

**DATA BASED
DOWNSIZING**

**A Case Study that
Exposes the Pitfalls!**

DATA BASED ADVISOR[®]

For Developing Desktop Database Applications

March 1992 \$3.95 Canada \$4.95

SPEED UP THAT APP

Tuning Tips, Tactics, and Trade-offs for:

- AREV
- Clarion
- Clipper
- DataEase
- dBASE IV
- EB
- FoxPro
- Interbase
- Oracle
- Paradox
- R:BASE
- SQL Server
- SQLBase
- SQLWindows

***Plans for growing your
VAR business***

***Nantucket's RDDs:
why they're
so late...***

(see page 134)



THE EXPERT SERIES

PARADOX

Options for Maintaining Indexes

More options for systems administrators—adding and deleting secondary indexes.

By Tim Colling 142

CLIPPER

Shady Characters

Add shadows to windows and boxes without writing C code.

By Mike Schinkel 144

WINDOWS

COMMon Problems

Trouble with Windows communications? Here are some solutions.

By Kenn Nesbitt 146

FOXPRO

Learning By Example

Use FoxPro's power tools to create a simple, scrolling list—and learn how the language works.

By Y. Alan Griver 148

dBASE IV

Using UDFs to Generate Reports

In much the same way you use UDFs to enhance editing features, use them to generate reports.

By Michael Liczbanski and Susan Perschke 150

TEST DRIVES

SOFTWARE UPDATE: Dr. Switch Adds Bells and Whistles

When you want to make your Xbase application a TSR, Dr. Switch-ASE can do. By Jeffrey Bersin

18

SOFTWARE UPDATE: MemoPlus Takes Care of Pesky Memos

Fix, pack, or modify your memo fields with this handy set of utilities.

By James Powell 20

FIELD REPORT: Clipper Developers Pack First Russian DevCon

An interesting perspective of the first Nantucket DevCon held in Russia.

By Michail Mikhailov 22

SOFTWARE PREVIEW: What's new in dBASE IV 1.5

A quick rundown of the new commands and functions.

By Michael Liczbanski and Susan Perschke 24



Page 58

DEPARTMENTS

FEEDBACK

Letters from our readers. 16

SOFTWATCH

26

DB CONNECTIONS

Keeping Up With the Pack

Stay on top of what's current with CIS and the DBADVISOR Forum.

By Bill House 158

EXPRESS LANE

161

USER GROUP

167

ADVERTISING INDEX

169

CALENDAR OF EVENTS

168

CLIENT/SERVER ADVISOR

FRONT END TOOLS

Looking Beneath the Surface

With so many front ends on the market, it's hard to choose. Here's a list of considerations that go beyond the interfaces...

By David McGovern 58

CORPORATE COMPUTING

The Time It Takes

Don't get caught up in the hype; allow extra time in laying the foundations for future client/server projects.

By Bob Zurek 62

MEMO TO MANAGEMENT

Client/Server Applications as Corporate Assets

What makes an application valuable? Here are some important application design considerations.

By Tipton Cole 64

WINDOWS FRONT ENDS

Ease Windows Development with SQLWindows

It's no secret that Windows is easy on users and hard on developers. Take a look at what Gupta's SQLWindows has done to improve developers' outlooks.

By Ramesh Balwani 66

SERVER TUNING TIPS

Emerald Bay

By John F. Hay 70

Interbase

By Michael Masterson 74

Oracle

By David McGovern 76

SQLBase

By Douglas Goddard 78

Microsoft SQL Server

By Kelly Gillespie 80

DATA BASED DOWNSIZING

The Ups and Downs of Downsizing

Here's how one development team used C to downsize a "corporate-critical" application from a mainframe to PCs.

By James Foye 82

How to Specify a SQL Server Sort Order

88

SYSTEMS INTEGRATION

Remote Client/Server Computing

By Kelly Gillespie 90

SERVER SCENE

The latest database server news. 93

PACKETS

Late-breaking client/server news. 94

TUNING TIPS

Oracle Server



By David
McGoveran

Relational database application tuning has a number of dimensions, each of which needs to be understood and exploited to achieve the desired effect. There are many possible goals in tuning a database server: performance, concurrency, memory usage, storage allocation, etc. Each of these involve trade-offs. For example, it may be possible to achieve some levels of performance only at the expense of concurrency or some levels of concurrency only at the expense of database integrity guarantees.

