowledge of performance evaluation methodologies, techniques, and tools.

All computer installations have to deal with problems which require a considerable involvement in performance-evaluation activities. Examples of such problems are procurement, configuration design, system tuning, upgrading, accounting and pricing, scheduling and operations management, and short- and long-term planning. Practically all organizations which make use of computer

systems must take performance into account in some of their decision-making processes. A system which does not perform as expected, or whose cost/performance ratio can be decreased, causes a waste of resources which must be avoided or eliminated.

The recognition of its practical importance and scope and the fact that it may be seen as the application of the scientific method to the study of computer systems make performance evaluation an attractive subject for an increasing number of researchers and teachers. However, the very fact that performance evaluation is being studied as a discipline distinct from system design is a symptom of an unsatisfactory situation. In other more traditional branches of engineering there are very few cases in which, as quite frequently happens with computer systems, the performance of a system can be improved even by primitive techniques by a very large factor after the system has been designed and implemented. Very few engineers specialize in, or even talk about, the evaluation of the performance of cars, planes, bridges, industrial processes, and electric circuits, since this is an essential aspect of any type of engineering design.

Computer systems engineering is not mature enough yet. It cannot base itself on quantitative laws similar to those which constitute the scientific foundations of other types of engineering: the laws of mechanics, electromagnetism, thermodynamics, and so on. Thus, performance evaluation problems should be studied with the ultimate objective of incorporating the results into design, implementation, and usage methodologies, that is, of making such a separate study meaningless.

In the long run it is reasonable to expect that the quantitative approach will permeate the core courses of computer science and engineering curricula, thereby making the offering of specific courses on performance evaluation unnecessary. However, reaching this point will require extensive and successful research, and even more important and difficult to achieve, a radical change of mentality. The distance between the present state of affairs and the goal mentioned above is perhaps best exemplified by the almost completely qualitative and descriptive approach that is currently taken to the teaching of computer system organization. In this situation it seems indispensable to offer courses entirely dedicated to performance evaluation. These courses should provide students with a new and very important viewpoint on computer systems, and with a picture of this rapidly growing field, its main themes, its problems, and the known approaches and solutions to them.

This book was born from one of such courses, which has been offered several times to Berkeley students. The level of the course, and therefore also the one of the book, has been chosen so as to be appropriate for computer science seniors and first-year graduate students. As for taking the course, the prerequisites for understanding the book are some familiarity with the material usually covered in an undergraduate two-quarter course on computer systems organization and/or operating systems, as well as with the principles and the basic techniques of computer programming and some elementary background in dis-

crete mathematics, statistics, and probability theory. Computer professionals will not have any problems with the first two prerequisites (computer systems and programming) and will find references in the bibliography to a few textbooks that adequately cover the mathematical and statistical material required to make good use of this book.

I have emphasized the conceptual aspects of performance evaluation techniques and problems that I believe should always be given maximum priority. However, I have not neglected those informative aspects that, in my opinion, significantly contribute to providing the reader with a comprehensive view of the state of the art both in research and in the commercial world. The resulting picture is, of course, profoundly subjective and biased. I have stressed those techniques, viewpoints, and approaches that I think will become more and more important in the near future, rather than trying to provide a faithful snapshot of the present situation. I have presented methodologies and tools borrowed from other fields as if they had been invented for evaluating computer systems, and discussed their characteristics only from this standpoint. I have committed a number of other major and minor sins which the hopefully forgiving reader will soon discover.

The unifying theme of the book is a very pragmatic one: the *evaluation study* seen as a set of procedures whose end goal is to gather information on the system being evaluated so as to be able to answer certain performance-related questions. Thus, the book is mostly concerned with the evaluation studies required to solve the main problems which arise in computer systems engineering and with the techniques to be used in these studies.

Chapter 1 introduces the subject of the book by describing the viewpoint from which computer systems will be regarded. It discusses the concept of performance, the most important performance indices, and the most popular classifications of evaluation problems and techniques.

The next three chapters are devoted to the major evaluation techniques and tools. Most of these techniques are well-known in other scientific and technical fields. Except for some minimal amount of background information, these chapters deal only with their applications to computer systems. Chapter 2 describes measurement studies, measurement tools, the design of experiments, and the application of empirical modeling methods to the problem of experimental results interpretation. Chapter 3 deals with the simulation of computer systems. Model formulation, construction, calibration and validation are discussed, as well as the design of simulation experiments and the interpretation of simulation results. Chapter 4 is devoted to analytic modeling. Both deterministic and probabilistic models of systems are treated, with a particular emphasis on the application of queuing network-modeling techniques to performance analysis.

