

WELCOME TO ISSUE 3, 2011



Already a subscriber?

Turn the page or click here for instructions.

Like what you're reading?

Share it with a colleague now!

Not a subscriber?

Click here to sign up now and enjoy these exclusive digital-edition benefits:

- › Search through the entire issue
- › Access back-issue archives
- › Share articles with others
- › Download a PDF copy

IBM

ISSUE 3, 2011 / VOLUME 16 NO. 3

data management

IBM.COM/DMMAGAZINE

KNOWLEDGE. PERFORMANCE. RESULTS.



ANSWER

Press the
answer
button

WHAT MAKES THE IBM NETEZZA
DATA WAREHOUSE APPLIANCE TICK?

FASTER THAN FAST

Tested: Informix Warehouse Accelerator

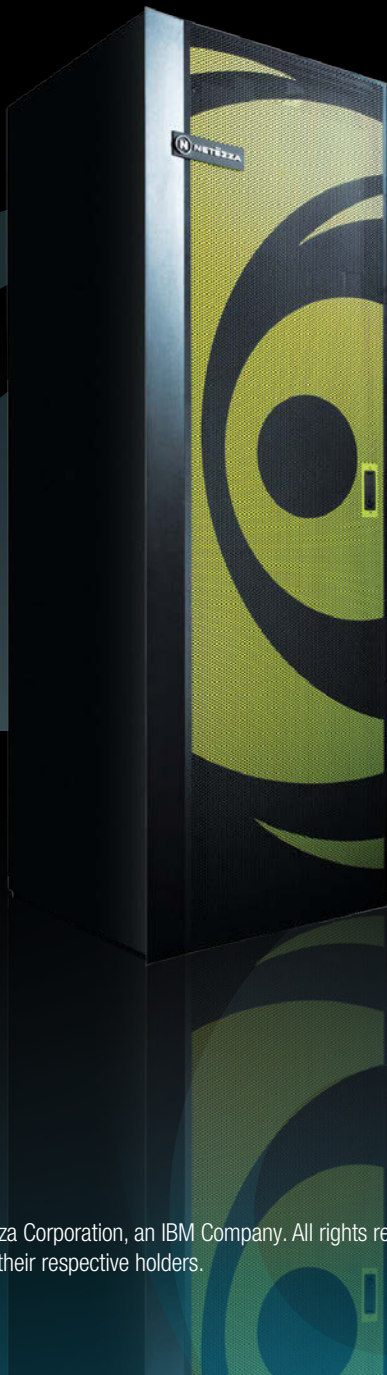
DATA PROS MEAN BUSINESS

Developer-DBAs go way beyond tuning

SOLID-STATE SURPRISE

Is your DBMS working harder than
it needs to?

It's a simple concept:
When you get answers faster,
you can ask more questions.



IBM Netezza: The Simple Appliance Built for Serious Analytics

IBM Netezza high-performance data warehouse appliances are purpose-built to make advanced analytics on big data simpler, faster and more accessible. By bringing simplicity to advanced analytics, we help companies maximize the full potential of their data in ways that were previously unthinkable—fueling entirely new, game-changing ways of working.

To learn more and find out how you can test drive an IBM Netezza data warehouse appliance on your site, with your data, visit www.netezza.com.



data management

KNOWLEDGE. PERFORMANCE. RESULTS.

Features

Press the Answer Button

The Netezza appliance looks simple on the outside.

What amazing tech lies within?

10



27

Your New Best Friend

Solid-state storage may change how you work with data

31

Go Back in Time

Temporal tables in DB2 10 for z/OS

Sponsor Article

Maximize Data Analytics Performance.....**23-26**
ibm.com/software/data/informix/warehouse

18

Modeling Trains with SPSS

Can you predict when a train will break down? SPSS and DB2 can.



Departments

- 2 **Editor's Note**
By Cameron Crotty
- 4 **NewsBytes**
- 8 **IIUG User View**
By Stuart Litel
- 9 **IDUG User View**
By Jeremy Dodd
- 35 **Data Architect**
The Rise of the Developer-DBA
By Robert Catterall
- 37 **Distributed DBA**
Automatic Storage Databases
By Roger E. Sanders
- 40 **Programmers Only**
PLANS + COLLECTIONS + VERSIONS + PACKAGES + DBRMs = Confusion
By Bonnie Baker
- 42 **Informix DBA**
Informix Warehouse Accelerator
By Lester Knutsen
- 44 **Smarter is...**
Saving Minds. Saving Lives
By David Geer

Sponsor Index

- Applied Analytix, Inc. **17**
applied-analytix.com/tm1checkup
- DBI **C3, 17**
dbisoftware.com/smarter
- IBM..... **C4**
ibm.com/facts
- IBM Information On Demand **3, 17**
ibm.com/events/InformationOnDemand
- International DB2 Users Group (IDUG)..... **13**
idug.org
- Melissa Data..... **17, 29**
melissadata.com/myibm
- Netezza..... **C2**
netezza.com
- Niteo Partners..... **17**
niteo.com/SmartPredictHC
- QueBIT **17**
quebit.com
- Quest Software **7**
quest.com/ToadDB2forDBAs

IBM data management

KNOWLEDGE. PERFORMANCE. RESULTS.

EDITOR

Cameron Crotty
editor@tdagroup.com

MANAGING EDITOR

Stephanie S. McLoughlin

ART DIRECTOR

Iva Frank

DESIGNER

Lalaine Gagni

CONTRIBUTING WRITERS

Bonnie Baker, Howard Baldwin, Robert Catterall, Jeremy Dodd, Marc Fiammante, David Geer, Michael Kearney, Lester Knutsen, Stuart Litel, Grégory Neuvéglise, Paul Pendle, Sasirekha Rameshkumar, Roger E. Sanders

EDITORIAL BOARD OF DIRECTORS

Jeannie Cramer, Jeff Jones, Steve Miller, Jennifer Reese, Bernie Spang

AD SALES

Jill Thiry
advertise@tdagroup.com

AD COORDINATORS

Katherine Hartlove, Kim Johnsen

SPECIAL THANKS TO

Lea Anne Bantsari

SUBSCRIPTION SERVICES AND REPRINTS

To subscribe to the print or digital version of *IBM Data Management* magazine, change your address, or make other updates to your information, please go to ibm.com/dmmagazine. For instant access to the *IBM Data Management* magazine digital edition, visit ibmdmmagazinedigital.com. For information about reprints, please send an e-mail to customerservice@tdagroup.com.

IBM and the IBM logo are registered trademarks of the International Business Machines Corporation and are used by TDA Group under license.

Material published in *IBM Data Management* magazine copyright © 2011, International Business Machines. Reproduction of material appearing in *IBM Data Management* magazine is forbidden without prior written permission from the editor.



