# DATABASE TUNING

Principles, Experiments, and Troubleshooting Techniques

#### DENNIS SHASHA & PHILIPPE BONNET

FOREWORD BY JIM GRAY (MICROSOFT)

### Database Tuning

PRINCIPLES, EXPERIMENTS, AND TROUBLESHOOTING TECHNIQUES

### The Morgan Kaufmann Series in Data Management Systems *Series Editor:* Jim Gray, Microsoft Research

Database Tuning: Principles, Experiments, and Troubleshooting Techniques Dennis Shasha and Philippe Bonnet

SQL:1999—Understanding Relational Language Components Jim Melton and Alan R. Simon

Information Visualization in Data Mining and Knowledge Discovery Edited by Usama Fayyad, Georges G. Grinstein, and Andreas Wierse

Transactional Information Systems: Theory, Algorithms, and Practice of Concurrency Control and Recovery Gerhard Weikum and Gottfried Vossen

Spatial Databases: With Application to GIS Philippe Rigaux, Michel Scholl, and Agnes Voisard

Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design Terry Halpin

Component Database Systems Edited by Klaus R. Dittrich and Andreas Gepper.

Managing Reference Data in Enterprise Databases: Binding Corporate Data to the Wider World Malcolm Chisholm

Data Mining: Concepts and Techniques Jiawei Han and Micheline Kamber

Understanding SQL and Java Together: A Guide to SQLJ, JDBC, and Related Technologies Jim Melton and Andrew Eisenberg

Database: Principles, Programming, and Performance, Second Edition Patrick and Elizabeth O'Neil

*The Object Data Standard: ODMG 3.0* Edited by R. G. G. Cattell and Douglas K. Barry Data on the Web: From Relations to Semistructured Data and XML Serge Abiteboul, Peter Buneman, and Dan Suciu

Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations Ian Witten and Eibe Frank

Joe Celko's SQL for Smarties: Advanced SQL Programming, Second Edition Joe Celko

Joe Celko's Data and Databases: Concepts in Practice Joe Celko

Developing Time-Oriented Database Applications in SQL Richard T. Snodgrass

Web Farming for the Data Warehouse Richard D. Hackathorn

Database Modeling & Design, Third Edition Toby J. Teorey

Management of Heterogeneous and Autonomous Database Systems Edited by Ahmed Elmagarmid, Marek Rusinkiewicz, and Amit Sheth

Object-Relational DBMSs: Tracking the Next Great Wave, Second Edition Michael Stonebraker and Paul Brown with Dorothy Moore

A Complete Guide to DB2 Universal Database Don Chamberlin

Universal Database Management: A Guide to Object/Relational Technology Cynthia Maro Saracco *Readings in Database Systems, Third Edition* Edited by Michael Stonebraker and Joseph M. Hellerstein

Understanding SQL's Stored Procedures: A Complete Guide to SQL/PSM Jim Melton

Principles of Multimedia Database Systems V. S. Subrahmanian

Principles of Database Query Processing for Advanced Applications Clement T. Yu and Weiyi Meng

Advanced Database Systems Carlo Zaniolo, Stefano Ceri, Christos Faloutsos, Richard T. Snodgrass, V. S. Subrahmanian, and Roberto Zicari

Principles of Transaction Processing Philip A. Bernstein and Eric Newcomer

Using the New DB2: IBMs Object-Relational Database System Don Chamberlin

Distributed Algorithms Nancy A. Lynch

Active Database Systems: Triggers and Rules For Advanced Database Processing Edited by Jennifer Widom and Stefano Ceri

Migrating Legacy Systems: Gateways, Interfaces, & the Incremental Approach Michael L. Brodie and Michael Stonebraker

Atomic Transactions Nancy Lynch, Michael Merritt, William Weihl, and Alan Fekete

Query Processing for Advanced Database Systems Edited by Johann Christoph Freytag, David Maier, and Gottfried Vossen

Transaction Processing: Concepts and Techniques Jim Gray and Andreas Reuter Building an Object-Oriented Database System: The Story of  $O_2$ Edited by François Bancilhon, Claude Delobel, and Paris Kanellakis

Database Transaction Models for Advanced Applications Edited by Ahmed K. Elmagarmid

A Guide to Developing Client/Server SQL Applications Setrag Khoshafian, Arvola Chan, Anna Wong, and Harry K. T. Wong

The Benchmark Handbook for Database and Transaction Processing Systems, Second Edition Edited by Jim Gray

