

# Transactional Information Systems

# Theory, Algorithms, and the Practice of Concurrency Control and Recovery

Gerhard Weikum

University of the Saarland, Germany

# Gottfried Vossen

University of Münster, Germany

Technische Universität Darmstadt  
FACHBEREICH INFORMATIK  
B I B L I O T H E K  
Inventar-Nr.: 1101-00186  
Sachgebiete: \_\_\_\_\_  
Standort: \_\_\_\_\_

MK®

MORGAN KAUFMANN PUBLISHERS

AN IMPRINT OF ACADEMIC PRESS

A Harcourt Science and Technology Company

SAN FRANCISCO   SAN DIEGO   NEW YORK   BOSTON  
LONDON   SYDNEY   TOKYO

# Contents

<b>Foreword</b>	ix
Jim Gray, Microsoft, Inc.	
<b>Preface</b>	xxi

## **PART ONE**

### **BACKGROUND AND MOTIVATION**

<b>Chapter 1 What Is It All About?</b>	<b>3</b>
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

<b>Chapter 2</b>	<b>Computational Models</b>	<b>41</b>
2.1	Goal and Overview	41
2.2	Ingredients	42
2.3	The Page Model	43
2.4	The Object Model	47
2.5	Road Map of the Book	53
2.6	Lessons Learned	56
	Exercises	56
	Bibliographic Notes	57

## PART TWO

### CONCURRENCY CONTROL

<b>Chapter 3</b>	<b>Concurrency Control: Notions of Correctness for the Page Model</b>	<b>61</b>
3.1	Goal and Overview	61
3.2	Canonical Concurrency Problems	62
3.3	Syntax of Histories and Schedules	65
3.4	Correctness of Histories and Schedules	71
3.5	Herbrand Semantics of Schedules	73
3.6	Final State Serializability	76
3.7	View Serializability	82
3.7.1	View Equivalence and the Resulting Correctness Criterion	83
3.7.2	On the Complexity of Testing View Serializability	86
3.8	Conflict Serializability	92
3.8.1	Conflict Relations	93
3.8.2	Class CSR	94
3.8.3	Conflicts and Commutativity	99
3.8.4	Restrictions of Conflict Serializability	101
3.9	Commit Serializability	105
3.10	An Alternative Correctness Criterion: Interleaving Specifications	108
3.11	Lessons Learned	119
	Exercises	120
	Bibliographic Notes	121

<b>Chapter 4</b>	<b>Concurrency Control Algorithms</b>	<b>125</b>
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
<b>Chapter 5</b>	<b>Multiversion Concurrency Control</b>	<b>185</b>
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

<b>Chapter 6</b>	<b>Concurrency Control on Objects:</b>	
	Notions of Correctness	217
6.1	Goal and Overview	217
6.2	Histories and Schedules	218
6.3	Conflict Serializability for Flat Object Transactions	223
6.4	Tree Reducibility	228
6.5	Sufficient Conditions for Tree Reducibility	233
6.6	Exploiting State Based Commutativity	240
6.7	Lessons Learned	246
	Exercises	247
	Bibliographical Notes	250
<b>Chapter 7</b>	<b>Concurrency Control Algorithms on Objects</b>	251
7.1	Goal and Overview	251
7.2	Locking for Flat Object Transactions	251
7.3	Layered Locking	252
7.4	Locking on General Transaction Forests	259
7.5	Hybrid Algorithms	265
7.6	Locking for Return Value Commutativity and Escrow Locking	267
7.7	Lessons Learned	271
	Exercises	272
	Bibliographic Notes	274
<b>Chapter 8</b>	<b>Concurrency Control on Relational Databases</b>	277
8.1	Goal and Overview	277
8.2	Predicate-Oriented Concurrency Control	278
8.3	Relational Update Transactions	285
	8.3.1 Syntax and Semantics	285
	8.3.2 Commutativity and Simplification Rules	287
	8.3.3 Histories and Final State Serializability	288
	8.3.4 Conflict Serializability	291
	8.3.5 Extended Conflict Serializability	293
	8.3.6 Serializability in the Presence of Functional Dependencies	295
	8.3.7 Summary	298
8.4	Exploiting Transaction Program Knowledge	299
	8.4.1 Motivating Example	299
	8.4.2 Transaction Chopping	301
	8.4.3 Applicability of Chopping	306

8.5	Lessons Learned	308
	Exercises	308
	Bibliographic Notes	311
<b>Chapter 9</b>	<b>Concurrency Control on Search Structures</b>	<b>313</b>
9.1	Goal and Overview	313
9.2	Implementation of Search Structures by B <sup>+</sup> Trees	315
9.3	Key Range Locking at the Access Layer	320
9.4	Techniques for the Page Layer	327
9.4.1	Lock Coupling	328
9.4.2	Link Technique	337
9.4.3	Giveup Technique	339
9.5	Further Optimizations	340
9.5.1	Deadlock-Free Page Latching	340
9.5.2	Enhanced Key Range Concurrency	341
9.5.3	Reduced Locking Overhead	343
9.5.4	Exploiting Transient Versioning	344
9.6	Lessons Learned	344
	Exercises	345
	Bibliographic Notes	347
<b>Chapter 10</b>	<b>Implementation and Pragmatic Issues</b>	<b>349</b>
10.1	Goal and Overview	349
10.2	Data Structures of a Lock Manager	349
10.3	Multiple Granularity Locking and Dynamic Escalation	352
10.4	Transient Versioning	354
10.5	Nested Transactions for Intra-transaction Parallelism	357
10.6	Tuning Options	359
10.6.1	Manual Locking	359
10.6.2	SQL Isolation Levels	360
10.6.3	Short Transactions	364
10.6.4	Limiting the Level of Multiprogramming	367
10.7	Overload Control	369
10.7.1	Feedback-Driven Method	369
10.7.2	Wait-Depth Limitation	373
10.8	Lessons Learned	374
	Exercises	375
	Bibliographic Notes	375

