Contents

Foreword
Jim Gray, Microsoft, Inc. ix

Preface xxii

PART ONE
BACKGROUND AND MOTIVATION

Chapter 1 What is It All About? 3
1.1 Goal and Overview 3
1.2 Application Examples 4
1.2.1 Online Transaction Processing: Debit/Credit Example 5
1.2.2 Electronic Commerce Example 9
1.2.3 Workflow Management: Travel Planning Example 12
1.3 System Paradigms 16
1.3.1 Three-Tier and Two-Tier Architectures 16
1.3.2 Federations of Servers 20
1.4 Virtues of the Transaction Concept 22
1.4.1 Transaction Properties and the Transaction Programming Interface 22
1.4.2 Requirements on Transactional Servers 26
1.5 Concepts and Architecture of Database Servers 27
1.5.1 Architectural Layers of Database Systems 27
1.5.2 How Data Is Stored 30
1.5.3 How Data Is Accessed 32
1.5.4 How Queries and Updates Are Executed 35
1.6 Lessons Learned 37
Exercises 38
Bibliographic Notes 38
Chapter 2  Computational Models  
2.1  Goal and Overview  
2.2  Ingredients  
2.3  The Page Model  
2.4  The Object Model  
2.5  Road Map of the Book  
2.6  Lessons Learned  
Exercises  
Bibliographic Notes

PART TWO  
CONCURRENCY CONTROL  

Chapter 3  Concurrency Control: Notions of Correctness for the Page Model  
3.1  Goal and Overview  
3.2  Canonical Concurrency Problems  
3.3  Syntax of Histories and Schedules  
3.4  Correctness of Histories and Schedules  
3.5  Herbrand Semantics of Schedules  
3.6  Final State Serializability  
3.7  View Serializability  
3.7.1  View Equivalence and the Resulting Correctness Criterion  
3.7.2  On the Complexity of Testing View Serializability  
3.8  Conflict Serializability  
3.8.1  Conflict Relations  
3.8.2  Class CSR  
3.8.3  Conflicts and Commutativity  
3.8.4  Restrictions of Conflict Serializability  
3.9  Commit Serializability  
3.10  An Alternative Correctness Criterion: Interleaving Specifications  
3.11  Lessons Learned  
Exercises  
Bibliographic Notes
Chapter 4 Concurrency Control Algorithms

4.1 Goal and Overview 125
4.2 General Scheduler Design 126
4.3 Locking Schedulers 130
   4.3.1 Introduction 130
   4.3.2 The Two-Phase Locking Protocol 133
   4.3.3 Deadlock Handling 138
   4.3.4 Variants of 2PL 142
   4.3.5 Ordered Sharing of Locks 144
   4.3.6 Altruistic Locking 150
   4.3.7 Non-Two-Phase Locking Protocols 155
   4.3.8 On the Geometry of Locking 162
4.4 Nonlocking Schedulers 166
   4.4.1 Timestamp Ordering 166
   4.4.2 Serialization Graph Testing 168
   4.4.3 Optimistic Protocols 170
4.5 Hybrid Protocols 175
4.6 Lessons Learned 179
   Exercises 180
   Bibliographic Notes 182

Chapter 5 Multiversion Concurrency Control 185

5.1 Goal and Overview 185
5.2 Multiversion Schedules 186
5.3 Multiversion Serializability 189
   5.3.1 Multiversion View Serializability 189
   5.3.2 Testing Membership in MVSR 193
   5.3.3 Multiversion Conflict Serializability 197
5.4 Limiting the Number of Versions 201
5.5 Multiversion Concurrency Control Protocols 203
   5.5.1 The MVTO Protocol 203
   5.5.2 The MV2PL Protocol 205
   5.5.3 The MVSGT Protocol 209
   5.5.4 A Multiversion Protocol for Read-Only Transactions 211
5.6 Lessons Learned 213
   Exercises 214
   Bibliographic Notes 215
Chapter 6 Concurrency Control on Objects:  
Notions of Correctness  
6.1 Goal and Overview  
6.2 Histories and Schedules  
6.3 Conflict Serializability for Flat Object Transactions  
6.4 Tree Reducibility  
6.5 Sufficient Conditions for Tree Reducibility  
6.6 Exploiting State Based Commutativity  
6.7 Lessons Learned  
Exercises  
Bibliographical Notes  

Chapter 7 Concurrency Control Algorithms on Objects  
7.1 Goal and Overview  
7.2 Locking for Flat Object Transactions  
7.3 Layered Locking  
7.4 Locking on General Transaction Forests  
7.5 Hybrid Algorithms  
7.6 Locking for Return Value Commutativity and Escrow Locking  
7.7 Lessons Learned  
Exercises  
Bibliographical Notes  

Chapter 8 Concurrency Control on Relational Databases  
8.1 Goal and Overview  
8.2 Predicate-Oriented Concurrency Control  
8.3 Relational Update Transactions  
8.3.1 Syntax and Semantics  
8.3.2 Commutativity and Simplification Rules  
8.3.3 Histories and Final State Serializability  
8.3.4 Conflict Serializability  
8.3.5 Extended Conflict Serializability  
8.3.6 Serializability in the Presence of Functional Dependencies  
8.3.7 Summary  

8.4 Exploiting Transaction Program Knowledge  
8.4.1 Motivating Example  
8.4.2 Transaction Chopping  
8.4.3 Applicability of Chopping  
Exercises  
Bibliographical Notes
## 8.5 Lessons Learned
Exercises
Bibliographic Notes