PRESIDENT

Paul Gustafson

VICE PRESIDENT, STRATEGY AND PROGRAMS

Nicole Sommerfeld

VICE PRESIDENT, EDITORIAL DIRECTOR

Debra McDonald

VICE PRESIDENT, CONTENT SERVICES

Paul Carlstrom

Printed in the U.S.A.

EDITOR'S NOTE / CAMERON CROTTY

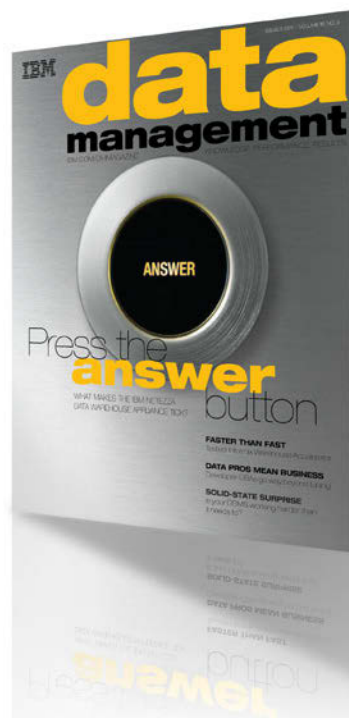
IBM.COM/DMMAGAZINE

It's all about prioritizing, or triage, or just getting things done. No matter how you say it, you make hundreds of decisions each day about what's important and what isn't. The goal is to focus the maximum possible energy on the things that really matter. In business, what matters are answers, and our cover story digs into how the IBM Netezza data warehouse appliance is built to deliver those answers.

In the magazine business, what matters is information, and we rely on our incredible team of writers to deliver that information. In this issue, we say a fond farewell to Dave Beulke, who has written the IDUG User View column for many years—we'll miss Dave's voice and insight. Taking over the IDUG column is Jeremy Dodd, the incoming International DB2 Users Group (IDUG) president. Please join me in welcoming Jeremy!

As always, send your thoughts, comments, suggestions, brainstorm, stories, and article ideas to us at editor@tdagroup.com, and subscribe to our electronic edition at ibmdmmagazinedigital.com.

Cameron Crotty



Subscribe

- ▶ Print
- ▶ Digital
- ▶ Web
- ▶ E-mail



Scan the code and subscribe now
ibmdmmagazinedigital.com

IBM Software

Information On Demand 2011

Turn Insight Into Action

October 23–27, 2011

Mandalay Bay | Las Vegas, Nevada

Please join us at Information On Demand 2011, October 23 - 27 in Las Vegas to learn how IBM can help you optimize business performance and achieve breakaway results. Discover the latest IBM products and solutions and receive the very best technical and business education. You'll want to be among the more than 10,000 expected attendees learning new and innovative strategies to accelerate information initiatives and gain the practical know-how to maximize the value of your solutions.

HIGHLIGHTS

- More than 700 Technical Education sessions
- Industry-focused Business & IT Leadership sessions
- Featuring Hardware, Software and Services Solutions
- 300 Customer speakers
- Complimentary Certification Tests
- IBM's largest EXPO
- IBM and Industry-renowned speakers
- Networking opportunities

TOP 5 REASONS TO ATTEND!

- 1 Improve your skills -**
Get deep technical education and the best strategic insight and analysis
- 2 Learn what's new -**
Explore the latest advances in IBM Information Management, Business Analytics, and Enterprise Content Management software and solutions
- 3 Get best practices -**
Hear from more than 300 industry leaders
- 4 Experience unrivaled networking -**
Interact with peers, industry experts, Business Partners and IBM executives
- 5 Take action -**
Make a direct impact on your organization with insights and actions you can quickly put to work for immediate return

Register Today - Save \$300 with the Early Bird rate!

Visit the website today to register or for more information: ibm.com/events/InformationOnDemand

Roundup: What's New in the InfoSphere

Terabyte pricing,
ready-to-run Warehouse Packs,
and Streams 2.0



You could have seen this coming. As organizations collect more information digitally, they're wrestling with the looming issue of "big data"—high volume, high velocity, and wildly varied information about customers, business processes, and operations that must be stored, analyzed, and integrated. To help, IBM is climbing into the ring with some new muscle for its InfoSphere integration, analytics, and data warehouse tools. These tools give businesses the takedown tactics they need, both financially and technologically.

Processor pricing punched

It's now clear that in many cases, paying by processor for a data warehouse limits organizations' abilities to exploit new hardware innovations and processor technologies. In response, IBM is offering the latest version of InfoSphere Warehouse (9.7.3) under a simpler

terabyte-pricing option. This option allows organizations to pay only for the volume of compressed user data they manage in their warehouse. The benefit: a transparent, predictable pricing structure that ties pricing more closely to data, not hardware.

In other developments, data professionals can now deploy a virtual image of the InfoSphere Warehouse Departmental Edition on any Intel/AMD-based platform. This allows DBAs to bypass install and setup activities, reducing time to value and thus freeing up DBA resources to focus on activities that can provide competitive advantage. And InfoSphere Warehouse 9.7.3 now provides five user licenses of Cognos 10 for reporting against InfoSphere Warehouse Cubing Services.

Productivity pumped up

IBM is also giving data professionals an edge in their game with new analysis tools. InfoSphere

Streams 2.0 includes Streams Processing Language enhancements that improve productivity and agility; new database support for the IBM Netezza data warehouse appliance, Microsoft SQL Server, and MySQL; health and metrics monitoring of current system status via an XML-based programmatic interface; new capabilities for application modularity designed to support larger and/or distributed application development teams; the freedom to nest and aggregate data types within a single stream definition; and enhancements in InfoSphere Streams Studio to speed writing, testing, and debugging Streams Processing Language applications.

In conjunction with InfoSphere Streams, InfoSphere BigInsights helps data professionals manage and analyze Internet-scale volumes of structured and unstructured data. Built on the open source Apache Hadoop software framework, BigInsights adds useful

Informix: Speed Thrills

New capabilities accelerate development and boost performance

No one ever complained about a database that was too fast. IBM Informix has goosed its systems in two key areas: development and queries.

On the IDE side, Informix Genero V2.32 lets developers tackle application modernization easily and take advantage of simple business logic to speed development and time to market. Genero V2.32 delivers both cost and time savings by helping developers transform legacy code into a 21st-century online transaction processing (OLTP) environment, whether it's browser based, uses a GUI, or targets mobile or cloud environments.

On the query side, Informix Ultimate Warehouse Edition (UWE) features Informix Warehouse Accelerator and Informix Flexible Grid technologies, and can accelerate warehouse queries as much as 100 times faster than previously possible, while at the same time lowering data warehouse costs. UWE includes support for unlimited data storage, processors, and memory, as well as deep compression that can reduce disk space usage as much as 80 percent while further enhancing performance.