**PART THREE**  
**RECOVERY**

<b>Chapter 11</b>	<b>Transaction Recovery</b>	<b>379</b>
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
<b>Chapter 12</b>	<b>Crash Recovery: Notion of Correctness</b>	<b>427</b>
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	444
Exercises	444
Bibliographic Notes	445
<b>Chapter 13 Page Model Crash Recovery Algorithms</b>	<b>447</b>
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
<b>Chapter 14 Object Model Crash Recovery</b>	<b>531</b>
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	536
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



<b>Chapter 15</b>	<b>Special Issues of Recovery</b>	<b>561</b>
15.1	Goal and Overview	561
15.2	Logging and Recovery for Indexes and Large Objects	562
15.2.1	Logical Log Entries for the Redo of Index Page Splits	562
15.2.2	Logical Log Entries and Flush Ordering for Large-Object Operations	566
15.3	Intra-transaction Savepoints and Nested Transactions	571
15.4	Exploiting Parallelism during Restart	577
15.5	Special Considerations for Main-Memory Data Servers	580
15.6	Extensions for Data-Sharing Clusters	583
15.7	Lessons Learned	589
	Exercises	589
	Bibliographic Notes	591
<b>Chapter 16</b>	<b>Media Recovery</b>	<b>593</b>
16.1	Goal and Overview	593
16.2	Log-Based Method	596
16.2.1	Database Backup and Archive Logging during Normal Operation	597
16.2.2	Database Restore Algorithms	599
16.2.3	Analysis of the Mean Time to Data Loss	602
16.3	Storage Redundancy	606
16.3.1	Techniques Based on Mirroring	607
16.3.2	Techniques Based on Error-Correcting Codes	610
16.4	Disaster Recovery	618
16.5	Lessons Learned	620
	Exercises	621
	Bibliographic Notes	621
<b>Chapter 17</b>	<b>Application Recovery</b>	<b>623</b>
17.1	Goal and Overview	623
17.2	Stateless Applications Based on Queues	625
17.3	Stateful Applications Based on Queues	632
17.4	Workflows Based on Queues	637
17.4.1	Failure-Resilient Workflow State and Context	639
17.4.2	Decentralized Workflows Based on Queued Transactions	640
17.5	General Stateful Applications	642
17.5.1	Design Considerations	643
17.5.2	Overview of the Server Reply Logging Algorithm	646

17.5.3	Data Structures	648
17.5.4	Server Logging during Normal Operation	650
17.5.5	Client Logging during Normal Operation	653
17.5.6	Log Truncation	655
17.5.7	Server Restart	657
17.5.8	Client Restart	659
17.5.9	Correctness Reasoning	662
17.5.10	Applicability to Multi-tier Architectures	666
17.6	Lessons Learned	667
	Exercises	668
	Bibliographic Notes	669

## PART FOUR

### COORDINATION OF DISTRIBUTED TRANSACTIONS

<b>Chapter 18</b>	<b>Distributed Concurrency Control</b>	<b>673</b>
18.1	Goal and Overview	673
18.2	Concurrency Control in Homogeneous Federations	676
18.2.1	Preliminaries	676
18.2.2	Distributed 2PL	679
18.2.3	Distributed TO	680
18.2.4	Distributed SGT	683
18.2.5	Optimistic Protocols	685
18.3	Distributed Deadlock Detection	686
18.4	Serializability in Heterogeneous Federations	690
18.4.1	Global Histories	691
18.4.2	Global Serializability	694
18.4.3	Quasi Serializability	696
18.5	Achieving Global Serializability through Local Guarantees	698
18.5.1	Rigorousness	698
18.5.2	Commitment Ordering	700
18.6	Ticket-Based Concurrency Control	702
18.6.1	Explicit Tickets for Forcing Conflicts	702
18.6.2	Implicit Tickets	706
18.6.3	Mixing Explicit and Implicit Tickets	707
18.7	Object Model Concurrency Control in Heterogeneous Federations	708
18.8	Coherency and Concurrency Control for Data-Sharing Systems	710
18.9	Lessons Learned	716

Exercises	717
Bibliographic Notes	719
<b>Chapter 19 Distributed Transaction Recovery</b>	<b>723</b>
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
 <b>PART FIVE</b>	
<b>APPLICATIONS AND FUTURE PERSPECTIVES</b>	
<b>Chapter 20 What Is Next?</b>	<b>771</b>
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
 <b>References</b>	 <b>791</b>
<b>Index</b>	<b>829</b>
<b>About the Authors</b>	<b>853</b>