High-Performance Parallel Database Processing and Grid Databases

David Taniar  
*Monash University, Australia*

Clement H.C. Leung  
*Hong Kong Baptist University and Victoria University, Australia*

Wenny Rahayu  
*La Trobe University, Australia*

Sushant Goel  
*RMIT University, Australia*
Contents

Preface xv

Part I Introduction

1. Introduction 3

1.1. A Brief Overview: Parallel Databases and Grid Databases 4
1.2. Parallel Query Processing: Motivations 5
1.3. Parallel Query Processing: Objectives 7
1.3.1. Speed Up 7
1.3.2. Scale Up 8
1.3.3. Parallel Obstacles 10
1.4. Forms of Parallelism 12
1.4.1. Interquery Parallelism 13
1.4.2. Intraquery Parallelism 14
1.4.3. Intraoperation Parallelism 15
1.4.4. Interoperation Parallelism 15
1.4.5. Mixed Parallelism—A More Practical Solution 18
1.5. Parallel Database Architectures 19
1.5.1. Shared-Memory and Shared-Disk Architectures 20
1.5.2. Shared-Nothing Architecture 22
1.5.3. Shared-Something Architecture 23
1.5.4. Interconnection Networks 24
1.6. Grid Database Architecture 26
1.7. Structure of this Book 29
1.8. Summary 30
1.9. Bibliographical Notes 30
1.10. Exercises 31
2. Analytical Models

2.1. Cost Models 33
2.2. Cost Notations 34
  2.2.1. Data Parameters 34
  2.2.2. Systems Parameters 36
  2.2.3. Query Parameters 37
  2.2.4. Time Unit Costs 37
  2.2.5. Communication Costs 38
2.3. Skew Model 39
2.4. Basic Operations in Parallel Databases 43
  2.4.1. Disk Operations 44
  2.4.2. Main Memory Operations 45
  2.4.3. Data Computation and Data Distribution 45
2.5. Summary 47
2.6. Bibliographical Notes 47
2.7. Exercises 47

Part II  Basic Query Parallelism

3. Parallel Search 51

3.1. Search Queries 51
  3.1.1. Exact-Match Search 52
  3.1.2. Range Search Query 53
  3.1.3. Multiattribute Search Query 54
3.2. Data Partitioning 54
  3.2.1. Basic Data Partitioning 55
  3.2.2. Complex Data Partitioning 60
3.3. Search Algorithms 69
  3.3.1. Serial Search Algorithms 69
  3.3.2. Parallel Search Algorithms 73
3.4. Summary 74
3.5. Bibliographical Notes 75
3.6. Exercises 75

4. Parallel Sort and GroupBy 77

4.1. Sorting, Duplicate Removal, and Aggregate Queries 78
  4.1.1. Sorting and Duplicate Removal 78
  4.1.2. Scalar Aggregate 79
  4.1.3. GroupBy 80
4.2. Serial External Sorting Method 80
### 4.3. Algorithms for Parallel External Sort

- **4.3.1. Parallel Merge-All Sort** 83
- **4.3.2. Parallel Binary-Merge Sort** 85
- **4.3.3. Parallel Redistribution Binary-Merge Sort** 86
- **4.3.4. Parallel Redistribution Merge-All Sort** 88
- **4.3.5. Parallel Partitioned Sort** 90

### 4.4. Parallel Algorithms for GroupBy Queries

- **4.4.1. Traditional Methods (Merge-All and Hierarchical Merging)** 92
- **4.4.2. Two-Phase Method** 93
- **4.4.3. Redistribution Method** 94

### 4.5. Cost Models for Parallel Sort

- **4.5.1. Cost Models for Serial External Merge-Sort** 96
- **4.5.2. Cost Models for Parallel Merge-All Sort** 98
- **4.5.3. Cost Models for Parallel Binary-Merge Sort** 100
- **4.5.4. Cost Models for Parallel Redistribution Binary-Merge Sort** 101
- **4.5.5. Cost Models for Parallel Redistribution Merge-All Sort** 102
- **4.5.6. Cost Models for Parallel Partitioned Sort** 103

### 4.6. Cost Models for Parallel GroupBy

- **4.6.1. Cost Models for Parallel Two-Phase Method** 104
- **4.6.2. Cost Models for Parallel Redistribution Method** 107