› MORE INFORMATION

Informix Genero: ibm.com/support/docview.wss?uid=swg27020968

IBM Informix Ultimate Warehouse Edition: ibm.com/software/data/informix/ultimate-warehouse-edition

administrative, workflow, provisioning, and security features, along with sophisticated analytical capabilities from IBM Research. BigInsights is now available as a free Basic Edition and an Enterprise Edition that includes additional support and IBM extensions. Together, Streams and BigInsights deliver a more developer- and user-friendly solution for complex, large-scale analytics.

Packs prebuilt and prepped

When end users are anxious to start attacking problems, data professionals can get them closer to the starting bell with IBM InfoSphere Warehouse Packs for both InfoSphere Warehouse Advanced Enterprise Edition and InfoSphere Warehouse Advanced Workgroup Edition. These new packs give data professionals a data warehouse with prebuilt data models; extract, transform, and

load (ETL) guidance based on those data models; and more than 60 prebuilt Cognos business intelligence reports, all ready to rumble. Once the warehouse is installed and data loaded, users of the advanced edition get immediate access to reports and dashboards in such areas as customer churn, market basket analysis, and vendor performance.

IBM InfoSphere Warehouse Advanced Edition incorporates all of the latest advances of InfoSphere Warehouse, including the new virtualization deployment option in the Departmental Edition, and new Optim backup and recovery tools (for more on those tools, see the next section).

Recovery and restore rolled out

Organizations that want to beef up their backup and recovery tools have three new moves as

part of the new InfoSphere Advanced Recovery Solutions. First, DB2 Merge Backup provides an automated backup tool that can dramatically decrease the time required to complete full data backups. Then, on the recovery side, DB2 Recovery Expert assists DBAs in accelerating the recovery process if an unplanned outage occurs. Finally, InfoSphere Optim High Performance Unload provides a rapid data extraction tool that can significantly reduce restoration and migration times for large volumes of data.

Even as big data gets bigger, IBM InfoSphere tools help data professionals cut it down to size.

› MORE INFORMATION

ibm.com/bigdata

ibm.com/software/data/infosphere

ibm.com/software/data/optim



IBM Partner News

DataKinetics I/O Optimizer, Dead Ahead!

Does your company live and die by Java Database Connectivity (JDBC) performance? IBM partner DataKinetics unveiled the Java to DB2 Optimizer at last May's International DB2 Users Group (IDUG) conference.

The product has a two-pronged goal: decrease the data access request time as well as the load on DB2 from Java access. To develop the optimizer, DataKinetics developers first reduced the number of I/Os to DB2 for read-only data and then reduced the load on DB2 by replacing dynamic SQL queries with stored procedures. Finally, using DataKinetics connectors, they further offloaded DB2 by providing an efficient buffering of the results of the SQL query, eliminating the need to re-drive DB2 SQL queries to obtain additional results. The improvements help minimize costly I/O and can deliver both faster database performance and faster system performance.

MORE INFORMATION
dki.com

Eliminate InfiniBand Limitations

One of the big challenges in data centers—especially as new architectures affect their hierarchical layers—is arranging high-speed connections to accommodate cabling lengths. IBM partner Obsidian Strategics has developed technology, called Longbow, that allows an InfiniBand fabric, normally a short-range network used in high-performance computing, to extend via optical fiber over varying distances, including campus, metro, and global networks.

IBM has now validated the Obsidian Longbow technology on DB2 pureScale database clusters. By combining Longbow with IBM Geographically Dispersed DB2 pureScale Cluster (GDPC), organizations can distribute pureScale servers and underlying mirrored general parallel file system (GPFS) storage across remote sites.

For the first time, this connection allows each half of the cluster to maintain synchronization across metro-area distances without a significant performance penalty—even when the links are encrypted. If a site-wide failure occurs, the surviving site always has a mirrored, accurate image of the database.

MORE INFORMATION
obsidianstrategics.com
ibm.com/db2/purescale

IBM Automates Private Cloud Provisioning

The more you can automate rote tasks, the more time you can apply to the real challenges. To help make you more productive, IBM has developed workload patterns—full stacks of software tailored for a particular purpose and designed to make deployment go faster. Newly available in June 2011 is the IBM Workload Deployer Pattern for DB2 Workgroup Server Edition V1.0, providing capabilities for the automated provisioning and management of database infrastructure for data-centric applications in a secure, private cloud.

Once IT configures the database pattern in the Workload Deployer appliance (previously known as WebSphere CloudBurst Appliance), the application group can quickly and easily deploy a database with a few mouse clicks to support departmental online transaction processing (OLTP) applications. Workload Deployer enables IT to register hardware for the private cloud on which IBM Workload Deployer Pattern for DB2 Workgroup Server Edition deploys the database infrastructure. The Workload Deployer appliance then tracks resource usage and requirements in the private cloud.

MORE INFORMATION
ibm.com/software/webservers/workload-deployer/#

Financing and Technical Help for Oracle Migrations

If you're investigating alternatives to Oracle—and many are in these turbulent times—IBM is now providing extensive technical, business, and financial assistance. For example, you can receive a no-charge, detailed financial analysis of the business case for switching from Oracle and a no-charge, customized technical evaluation and conversion plan. IBM is also offering side-by-side cost/performance comparisons between Oracle and IBM software at an IBM Software Lab or client site. IBM has added more than 100 online and in-person training courses to quickly bring Oracle data professionals and developers up to speed on IBM software. Finally, IBM Global Financing is offering 0 percent financing on IBM software, with no hardware purchase required. To find out if your organization qualifies, send a message to askdata@ca.ibm.com.

MORE INFORMATION
Financing: ibm.com/financing/us/lifecycle/acquire/itsoftware.html
Free Information Management workshops: ibm.com/developerworks/data/bootcamps
Free two-day session, IBM DB2 Workshop for Oracle Professionals (includes free certification): ibm.com/developerworks/wikis/display/im/IBM+DB2+Workshop
Free four-day session, IBM DB2 9.7 Bootcamp and Oracle to DB2 9.7 Migration Clinic (includes free certification): ibm.com/developerworks/wikis/display/im/DB2+9.7+Bootcamp+and+Oracle+to+DB2+9.7+Migration+Clinic

Education and Books

Education

New DB2 for z/OS Certification

DB2 10 System Administrator for z/OS: Plan, install, manage, tune, and secure DB2 for z/OS environments.

> MORE INFORMATION

ibm.com/certify/tests/ovr617.shtml

Pre-Conference Certification Crammer Courses at Information On Demand 2011

IBM will offer full-day "crammer" courses on Sunday, October 23, 2011, from 8:30 a.m. to 4:30 p.m. at the Mandalay Bay Convention Center. Take advantage of preferred attendee pricing of US\$499 and learn from professional IBM instructors and Information Champions.

> MORE INFORMATION

ibm.com/software/data/education/iod2011.html?ca=dmmag

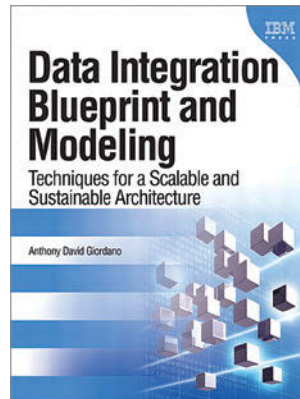
New Information Management Training Courses

See the latest Information Management course offerings from IBM and updates about existing courses at this dedicated site.

