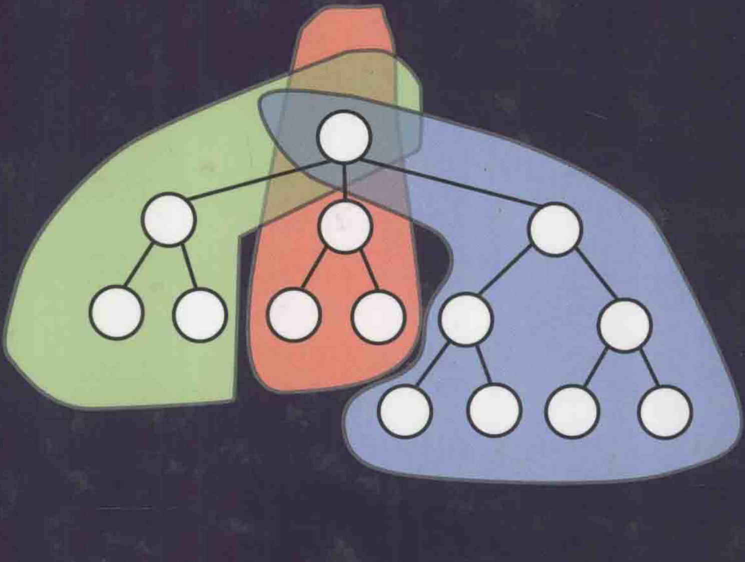
High-Performance Parallel Database Processing and Grid Databases



DAVID TANIAR • CLEMENT H. C. LEUNG

WENNY RAHAYU • SUSHANT GOEL

WILEY

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

江苏工业学院图书馆
藏书章

WILEY

A John Wiley & Sons, Inc., Publication

Copyright © 2008 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400, fax 978-646-8600, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993 or fax 317-572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print, however, may not be available in electronic formats.

Library of Congress Cataloging-in-Publication Data:

Taniar, David.
High-performance parallel database processing and grid databases / by David Taniar, Clement Leung, Wenny Rahayu.
p. cm.
Includes bibliographical references.
ISBN 978-0-470-10762-1 (cloth : alk. paper)
1. High performance computing. 2. Parallel processing (Electronic computers)
3. Computational grids (Computer systems) I. Leung, Clement H. C. II. Rahayu, Johanna Wenny. III. Title.
QA76.88.T36 2008
004'.35—dc22

2008011010

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

High-Performance Parallel Database Processing and Grid Databases

WILEY SERIES ON PARALLEL AND DISTRIBUTED COMPUTING

Editor: Albert Y. Zomaya

A complete list of titles in this series appears at the end of this volume.

此为试读，需要完整PDF请访问：www.ertongbook.com

Preface

The sizes of databases have seen exponential growth in the past, and such growth is expected to accelerate in the future, with the steady drop in storage cost accompanied by a rapid increase in storage capacity. Many years ago, a terabyte database was considered to be large, but nowadays they are sometimes regarded as small, and the daily volumes of data being added to some databases are measured in terabytes. In the future, petabyte and exabyte databases will be common.

With such volumes of data, it is evident that the sequential processing paradigm will be unable to cope; for example, even assuming a data rate of 1 terabyte per second, reading through a petabyte database will take over 10 days. To effectively manage such volumes of data, it is necessary to allocate multiple resources to it, very often massively so. The processing of databases of such astronomical proportions requires an understanding of how high-performance systems and parallelism work. Besides the massive volume of data in the database to be processed, some data has been distributed across the globe in a Grid environment. These massive data centers are also a part of the emergence of Cloud computing, where data access has shifted from local machines to powerful servers hosting web applications and services, making data access across the Internet using standard web browsers pervasive. This adds another dimension to such systems.

Parallelism in databases has been around since the early 1980s, when many researchers in this area aspired to build large special-purpose database machines—databases employing dedicated specialized parallel hardware. Some projects were born, including Bubba, Gamma, etc. These came and went. However, commercial DBMS vendors quickly realized the importance of supporting high performance for large databases, and many of them have incorporated parallelism and grid features into their products. Their commitment to high-performance systems and parallelism, as well as grid configurations, shows the importance and inevitability of parallelism.

In addition, while traditional transactional data is still common, we see an increasing growth of new application domains, broadly categorized as data-intensive applications. These include data warehousing and online analytic processing (OLAP) applications, data mining, genome databases, and multiple media databases manipulating unstructured and semistructured data. Therefore, it is critical to understand the underlying principle of data parallelism, before specialized and new application domains can be properly addressed.

This book is written to provide a fundamental understanding of parallelism in data-intensive applications. It features not only the algorithms for database operations but also quantitative analytical models, so that performance can be analyzed and evaluated more effectively.