Oracle Server for OS/2 offers a wide variety of tuning opportunities and a number of facilities for the purpose. Tuning can be performed at various levels of "granularity" for a particular SQL statement, a critical module, a particular application, some portion of the database, or globally (as for throughput, storage demands, etc.). As a result, tuning Oracle Server requires planning and experience. Here are a few tips for doing that planning and gaining that experience, most of which could be applied to any database server with some minor changes.

► **1.** Throw as much memory and as many disk drives at the problem as you can afford. Oracle Server likes resources, and essentially uses all that are available. On the other hand, don't expect too much. Oracle Server for OS/2 will support about 10 concurrent users at high performance in practice (despite TPC benchmark numbers) before something gives.

► **2.** Try to reduce the granularity of the tuning problem. Profilers can be used to determine critical code segments and point to high cost database operations. Oracle's tuning facilities can help identify specific statements that are performing poorly or determine if additional memory would help.

► **3.** Plan for and develop tuning metrics, both within the application and independent of it. The first can be done with special routines within the applica-

tion that are conditionally compiled or that are "enabled" via a command line argument or a special command file opened when the application is invoked. The second can generally be done with SQL*Plus scripts.

► **4.** Prioritize the goals, and reduce the granularity of the problem as much as possible. Don't get overly involved in the details if what you need is system throughput. And don't spend time optimizing every SQL statement if one statement is consuming all the resources. Think globally, act locally.

► **5.** Study the *Database Administrator's Guide* and the *Advanced Performance Tuning Guide* to become familiar with the various tuning facilities and techniques. Read between the lines. Like all vendors, Oracle isn't eager to state that certain functionality doesn't perform well. Think through each suggestion, and question its implications. The more you understand about how Oracle works internally, the better you can tune Oracle Server. Also, remember that even the best Oracle expert may not understand your specific application requirements. You're probably best qualified to determine if advice is appropriate or how it needs to be interpreted. Among the facilities you should be familiar with are the following tables: V\$ROWCACHE, V\$WAITSTAT, X\$KCBCBH, X\$KCBRBH, and PLAN_TABLE. Also be familiar with SQL*DBA, the SQL trace facility, TKPROF, and EXPLAIN PLAN. In addition, familiarity with the system catalog and INIT.ORA parameters is essential.

► **6.** When tuning SQL statements, remember that phrasing is critical to Oracle Server. Unlike most RDBMS optimizers, Oracle's optimizer is syntax-sensitive. It examines table order in the FROM clause to determine the order of processing, and doesn't use a beneficial index on a column or columns unless they're explicitly referenced in the WHERE clause. The column must not be used inside a function. Because the optimizer doesn't decide between alternative methods of processing a SQL statement, it may use an index that it shouldn't. In this case, the index can be disabled by preventing the optimizer from recognizing a column of the index

by concatenating a NULL or a space a character column or adding zero to numerical column.

► **7.** If a critical request needs access to one or two more columns than are in a useful index, try adding these to the index. During reads, only the index need be accessed. This is especially beneficial if indexed access results in a large amount of random data block I/O.

► **8.** Avoid the use of NULL wherever possible. NULLs introduce anomalies in SQL, and make it hard to interpret results. They usually disable the use of an otherwise beneficial index.

► **9.** Denormalization may be necessary to reduce the number of tables in common joins. Oracle doesn't respond well to more than five tables in a single SQL statement. However, you might be able to avoid denormalization by creating intermediate results in temporary tables and subsequently joining these intermediate results.

► **10.** Distribute data. Place rollback segments and redo log files on separate disk drives to reduce drive contention. Proper sizing can reduce storage sag and dynamic extension of storage space. The number of rollback segments is important to concurrency.

► **11.** Tune the INIT.ORA parameters (DB_BLOCK_WRITE_BATCH and DB_BLOCK_MAX_SCAN_CNT) to improve multi-block writes. You can also tune DB_FILE_MULTIBLOCK_READ_COUNT to improve multi-block reads. Some of the other important parameters to consider include the integrity/concurrency trade-off (with ROW_LOCKING and SERIALIZABLE), the number of ROLLBACK_SEGMENTS, and statistics gathering parameters (avoid these if performance is important).