Camelot and Avalon: A Distributed Transaction Facility Edited by Jeffrey L. Eppinger, Lily B. Mummert, and Alfred Z. Spector

Readings in Object-Oriented Database Systems Edited by Stanley B. Zdonik and David Maier

## Database Tuning

PRINCIPLES, EXPERIMENTS, AND TROUBLESHOOTING TECHNIQUES

Dennis Shasha courant institute of mathematical sciences new york university

Philippe Bonnet UNIVERSITY OF COPENHAGEN



AN IMPRINT OF ELSEVIER SCIENCE AMSTERDAM BOSTON LONDON NEW YORK OXFORD PARIS SAN DIEGO SAN FRANCISCO SINGAPORE SYDNEY TOKYO PUBLISHING DIRECTOR Diane D. Cerra ASSISTANT PUBLISHING SERVICES MANAGER Edward Wade SENIOR PRODUCTION EDITOR Cheri Palmer SENIOR DEVELOPMENTAL EDITOR Belinda Breyer EDITORIAL ASSISTANT Mona Buehler COVER DESIGN Yvo Riezebos COVER IMAGSE Dr. Mike Hill (Cheetah) and EyeWire Collection (Violin)/gettyimages TEXT DESIGN Frances Baca TECHNICAL ILLUSTRATION Technologies 'n' Typography COMPOSITION International Typesetting and Composition COPYEDITOR Robert Fiske PROOFREADER Jennifer McClain INDEXER TY Koontz PRINTER Edwards Brothers Incorporated

Designations used by companies to distinguish their products are often claimed as trademarks or registered trademarks. In all instances in which Morgan Kaufmann Publishers is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

Morgan Kaufmann Publishers An Imprint of Elsevier Science 340 Pine Street, Sixth Floor San Francisco, CA 94104-3205, USA www.mkp.com

©2003 by Elsevier Science (USA) All rights reserved. Printed in the United States of America

07 06 05 04 03 03 5 4 3 2 1

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, or otherwise—without the prior written permission of the publisher.

Library of Congress Control Number: 2001099791 ISBN 1-55860-753-6

This book is printed on acid-free paper.

To Karen, who always sings in tune D.S.

To Annie, Rose, and to the memory of Louise and Antoine Bonnet P.B.

#### FOREWORD

by Jim Gray, *Microsoft Research* Series Editor, Morgan Kaufmann Series in Data Management Systems

Each database product comes with an extensive tuning and performance guide, and then there are the after-market books that go into even more detail. If you carefully read all those books, then did some experiments, and then thought for a long time, you might write this book. It abstracts the general performance principles of designing, implementing, managing, and use of database products. In many cases, it exemplifies the design trade-offs by crisp experiments using all three of the most popular products (IBM's DB2, Oracle's Oracle, and Microsoft's SQLServer). As such, it makes interesting reading for an established user as well as the novice. The book draws heavily from the author's experience working with Wall Street customers in transaction processing, data warehousing, and data analysis applications. These case studies make many of the examples tangibl2.

For the novice, the book gives sage advice on the performance issues of SQLlevel logical database design that cuts across all systems. For me at least, the physical database design was particularly interesting, since the book presents the implications of design choices on the IBM, Oracle, and Microsoft systems. These systems are quite different internally, and the book's examples will surprise even the systems' implementers. It quantifies the relative performance of each design choice on each of the three systems. Not surprisingly, no system comes out "best" in all cases; each comes across as a strong contender with some blind spots.

The book can be read as a tutorial (it has an associated Web site at *www.mkp.com* /dbtune/), or it can be used as a reference when specific issues arise. In either case, the writing is direct and very accessible. The chapters on transaction design and transaction chopping, the chapters on time series data, and the chapters on tuning will be of great interest to practitioners.

#### CONTENTS

Forewor	ď			ix
Preface				xix
Chapter	1:	BASIC PI	RINCIPLES	1
	1.1	The Po	wer of Principles	1
	1.2	Five Ba	usic Principles	2
		1.2.1	Think Globally; Fix Locally	2
		1.2.2	Partitioning Breaks Bottlenecks	3
		1.2.3	Start-Up Costs Are High; Running Costs	
			Are Low	4
		1.2.4	Render unto Server What Is Due unto Server	5
		1.2.5	Be Prepared for Trade-Offs	6
	1.3	Basic F	Principles and Knowledge	7
Chapter	2:	TUNING	THE GUTS	9
	2.1	Goal o	f Chapter	9
	2.2 Locking and Concurrency Control		9	
		2.2.1	Correctness Considerations	12
		2.2.2	Lock Tuning	16
	2.3	Loggin	g and the Recovery Subsystem	36
		2.3.1	Principles of Recovery	37
		2.3.2	Tuning the Recovery Subsystem	42
	2.4	Operat	ing System Considerations	49
		2.4.1	Scheduling	50
		2.4.2	Database Buffer	52
		2.4.3	How Much Memory Is Economical	54
		2.4.4	Multiprogramming Level	56
		2.4.5	Files: Disk Layout and Access	57
	2.5	Hardwa	are Tuning	59
		2.5.1	Tuning the Storage Subsystem	59
		2.5.2	Enhancing the Hardware Configuration	66