The present book brings into a single volume the latest techniques and principles of parallel and grid database processing. It provides a much-needed, self-contained advanced text for database courses at the postgraduate or final year undergraduate levels. In addition, for researchers with a particular interest in parallel databases and related areas, it will serve as an indispensable and up-to-date reference. Practitioners contemplating building high-performance databases or seeking to gain a good understanding of parallel database technology too will find this book valuable for the wealth of techniques and models it contains.

STRUCTURE OF THE BOOK

This book is divided into five parts. Part I gives an introduction to the topic, including the rationale behind the need for high-performance database processing, as well as basic analytical models that will be used throughout the book.

Part II, consisting of three chapters, describes parallelism for basic query operations. These include parallel searching, parallel aggregate and sorting, and parallel join. These are the foundation of query processing, whereby complex queries can be decomposed into any of these atomic operations.

Part III, consisting of the next four chapters, focuses on more advanced query operations. This part covers groupby-join operations, parallel indexing, parallel object-oriented query processing, in particular, collection join, and query scheduling and optimization.

Just as the previous two parts deal with parallelism of read-only queries, the next part, Part IV, concentrates on transactions, also known as write queries. We use the grid environment to study transaction management. In grid transaction management, the focus is mainly on grid concurrency control, atomic commitment, durability, as well as replication.

Finally, Part V introduces other data-intensive applications, including data warehousing, OLAP, business intelligence, and parallel data mining.

ACKNOWLEDGMENTS

The authors would like to thank the publisher, John Wiley & Sons, for agreeing to embark on this exciting journey. In particular, we would like to thank Paul Petralia, Senior Editor, for supporting this project. We would also like to thank Whitney Lesch and Anastasia Wasko, Assistants to the Editor, for their endless efforts to ensure that we remained on track from start to completion. Without their encouragement and reminders, we would not have been able to finish this book.

We also thank Bruna Pomella, who proofread the entire manuscript, for commenting on ambiguous sentences and correcting grammatical mistakes.

Finally, we would like to express our sincere thanks to our respective universities, Monash University, Victoria University, Hong Kong Baptist University, La Trobe University, and RMIT, where the research presented in this book was conducted. We are grateful for the facilities and time that we received during the writing of this book. Without these, the book would not have been written in the first place.

David Taniar
Clement H.C. Leung
Wenny Rahayu
Sushant Goel

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 **33**

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 **83**

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 **92**

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 **96**

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 **104**

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

4.7. Summary **109**
4.8. Bibliographical Notes **110**
4.9. Exercises **110**

5. Parallel Join **112**

5.1. Join Operations **112**
5.2. Serial Join Algorithms **114**

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 **120**

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 **128**

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

此为试读，需要完整PDF请访问：www.ertongbook.com

5.5. Parallel Join Optimization **132**

5.5.1. Optimizing Main Memory 132
5.5.2. Load Balancing 133

5.6. Summary **134**
5.7. Bibliographical Notes **135**
5.8. Exercises **136**

Part III Advanced Parallel Query Processing

6. Parallel GroupBy-Join **141**

6.1. Groupby-Join Queries **141**

6.1.1. Groupby Before Join 142
6.1.2. Groupby After Join 142

6.2. Parallel Algorithms for Groupby-Before-Join Query Processing **143**

6.2.1. Early Distribution Scheme 143
6.2.2. Early GroupBy with Partitioning Scheme 145
6.2.3. Early GroupBy with Replication Scheme 146

6.3. Parallel Algorithms for Groupby-After-Join Query Processing **148**

6.3.1. Join Partitioning Scheme 148
6.3.2. GroupBy Partitioning Scheme 150

6.4. Cost Model Notations **151**
6.5. Cost Model for Groupby-Before-Join Query Processing **153**

6.5.1. Cost Models for the Early Distribution Scheme 153
6.5.2. Cost Models for the Early GroupBy with Partitioning Scheme 156
6.5.3. Cost Models for the Early GroupBy with Replication Scheme 158

6.6. Cost Model for "Groupby-After-Join" Query Processing **159**

6.6.1. Cost Models for the Join Partitioning Scheme 159
6.6.2. Cost Models for the GroupBy Partitioning Scheme 161

6.7. Summary **163**
6.8. Bibliographical Notes **164**
6.9. Exercises **164**

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

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 427

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

17. Parallel Clustering and Classification 464

17.1. Clustering and Classification **464**
17.1.1. Clustering 464
17.1.2. Classification 465
17.2. Parallel Clustering **467**
17.2.1. Clustering: Concepts 467
17.2.2. *k*-Means Algorithm 468
17.2.3. Parallel *k*-Means Clustering 471
17.3. Parallel Classification **477**
17.3.1. Decision Tree Classification: Structures 477
17.3.2. Decision Tree Classification: Processes 480
17.3.3. Decision Tree Classification: Parallel Processing 488
17.4. Summary **495**
17.5. Bibliographical Notes **498**
17.6. Exercises **498**

Permissions 501

List of Conferences and Journals 507

Bibliography 511

Index 541

Part I

Introduction

此为试读，需要完整PDF请访问：www.ertongbook.com