Chapter 5 is concerned with a problem of crucial importance in any evaluation study, the one of work-load characterization. Our lack of knowledge in this area is probably the most serious obstacle to progress in performance evaluation. A separate chapter has been devoted to it since the basic problems in

work-load characterization are conceptually the same for all techniques. These problems, together with the known approaches to work-load modeling, are explored in this chapter.

Having been exposed to the techniques which constitute an essential part of any evaluation study, the reader should be ready to apply them to various types of evaluation problems. Three broad classes of problems are examined in the next three chapters. Chapter 6 describes the technical aspects of computer selection and discusses the adequacy of the known techniques and tools with respect to this problem. Chapter 7 deals with performance improvement, the area in which the most striking successes have been obtained, but whose importance is hopefully going to decrease as better system and installation design methodologies become available. Chapter 8 discusses the role of performance evaluation in computer system design. The techniques exposed in the first part of the book are applied there to various design problems.

Finally, Chapter 9 contains a discussion of program performance evaluation. The indices, techniques, and problems related to the performance aspects of programs are briefly described following the organizational scheme of the rest of the book.

Needless to say, this book may be correctly characterized as an introduction to performance evaluation. The reader will easily realize that each chapter might be seen as the introductory chapter, or an extended summary, of an entire book that could be written on the same subject. Besides being usable as a textbook in courses on computer systems performance evaluation or as a reference in courses on computer organization, computer systems design, and operating systems, the book, possibly supplemented with readings from periodical literature and other books, can serve as a textbook in specialized courses or short courses on measurement, modeling of computer systems, computer selection techniques, tuning techniques, and software evaluation.

Many individuals and organizations gave me substantial help in the conception and preparation of this book. The necessary background was provided by all my teachers. Among the most effective of them, I must include my parents, who taught me their philosophy of life, and my students at Berkeley, whose motivation has been a constant source of inspiration and encouragement. A number of my colleagues in the Department of Electrical Engineering and Computer Sciences at Berkeley are also to be thanked for their scientific and moral support and for the continuous challenge of their example. Several people have stimulated my work with their research and their interest in the book. I am really indebted to all the authors listed in the bibliography and to many other workers in the performance evaluation area. My particular gratitude goes to all the individuals who gave me valuable advice on the manuscript or some of its parts, especially to the reviewers, Harold Heath and Daniel Siewiorek, and to Steve Kimbleton, Frank Palermo and Felix Lam. Karl Karlstrom's encouragement and editorial support were extremely helpful throughout my effort. The typing of the manuscript was done with great care, rare patience, and excellent results by Ruth Suzuki, Edith Purser, and Mary

Ann Ratch. Finally, I would like to thank the birds, the squirrels, the trees, and the flowers of my garden in Berkeley, in whose company most of this book has been written.

The Regents of the University of California supported the preparation of this book by granting me a sabbatical quarter and a Regent's Summer Fellowship. The Universities of Pisa and Genoa and the Polytechnic Institute of Milan are also to be credited for their support. Some of the work reported in Chapter 2 was sponsored by the Advanced Research Projects Agency under contract DAHC15-70-C-0274. Some of the material reported in Chapter 9 was supported by the Division of Computing Research of the National Science Foundation under grant DCR74-18375 and the Research Division of the IBM Corporation.

For all the errors and omissions, I am indebted to no one but myself.

DOMENICO FERRARI



# Contents

## List of Examples

xi

## Preface

xiv

## 1

### The Performance Evaluation Viewpoint

1

- 1.1 Introduction 1
  - 1.2 The Objects of Evaluation Studies 3
  - 1.3 The Phases of an Evaluation Study 9
  - 1.4 Performance Indices 11
  - 1.5 A Classification of Evaluation Studies 17
  - 1.6 An Overview of Evaluation Techniques 19
- Problems 23

