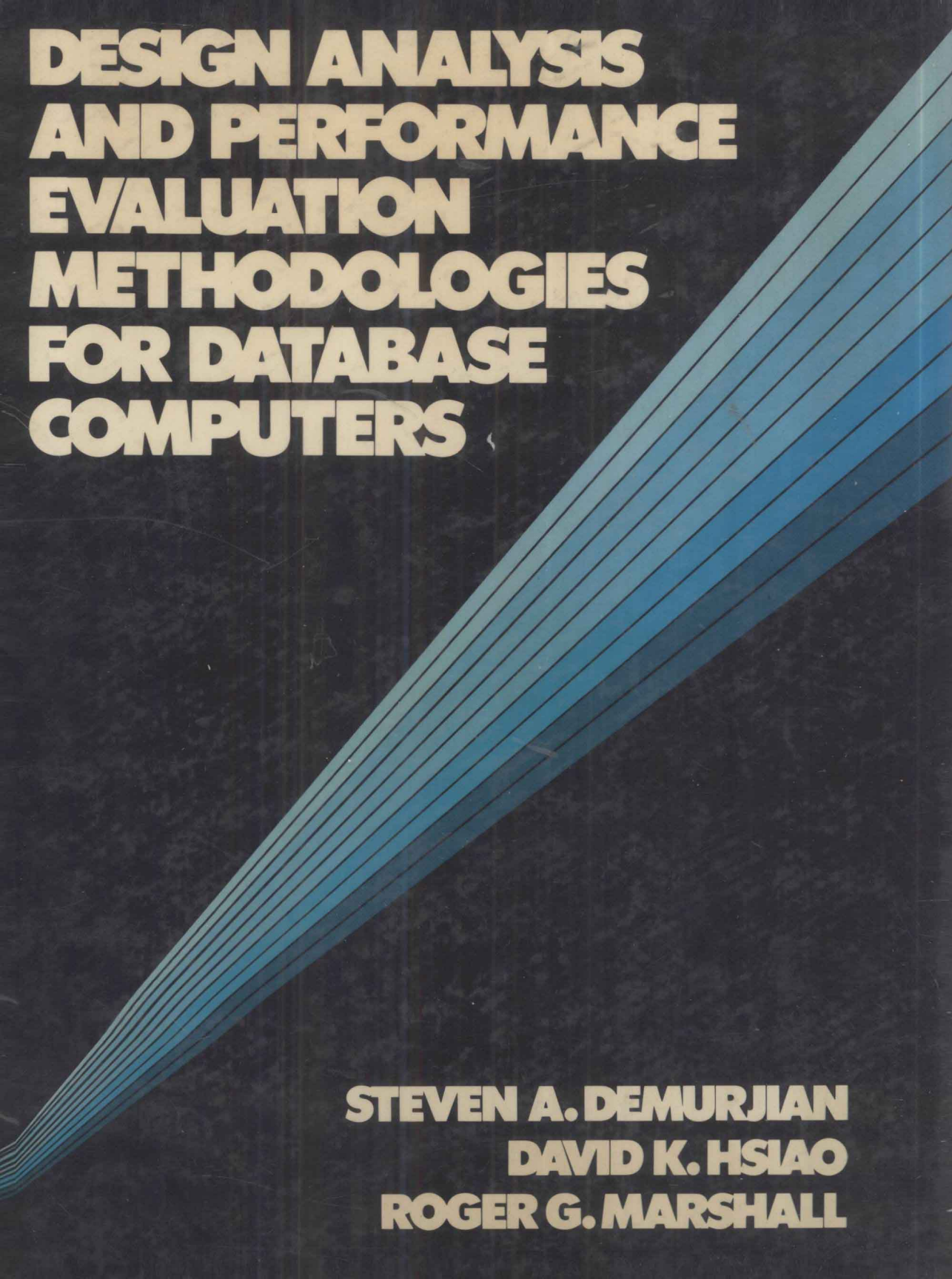
DESIGN ANALYSIS AND PERFORMANCE EVALUATION METHODOLOGIES FOR DATABASE COMPUTERS



STEVEN A. DEMURJIAN
DAVID K. HSIAO
ROGER G. MARSHALL

DESIGN ANALYSIS AND PERFORMANCE EVALUATION METHODOLOGIES FOR DATABASE COMPUTERS

STEVEN A. DEMURJIAN
Department of Computer and Information Sciences
The Ohio State University

