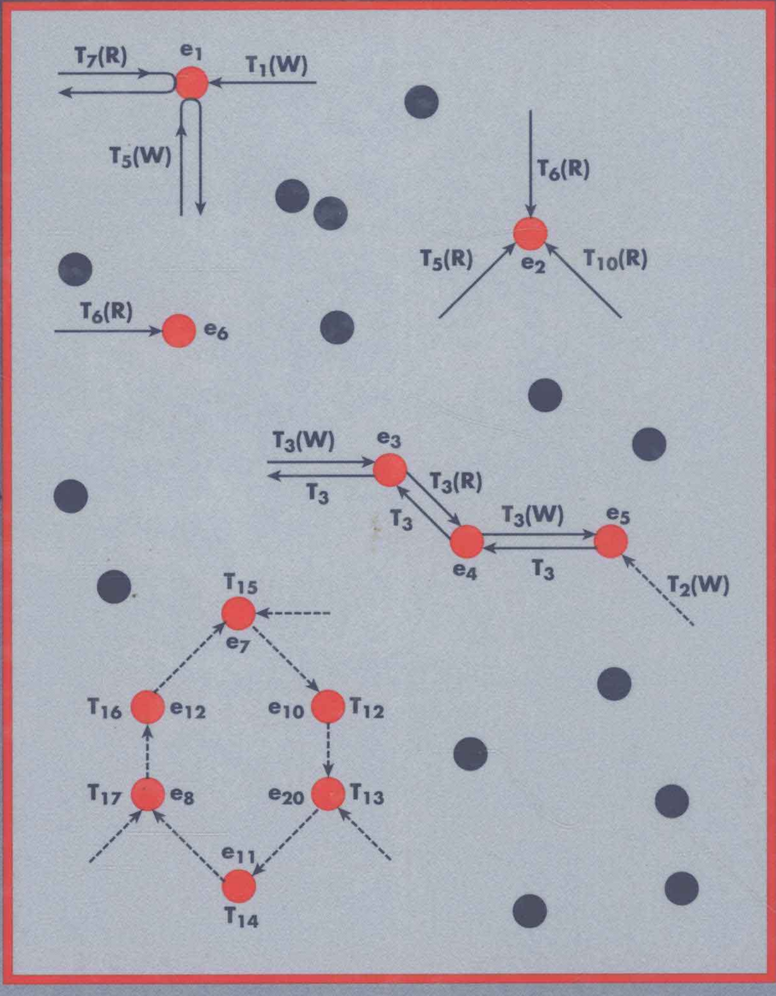
PERFORMANCE OF Concurrency Control Mechanisms in Centralized Database Systems

Rakesh Agrawal
Paul Bober
Michael Carey
Panos Chrysanthis
Margaret Eich
Vibby Gottemukkala
Jayant Haritsa
Meichun Hsu
Sushil Jajodia
David Langworthy
Juhnyoung Lee
Toby Lehman
Miron Livny
Vijay Kumar
C. Mohan
Ravi Mukkamala
Patrick O'Neil
Calton Pu
Krithi Ramamritham
Andreas Reuter
Friedemann Schwenkreis
Sang Son
Alexander Thomasian
David Wells
Shehan Xavier
Phillip Yu
Stanley Zdonik



VIJAY KUMAR

PERFORMANCE OF

Concurrency Control Mechanisms in Centralized Database Systems

VIJAY KUMAR, editor
University of Missouri—Kansas City

Prentice Hall, Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging-in-Publication Data

Performance of concurrency control mechanisms in centralized database systems / edited by Vijay Kumar.
p. cm.
Includes bibliographical references and index.
ISBN 0-13-065442-6
1. Database management. 2. Distributed databases. 3. Parallel processing (Electronic computers) I. Vijay Kumar
QA76.9.D3P457 1996
005.75--dc20 95-47126
CIP

Publisher: Alan Apt
Production Editor: Mona Pompili
Cover Designer: Bruce Kenselaar
Production Coordinator: Donna Sullivan
Editorial Assistant: Shirley McGuire

© 1996 by Prentice-Hall, Inc.
Simon & Schuster/A Viacom Company
Englewood Cliffs, New Jersey 07632

The author and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

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

ISBN 0-13-065442-6

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*
SIMON & SCHUSTER ASIA PTE. LTD., *Singapore*
EDITORA PRENTICE-HALL DO BRASIL, LTDA., *Rio de Janeiro*

The following papers were reprinted with permission.