## 2

### Measurement Techniques

26

- 2.1 Measurement Studies 26
- 2.2 An Introduction to Measurement Tools 29

- 2.3 Hardware Tools 32
  - 2.3.1 Introduction 32
  - 2.3.2 Fixed Hardware Tools 33
  - 2.3.3 Wired-Program Tools 34
  - 2.3.4 Stored-Program Tools 40
- 2.4 Software and Firmware Tools 44
  - 2.4.1 Introduction 44
  - 2.4.2 Internally Driven Tools 47
  - 2.4.3 Externally Driven (Sampling) Tools 56
- 2.5 Instrumentation Systems 64
- 2.6 The Design of Measurement Experiments 66
  - 2.6.1 Problem Definition 66
  - 2.6.2 The Identification of the Factors 67
  - 2.6.3 The Selection of the Levels for Each Session 69
  - 2.6.4 The Time and Duration of a Session 73
- 2.7 The Interpretation of Measurement Results 80
  - 2.7.1 Data Reduction and Presentation Techniques 80
  - 2.7.2 The Analysis of Variance 82
  - 2.7.3 Regression Analysis 89
- Problems 96

### 3

#### Simulation Techniques

- 3.1 Simulation Studies 100
- 3.2 Model Formulation and Construction 108
  - 3.2.1 The Basic Problems 108
  - 3.2.2 The Structure of a Simulator 110
  - 3.2.3 Alternative Organizations 120
  - 3.2.4 The Simulation of the Work Load 123
  - 3.2.5 The Implementation of a Simulator 132
- 3.3 Model Calibration and Validation 134
  - 3.3.1 Problem Definition 134
  - 3.3.2 The Calibration of a Simulator 139
  - 3.3.3 A Calibration Methodology 141
- 3.4 The Design of Simulation Experiments 145
  - 3.4.1 Experimental Design in Simulation 145
  - 3.4.2 The Duration of Simulation Runs 147
- 3.5 The Interpretation of Simulation Results 155
  - Problems 156

<b>4</b>		<b>160</b>
	<b>Analytic Techniques</b>	
4.1	Analytic Studies 160	
4.2	Deterministic Models 163	
4.2.1	Introduction 163	
4.2.2	A Model of CPU-I/O Overlap 165	
4.2.3	A Mean-Value Model of a Multiprogramming System 170	
4.3	Probabilistic Models 173	
4.3.1	Markov Models and Queuing Models 173	
4.3.2	Nonpreemptive Single-Service-Center Models 181	
4.3.3	Preemptive Single-Service-Center Models 191	
4.3.4	Open-Network Models 196	
4.3.5	Closed-Network Models 202	
4.4	Analytic Model Calibration and Validation 214	
	Problems 217	
<b>5</b>		<b>221</b>
	<b>Work-Load Characterization</b>	
5.1	Introduction 221	
5.2	The Formulation of a Work-Load Model 225	
5.2.1	Approaches to Work-Load Characterization 225	
5.2.2	The Problem of Representativeness 228	
5.2.3	The Problem of System-Independent Characterization 232	
5.3	The Construction of Work-Load Models 237	
5.3.1	Basic Definitions 237	
5.3.2	Natural Work-Load Models 238	
5.3.3	Nonexecutable Artificial Models 241	
5.3.4	Executable Artificial Models 245	
5.3.5	The Design of an Executable Artificial Model 255	
5.4	Work-Load Model Calibration and Validation 270	
	Problems 273	
<b>6</b>		<b>276</b>
	<b>Performance Evaluation in Selection Problems</b>	
6.1	Selection Problems 276	
6.2	Selection Methodologies 279	
6.2.1	A Competitive Selection Procedure 279	
6.2.2	The Role of Performance Evaluation in Selection Procedures 282	

- 6.3 Performance Comparisons 285
  - 6.3.1 The Comparison Problem 285
  - 6.3.2 Processing Mode Comparisons 289
  - 6.3.3 Vendor Comparisons 293
  - 6.3.4 Installation Comparisons 307
  - 6.3.5 System Component Comparisons 313
- 6.4 Evaluation Techniques for Selection Studies 315
- 6.5 Work-Load Characterization for Selection Studies 316
  - Problems 328