	Bibliogra	uphy	71
	Exercises		74
Chapter	3: IN	DEX TUNING	77
	3.1	Goal of Chapter	77
	3.2	Types of Queries	77
	3.3	Key Types	81
	3.4	Data Structures	81
		3.4.1 Structures Provided by Database Systems	82
		3.4.2 Data Structures for In-Memory Data	87
	3.5	Sparse Versus Dense Indexes	89
	3.6	To Cluster or Not to Cluster	90
		3.6.1 Evaluation of Clustering Indexes	92
		3.6.2 Nonclustering Indexes	96
		3.6.3 Composite Indexes	100
	3.7	Joins, Foreign Key Constraints, and Indexes	102
	3.8	Avoid Indexes on Small Tables	105
	3.9	Summary: Table Organization and Index Selection	105
	3.10	Distributing the Indexes of a Hot Table	110
	3.11	General Care and Feeding of Indexes	111
	Bibliogra	phy	115
	Exercises		116
Chapter	4: TU	NING RELATIONAL SYSTEMS	123
	4.1	Goal of Chapter	123
	4.2	Table Schema and Normalization	124
		4.2.1 Preliminary Definitions	124
		4.2.2 Some Schemes Are Better Than Others	125
		4.2.3 Normalization by Example	127
		4.2.4 A Practical Way to Design Relations	129
		4.2.5 Functional Dependency Test	131
		4.2.6 Tuning Normalization	131
		4.2.7 Tuning Denormalization	136
	4.3	Clustering Two Tables	137
	4.4	Aggregate Maintenance	138
	4.5	Record Layout	140

	4.6	Query	Tuning	143
		4.6.1	Minimizing DISTINCTs	151
		4.6.2	Rewriting of Nested Queries	153
	4.7	Trigge	rs	158
		4.7.1	Uses of Triggers	158
		4.7.2	Trigger Performance	160
	Bibliog	graphy		161
	Exercis	es		162
Chapte	r 5: C	сомми	NICATING WITH THE OUTSIDE	165
	5.1	Talkin	g to the World	165
	5.2	Client	-Server Mechanisms	167
	5.3	Object	ts, Application Tools, and Performance	168
		5.3.1	Beware of Small Objects	169
		5.3.2	Beware of Application Development Tools	171
	5.4	Tuning	g the Application Interface	171
		5.4.1	Avoid User Interaction Within a Transaction	172
		5.4.2	Minimize the Number of Round-Trips	
			Between the Application and the	
			Database Server	172
		5.4.3	Retrieve Needed Columns Only	174
		5.4.4	Retrieve Needed Rows Only	175
		5.4.5	Minimize the Number of Query	
			Compilations	176
	5.5	Bulk I	.oading Data	179
	5.6	Access	ing Multiple Databases	180
	Bibliog	graphy		183
Chapte	r6: C	ASE ST	UDIES FROM WALL STREET	185
	6.1	Techni	ques for Circumventing Superlinearity	185
	6.2	Perfor	m Data Integrity Checks at Input Time	188
	6.3	Distril	oution and Heterogeneity	188
		6.3.1	Interoperability with Other Databases	188
		6.3.2	Global Systems	190
		6.3.3	Managing Connections Socialistically in a	
			Distributed Setting	192