> MORE INFORMATION

ibm.com/software/data/education/courses.html?ca=dmmag

Books



Data Integration Blueprint and Modeling: Techniques for a Scalable and Sustainable Architecture, by Anthony David Giordano

Data integration now accounts for a major part of the expense and risk of typical data warehousing and business intelligence projects—and, as businesses increasingly rely on analytics, the need for a blueprint for data integration is greater now than

ever. Author Anthony David Giordano, leader of the IBM Global Data Integration Practice for Business Intelligence (BI), writes on the important topic of improving data integration, presenting a clear, consistent approach to defining, designing, and building data integration components to reduce cost, simplify management, enhance quality, and improve effectiveness.

> MORE INFORMATION

ibmpressbooks.com/title/0137084935

The Power of IBM DB2® at Your Fingertips

Simplify the Management of Your Complex DB2 Environment With Toad®



Toad® for IBM DB2 simplifies SQL development and administration, automates code optimization and delivers at-a-glance catalog browsing. And DB2 object management and reporting? It's a snap.

More than one million Toad users can't be wrong – after all, Toad also supports Oracle, SQL Server, Sybase, MySQL and other database platforms.

Learn about the best features of Toad for DB2 in "The Top 10 Things a DBA Should Know About Toad for DB2" at:

www.quest.com/ToadDB2forDBAs





Stuart Litel
is president of
the International
Informix Users

Group (IIUG; iiug.org/president), CTO of Kazer Technologies (kazer.com), an IBM Gold Consultant, member of the IBM Data Champion Inaugural 2008 class, and recipient of the 2008 IBM Data Professional of the Year award.

Happy 10th Anniversary to IBM and Informix

Ten years later, it's still the technology that matters

Two of the biggest days in my professional life occurred in 2001. On April 24, IBM announced it was buying the assets of the database operation of Informix Corporation, including software and personnel. On July 1, the transfer of the assets to IBM was completed. How have the last 10 years been for the Informix world?

I know what most of you think is coming next. You're thinking that I'm about to go off on a rant about Informix being just one of many products at IBM, and how much different that is from the old Informix Corporation, which revolved around one main product, and how Informix is now lost in a sea of more than 1,000 software products with almost no advertising.

And maybe there's a part of me that would like to say that. But here's the thing: *it doesn't matter*. When you pick a technology product, you don't buy it for the marketing. You buy it for what needs and problems you have to solve. And when it comes to technology and Informix, IBM has put its money where its mouth is, and Informix is still way out in front.

In fact, let's call it what it is: complete and utter domination. It's possible that Informix has produced more successful technological milestones in the last 10 years than in the 20 before IBM bought it. It's been released on more platforms (and with fewer bugs) faster than ever before. The Informix technology team has truly outdone

itself, from creating one of the fastest and most reliable online transaction processing (OLTP) engines out there to developing the latest Ultimate Warehouse Edition.

Back in 1984 when I first heard of Informix, my company went to see what the industry had to offer. We set up tests that fit what we were trying to accomplish, and we chose the product that best met our needs. I am absolutely convinced that if we had to do the same tests for the same situation today, Informix would still be the product we would pick.

This past May, the fourth annual IIUG Informix Conference had more than 85 tech sessions. As I read through the conference surveys, it seems we were a big hit. Dr. David Ferrucci, the principal investigator for the recent Watson *Jeopardy!* project, was a keynote speaker. I was concerned about starting his keynote late in the afternoon, thinking that people would have only dinner on their minds. Boy was I wrong.

When it comes to technology and Informix, IBM has put its money where its mouth is.

David's session went long, but nobody wanted to leave. We just sat there and listened. This was truly an eye-opening session. Thank you, David, for coming to speak and thank you, Ambuj Goyal, for helping us get David. And most of all, congratulations to David on your success with this project and on becoming an IBM Fellow.

Congratulations are also in order to the four newest members of the IBM 2011 Information Champions club from the Informix community: Andrew Ford, David Link, Bruce Simms, and Kernoal Stephens. For more on the champions, visit ibm.com/developerworks/champion.

Finally, it was more than five years ago when my phone rang with a call from a buddy of mine, David Beulke, who was president of the International DB2 Users Group (IDUG). David said that someone from some IBM magazine wanted to know if I would write an article about Informix. I talked to the editor and took on the task. I mean, what did I know about writing? (I still don't know much, but have had some great editors.) Well, the rest is history as one article turned into a regular column (and a regular headache for the editors of this magazine dealing with me). So my last thank you is to David Beulke for making me do something I never thought I could do and sharing these pages with me every quarter. Next time we are together, the drinks are on me! *

A Little Help from Your Friends

If you want to learn more about DB2, IDUG forums are the place to be



Jeremy Dodd

is president of the International DB2 Users Group (IDUG) and assistant vice president at Barclays Capital with over 20 years of relational database experience and more than 10 working with DB2 for Linux, UNIX, and Windows (LUW).

The IDUG North America DB2 Tech Conference in Anaheim was quite a week. Seeing DB2 users sharing their experiences with each other was a great way to begin my time as president of IDUG. Before I continue, I would like to thank David Beulke, a past president of IDUG, for writing this article on behalf of IDUG for a number of years.

It was clear in Anaheim that DB2 10 for z/OS is foremost in the thoughts of many organizations. One useful element of this version is the ability to skip a release, as a significant number of organizations run DB2 8.

IDUG has launched a forum for discussing DB2 10 for z/OS migrations, including best practices and skip release experiences. This is just one example of how the DB2 community offers peer-to-peer help and assists all users in their careers. A number of industry experts have agreed to contribute to this forum, ensuring that there will be great information: Joe Borges, John Campbell, Ian Cook, Chris Crone, Michael Dewert, Gareth Jones, Jeff Josten, Roger Miller, Bryan Paulsen, Suresh Sane, and Julian Stuhler. Come visit the forum on idug.org, and here's a taste of an early discussion (posts edited for length and clarity).

Forum Member: *We are what I would classify as a young vanilla DB2 user and we had no issues migrating from DB2 7 to DB2 8. Here is my dilemma... Leapfrog DB2 9 to*

DB2 10 so we can get current (won't do that until at least nine months after DB2 10-GA), which will give us immediate CPU savings and all the other goodies and then have more time to take advantage of the new features and educate everyone? Or keep my job safe by going to DB2 9?