## 7

### Performance Evaluation in Improvement Problems

332

- 7.1 Improvement Problems 332
- 7.2 Improvement Methodologies 334
- 7.3 Diagnosis 341
  - 7.3.1 The Basic Problem 341
  - 7.3.2 An Analytic Study of Bottlenecks 349
  - 7.3.3 Bottleneck Detection 357
  - 7.3.4 A Statistical Diagnosis Method 367
- 7.4 Therapy 369
  - 7.4.1 Tuning Therapies 369
  - 7.4.2 Self-Tuning Mechanisms 379
  - 7.4.3 Upgrading Therapies 385
- 7.5 Evaluation Techniques in Improvement Studies 390
- 7.6 Work-Load Characterization in Improvement Studies 391
  - Problems 392

## 8

### Performance Evaluation in Design Problems

396

- 8.1 Design Problems 396
- 8.2 Design Methodologies 400
- 8.3 Performance Considerations in Iterative Design 403
  - 8.3.1 Analytic Techniques for the Design of an Initial Configuration 403
  - 8.3.2 The Design of the Memory Hierarchy 411
  - 8.3.3 Exploring Design Trade-offs 424
- 8.4 Performance Considerations in Hierarchical Design 431
- 8.5 Evaluation Techniques for Design Studies 442
- 8.6 Work-Load Characterization for Design Studies 444
  - Problems 447

<b>9</b>		
<b>The Evaluation of Program Performance</b>		<b>451</b>
9.1	Introduction	451
9.2	Program Performance Indices	453
	9.2.1 Time and Space	453
	9.2.2 Indices of Program Behavior	462
9.3	Program Evaluation Techniques	472
	9.3.1 Program Measurement	472
	9.3.2 Program Modeling	482
9.4	Program Evaluation Problems	508
	9.4.1 Program Selection	508
	9.4.2 Program Improvement	510
	9.4.3 Program Design	522
	Problems	525
<b>Bibliography</b>		<b>530</b>
<b>Index</b>		<b>541</b>

# List of Examples

## **Chapter 1**

- 1.1 Definition and Calculation of Performance Indices for an Interactive System 13
- 1.2 Repetition of the Calculations of Example 1.1 with a Different Time Quantum 16
- 1.3 Analytic Versus Simulation Models 21

## **Chapter 2**

- 2.1 Measurement of the Mean Throughput Rate of a Batch-Processing System by Fixed Hardware Tools 33
- 2.2 Measurement of Utilizations and Overlap Factors by a Wired-Program Hardware Tool 37
- 2.3 Measurement of the Utilizations of Software Resources by a Stored-Program Hardware Tool 41
- 2.4 Measurement of the Distribution of Interactive Command Types by an Internally Driven, Checkpoint-Based Software Tool 48
- 2.5 The Use of an Interactive Software Measurement Tool (the SMT) to Instrument a Program 54

- 2.6 Measurement of Utilizations by a Sampling Tool 60
- 2.7 Identification of the Factors for a Paging Experiment 68
- 2.8 Layout and Level Selection for a Paging Experiment 72
- 2.9 Determining Session Duration by the Method of Subsamples for an Experiment to Evaluate Interactive Scheduling Algorithms 77
- 2.10 Analysis of Variance Applied to the Results of a Factorial Paging Experiment 82
- 2.11 Regression Analysis Applied to the Estimation of the Effects of Operating System Changes on an Interactive System's Performance 89

**Chapter 3**

- 3.1 Simulation Model for Determining the Execution Time of a Program in a Uniprogramming System 102
- 3.2 Construction of a Simulator to Evaluate the Effects of Hardware Changes on the Performance of a Batch-Processing System 110
- 3.3 Work-Load Simulator for Example 3.1 125
- 3.4 Work-Load Simulator for Example 3.2 129
- 3.5 The Calibration of the Simulator Designed in Examples 3.2 and 3.4 139
- 3.6 Design of a Factorial Experiment with the Simulator Formulated in Example 3.2 146
- 3.7 Determining the Durations of the Runs for the Experiment Designed in Example 3.6 153

**Chapter 4**

- 4.1 A Deterministic Model of a Uniprogramming System for the Evaluation of Various Overlap Schemes 165
- 4.2 A Deterministic Mean-Value Model of a Multiprogramming System 170
- 4.3 A Discrete-Time Markov Model of the CPU of a Multiprogramming System 176
- 4.4 An  $M/M/1$  Queuing Model of a Uniprogramming Interactive System with Infinite Time Quantum 186
- 4.5 An  $M/G/1$  Queuing Model of the Uniprogramming Interactive System of Example 4.4 191
- 4.6 A Preemptive Single-Service-Center Model of a Uniprogramming Interactive Time-Sharing System with Round-Robin Scheduling 194
- 4.7 A Single-Service-Center Processor-Sharing Model of a Time-Sharing System 195
- 4.8 Formulation and Solution of a Markovian Open-Network Queuing Model for a Multiprogramming System 197