DAVID K. HSIAO
Department of Computer Science
Naval Postgraduate School

ROGER G. MARSHALL
Department of Computer Science
Loyola College

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

Library of Congress Catalog Number: 87-60025

Editorial/production supervision: Barbara Marttine Webber
Cover design: Jeannette Jacobs
Manufacturing buyer: S. Gordon Osbourne

The publisher offers discounts on this book when ordered in bulk quantities. For more information, write:

Special Sales/College Marketing
Prentice-Hall, Inc.
College Technical and Reference Division
Englewood Cliffs, NJ 07632

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-199902-8 025

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall Hispanoamericana, S.A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Prentice-Hall of Southeast Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

DESIGN ANALYSIS AND PERFORMANCE EVALUATION METHODOLOGIES FOR DATABASE COMPUTERS

This book is dedicated to Professor Hsiao's former students and colleagues.

FOREWORD

The explosive growth in the use of computers in all parts of society has caused a very rapid increase in the use of huge databases to track vital information in business, government, and academia. The need to rapidly manipulate very large amounts of data of different types has created the need for a special class of machines to manage the high volume of work that must be done in a very short time to provide meaningful query response. More important, there is also a need for design and evaluation methodologies for these database machines. Such methodologies are becoming generally available for general-purpose computers, but so far have not been developed for database machines. Until now, the design and evaluation of database machines has been an art. This book, by describing methodologies for design analysis and performance evaluation, including the application of quantitative principles, moves the field away from art and toward engineering science. The methodologies will help us achieve more effective database machines upon which we will become increasingly dependent in this information age.

Database computers are radically different from general-purpose machines. They are uniquely tailored to manipulate large volumes of information and to analyze the relationships between pieces of information in the database. They must handle extremely high volumes of transactions between the processor and the data stored on external devices and at the same time be efficient at the computation-intensive tasks involving analysis and manipulation of the relationships in the database. Tailoring the hardware to handle these different tasks and designing software to take maximum advantage of the machine and provide satisfactory response to the user present a unique set of challenges to the designer, who must understand the models used to describe databases and the model-based languages used to develop software for database systems. This book

assists the designer and evaluator of these unique systems by providing quantitative methodologies for design analysis and performance evaluation.

This book presents five categories of methodologies for the design and evaluation of database computers. These are (1) closed (tasks neither enter nor leave the system) and open (jobs enter and leave the system) queueing network models, (2) complexity analyses based on time (where we are concerned with the time for an algorithm to be carried out) and space (where we are concerned about memory or other resources required for an algorithm), (3) simulations, (4) benchmarking, and (5) checkpointing. Both analytic and experimental approaches are covered.

These methodologies are applicable to all of the stages of the life cycle of database computers. They may be applied during design for analysis of alternative approaches and estimation of performance. They may be applied during implementation. For example, checkpoints can assist in performance evaluation. They are useful at the completion stage for verifying the decisions made during the design process and for evaluating the capabilities of the finished computer.

This book makes a unique contribution to our understanding of these complex computers by applying the methodologies developed on two well-documented database computers. One of these is the much published DBC machine, and the other is a carefully prototyped software system called the multi-backend database system (MBDS). Other database computers are also used where appropriate to demonstrate the methodologies.

This book is of value to the entire computing community. It brings engineering and scientific principles to bear on our ever increasing computing requirements. The processing and data-handling requirements of our future systems will not be satisfied by advances in circuit technology alone. Advances in architecture that provide real-time access to large databases and real-time resolution of queries to those databases, and the ability to analyze new architectures are critical to meeting future system needs.

Dr. Edward Lieblein
Director of Computer Software and Systems
Office of the Under Secretary for
Research and Engineering
Department of Defense

PREFACE

This book is based on the outline of a paper of the same title. The paper was recently published as a chapter in the 25th anniversary volume of *Advances in Computers* (editor, Marshall Yovits; Academic Press, 1986). In the course of the development of that paper, we discovered that we had covered a wide range of design-analysis and performance-evaluation methodologies. We had also covered a decade of our work and experience in the application of these methodologies. Due to the limited space in the chapter of that volume, we had to exclude considerable material. We felt that the exclusion gave the reader a narrow impression of our work and application. Further, it did not allow us to relate our complete work and entire experience to the reader. Therefore, using the outline of the paper, we decided to provide a comprehensive and thorough presentation of our work and experience in design analysis and performance evaluation of database computers.