Julian Stuhler: *[...] If you're well established on DB2 8 and have the resource/approval for an upgrade project in the very short term, it might be worth going to DB2 9 first while the DB2 10 codebase continues to mature. [...] If it's going to take you some months to get the resource/approval for the upgrade, or if you're in need of specific features in DB2 10, then the skip migration might be for you. The critical factor is the announced EOS [end of service] date for DB2 8, which is April 30, 2012. If you're not off DB2 8 by then, you're going to have to talk to IBM about extended support. [...]*

Roger Miller: *I tried to put my flavor of choices into a paper, and I often append it to migration presentations, like this one: <ftp://ftp.software.ibm.com/software/data/db2/zos/presentations/migration/db2-10-migration-planning-webcast-2011-feb-miller.pdf>*

Early migrations need to use best practices for service and testing. We have been keeping changes on these pages: ibm.com/developerworks/data/bestpractices/db2zos

Bjarne Nelson: *Unless your (or the company's) risk appetite would fit being an early adopter of a new release of DB2, my take would be that you can alone justify the migration from DB2 8 to 9 based on normal software life cycle needs. [...] The business case for going through another software life cycle shortly thereafter could then be easily justifiable based on CPU savings and all the nice new features being made available in DB2 10. You will anyway have to learn and adapt new features and functions in both DB2 9 and 10, so perhaps a step-by-step approach would be the right way forward considering all factors.*

IDUG has launched a forum for discussing DB2 10 for z/OS migrations, including best practices and skip release experiences.

This is just one of many conversations on a wide range of topics that you'll find in the IDUG forums. It's the place to be if you want to learn more about DB2. I'm looking forward to seeing you there! *



Press the
answer
button

By Michael Kearney

WE GO DEEP INSIDE THE NETEZZA
DATA WAREHOUSE APPLIANCE TO
FIND OUT WHAT MAKES IT TICK.



a refrigerator isn't complicated to use. Neither is a dishwasher nor a clothes dryer. These appliances have one or two simple controls, they don't need a lot of maintenance, and they do one thing better than any other type of device.

●● The Netezza appliance integrates database software, servers, and storage into a single system.

Making a data warehouse that is so focused and low maintenance that it can fairly be called an appliance is no easy task, but the IBM Netezza data warehouse appliance has proven that this approach works well.

The best way to really appreciate an appliance is to pull it apart and see what makes it go. In this article, you'll take a tour of the Netezza appliance from the inside out and learn some of the ways that it advances the state of the art in data warehousing and analytics.

How to design an integrated appliance

The Netezza appliance integrates database software, servers, and storage into a single system. Data is loaded through standard extract, transform, and load (ETL) connectors, and is then available for queries from business intelligence and analytics applications.

Inside the appliance are two major components: the host servers, which manage the connection between the appliance and the outside world, and the Snippet Blades (S-Blades), which handle the database heavy lifting.

The host servers use symmetric multiprocessing (SMP) and run a SQL relational database management system (RDBMS) on Linux. Their active/passive configuration provides full redundancy should one server fail. Whichever server is active presents a standardized interface (ANSI SQL, JDBC, ODBC) to external tools and applications. It also compiles SQL queries into executable code segments called *snippets*, distributes the snippets for execution, and creates optimized query plans.

The S-Blades are processing nodes that consist of an independent server containing several multicore CPUs, multiengine field programmable gate arrays (FPGAs), and RAM. Depending on its size, a single Netezza appliance can hold more than a hundred S-Blades. Together, the fleet of S-Blades represents a massively parallel processing (MPP) grid behind an SMP front end. This two-tier architecture is called Asymmetric Massively Parallel Processing (AMPP), as shown in Figure 1.

The host servers and S-Blades are connected to a series of high-density, high-performance, highly redundant, independent disk arrays. A high-speed network fabric running a customized IP-based protocol and optimized to scale to more than a thousand nodes connects all of the hardware components. Each node can transfer data sets simultaneously to every other node. All of the hardware components are hot-swappable.

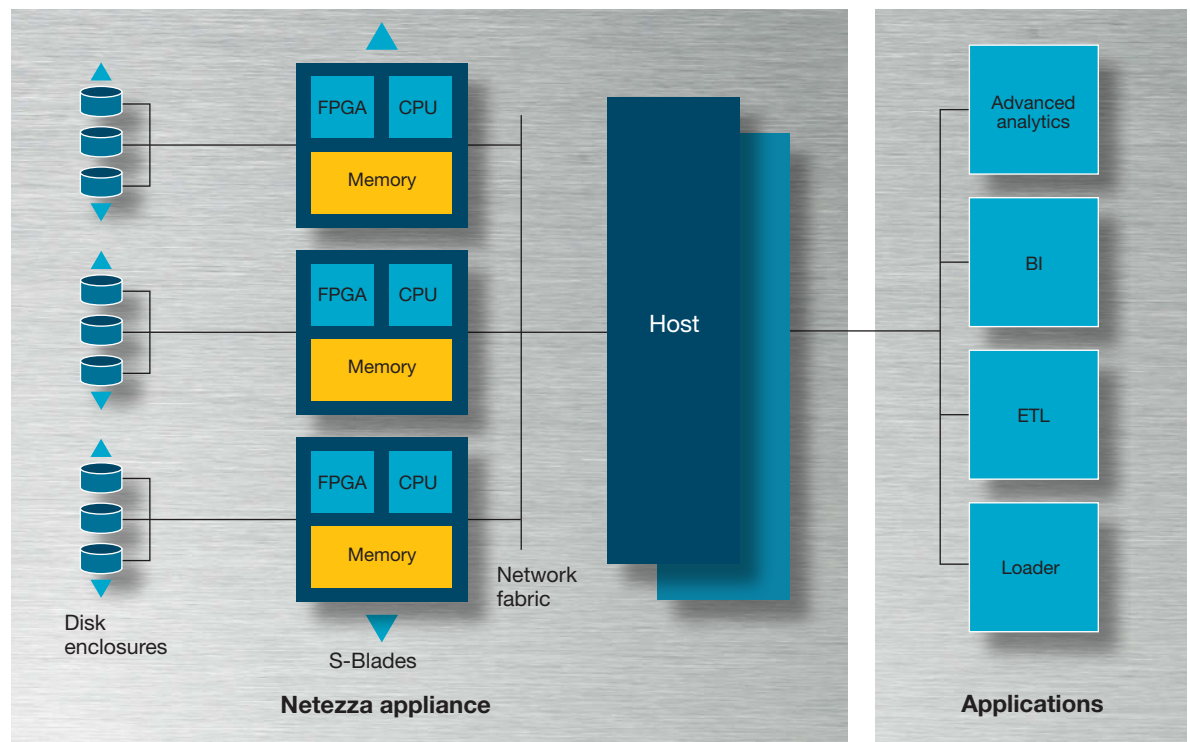
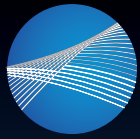


Figure 1: Netezza AMPP architecture



IDUG

The Worldwide
DB2 User Community

Mentor Program Expanded

Now even more people qualify as IDUG Mentors!

Have you been to **three** IDUG DB2 technical conferences? That qualifies you to bring a first-time attendee with you at a huge 80% discount.