### 4.7. Summary

### 4.8. Bibliographical Notes

### 4.9. Exercises

### 5. Parallel Join

#### 5.1. Join Operations

#### 5.2. Serial Join Algorithms

- **5.2.1. Nested-Loop Join Algorithm** 114
- **5.2.2. Sort-Merge Join Algorithm** 116
- **5.2.3. Hash-Based Join Algorithm** 117
- **5.2.4. Comparison** 120

#### 5.3. Parallel Join Algorithms

- **5.3.1. Divide and Broadcast-Based Parallel Join Algorithms** 121
- **5.3.2. Disjoint Partitioning-Based Parallel Join Algorithms** 124

#### 5.4. Cost Models

- **5.4.1. Cost Models for Divide and Broadcast** 128
- **5.4.2. Cost Models for Disjoint Partitioning** 129
- **5.4.3. Cost Models for Local Join** 130
CONTENTS

7. Parallel Indexing 167

7.1. Parallel Indexing—an Internal Perspective on Parallel Indexing Structures 168

7.2. Parallel Indexing Structures 169

- 7.2.1. Nonreplicated Indexing (NRI) Structures 169
- 7.2.2. Partially Replicated Indexing (PRI) Structures 171
- 7.2.3. Fully Replicated Indexing (FRI) Structures 178

7.3. Index Maintenance 180

- 7.3.1. Maintaining a Parallel Nonreplicated Index 182
- 7.3.2. Maintaining a Parallel Partially Replicated Index 182
- 7.3.3. Maintaining a Parallel Fully Replicated Index 188
- 7.3.4. Complexity Degree of Index Maintenance 188

7.4. Index Storage Analysis 188

- 7.4.1. Storage Cost Models for Uniprocessors 189
- 7.4.2. Storage Cost Models for Parallel Processors 191

7.5. Parallel Processing of Search Queries using Index 192

- 7.5.1. Parallel One-Index Search Query Processing 192
- 7.5.2. Parallel Multi-Index Search Query Processing 195

7.6. Parallel Index Join Algorithms 200

- 7.6.1. Parallel One-Index Join 200
- 7.6.2. Parallel Two-Index Join 203

7.7. Comparative Analysis 207

- 7.7.1. Comparative Analysis of Parallel Search Index 207
- 7.7.2. Comparative Analysis of Parallel Index Join 213

7.8. Summary 216

7.9. Bibliographical Notes 217

7.10. Exercises 217

8. Parallel Universal Qualification—Collection Join Queries 219

8.1. Universal Quantification and Collection Join 220

8.2. Collection Types and Collection Join Queries 222

- 8.2.1. Collection-Equi Join Queries 222
- 8.2.2. Collection-Intersect Join Queries 223
- 8.2.3. Subcollection Join Queries 224

8.3. Parallel Algorithms for Collection Join Queries 225

8.4. Parallel Collection-Equi Join Algorithms 225

- 8.4.1. Disjoint Data Partitioning 226
8.4.2. Parallel Double Sort-Merge Collection-Equi Join Algorithm 227
8.4.3. Parallel Sort-Hash Collection-Equi Join Algorithm 228
8.4.4. Parallel Hash Collection-Equi Join Algorithm 232

8.5. Parallel Collection-Intersect Join Algorithms 233
8.5.1. Non-Disjoint Data Partitioning 234
8.5.2. Parallel Sort-Merge Nested-Loop Collection-Intersect Join Algorithm 244
8.5.3. Parallel Sort-Hash Collection-Intersect Join Algorithm 245
8.5.4. Parallel Hash Collection-Intersect Join Algorithm 246

8.6. Parallel Subcollection Join Algorithms 246
8.6.1. Data Partitioning 247
8.6.2. Parallel Sort-Merge Nested-Loop Subcollection Join Algorithm 248
8.6.3. Parallel Sort-Hash Subcollection Join Algorithm 249
8.6.4. Parallel Hash Subcollection Join Algorithm 251

8.7. Summary 252
8.8. Bibliographical Notes 252
8.9. Exercises 254

9. Parallel Query Scheduling and Optimization 256

9.1. Query Execution Plan 257
9.2. Subqueries Execution Scheduling Strategies 259
9.2.1. Serial Execution Among Subqueries 259
9.2.2. Parallel Execution Among Subqueries 261

9.3. Serial vs. Parallel Execution Scheduling 264
9.3.1. Nonskewed Subqueries 264
9.3.2. Skewed Subqueries 265
9.3.3. Skewed and Nonskewed Subqueries 267