► **12.** Look out for "deadlocks," especially if you aren't using the transaction processing option. Oracle treats any process waiting for a resource as though it is possibly involved in a deadly embrace. Resource waits can be used to identify high contention resources. Requests for high contention resources should receive special tuning attention. If this doesn't work, try staggering requests for these resources, further nor-

malization, or even duplicating the resource (assuming you can do so without endangering database integrity).

► **13.** Use SEQUENCE generators to eliminate application processing and therefore unnecessary requests to the database server. Note that you must trade-off between speed due to caching of sequence numbers and the fact that entries in cache are lost in an instance failure.

► **14.** Tune the System Global Area (SGA). The SGA should fit into real memory, but the larger the buffer cache and data dictionary cache, the less disk I/O.

Editor's Note: The material in this article was adapted from a chapter on performance tuning in the author's book, *An Advanced Guide to Client/Server Applications*, in preparation. Copyright 1992 by David McGoveran. All Rights Reserved.

David McGoveran is president of Alternative Technologies in Santa Cruz, CA. The firm specializes in relational databases. David can be reached at (408) 425-1859. □

The Information Management Group

Training Programs and Consulting Services

As an outgrowth of our systems consulting practice, The Information Management Group provides end-user and developer training on major PC and Macintosh database systems. We emphasize "learning by doing". At our Chicago training facility, students are equipped with workstations, and are guided through system functions and application examples. We also offer custom on-site training and application development workshops.

For More Information,
Call (312) 280-1007.

**Ask About Our
Mac Courses!**

The Information Management Group • 720 N. Franklin • Suite 300 • Chicago IL • 60610

FoxPro 2.0
Courses

End-User Interactive

Two days: focuses on FoxPro 2.0 interactive features, with an in-depth review of database management, report-writing, and an introduction to menu and screen building.

Developer Basics

Two days: designed for entry-level programmers, this course reviews FoxPro commands & functions and programming tools in a structured development setting.

Developer Intermediate

Three days: for developers familiar with the basics of FoxPro programming, this course provides an in-depth review of FoxPro 2.0 tools and development techniques.

Developer Advanced

Two days: for experienced FoxPro developers, with the focus on optimizing application performance and creating state-of-the-art FoxPro 2.0 application interfaces.

FoxPro 2.0 Power Tools

Two days: for FoxPro developers, this course focuses exclusively on 2.0 power tools, including "snippet" programming techniques and user interface design.

BUSINESS OPPORTUNITY...

**Wanted. VARs, Programmers, and Resellers to
Market Medical & Dental Software Royalty-Free.
Income Potential of 5 or 6 Figures to Full-Time VARs!**

DEMO SYSTEM

\$1000 ea.

100% Functional,
Limited to 250
Patients.

So Good,
You Can SELL
From This DEMO!

Foreign orders include C.O.D. shipping

- Complete source code (45,000+ lines) in easy to modify FoxBASE+/FoxPro!
- Benefit from our 5 years of experience, save 1000's of research and development hours!
- Current user base of 100's of happy doctors with systems costs ranging from \$2995 to over \$20,000!
- VARs and doctors vote this system the best on the market at any price!
- Declared the finest user interface created for medical management.
- 1990's technology utilizes windows, macros, arrays, UDF's and scrolling lookups!
- Addresses 1 million patients, 99 doctors, and produces 1,000's of reports!
- FAST!! Locate 1 patient out of 100,000 in less than 1/2 second!
- Integrated general ledger, full security system, and multi-tasking capability!
- Clinical features unavailable in any other package, BUILT-IN!
- Full LAN support and complete desktop management system with 10 pop-up features!
- This market is HOT and recession resistant... successful VARs install 50+ systems per year!
- Full VAR sales support: color brochures, demonstration video and MORE!



Complete Software, 45,000+ Lines of Source Code, 485 Page Manual,
Royalty - Free License, For a Small 1 Time Investment



16601 BLANCO, SUITE 206 • SAN ANTONIO, TEXAS 78252
512/492-2225 • FAX 512/492-2827

Top-Rated Medical
& Dental Software
FOXMed
FOXdent