This book nearly triples the page count and the material of the original paper. It also provides more background material on mathematical elements of the methodologies, more elaboration on the methodologies, and more applications of the methodologies. Some of the methodologies and applications are new and recent. Most of the applications have not been published in the literature, since innovative utilizations of the methodologies do not lend themselves easily to scholarly publications. However, we believe that the utilization of the methodologies is at least as important as the methodologies themselves. In this book we have made contributions to the discovery of new methodologies as well as to the application of both the new and old methodologies.

One of the most important contributions of this book is to relate the methodologies and applications to the database computers. Such an effort can benefit the database computer community for the years to come.

As a new discipline, the database computer has recently prompted many designs, prototypes, and products. We need to have a wide range of methodologies for its evaluation. Also, because database computers are new and different, we need to have concrete experiences in the application of these methodologies. This book addresses both of these needs.

Steven A. Demurjian
David K. Hsiao
Roger G. Marshall

Monterey, California

ACKNOWLEDGMENTS

The research on the database machine, DBC, was carried on from 1976 through 1980. The concept of DBC and the research on DBC were initiated by D. K. Hsiao, advanced by R. I. Baum (now Dr. Baum of IBM Poughkeepsie Lab), expanded by K. Kannan (now Dr. Kannan of IBM Thomas J. Watson Research Center), continued by J. Banerjee (now Dr. Banerjee of MCC in Austin, Texas), and completed by J. Menon (now Dr. Menon of IBM San Jose Research Lab) under the supervision of D. K. Hsiao. The reference articles (Banerjee, et al., 1978; Banerjee, et al., 1979) may serve as an overview of the organization of DBC. All of the research on DBC was conducted at the Ohio State University (OSU) with support from the Office of Naval Research (ONR).

The research and development of the multi-backend database system, MBDS, with over 20,000 lines of C code, has involved many individuals and organizations. We would like to acknowledge their support, contribution, and roles. The original idea of the software multiple-backend approach to database management for performance gains, capacity growth, and hardware upgrade was due to D. K. Hsiao in his equipment proposal for MBDS to Digital Equipment Corporation (DEC) at the beginning of 1980. The design and analysis of MBDS which began in 1980 was due to J. Menon as part of his doctoral dissertation (Menon, 1981). The research and development on MBDS has been supported by ONR from the beginning. The three interim computers (namely, one VAX-11/780 and two PDP-11/44s), the communications buses (PCLs), the disk drives (RM02s), the terminals (VT100s), the line printer, the tape drive, and other peripherals for MBDS were funded by DEC and ONR. OSU provided an air-conditioned room for the equipment. The detailed specification and implementation of MBDS were due mainly to A. Orooji as part of his study of modern software engineering techniques and their applications to database-systems design and implementation. The implementation work

began in 1981 and was supervised jointly by D. K. Hsiao and D. S. Kerr until 1982 and by D. S. Kerr and A. Orooji until 1983. The design and implementation efforts are documented in a series of reports (Kerr, et al., 1982; He, et al., 1982; Boyne, et al., 1983a; Demurjian, et al., 1984). A larger number of implementors are acknowledged therein.

In 1983, MBDS was moved from OSU to the Naval Postgraduate School (NPS), along with all of the hardware except the VAX-11/780 and a terminal, which remained at OSU. NPS provided the moving expenses and a VAX-11/780 for MBDS. D. S. Kerr spent the entire academic year of 1983 at NPS debugging MBDS with the help of S. Demurjian. Currently, MBDS is being utilized as a research vehicle for the study of multi-lingual and benchmarking issues (Demurjian and Hsiao, 1985d; Demurjian and Hsiao, 1987). The new hardware and the multi-lingual and benchmarking work are supported by the Department of Defense STARS program.

CONTENTS

LIST OF FIGURES xv

LIST OF TABLES xvii

FOREWORD xxi

PREFACE xxiii

ACKNOWLEDGMENTS xxv

1 INTRODUCTION 1

1.1. The Need of Methodologies for the Design Analysis and Performance Evaluation of Database Computers *2*
1.2. The Differences and Uniqueness of the Application of the Methodologies to the Database Computers *3*
1.3. The Organization of the Remaining Chapters *4*

2 BACKGROUND MATERIAL ON TEST BEDS AND TEST ELEMENTS 7

2.1. The First Test Bed -- The Architecture of a Software Database System, MBDS *7*
2.1.1. The Data Structures and the Primary Operations *9*
2.1.1.1. The Database Structure *9*
2.1.1.2. The Directory Structure *11*
2.1.1.3. The Primary Database Operations *13*
2.1.2. The Placement of Data *15*
2.1.2.1. The Directory Placement *15*