9.4. Scheduling Rules 269
9.5. Cluster Query Processing Model 270
9.5.1. Overview of Dynamic Query Processing 271
9.5.2. A Cluster Query Processing Architecture 272
9.5.3. Load Information Exchange 273

9.6. Dynamic Cluster Query Optimization 275
9.6.1. Correction 276
9.6.2. Migration 280
9.6.3. Partition 281

9.7. Other Approaches to Dynamic Query Optimization 284
9.8. Summary 285
10. Transactions in Distributed and Grid Databases

10.1. Grid Database Challenges 292
10.2. Distributed Database Systems and Multidatabase Systems 293
   10.2.1. Distributed Database Systems 293
   10.2.2. Multidatabase Systems 297
10.3. Basic Definitions on Transaction Management 299
10.4. Acid Properties of Transactions 301
10.5. Transaction Management in Various Database Systems 303
   10.5.1. Transaction Management in Centralized and Homogeneous
           Distributed Database Systems 303
   10.5.2. Transaction Management in Heterogeneous Distributed Database
           Systems 305
10.6. Requirements in Grid Database Systems 307
10.7. Concurrency Control Protocols 309
10.8. Atomic Commit Protocols 310
   10.8.1. Homogeneous Distributed Database Systems 310
   10.8.2. Heterogeneous Distributed Database Systems 313
10.9. Replica Synchronization Protocols 314
   10.9.1. Network Partitioning 315
   10.9.2. Replica Synchronization Protocols 316
10.10. Summary 318
10.11. Bibliographical Notes 318
10.12. Exercises 319

11. Grid Concurrency Control

11.1. A Grid Database Environment 321
11.2. An Example 322
11.3. Grid Concurrency Control 324
   11.3.1. Basic Functions Required by GCC 324
   11.3.2. Grid Serializability Theorem 325
   11.3.3. Grid Concurrency Control Protocol 329
   11.3.4. Revisiting the Earlier Example 333
   11.3.5. Comparison with Traditional Concurrency Control
           Protocols 334
CONTENTS

11.4. Correctness of GCC Protocol 336
11.5. Features of GCC Protocol 338
11.6. Summary 339
11.7. Bibliographical Notes 339
11.8. Exercises 339

12. Grid Transaction Atomicity and Durability 341

12.1. Motivation 342
12.2. Grid Atomic Commit Protocol (Grid-ACP) 343
   12.2.1. State Diagram of Grid-ACP 343
   12.2.2. Grid-ACP Algorithm 344
   12.2.3. Early-Abort Grid-ACP 346
   12.2.4. Discussion 348
   12.2.5. Message and Time Complexity Comparison Analysis 349
   12.2.6. Correctness of Grid-ACP 350
12.3. Handling Failure of Sites with Grid-ACP 351
   12.3.1. Model for Storing Log Files at the Originator and Participating Sites 351
   12.3.2. Logs Required at the Originator Site 352
   12.3.3. Logs Required at the Participant Site 353
   12.3.4. Failure Recovery Algorithm for Grid-ACP 353
   12.3.5. Comparison of Recovery Protocols 359
   12.3.6. Correctness of Recovery Algorithm 361
12.4. Summary 365
12.5. Bibliographical Notes 366
12.6. Exercises 366

13. Replica Management in Grids 367

13.1. Motivation 367
13.2. Replica Architecture 368
   13.2.1. High-Level Replica Management Architecture 368
   13.2.2. Some Problems 369
13.3. Grid Replica Access Protocol (GRAP) 371
   13.3.1. Read Transaction Operation for GRAP 371
   13.3.2. Write Transaction Operation for GRAP 372
   13.3.3. Revisiting the Example Problem 375
   13.3.4. Correctness of GRAP 377
13.4. Handling Multiple Partitioning 378
   13.4.1. Contingency GRAP 378
   13.4.2. Comparison of Replica Management Protocols 381
   13.4.3. Correctness of Contingency GRAP 383
13.5. Summary 384
13.6. Bibliographical Notes 385
13.7. Exercises 385

14. Grid Atomic Commitment in Replicated Data 387

14.1. Motivation 388
14.1.1. Architectural Reasons 388
14.1.2. Motivating Example 388

14.2. Modified Grid Atomic Commitment Protocol 390
14.2.1. Modified Grid-ACP 390
14.2.2. Correctness of Modified Grid-ACP 393

14.3. Transaction Properties in Replicated Environment 395
14.4. Summary 397
14.5. Bibliographical Notes 397
14.6. Exercises 398