	6.4	Trading Space for Time in History-Dependent		
		Querie	S	193
	6.5	Chopping to Facilitate Global Trades		
	6.6	Cluster	ring Index Woes	195
	6.7	Beware	e the Optimization	195
	6.8	Disaster Planning and Performance		
	6.9	Keepin	g Nearly Fixed Data Up to Date	201
	6.10	Deletio	ons and Foreign Keys	202
	6.11	Partitio	oning Woes: The Hazards	
		of Mea	ningful Keys	203
	6.12	The Pr	oblem of Time	203
		6.12.1	Present Value	203
		6.12.2	Regular Time Series and Statistics	204
		6.12.3	Irregular Time Series and Frequency Counting	205
		6.12.4	Bitemporality	206
		6.12.5	What to Do with Time	207
	Bibliogra	aphy		209
	Exercises			210
Chapter	r7: TR	OUBL	ESHOOTING	213
1		by Albert	o Lerner	
	7.1	Introdu	iction	213
		7.1.1	A Consumption Chain Approach	214
		7.1.2	The Three Questions	217
	7.2	How to	Gather Information: The Tools	217
		7.2.1	Query Plan Explainers	217
		7.2.2	Performance Monitors	221
		7.2.3	Event Monitors	223
		7.2.4	Now What?	224
	7.3	Querie	s from Hell	226
		7.3.1	Finding "Suspicious" Queries	226
		7.3.2	Analyzing a Query's Access Plan	227
		7.3.3	Profiling a Query Execution	229
	7.4	Are DE	BMS Subsystems Working Satisfactorily?	231
		7.4.1	Disk Subsystem	231
		7.4.2	Buffer (Cache) Manager	233

		7.4.3	Logging Subsystem	234
		7.4.4	Locking Subsystem	235
7.	5	Is the <b>I</b>	BMS Getting All It Needs?	236
		7.5.1	Checking on CPU	237
		7.5.2	Checking on Disks and Controllers	237
		7.5.3	Checking on Memory	239
		7.5.4	Checking the Network	239
7.	6	Conclu	sion	240
Bı	iblioz	graphy		240
Chapter 8	: 1	UNING	E-COMMERCE APPLICATIONS	243
8.	1	Goal		243
8.	2	E-com	nerce Architecture	243
8.	3	Tuning	the E-commerce Architecture	246
		8.3.1	Caching	246
		8.3.2	Connection Pooling	249
		8.3.3	Indexing	249
8.	.4	Case St	udy: Shop Comparison Portal	250
8.	5	Capaci	ty Plannin; in a Nutshell	253
		8.5.1	Capacity Planning Essentials	254
		8.5.2	What to Purchase	256
Bi	iblioz	graphy		257
Es	xerci.	ses		258
Chapter 9:	: (	CELKO O	N DATA WAREHOUSES:	
	-	TECHNI	QUES, SUCCESSES,	
		AND MIS	STAKES	261
		by Joe Ce	lko	

9.1	Early History	261	
9.2	Forget What the Elders Taught Ye	<b>Du</b> 262	
9.3	Building a Warehouse Is Hard		
9.4	The Effect on the Bottom Line	268	
	9.4.1 Wal-Mart	269	
	9.4.2 Supervalu	271	
	9.4.3 Harrah's	273	

Chapter 10:	DATA WAREHOUSE TUNING	275
10.1	What's Different About Data Warehouses	275
	10.1.1 Uses of Data Warehouses	276
	10.1.2 Technology for Data Warehousing	278
10.2	Tuning for Customer Relationship	
	Management Systems	289
10.3	Federated Data Warehouse Tuning	294
10.4	Product Selection	295
Biblio	graphy	297
Exerci	ises	299
Appendix A:	REAL-TIME DATABASES	301
A.1	Overview	301
A.2	Replicated State Machine Approach	303
Appendix B:	TRANSACTION CHOPPING	305
<b>B.1</b>	Assumptions	305
<b>B.2</b>	Correct Choppings	307
B.3	Finding the Finest Chopping	312
<b>B.4</b>	Optimal Chopping Algorithm	315
B.5	Application to Typical Database Systems	317
	B.5.1 Chopping and Snapshot Isolation	318
<b>B.6</b>	Related Work	319
Biblio	graphy	323
Appendix C:	TIME SERIES, ESPECIALLY FOR	
	FINANCE	325
C.1	Setting Up a Time Series Database	326
C.2	FAME	327
C.3	S-Plus	329
<b>C.4</b>	SAS	329
C.5	KDB	330
C.6	Oracle 8i Time Series	332
<b>C.</b> 7	Features You Want for Time Series	332
<b>C.8</b>	Time Series Data Mining	333

CONTENTS XVII

Appendix D:	UNDERSTANDING ACCESS PLANS	337
<b>D.1</b>	Data Access Operators	339
D.2	Query Structure Operators	342
D.3	Auxiliary Operators	346
Biblio	graphy	348
Appendix E:	CONFIGURATION PARAMETERS	349
<b>E.1</b>	Oracle	350
<b>E.2</b>	SQL Server	354
E.3	DB2 UDB	355
Glossary		361
Index		387

#### PREFACE

#### **Goal of Book**

Database tuning is the activity of making a database application run more quickly. "More quickly" usually means higher throughput though it may mean lower response time for some applications.