Chapter 5: Rakesh Agrawal, Michael Carey, and Miron Livny, "Concurrency Control Performance Modeling: Alternative and Implications," *ACM Transactions on Database Systems*, Vol. 12, No. 4, December 1987, pp. 609–654.

Chapter 8: Alexander Thomasian, "Two-Phace Locking Performance and its Thrashing Behavior," *ACM Transactions on Database Systems*, Vol. 18, No. 4, December 1993, pp. 579–625.

Chapter 9: Margaret H. Eich and David L. Wells, "Database Concurrency Control Using Data Flow Graph," *ACM Transactions on Database Systems*, Vol. 13, No. 2, June 1988, pp. 197–227.

Chapter 11: C. Mohan, "Commit_LSN: A Novel and Simple Method for Reducing Locking and Latching in Transaction Processing Systems," *Proceedings of the 16th International Conference on Very Large Data Bases*, Brisbane, August 1990.

Chapter 14: Sushil Jajodia and Ravi Mukkamala, "Reduction in Transaction conflicts Using Semantics-Based Concurrency Control," *Information Sciences*, Vol. 68, No. 1/2, February 1993, pp. 91–111.

Chapter 17: Jayant Haritsa, "Firm Real-Time Concurrency Control," *The Journal of Real-Time Systems*, Vol. 4, 1992, pp. 203–241.

Chapter 19: Michael J. Carey and Miron Livny, "Conflict Detection Tradeoffs for replicated Data," *ACM Transactions on Database Systems*, Vol. 16, No. 4, December 1991, pp. 703–746.

Chapter 20: Paul M. Bober and Michael J. Carey, "On Mixing Queries and Transsactions via Multiversion Locking," *Proceedings of the Eighth IEEE International Conference on Data Engineering*, February 1992, pp. 535–545.

Foreword

Concurrency control has moved from an isolated database specialty to a central role in modern computing. Geographically distributed systems and parallel computer systems execute computations concurrently. Hence, concurrency control is now a cornerstone of modern computer system architecture.

This book collects in one volume our best understanding of models and techniques to concurrently execute application programs. Vijay Kumar has collected the work of the leading lights in the field, asking each to describe his specialty and contributions. Chapters cover the entire spectrum of issues: conceptual models, implementation techniques, and performance evaluation. As such, the book is an excellent snapshot of the state of the art. It collects the best ideas in one volume. Perhaps even more valuable, the book describes many unsolved problems, and so forms a research agenda for the concurrency control field.

The book begins with abstract models that capture application domains ranging from transaction processing to the more demanding areas of workflow, software-engineering, and real-time process control. It covers both the classical database lore and the more recent work on main-memory databases, object-oriented databases, workflow, and contracts.

Complementing these concepts chapters are a compendium of chapters giving implementation techniques. One extensive chapter explains the popular ARIES technique with emphasis on the interactions between concurrency control and recovery issues. Another chapter treats the hot-spot problem in detail. Other chapters cover concurrency control techniques appropriate for object-oriented databases, real-time databases, replicated databases, and transaction scheduling in a mixed workload environment.

The concepts and techniques chapters are kept honest by chapters that benchmark performance of various concurrency control techniques. Approaches are evaluated by both analytic models and by empirical simulations. These analytic and empirical papers summarize the latest performance models of concurrency control techniques. They also contain results of extensive simulations comparing the throughput and response time offered by the various approaches under a wide spectrum of workloads.