- 4.9 Formulation and Solution of a Markovian Closed-Network Queuing Model for a Multiprogramming System 203
- 4.10 A Closed-Network Model of an Interactive System 210

**Chapter 5**

- 5.1 Work-Load Model Driving the Batch-Processing System Simulator Formulated in Example 3.2 225
- 5.2 The Work-Load Model Which Drives the Central-Server Model of a Multiprogramming System Discussed in Example 4.9 227
- 5.3 Work-Load Equivalence for the Central-Server Model of Example 4.9 229
- 5.4 Adapting a Trace Model of a Job to Changes in the Job's Behavior 243
- 5.5 Use of an Instruction Mix to Compare the Mean Instruction Execution Rates of Two CPU's 248
- 5.6 Driving an Interactive System by an Executable Trace 250
- 5.7 Design of an Executable Artificial Work-Load Model that Reproduces the Joint Probability Distribution of Job Parameters 259
- 5.8 Design of a Simple Synthetic Job 264
- 5.9 The Relationships Between Arguments and Behavior for the Synthetic Job Designed in Example 5.8 272

**Chapter 6**

- 6.1 A Simple Comparison Between Centralized and Decentralized Solutions in Information-Processing System Design 296
- 6.2 Benchmark-Based Selection of a Vendor for a General-Purpose Installation 302
- 6.3 The Selection of an Installation for a Given Set of Application Programs 309
- 6.4 Work-Load Characterization for Comparisons of CPU Speeds 319
- 6.5 Work-Load Characterization for a Comparison of Interactive Operating Systems 321
- 6.6 Forecasting the Work Load of a Batch-Processing Computer Center 325

**Chapter 7**

- 7.1 Throughput-Rate Bottlenecks in a Central-Server Model of a Batch-Processing System 343
- 7.2 Turnaround-Time Bottlenecks in a Central-Server Model of a Batch-Processing System 348



- 7.3 The Thrashing Curve of a Virtual-Memory System and Its Relationship to Bottlenecks 352
- 7.4 Diagnosis of a Single-Channel Batch System Based on a System Utilization Profile or a Kiviat Graph 360
- 7.5 Diagnosis of the System Studied in Example 7.1 Based on a System Utilization Profile or a Kiviat Graph 362
- 7.6 Diagnosis of a Two-Channel Batch System Based on a System Utilization Profile or a Kiviat Graph 364
- 7.7 Bottleneck Detection in an Interactive System by Simulation 366
- 7.8 Predicting the Effects of a Tuning Therapy on the System Studied in Example 7.6 by Using a System Utilization Profile 371
- 7.9 Improvement Predictions Based on a Central-Server Model of a System to Which a Tuning Therapy is to be Applied 372
- 7.10 The Effects of Changes in the Memory Policy or in Program Behavior on the Performance of a Virtual-Memory System 374
- 7.11 Choosing the Best Upgrading Strategy for a Batch-Processing System Modeled As a Central-Server Queuing Network 387
- 7.12 Predicting the Effects of Upgrading Therapies on the Performance of the System Dealt with in Example 7.6 388
- 7.13 An Upgrading Study for the System Analyzed in Example 7.9 389

## **Chapter 8**

- 8.1 A Design Procedure for the Initial Configuration of a Medium-Small Batch System 404
- 8.2 The Design of an Initial Configuration for a Virtual-Memory Version of the System Dealt with in Example 8.1 408
- 8.3 Initial Design of an Interactive System 410
- 8.4 The Design of the Memory Hierarchy for the Batch-Processing System of Example 8.1 414
- 8.5 The Design of the Memory Hierarchy for the Virtual-Memory System of Example 8.2 416
- 8.6 Introducing a Cache Memory into the Virtual-Memory System Configuration Designed in Example 8.2 418
- 8.7 The Calculation of Primary Memory Request Rates in the Systems Designed in Examples 8.1 and 8.6 422
- 8.8 The Evaluation of Trade-offs in the Design of a Computer Graphics System 425
- 8.9 A Simulator-Based Hierarchical Procedure for the Design of a Multiprocessor System 434