2.1.2.2. The Database Placement *16*
2.1.3. The Process Structure *17*
2.1.3.1. The Test-Interface Process *17*
2.1.3.2. The Processes of the Controller *17*
2.1.3.3. The Processes of Each Backend *19*
2.2. The Second Test Bed -- The Organization of a Hardware Database Machine, DBC *20*
2.2.1. The Major Components of DBC *20*
2.2.2. The Command Execution of DBC *23*
2.3. Elements of the Testing Theory *25*
2.3.1. Probability Density and Distribution Functions *25*
2.3.2. Continuous versus Discrete Distributions *26*
2.3.3. Examples of Continuous Distributions *27*
2.3.4. Examples of Discrete Distributions *28*
2.3.5. Application Criteria *30*
2.3.6. The Constant Distribution *31*
2.3.7. Concluding Remarks *31*

3 QUEUEING THEORY AS A DESIGN-ANALYSIS METHODOLOGY 33

3.1. An Introduction to Queueing Network Models *33*
3.1.1. Basic Server Models *35*
3.1.2. A Characterization of Queueing Models *42*
3.1.3. Birth and Death Processes *43*
3.1.4. The Quasi-Random Input *48*
3.1.5. The Single-Server Queueing Model *49*
3.1.5.1. The M/M/1/D Model *50*
3.1.5.2. The M/M/1/W Model *52*
3.1.6. The Multiple-Server Queueing Model *54*
3.1.6.1. More on Quasi-Random Input *55*
3.1.6.1.1. Non-Queueing of Arrivals *55*
3.1.6.1.2. The Queueing of Requests *57*
3.2. A Closed Queueing Network Model for the Design Analysis of a Database System Element *58*
3.2.1. The Eight Directory Management Strategies *59*
3.2.2. A Closed Queueing Network Model for Evaluating the Directory Management Strategies *61*
3.2.2.1. The Model for Strategy A *66*

3.2.2.2. The Model for Strategy B *67*
3.2.2.3. The Model for Strategy C *67*
3.2.2.4. The Model for Strategy D *68*
3.2.2.5. The Model for Strategy E *69*
3.2.2.6. The Model for Strategy F *70*
3.2.2.7. The Model for Strategy G *70*
3.2.2.8. The Model for Strategy H *71*
3.2.3. The Closed Queueing Network Results for the Directory Management Strategies *71*
3.2.4. Conclusions and Observations *74*
3.3. An Open Queueing Network Model for the Design Analysis of a Database Machine Component *75*
3.3.1. The Join Hardware and Its Operation *76*
3.3.2. Analyzing the Join Hardware of the DBC *78*
3.3.2.1. An Analysis of the A-Memory *78*
3.3.2.2. An Open Queueing Network Model for the B-Memory *80*
3.3.3. The Open Queueing Network Results for the Join Hardware of the DBC *81*
3.3.3.1. The Results of the B-Memory Analysis *82*
3.3.3.2. The Results of the Integrated A-Memory/B-Memory Analysis *84*
3.3.4. Conclusions and Observations *85*

4 TIME COMPLEXITY AS A DESIGN-ANALYSIS METHODOLOGY 87

4.1. An Introduction to Time Complexity *88*
4.2. The Application of Time Complexity for the Design Analysis of a Database System Element *89*
4.2.1. The Time Complexities of the Directory Management Strategies *90*
4.2.2. The Time-Complexity Results for the Directory Management Strategies *94*
4.2.3. Conclusions and Observations *96*
4.3. The Application of Time Complexity for the Implementation Alternatives of a Database System Element *96*
4.3.1. The RETRIEVE-COMMON Operation *97*
4.3.2. Design Alternatives for

the RETRIEVE-COMMON Operation *97*
4.3.3. Implementation Alternatives for the RETRIEVE-COMMON Operation *98*
4.3.3.1. A Straightforward Merge *99*
4.3.3.2. A Sort-Match Merge *99*
4.3.3.3. A Bucket-Hashing Merge *100*
4.3.4. The Time-Complexity Analyses of the Implementation Alternatives *101*
4.3.4.1. Analysis of the Straightforward Merge *102*
4.3.4.2. Analysis of the Sort-Match Merge *103*
4.3.4.3. Analysis of the Bucket-Hashing Merge *104*
4.3.5. Conclusions and Observations *106*
4.4. The Application of Time Complexity for the Design Analysis of a Database Machine Component *106*
4.4.1. The Time Complexity of the DBC Join Operation *107*
4.4.2. Comparing Join Algorithms *109*
4.4.3. Conclusions and Observations *111*