Part V Other Data-Intensive Applications

15. Parallel Online Analytic Processing (OLAP) and Business Intelligence 401

15.1. Parallel Multidimensional Analysis 402
15.2. Parallelization of ROLLUP Queries 405
15.2.1. Analysis of Basic Single ROLLUP Queries 405
15.2.2. Analysis of Multiple ROLLUP Queries 409
15.2.3. Analysis of Partial ROLLUP Queries 411
15.2.4. Parallelization Without Using ROLLUP 412

15.3. Parallelization of CUBE Queries 412
15.3.1. Analysis of Basic CUBE Queries 413
15.3.2. Analysis of Partial CUBE Queries 416
15.3.3. Parallelization Without Using CUBE 417

15.4. Parallelization of Top-N and Ranking Queries 418
15.5. Parallelization of Cume_Dist Queries 419
15.6. Parallelization of NTILE and Histogram Queries 420
15.7. Parallelization of Moving Average and Windowing Queries 422
15.8. Summary 424
15.9. Bibliographical Notes 424
15.10. Exercises 425
### 16. Parallel Data Mining—Association Rules and Sequential Patterns

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.1. From Databases To Data Warehousing To Data Mining: A Journey</td>
<td>428</td>
</tr>
<tr>
<td>16.2. Data Mining: A Brief Overview</td>
<td>431</td>
</tr>
<tr>
<td>16.2.1. Data Mining Tasks</td>
<td>431</td>
</tr>
<tr>
<td>16.2.2. Querying vs. Mining</td>
<td>433</td>
</tr>
<tr>
<td>16.2.3. Parallelism in Data Mining</td>
<td>436</td>
</tr>
<tr>
<td>16.3. Parallel Association Rules</td>
<td>440</td>
</tr>
<tr>
<td>16.3.1. Association Rules: Concepts</td>
<td>441</td>
</tr>
<tr>
<td>16.3.2. Association Rules: Processes</td>
<td>444</td>
</tr>
<tr>
<td>16.3.3. Association Rules: Parallel Processing</td>
<td>448</td>
</tr>
<tr>
<td>16.4. Parallel Sequential Patterns</td>
<td>450</td>
</tr>
<tr>
<td>16.4.1. Sequential Patterns: Concepts</td>
<td>452</td>
</tr>
<tr>
<td>16.4.2. Sequential Patterns: Processes</td>
<td>456</td>
</tr>
<tr>
<td>16.4.3. Sequential Patterns: Parallel Processing</td>
<td>459</td>
</tr>
<tr>
<td>16.5. Summary</td>
<td>461</td>
</tr>
<tr>
<td>16.6. Bibliographical Notes</td>
<td>461</td>
</tr>
<tr>
<td>16.7. Exercises</td>
<td>462</td>
</tr>
</tbody>
</table>

### 17. Parallel Clustering and Classification

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.1. Clustering and Classification</td>
<td>464</td>
</tr>
<tr>
<td>17.1.1. Clustering</td>
<td>464</td>
</tr>
<tr>
<td>17.1.2. Classification</td>
<td>465</td>
</tr>
<tr>
<td>17.2. Parallel Clustering</td>
<td>467</td>
</tr>
<tr>
<td>17.2.1. Clustering: Concepts</td>
<td>467</td>
</tr>
<tr>
<td>17.2.2. (k)-Means Algorithm</td>
<td>468</td>
</tr>
<tr>
<td>17.2.3. Parallel (k)-Means Clustering</td>
<td>471</td>
</tr>
<tr>
<td>17.3. Parallel Classification</td>
<td>477</td>
</tr>
<tr>
<td>17.3.1. Decision Tree Classification: Structures</td>
<td>477</td>
</tr>
<tr>
<td>17.3.2. Decision Tree Classification: Processes</td>
<td>480</td>
</tr>
<tr>
<td>17.3.3. Decision Tree Classification: Parallel Processing</td>
<td>488</td>
</tr>
<tr>
<td>17.4. Summary</td>
<td>495</td>
</tr>
<tr>
<td>17.5. Bibliographical Notes</td>
<td>498</td>
</tr>
<tr>
<td>17.6. Exercises</td>
<td>498</td>
</tr>
</tbody>
</table>

### Permissions

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>501</td>
</tr>
</tbody>
</table>

### List of Conferences and Journals

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>507</td>
</tr>
</tbody>
</table>

### Bibliography

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>511</td>
</tr>
</tbody>
</table>

### Index

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>541</td>
</tr>
</tbody>
</table>