A week of technical presentations and education for less than \$400!

Get all the details at www.IDUG.org/Mentor

Looking for the Tech Conference?

It's back – **at IDUG!** An IDUG DB2 Tech Conference is where DB2 is The MAIN EVENT, not a side show. Through these independent events, attendees get user-delivered, unbiased DB2 content featuring:

- User-driven technical sessions to sharpen skills
- Best practices to drive efficiencies and optimize investments
- Direction on new and existing database and related product issues
- Candid discussions with customers, IBM experts, developers and DB2 consultants

IDUG DB2 Tech Conferences

Melbourne, Australia | 14-16 September 2011

Prague, Czech Rep. | 13-18 November 2011

Denver, Colorado | 14-18 May 2012

MENTORING

CERTIFICATION

DB2
EXPERTS

NETWORKING

EDUCATION

Be a Part of the IDUG Community Strength in Numbers

more than **17,000** members representing **100** countries globally
thousands of DB2 user experiences
more than a **handful** of in-person events year-round

IDUG is an independent, user-driven community that helps DB2 users enhance their professional skills and network, drive efficiencies in their business, and optimize their DB2 technologies.

Experience IDUG year-round.
Sign-up is FREE at www.IDUG.org



●● The Netezza DBMS does not use indexes. Query plans are based on up-to-date statistics.

Processing a query

The action begins when an application sends a query to the appliance host server, which compiles the query and creates an optimized query execution plan. That may sound like a familiar process, but the Netezza appliance takes some very unfamiliar paths.

First, the Netezza database management system (DBMS), a highly optimized engine for the massively parallel Netezza environment, does not use indexes. The logic behind this design decision is that indexes add value only when the DBA can predict which queries will be run against the database. For truly exploratory analytics work, such predictions can be nearly impossible, as data analysis is iterative and unpredictable: ask a question; consider the response; ask a better question. The only reason to repeat questions exactly is to confirm or check the results of the analysis.

Instead, the optimizer gathers up-to-date statistics on every database table referenced in a query. The optimizer combines these statistics with detailed performance knowledge of the specific system components, making it possible to accurately measure the disk, processing, and network costs associated with an operation. The resulting query plans are based on data, rather than heuristics, and are thus extremely efficient. By using these statistics, the optimizer minimizes disk I/O and data movement, two factors that reduce performance in a data warehouse system. Another critical benefit of the non-indexed approach is that the Netezza appliance doesn't need to be tuned at all.

As part of the transformation, the optimizer also determines the correct table join order, rewrites expressions, and eliminates any redundancy from the SQL. The optimizer can use its knowledge of the appliance components to determine the best join order in a complex join. For example, when joining multiple small tables to a large fact table, the optimizer can choose to broadcast the small tables in their entirety to each of the S-Blades, while keeping the large table distributed across all snippet processors. This approach minimizes data movement while taking advantage of the AMPP architecture to parallelize the join.

Netezza in retail



Retail can be a tough market for data warehouse technology. Retailers operate on thin margins, and when they invest in technology they expect it

to work—no ifs and no buts. Their industry is highly competitive, so to succeed the warehouse must help the business by bringing new, value-creating applications rapidly into production.

Many data warehouses don't fully succeed: in their most recent *Magic Quadrant for Data Warehouse Database Management Systems* published on January 28, 2011, Gartner analysts estimate "Nearly 70% of data warehouses experience performance-constrained issues of various types."¹ Warehouses running on older database technology make it difficult, not easy, to develop new applications:

Converting and broadcasting the query

After the optimizer has done its work, the compiler converts the query plan into executable code segments called *snippets*. When compiling queries, the host server uses a feature called the *object cache* to accelerate query performance. This is a large cache of previously compiled snippet code that supports parameter variations. For example, a snippet with the clause `WHERE name = 'bob'` might use the same compiled code as a snippet with the clause `WHERE name = 'jim'` but with settings that reflect the different name.

Snippets are then sent to the S-Blades to be executed in parallel. To ensure efficient handling, a scheduler runs on the host servers, balancing execution across complex workloads to meet the needs of multiple users, while maintaining high utilization and throughput. The scheduler considers a number of factors—including query priority, size, and resource availability—in determining when to execute snippets on the S-Blades. Like the optimizer, the scheduler uses the appliance architecture to gather metrics about resource availability from each component of the system. Sophisticated algorithms enable the scheduler to maximize system throughput, utilizing nearly 100 percent

●● When compiling queries, the host server uses a cache of previously compiled code to save time. Code segments are then executed in parallel.

of the disk bandwidth, while ensuring that memory and network resources are not overloaded—a common cause of thrashing. When the scheduler gives the green light, the snippet is broadcast to the snippet processors, which reside on the S-Blades.

In addition to the scheduler on the host servers, the snippet processors have their own smart preemptive scheduler that allows snippets from multiple queries to execute simultaneously. To decide when and for how long to schedule a particular snippet for execution, the scheduler takes into account the priority of the query and the resources set aside for the user or group that issued it.

57 percent of respondents surveyed for a report published by The Data Warehousing Institute (TDWI)² in July 2011 expressed frustration with backlogs of requests for new business intelligence applications and IT's inability to satisfy new requests in a timely manner.

Retailers have been quick to embrace the Netezza appliance. “We are a very data-driven company, and we need to make information available to the business faster to drive decision making. We were experiencing performance issues with our legacy BI [business intelligence] platform and were looking to speed up our ability to crunch through and analyze the vast amounts of new data we generate daily,” says Ed Macri, vice president of advertising and business intelligence at CSN Stores. With the

Netezza appliance, CSN can answer in seconds or minutes questions that previously took hours or days.³ At another large U.S. retailer, slow data load jobs were causing problems, because their indexes needed to be destroyed and rebuilt each time new data was loaded. The Netezza appliance does not use indexes, and the retailer was able to reduce load times from more than five hours to less than three minutes. In these and many other cases, Netezza is helping retailers get answers faster.

¹ Complimentary copies of the Gartner report are available at netezza.com/gartnermq/2011.

² Claudia Imhoff and Colin White, *Self-Service Business Intelligence: Empowering Users to Generate Insights* (The Data Warehousing Institute, July 1, 2011).

³ IBM, “CSN Stores Enhances Shopping Experience with IBM Data Warehouse Appliance,” news release, January 11, 2011, netezza.com/releases/2011/release011111_2.aspx.

