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

This Page Intentiona Left Blank

Database Tuning

PRINCIPLES, EXPERIMENTS, AND TROUBLESHOOTING TECHNIQUES

Dennis Shasha
COURANT INSTITUTE OF MATHEMATICAL SCIENCES
NEW YORK UNIVERSITY

Philippe Bonnet
UNIVERSITY OF COPENHAGEN

MORGAN KAUFMANN PUBLISHERS
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.

This Page Intentiona Left Blank

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 tangible.

For the novice, the book gives sage advice on the performance issues of SQL-level 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.

This Page Intentiona Left Blank

CONTENTS

This Page Intentiona Left Blank

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 knowledge 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

This Page Intentiona Left Blank