5 A COMBINED METHODOLOGY OF QUEUEING THEORY AND TIME COMPLEXITY 113

5.1. Combining Queueing Theory and Time Complexity *113*
5.1.1. Time-Complexity Expressions for the Single-Server Model *114*
5.1.2. Waiting-Time Expressions for the Multiple-Server Model *117*
5.1.2.1. The Poisson Input *117*
5.1.2.2. The Quasi-Random Input *118*
5.1.3. A Simplified Queueing Network Model for the Multi-Backend Database System (MBDS) *120*
5.2. The Application of Single- and Multiple-Server Queueing Network Models for the Design Analysis of a Database Server for Office Automation *129*
5.2.1. The Characterization and Qualification of a Database Server *130*
5.2.2. The Integration of MBDS into the OAE *136*
5.2.2.1. Five Approaches to the Integration of a Database Server *136*
5.2.2.2. An Analysis of the Five Designs:

A Heuristic Approach *139*
5.2.3. A Formal Analysis of the Five Integration Designs *140*
5.2.3.1. The Queueing Equations *141*
5.2.3.2. A Queueing Analysis of the Five Designs *143*
5.2.3.3. The Total-Waiting-Time Equations for the Five Configurations *144*
5.2.4. A Sensitivity Analysis of the Queueing Expressions *147*
5.2.5. Conclusions and Observations *156*

6 SIMULATION AS A PERFORMANCE-ESTIMATION METHODOLOGY 159

6.1. Simulation Techniques, Languages, and Programs *159*
6.2. The Application of Simulations for the Study of a Database System's Performance *163*
6.2.1. A Simulation Model for the Database System MBDS *164*
6.2.2. The Simulation Results for the MBDS *168*
6.2.3. Conclusions and Observations *174*
6.3. The Application of Simulations for the Study of a Database Machine's Performance *175*
6.3.1. A Simulation Model of the Database Machine DBC *176*
6.3.1.1. The Structure-Loop Simulation Model *176*
6.3.1.2. The Data-Loop Simulation Model *179*
6.3.2. The Simulation Results for the DBC *183*
6.3.2.1. The Structure-Loop Simulation Results *183*
6.3.2.2. The Data-Loop Simulation Results *185*
6.3.3. Conclusions and Observations *187*

7 BENCHMARKING AS A PERFORMANCE-EVALUATION METHODOLOGY 189

7.1. Benchmarking Considerations for Database Computers *190*
7.1.1. A Strategy for Benchmark Specification *190*
7.1.2. Relative Performance-Evaluation Techniques *193*
7.1.3. Absolute Performance-Evaluation Techniques *194*
7.1.4. A Combination of the Two Techniques *194*
7.2. A Benchmarking Methodology for Multiple-Backend Database Computers *195*
7.2.1. The Measures (Benchmarks) of the

Multiple-Backend Database Computer *196*
7.2.1.1. A Measure (Benchmark) on Performance Gains *196*
7.2.1.2. A Measure (Benchmark) on Capacity Growth *199*
7.2.1.3. The Test (Benchmark) Transactions and the Test (Benchmark) Databases *201*
7.2.2. The Design and Generation of Test (Benchmark) Databases *202*
7.2.2.1. The Concept of Database-Size Multiples *202*
7.2.2.2. The Determination of the Possible Test Configurations *203*
7.2.2.3. The Choice of Database Sizes and Record Sizes *204*
7.2.2.4. The Calculation of the Database-Size Multiple *206*
7.2.2.5. The Consideration of Database Formats *207*
7.2.2.6. The Formation of Partitions (Clusters) *211*
7.2.2.7. The Definition of Record Templates and Directory Data *221*
7.2.2.8. A Summary of the Test-Database Methodology *228*
7.2.3. The Design and Generation of Test-Transaction Mixes *230*
7.2.3.1. The Emphasis on Generic, Primary Database Operations *231*
7.2.3.2. The Test-Transaction Mixes *233*
7.2.3.2.1. Data-Intensive and Overhead-Intensive Retrievals *233*
7.2.3.2.2. Simple, Data-Intensive Updates *234*
7.2.3.2.3. Data-Intensive and Overhead-Intensive Common Retrievals *236*
7.2.3.2.4. Simple and Complex Inserts *238*
7.2.3.2.5. Overhead-Intensive and Data-Intensive Deletes *238*
7.2.3.2.6. Complex Data-Intensive and Overhead-Intensive Updates *240*
7.2.3.2.7. The Benchmarking Sequence *242*
7.2.3.2.8. A Summary of the Test-