### Chapter 9 Concurrency Control on Search Structures
9.1 Goal and Overview
9.2 Implementation of Search Structures by B+ Trees
9.3 Key Range Locking at the Access Layer
9.4 Techniques for the Page Layer
  9.4.1 Lock Coupling
  9.4.2 Link Technique
  9.4.3 Giveup Technique
9.5 Further Optimizations
  9.5.1 Deadlock-Free Page Latching
  9.5.2 Enhanced Key Range Concurrency
  9.5.3 Reduced Locking Overhead
  9.5.4 Exploiting Transient Versioning
9.6 Lessons Learned
Exercises
Bibliographic Notes

### Chapter 10 Implementation and Pragmatic Issues
10.1 Goal and Overview
10.2 Data Structures of a Lock Manager
10.3 Multiple Granularity Locking and Dynamic Escalation
10.4 Transient Versioning
10.5 Nested Transactions for Intra-transaction Parallelism
10.6 Tuning Options
  10.6.1 Manual Locking
  10.6.2 SQL Isolation Levels
  10.6.3 Short Transactions
  10.6.4 Limiting the Level of Multiprogramming
10.7 Overload Control
  10.7.1 Feedback-Driven Method
  10.7.2 Wait-Depth Limitation
10.8 Lessons Learned
Exercises
Bibliographic Notes
PART THREE
RECOVERY

Chapter 11 Transaction Recovery 379
11.1 Goal and Overview 379
11.2 Expanded Schedules with Explicit Undo Operations 381
   11.2.1 Intuition and Overview of Concepts 381
   11.2.2 The Formal Model 382
11.3 Correctness Criteria for the Page Model 385
   11.3.1 Expanded Conflict Serializability 385
   11.3.2 Reducibility and Prefix Reducibility 387
11.4 Sufficient Syntactic Conditions 390
   11.4.1 Recoverability 391
   11.4.2 Avoiding Cascading Aborts 391
   11.4.3 Strictness 393
   11.4.4 Rigorousness 393
   11.4.5 Log Recoverability 398
11.5 Page Model Protocols for Schedules with Transaction Aborts 402
   11.5.1 Extending Two-Phase Locking for Strictness and Rigorousness 402
   11.5.2 Extending Serialization Graph Testing for Log Recoverability 403
   11.5.3 Extending Other Protocols for Log Recoverability 406
11.6 Correctness Criteria for the Object Model 407
   11.6.1 Aborts in Flat Object Schedules 407
   11.6.2 Complete and Partial Aborts in General Object Model Schedules 416
11.7 Object Model Protocols for Schedules with Transaction Aborts 419
11.8 Lessons Learned 420
Exercises 421
Bibliographic Notes 423

Chapter 12 Crash Recovery: Notion of Correctness 427
12.1 Goal and Overview 427
12.2 System Architecture and Interfaces 430
12.3 System Model 434
12.4 Correctness Criterion 437
12.5 Road Map of Algorithms 439
12.6 Lessons Learned
Exercises 444
Bibliographic Notes 445

Chapter 13 Page Model Crash Recovery Algorithms 447
13.1 Goal and Overview 447
13.2 Basic Data Structures 449
13.3 Redo-Winners Paradigm 453
13.3.1 Actions during Normal Operation 454
13.3.2 Simple Three-Pass Algorithm 458
13.3.3 Enhanced Algorithm: Log Truncation, Checkpoints, Redo Optimization 473
13.3.4 The Complete Algorithm: Handling Transaction Aborts and Undo Completion 491
13.4 Redo-History Paradigm 501
13.4.1 Actions during Normal Operation 501
13.4.2 Simple Three-Pass and Two-Pass Algorithms 501
13.4.3 Enhanced Algorithms: Log Truncation, Checkpoints, and Redo Optimization 510
13.4.4 Complete Algorithms: Handling Transaction Rollbacks and Undo Completion 510
13.5 Lessons Learned 518
13.5.1 Putting Everything Together 519
Exercises 526
Bibliographic Notes 528

Chapter 14 Object Model Crash Recovery 531
14.1 Goal and Overview 531
14.2 Conceptual Overview of Redo-History Algorithms 532
14.3 A Simple Redo-History Algorithm for Two-Layered Systems 536
14.3.1 Actions during Normal Operation 538
14.3.2 Steps during Restart 539
14.4 An Enhanced Redo-History Algorithm for Two-Layered Systems 545
14.5 A Complete Redo-History Algorithm for General Object Model Executions 552
14.6 Lessons Learned 556
Exercises 558
Bibliographic Notes 560
Chapter 19 Distributed Transaction Recovery

19.1 Goal and Overview 723
19.2 The Basic Two-Phase Commit Algorithm 725
  19.2.1 2PC Protocol 725
  19.2.2 Restart and Termination Protocol 733
  19.2.3 Independent Recovery 741
19.3 The Transaction Tree Two-Phase Commit Algorithm 744
19.4 Optimized Algorithms for Distributed Commit 748
  19.4.1 Presumed-Abort and Presumed-Commit Protocols 749
  19.4.2 Read-Only Subtree Optimization 756
  19.4.3 Coordinator Transfer 758
  19.4.4 Reduced Blocking 761
19.5 Lessons Learned 763
  Exercises 765
  Bibliographic Notes 766

PART FIVE
APPLICATIONS AND FUTURE PERSPECTIVES

Chapter 20 What Is Next? 771
20.1 Goal and Overview 771
20.2 What Has Been Achieved? 771
  20.2.1 Ready-to-Use Solutions for Developers 772
  20.2.2 State-of-the-Art Techniques for Advanced System Builders 773
  20.2.3 Methodology and New Challenges for Researchers 775
20.3 Data Replication for Ubiquitous Access 776
20.4 E-Services and Workflows 779
20.5 Performance and Availability Guarantees 783
  Bibliographic Notes 787

References 791
Index 829
About the Authors 853