This volume as a whole is a great achievement—computer science at its best. The results are both elegant and relevant. They pull together both theory and practice in one unified framework. The result is both a tribute to those who created this work and a research agenda for the future.

Jim Gray
San Francisco, California

此为试读,需要完整PDF请访问: www.ertongbook.com

Preface

This book is about the mechanisms that maintain consistency-preserving execution of concurrent transactions and their performance. These mechanisms are commonly known as Concurrency Control Mechanisms (CCMs). For some time CCM has been an important area of research. It seems, for this reason, that a large number of CCMs have appeared in a short span of time. This proliferation of CCMs hints that at some time or another every database researcher had a close encounter with CCMs!

The extensive research in CCMs for centralized database systems has given database researchers a very good understanding of the essential aspects of CCMs for these systems. Unfortunately, this may not be said for CCMs for distributed database systems. For this reason, this topic has not been discussed in detail here. This book, therefore, deals mainly with CCMs and their performance for centralized database systems. However, it does include some information about CCMs that are also suitable for distributed database systems.

The book compiles a number of articles on CCMs and their performance, some of which have been published previously elsewhere, and some of which have been written exclusively for the book. The compilation, I believe, covers a lot of useful things about CCMs.

A log of scientific discovery is difficult to compile, especially in the case of computer science. In preparing a history of the origin of locking, concurrency, etc., I have had, at times, to trap the personal or private communication lines! Sometimes these lines were very noisy and some messages were encrypted. And sometimes the messages related to these topics were mixed with pleasurable conversation, more interesting than locking, which I should not include here!

The untold history (private communications with database wizards, wire tapping, etc.) of transaction processing reveals a number of exciting facts about locking and concurrency. It discloses that the wonders of locking were first seen around 1965 with Charlie Bachman's IDMS system. For this, Bachman received the Turing Award. The history then indicates that Jim Gray had some affinity and dealings with locking and he wrote a few things about the magic of locking back in 1968.

Then in 1970, the first publication appeared that revealed the secrets of locking to the general public.

The history reveals that the concept of concurrency was originally described (pragmatically) by Thomas Work (IBM). The documentation, said to be sketchy, was labeled "IBM Confidential." So here ends the detail! Locking was a part of the functional specification for IMS/360 Version 2. (IBM's OS/360 had a file locking facility called ENQ/DEQ, which was a partial inspiration.)

The history further discloses that in 1969, the developers of IMS/360 proposed to use locking. It was a small team (Sid Kornelius, Vern Watts, Thomas Work, Tom Sawyer, and Ron Obermark) and all contributed in the development of locking. In particular, Thomas Work, Tom Sawyer, and Ron Obermark concentrated on the concurrency control aspects, and Ron Obermark wrote the lock manager, which did deadlock detection. The first implementation was publicly available in 1972, as a feature of IMS/VS Version 1.0 (called Program Isolation Feature). It was used on the segments of the IMS/VS DL-1 database. The gosssip is that the idea of locking was also thought about by some other IBM people.

People working in the area of process synchronization in operating systems (mostly Burroughs, IBM, and Multics) began to expand the domain of operating systems' semaphores for managing concurrent operations in database systems. It seems that this laid the foundation for CCMs. The idea of mutual exclusion was implemented in database systems using locking concept, and the scheme came to be known as concurrency control mechanisms.

To exploit the power of CCMs for database systems, a number of other important concepts came into existence. Many ideas were implicit in early systems (e.g., DBTG had the notion of "run-unit," which was somewhat like a transaction; IMS had the notion of "sync-point," which was transaction commit; etc.) Indeed, the relational gang later elevated this from the level of a feature to a concept.