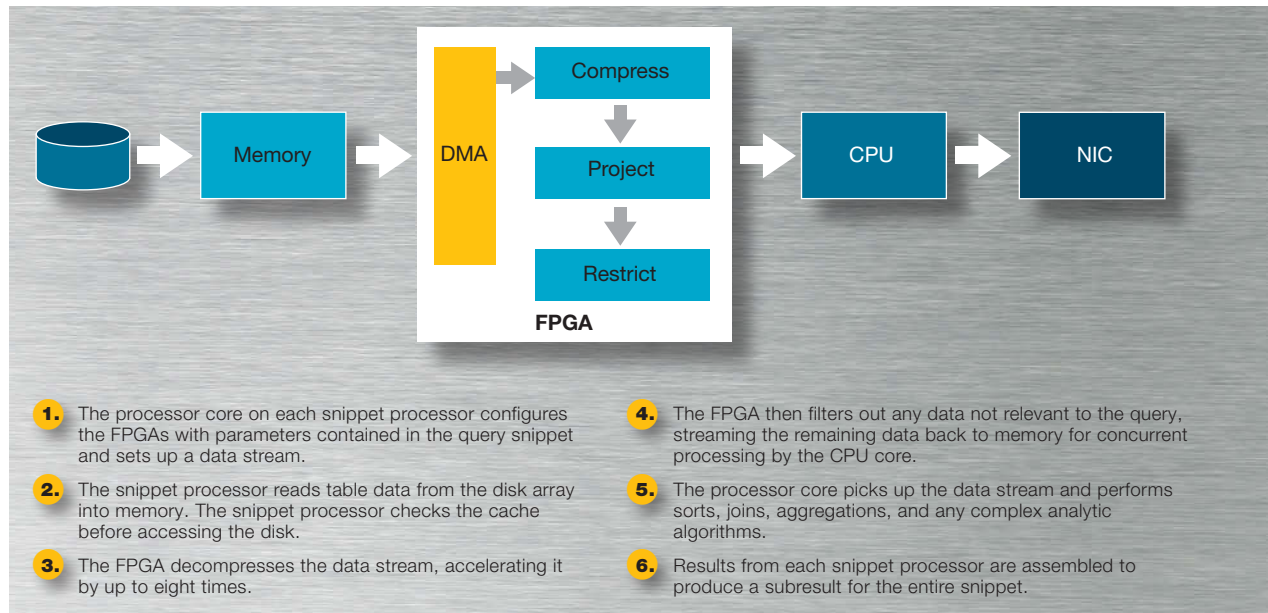


Figure 2: Inside a Netezza S-Blade, showing the software engines in the FPGA

Hot hardware SQL action at the S-Blade

Each snippet sent by the host server arrives at the S-Blade with two elements: compiled code to be executed by individual CPU cores, and a set of FPGA parameters used by four virtual engines that accelerate the overall processing of the query (see Figure 2). The Compress engine decompresses data streamed from disk. The Project and Restrict engines then reduce the scope of data that the rest of the query will act on, first eliminating columns not specified in the `SELECT` or `JOIN` clauses, and then removing rows based on restrictions specified in the `WHERE` clause. Finally, the Visibility engine filters other rows that should not be “seen” by a query—such as those locked by a transaction not yet committed—and maintains atomicity, consistency, isolation, and durability (ACID) compliance.

A processor core then picks up the uncompressed, filtered data block and performs fundamental database operations such as sorts, joins, and aggregations. It also applies any complex algorithms embedded by an analytics application. All this happens in parallel on hundreds of snippet processors, depending on the model of the appliance. Every node then sends its result over the network fabric to other S-Blades or the active host server, which communicates the resulting data set out from the appliance to the requesting application.

All snippet processors now have snippet results that must be assembled. The snippet processors use the intelligent network fabric to communicate flexibly with the host

servers and with each other to perform intermediate calculations and aggregations. The host server assembles the intermediate results received from the snippet processors, compiles the final result set, and returns it to the user’s application. Meanwhile, other queries are streaming through the system at various stages of completion.

One appliance, many answers

The Netezza system combines advanced hardware technology with careful study of how to maximize database performance in a purpose-built and optimized environment. The result is truly an appliance: simple to operate, easy to maintain, and better than just about anything else at what it does. Need answers? Just drop in your data and press the button. ✱

Michael Kearney is a product marketing director at Netezza, an IBM company.

RESOURCES

IBM Redguide, *The Netezza Data Appliance Architecture*:

www.redbooks.ibm.com/abstracts/redp4725.html?Open

Netezza appliances:

netezza.com/data-warehouse-appliance-products

Netezza and Oracle Exadata comparison:

netezza.com/exadata-netezza-compared

IBM invites the readers of *IBM Data Management* magazine to visit the Business Partner sites listed below.



QueBIT®

For effective forecasting, insurance companies need to align financial and actuarial planning processes. The challenge of modeling both underwriting and calendar years is critical.

QueBIT, an IBM Premier Partner, has the expertise in both IBM Cognos TM1 and the insurance industry to deliver positive results.

Learn more at www.quebit.com



Patient Care and Meaningful Use Dashboards

SmartPredict™: an award-winning solution providing information integration, performance dashboards and scenario-based analytics for healthcare companies to:

- Provide relevant information to the care team
- Increase the efficiency of the care team
- Improve patient satisfaction and quality of care
- Demonstrate meaningful use

www.niteo.com/SmartPredictHC



DBI Software is the leader in database performance tuning and optimization.

Our unique products enable DBAs to adopt a proactive methodology—escaping the trap of simple “reactive” tools and hardware addiction. Please also join us for our free educational webinar series *The DB2Night Show™*.

For more information visit www.DBISoftware.com



We are the TM1 Experts!

Is it time for your TM1 application to get a performance tune-up? Do you want to learn how to leverage your investment into new areas of your business? We have over 15 years of TM1 experience. Schedule a TM1 checkup, including a recommendations report, at www.applied-analytix.com/tm1checkup.

Register today and save \$300 with the early bird rate!

IBM Software

Information On Demand 2011

Turn Insight Into Action

October 23-27, 2011

Mandalay Bay | Las Vegas, Nevada

ibm.com/events/InformationOnDemand



Need to clean or enrich your database?

Get the IBM developer tools that make the job easy. Get your free trial at www.MelissaData.com/IBM

MELISSA DATA®

Your Partner in Data Quality



Modeling Trains



How sequential analysis can predict the future, and why DB2 might be smarter than you think

with SPSS

By Marc Fiammante and Grégory Neuvéglise



BY THEIR NATURE, TRAINS INVITE IMAGINATION. THEY CONJURE visions of exotic travel and long-distance commerce. But trains are also large, complicated, and expensive to repair when they break unexpectedly. For a railway operator, the ability to predict a failure before it happens would be tremendously valuable.

Recently, an IBM team worked with a European train manufacturer to develop exactly this capability, using IBM SPSS Modeler and IBM DB2 to analyze data generated by the trains. This article will show how to use SPSS to identify patterns in data. It also will show how IBM DB2 made an unexpected contribution to the project—handling a set of moderately complex mathematical calculations directly within SQL.