To make a system run more quickly, the database tuner may have to change the way applications are constructed, the data structures and parameters of a database system, the configuration of the operating system, or the hardware. The best database tuners, therefore, can solve problems requiring broad knowledge of an application and of computer systems.

This book aims to impart that I nowledge to you. It has three operational goals.

- 1. To help you tune your application on your database management system, operating system, and hardware.
- 2. To give you performance criteria for choosing a database management system, including a set of experimental data and scripts that may help you test particular aspects of systems under consideration.
- 3. To teach you the principles underlying any tuning puzzle.

The best way to achieve the first goal is to read this book in concert with the tuning guide that comes with your particular system. These two will complement each other in several ways.

 This book will give you tuning ideas that will port from one system to another and from one release to another. The tuning guide will mix such general techniques with system- and release-specific ones.

- This book embodies the experience and wisdom of professional database tuners (including ourselves) and gives you ready-made experimental case study scripts. Thus, it will suggest more ideas and strategies than you'll find in your system's tuning guide.
- This book has a wider scope than most tuning guides since it addresses such issues as the allocation of work between the application and the database server, the design of transactions, and the purchase of hardware.

>NOTE TO TEACHERS Although this book is aimed primarily at practicing professionals, you may find it useful in an advanced university database curriculum. Indeed, we and several of our colleagues have taught from this book's predecessor, *Database Tuning: A Principled Approach* (Prentice Hall, 1992).

Suppose your students have learned the basics of the external views of database systems, query languages, object-oriented concepts, and conceptual design. You then have the following choice:

- For those students who will design a database management system in the near future, the best approach is to teach them query processing, concurrency control, and recovery. That is the classical approach.
- For those students who will primarily use or administer database management systems, the best approach is to teach them some elements of tuning.

The two approaches complement each other well if taught together. In the classical approach, for example, you might teach the implementation of B-trees. In the tuning approach, you might teach the relevant benefits of B-trees and hash structures as a function of query type. To give a second example, in the classical approach, you might teach locking algorithms for concurrency control. In the tuning approach, you might teach techniques for chopping up transactions to achieve higher concurrency without sacrificing serializability.

We have tried to make the book self-contained inasmuch as we assume only a reading knowledge of the relational query language SQL, an advanced undergraduate-level course in data structures, and if possible, an undergraduate-level course in operating systems. The book discusses the principles of concurrency control, recovery, and query processing to the extent needed.

If you are using this book as a primary text for a portion of your classes, we can provide you with lecture notes if you e-mail us at *shasha@cs.nyu.edu* or *bonnet@ diku.dk*.

#### What You Will Learn

Workers in most enterprises have discovered that buying a database management system is usually a better idea than developing one from scratch. The old excuse—"The performance of a commercial system will not be good enough for my application" has given way to the new realization that the amenities offered by database management systems (especially, the standard interface, tools, transaction facilities, and data structure independence) are worth the price. That said, relational database systems often fail to meet expressability or performance requirements for some data-intensive applications (e.g., search engines, circuit design, financial time series analysis, and data mining). Applications that end up with a relational database system often have the following properties: large data, frequent updates by concurrent applications, and the need for fault tolerance.

The exceptional applications cited here escape because they are characterized by infrequent bulk updates and (often) loose fault tolerance concerns. Further, the SQL data model treats their primary concerns as afterthoughts.

But many notable applications fall within the relational purview: relational systems capture aggregates well, so they are taking over the data warehouse market from some of the specialized vendors. Further, they have extended their transactional models to support e-commerce.

Because relational systems are such an important part of the database world, this book concentrates on the major ones: DB2, SQL Server, Oracle, Sybase, and so on. On the other hand, the book also explores the virtues of specialized systems optimized for ordered queries (Fame, KDB) or main memory (TimesTen).

You will find that the same principles (combined with the experimental case study code) apply to many different products. For this reason, examples using one kind of system will apply to many.

After discussing principles common to all applications and to most data base systems, we proceed to discuss special tuning considerations having to do with Web support, data warehousing, heterogeneous systems, and financial time series. We don't discuss enterprise resource planning systems such as SAP explicitly, but since those are implemented on top of relational engines, the tuning principles we discuss apply to those engines.

#### How to Read This Book

The tone of the book is informal, and we try to give you the reasoning behind every suggestion. Occasionally, you will see quantitative rules of thumb (backed by