The names of some other formidable gangs are worth mentioning in this context. The System R group, the GE MadMan group (Daniel Rosenkrantz, Richard Sterns, and Philip Lewis), the CCA group (Phil Bernstein, Nathan Goodman, and others), Dave Lomet at Watson Research, and the Lampson-Sturgis file system group at Xerox PARC all had independent and similar ideas. The exact definition of consistency was much debated. The System R group (Jim Gray, Kapali Eswaran, Irving Traiger, R. Lorie, and others) argued strongly among themselves and concluded by offering the degrees of consistency story. Jim believed that the MadMan group had a similar experience, but for implementation purpose they needed degree 3 (full isolation). Bernstein and Goodman had a more relaxed/general definition of concurrency. Dave Lomet had an idea of atomic actions similar to degree 3.

The effort of these people laid the foundation for an important research area. The proliferation of CCMs necessitated the study of their performace. This required a re-examination of database processing and its structure from a concurrency viewpoint.

The book first deals with semantics and theoretical aspects of concurrency control. Then, an early paper by Andreas Reuter (Chapter 4) describes a framework of performance study of CCMs and also serves as an important link in the history of CCMs. There are some articles that do not contain any performance work. These papers are included because of their innovative concurrency control algorithms.

Managing concurrency in real-time systems has been a very active area of research. A number of papers on real-time concurrency control mechanisms have been compiled in the book. These papers present the current research direction in real-time database systems.

There are a couple of papers in the book that are related to CCMs for distributed database systems. These papers highlight the common elements of centralized and distributed transaction processing environments with respect to concurrency.

The book does not intend to serve as an encyclopedia of CCMs but compiles a subset of essential aspects of concurrency control for graduate students and database researchers. There is a large volume of completed work out there, and I believe that compiling it in more than one book is desirable.

Acknowledgements

This book would not have materialized without the contributions of a number of authors. Their help and cooperation shortened significantly the compilation and preparation time. The delay in its publication is mainly due to not finishing my portion of the job, which was to prepare the camera-ready copy.

I am grateful to many people for their help in completing this project. I will not provide a complete list since it would run through a number of pages. I, will, however, give a special mention to Mike Carey, Jim Gray, Meichun Hsu, Sushil Jajodia, C. Mohan, Krithi Ramamritham, Andreas Reuter, and Philip Yu. They were very generous in accepting my invitation to contribute their articles to the book. They did not need any persuasion. I e-mailed them my request and asked Jim if he could write the foreword. I was expecting NOs from most of them, but to my surprise they were all YESes. Mike was the first to respond, and he allowed me to communicate with him regularly for his useful advice. Jim not only wrote the foreword but also provided a good portion of the historical information on concurrency control mechanisms, locking, and so on. This boosted my hope and encouraged me to extend my invitations to a number of other contributors.

Preparing the camera-ready copy of the book has been a very time-consuming process. It would have been worse had I not been rescued by Dr. Panos Chrysanthis, Dr. Juhnyoung Lee, Dr. Deep Medhi, Dr. Sang H. Son, and Dr. Ken Shultis. Ken generously helped me with many LaTeX problems and his book LaTeX Notes, Prentice Hall, had been very useful. My colleague Deep spent hours with me in fixing my LaTeX mistakes. I owe him a lot. C. Mohan and Philip Yu were also very helpful in resolving my formatting errors.

Family members played a very helpful role in this time consuming project. My wife, Elizabeth Kumar continuously provided me encouragement. Without her help the design of the cover of the book, which illustrates the events in the execution life of concurrent transactions (blocking, conflict, deadlocks, roll-backs, and so on) would not have been possible. Our three-year-old son, Krishna, and one-year-old son Arjun, entertained and relieved me from the formatting stress on a regular basis.

My thanks to Beth Casey, Mona Pompili, and Alan Apt of Prentice Hall. Beth was instrumental in getting the project approved quickly.

Vijay Kumar
University of Missouri–Kansas City

Contents

1 Transactions and Database Processing **1**

1.1 Introduction 1

1.2 ACID Properties of Transactions 3

1.2.1 Transaction Execution 3

1.3 Management of Concurrent Execution 4

2 Serializability-Based Correctness Criteria **7**

2.1 Introduction 7

2.2 Correctness of Concurrent Executions 9

2.2.1 Transaction Interference 9

2.2.2 Serializability 12

2.3 Serializability Theory 13

2.3.1 Objects, Operations, and Conflicts 13

2.3.2 Histories of Transactions 14

2.3.3 Conflict Serializability 16

2.4 Atomicity and Durability 18

2.5 Overview of Concurrency Control Methods 20

2.6 Overview of Recovery Methods 22

2.7 Conclusion 25

3 Concurrency Control Mechanisms and Their Taxonomy **30**

3.1 Introduction 30

3.2 Taxonomy—Level 1 31

3.3 Taxonomy—Level 2 31
3.3.1 Two-Phase Locking (2PL) 32
3.3.2 Timestamping 34
3.3.3 Mixed Approach 34
3.4 Taxonomy—Level 3 35
3.4.1 Heuristic Approach 35

4 An Analytic Model of Transaction Interference 37
4.1 Introduction 37
4.2 A Basic Model of Transaction Concurrency 38
4.3 An Analytic Model for Two Concurrent Transactions 43
4.4 Simulation of the Dynamical Interference of Two Transactions 52
4.5 Extension of the Model 55

5 Concurrency Control Performance Modeling: Alternatives and Implications 58
5.1 Introduction 58
5.2 Concurrency Control Strategies 61
5.3 Performance Model 62
5.4 General Experiment Information 67
5.4.1 Performance Metrics 67
5.4.2 Parameter Settings 68
5.5 Resource-Related Assumptions 71
5.5.1 Experiment 1: Infinite Resources 71
5.5.2 Experiment 2: Resource-Limited Situation 74
5.5.3 A Brief Aside 79
5.5.4 Experiment 3: Multiple Resources 81
5.5.5 Experiment 4: Interactive Workloads 86
5.5.6 Resource Related Conclusions 91
5.6 Transaction Behavior Assumption 92
5.6.1 Experiment 6: Modeling Restarts 92
5.6.2 Experiment 7: Write Lock Acquisition 95
5.6.3 Transaction Behavior Conclusions 99
5.7 Conclusions and Implications 99

6 Modeling and Analysis of Concurrency Control Schemes 106
6.1 Introduction 107
6.2 Workload Model 108

此为试读，需要完整PDF请访问：www.ertongbook.com

6.3 Concurrency Control: General Methodology 110
6.3.1 Conflict Analysis 110
6.3.2 Conservation Property of Conflict 113
6.4 Analysis of OCC and Locking Schemes 114
6.4.1 Optimistic Protocols 114
6.4.2 Standard Locking 118
6.4.3 Hybrid OCC Schemes 120
6.4.4 Performance Results 121
6.5 The Locking with Deferred Blocking Scheme 125
6.5.1 The LDB Scheme 126
6.5.2 Analysis of LDB 129
6.5.3 Performance Results 132
6.6 Dynamic Time-stamp Certification 134
6.6.1 The TSH Scheme 135
6.6.2 Analysis of TSH 138
6.6.3 Performance Results 141
6.7 Summary and Conclusion 143

7 Modeling Performance Impact of Hot Spots 148
7.1 Introduction 149
7.2 The Transaction Model 150
7.2.1 The Basic Model 150
7.2.2 A Model of Nonuniform Access 152
7.3 Data Flow Analysis For Nonuniform Access 153
7.3.1 Data Flow Balance and Distribution of Locked Granules 154
7.3.2 Equal-Chance Abort Policies 155
7.3.3 The Database Reduction Factor 156
7.3.4 Solving the Model 158
7.3.5 Numerical Results 159
7.4 The Robustness Property 159
7.5 Conclusion 164

8 Two-Phase Locking Performance and Its Thrashing Behavior 166
8.1 Introduction 167
8.2 System Model 170
8.3 Analysis of 2PL with Identical Processing Times 175
8.3.1 Global Performance Measures 176
8.3.2 Probability of Lock Conflict and Mean Waiting Time 177

8.3.3 The Mean Waiting Time per Lock Conflict 179
8.3.4 Thrashing Behavior in 2PL 183
8.3.5 Analysis with a Finite Wait-Depth 187
8.4 Validation and System Behavior for Identical Transaction Steps 188
8.5 Different Processing Times for Transaction Steps 194
8.5.1 Analysis of the System 194
8.5.2 Validation and Parametric Studies 196
8.6 Load Control and Factors Contributing to Thrashing 200
8.6.1 Contributing Effects to Thrashing and Its Prevention 201
8.6.2 Methods to Reduce the Level of Lock Contention 205
8.7 Conclusions 206

9 Database Concurrency Control Using Data Flow Graphs 215
9.1 Introduction 215
9.2 DBFG Requirements 217
9.3 DBFG Definition 220
9.4 DBFG Operations 226
9.5 DBFG Scheduling 229
9.6 Performance Analysis 234
9.6.1 Simulation Model 234
9.6.2 Experiment Results—Impact of Number of Transactions 241

10 Concurrency Control and Recovery Methods for B$^+$-Tree Indexes: ARIES/KVL and ARIES/IM 248
10.1 Introduction 249
10.2 The Basics 252
10.2.1 Locks and Latches 252
10.2.2 Recovery and ARIES 254
10.2.3 Conventions and Storage Structures 256
10.3 A Simple Locking Algorithm 258
10.3.1 Simple Operations 258
10.3.2 More Powerful Fetch Operation 260
10.3.3 Problems 263
10.4 ARIES/KVL 265
10.4.1 Tree Traversal 268
10.4.2 Basic Operations 270
10.4.3 Discussion 280
10.4.4 Deadlocks 282

10.5 ARIES/IM 282
10.5.1 Locking 283
10.5.2 Recovery 286
10.6 System R's Algorithm 296
10.7 Summary and Conclusions 299

11 Commit_LSN: A Noval and Simple Method for Reducing Locking and Latching in Transaction Processing Systems 307
11.1 Introduction 308
11.1.1 Overview of the Method 309
11.1.2 Latches and Locks 310
11.1.3 Logging and Recovery 312
11.2 Goals 313
11.3 The Commit_LSN Method 314
11.4 Implementation in Different Systems 316
11.5 Object-Specific Commit_LSNs 318
11.6 Applications of the Method 319
11.7 Shared Disks Versus Partitioning 327
11.8 Summary 329

12 Synchronizing Long-Lived Computations 336
12.1 Introduction 336
12.2 An Execution Model 338
12.2.1 Definitions 338
12.2.2 Constraints and Behavior at Run Time 339
12.2.3 Recovery Issues 341
12.2.4 Correctness 342
12.3 Concurrency Control 342
12.3.1 Ensuring Compensation 342
12.3.2 Support for Successful Executions 344
12.4 Spheres of Isolation 344
12.4.1 Simple Approaches 344
12.4.2 Connecting Objects 346
12.4.3 A Concurrency Control Mechanism 347
12.4.4 Spheres — An Abstraction 348
12.5 Comparison With Serializability 349
12.5.1 The Basic Isolation Problems 350
12.5.2 Semantics of an Access 351

12.6 Summary 353

13 Implementation Considerations and Performance Evaluation of Object-Based Concurrency Control Protocols 356

13.1 Introduction 356
13.2 Object-Based Concurrency Control Semantics 358
13.3 Objects and Operations 360
13.3.1 Stack 360
13.3.2 Queue 360
13.3.3 Semiqueue 361
13.3.4 Directory 362
13.4 Overview of Concurrency Control Strategies 362
13.4.1 Two-Phase Locking Protocols 362
13.4.2 Optimistic Methods 363
13.5 Concurrency Control Implementations 365
13.5.1 Implementation of Concurrency Control for Objects 365
13.5.2 Optimistic Protocols 367
13.6 Performance Model and Metric 368
13.6.1 Performance Metric 371
13.7 Simulation Results 372
13.7.1 Experiment 1 — Ten Resource Units 372
13.7.2 Experiment Two — Resource Related Assumptions 377
13.7.3 Experiment 4 381
13.7.4 Summary of Simulation Results 383

14 Reduction in Transaction Conflicts Using Semantics-Based Concurrency Control 387

14.1 Introduction 387
14.2 Commutative Classes and Their Relevance 389
14.3 A Distributed Database Model 389
14.4 Detection of Transaction Conflicts Among Partitions 390
14.5 The Probablistic Model 391
14.6 Computation 393
14.6.1 Computation of Transaction Roll-backs 394
14.6.2 Computation of Transaction Roll-backs Due to Direct Dependency Conflicts 395
14.7 Results 396
14.7.1 An Illustration 396

14.7.2 Special Case 398
14.7.3 When is Commutativity Beneficial? 400
14.8 Summary of Observations 403

15 The Design and Performance Evaluation of a Lock Manager for a Memory-Resident Database System 406
15.1 Introduction 407
15.2 Design Overview 408
15.2.1 Latches 409
15.2.2 Locks 410
15.3 Implementation of the MMM Lock Manager 412
15.3.1 Escalation 413
15.3.2 De-escalation 415
15.3.3 De-escalation Daemon 415
15.3.4 Implications of the Design 417
15.4 Performance 418
15.4.1 The Results 419
15.4.2 The Bottom Line 421
15.5 Related Work 422
15.5.1 Cheap-First-Locks 422
15.5.2 Amoeba 423
15.5.3 Adjustable Lock Granularity 424
15.5.4 The Original MRDB Work 425
15.6 Interesting Alternatives 425
15.6.1 Predicate Locks 425
15.7 Conclusions 426

16 Performance of Concurrency Control Algorithms for Real-Time Database Systems 429
16.1 Introduction 429
16.2 Concurrency Control Algorithms 432
16.2.1 Locking-Based Algorithms 432
16.2.2 Optimistic Concurrency Control-Based Algorithms 434
16.2.3 Qualitative Analysis 436
16.2.4 Implementation Issues 437
16.3 RTDBS Model 438
16.3.1 Soft Deadline Versus Firm Deadline 438
16.3.2 Resource Availability 441

16.3.3 Other Issues 442
16.4 Experimental Environment 442
16.4.1 Simulation Model 442
16.4.2 Parameter Setting 444
16.4.3 Performance Metrics 446
16.5 Experiments and Results 446
16.5.1 Experiment 1: Soft Deadline and Finite Resources 446
16.5.2 Experiment 2: Soft Deadline and Infinite Resources 450
16.5.3 Experiment 3: Firm Deadline and Finite Resources 453
16.5.4 Experiment 4: Firm Deadline and Infinite Resources 455
16.6 Conclusions 457

17 Firm Real-Time Concurrency Control 461
17.1 Introduction 461
17.2 Real-Time Framework 464
17.3 Related Work 465
17.4 Concurrency Control Algorithms 466
17.4.1 Conventional Locking (2PL) 466
17.4.2 Real-Time Locking Algorithms 466
17.4.3 Conventional Optimistic Concurrency Control (OPT) 468
17.4.4 Real-Time Optimistic Algorithms 469
17.4.5 Summary 471
17.5 Real-Time DBMS Performance Model 472
17.5.1 Workload Model 473
17.5.2 System Model 474
17.5.3 Execution-Time Computation 475
17.5.4 Concurrency Control Overhead 475
17.6 Experiments and Results 475
17.6.1 Performance Metric 476
17.6.2 Parameter Settings 476
17.6.3 Experiment 1: Baseline Experiment 477
17.6.4 Experiment 2: Pure Data Contention 479
17.6.5 Experiment 3: Variable-Slack Ratio 481
17.6.6 Experiment 4: Wait-Control Mechanism 484
17.6.7 Experiment 5: Performance Crossover 487
17.7 Conclusions 489

此为试读，需要完整PDF请访问：www.ertongbook.com