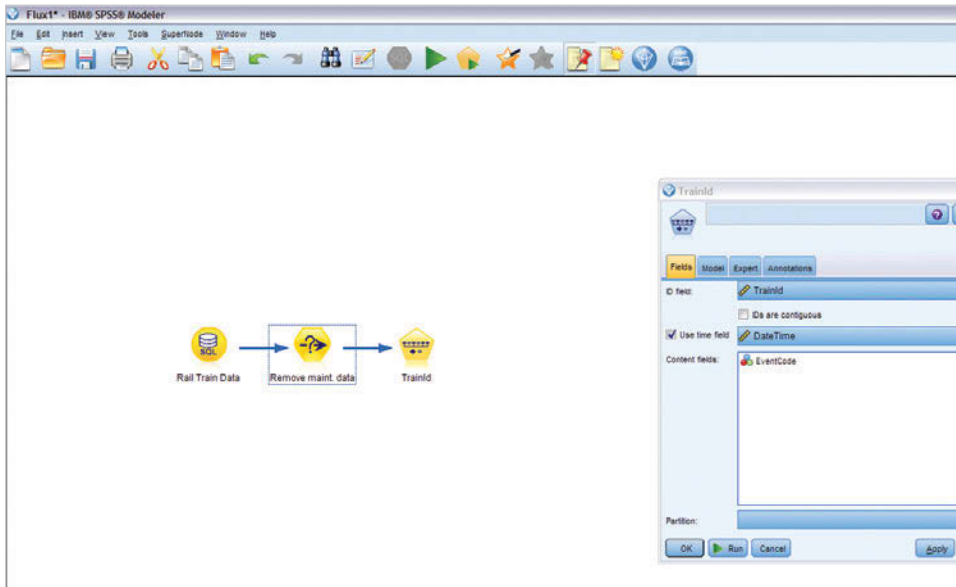


Figure 1: The ID field indicates which train is related to a sequence of events, the Time field describes the sequence order, and the Content field contains the code of the event that has occurred.

The screenshot shows the UNIT rule sets interface. It has tabs for 'Model', 'Settings', 'Summary', and 'Annotations'. The 'Model' tab is active, showing a table of rules. The table has columns for 'Antecedent', 'Consequent', 'Support %', and 'Confidence %'. The table is sorted by 'First Antecedent' and shows 12,358 rules. The first few rows are highlighted in yellow.

Antecedent	Consequent	Support %	Confidence %
21382	18232	51.923	96.296
21208	18232	55.769	93.103
21382	13202	53.046	92.857
13202	21336	100.0	92.300
21382	8335	100.0	86.538
21382	34201	100.0	82.692
21382	21336	82.692	79.07
34201	34201

Figure 2: The rule sets interface, showing the results of the analysis. The sequence node found 12,358 rules.

The screenshot shows the UNIT rule sets interface with a filter applied. The table shows rules for antecedent '2201' and consequent '2201'. Below the table, there are filter settings: 'consequent include any of 2201' (checked), 'antecedent include any of' (unchecked), 'confidence Above 60.0' (checked), and 'antecedent support Above 0.0' (unchecked).

Antecedent	Consequent	Support %	Confidence %
2201	2201	75.0	76.923
2201	2201	90.305	87.234
2201	2201	100.0	84.615

Figure 3: A filter applied for event code 2201. One rule shows that 87.2 percent of the time, event code 2201 was preceded by two occurrences of event code 2001.

Predictions from statistics

On a modern train, more than 1,000 different types of mechanical and electrical events are recorded, including operational events such as “opening door” or “train is braking,” warning events such as “line voltage frequency out of range” or “compression is low in compressor X,” and failure events such as “pantograph is out of order” or “inverter lockout.” More than 2,000 events are collected on each train every day.

IBM proposed to use sequential analysis to determine whether any of these events or event sequences indicate that a more severe problem is about to occur. Sequential analysis uses statistical modeling to detect patterns in data. For example, it can show that if particular grocery store customers purchase baby formula and diapers one day, they are highly likely to purchase stewed fruit (baby food) at some point in the future.

Sequential analysis of events

The IBM team started with the raw event data for 2010: a 20-million-row data set containing the event time, event code, and the GPS location of the event. The team used the sequence node algorithm in IBM SPSS Modeler to analyze the train data. The sequence analysis is a two-stage process for sequential pattern mining.¹ First, the algorithm mines the data for frequent sequences and computes an adjacency lattice. The user can then explore the lattice by using specified criteria, such as support and confidence bounds, or by placing restrictions on the antecedent sequence.

Using the IBM SPSS Modeler workbench (see Figure 1), the team defined the input table in a DB2 data source node, added a Select node to remove

¹ Rakesh Agrawal and Ramakrishnan Srikant, *Mining Sequential Patterns* (San Jose, California: IBM, 1995), rsrikant.com/papers/icde95.pdf.

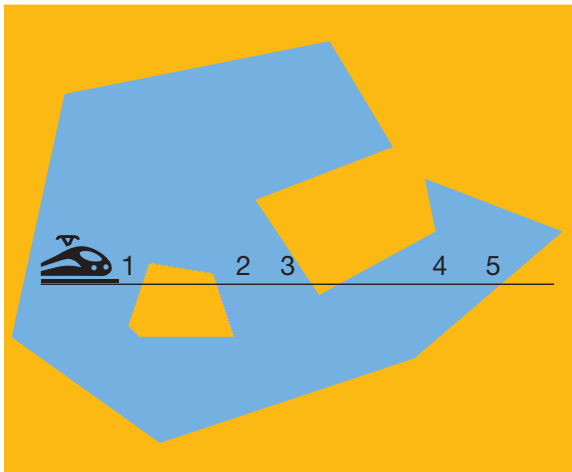


Figure 4: The solution to the point-in-polygon problem is to draw a horizontal line starting at the point in question. If the point is within the polygon, the line will intersect the edges of the polygon an odd number of times.

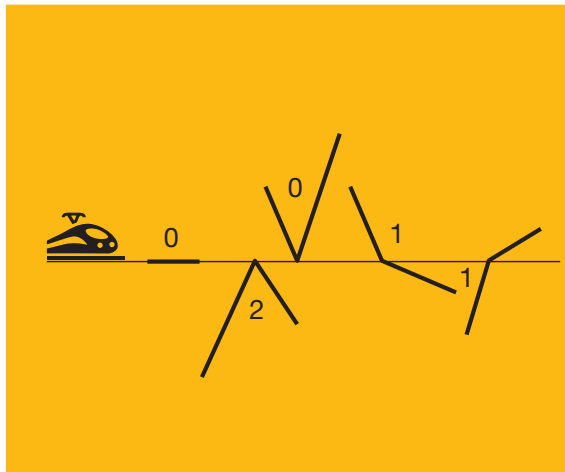


Figure 5: If the horizontal line drawn is aligned with a polygon vertex, then only the segments that are below the line are counted.

data that occurred while the trains were in maintenance depots, and defined the input columns in the sequence node algorithm. The team also set the time period for which the sequences were to be studied, specifying that only events occurring in a two-day timeframe were to be considered.

The SPSS Modeler rule sets interface shows the rules found by SPSS after the analysis has been completed (see Figure 2). The highlighted rule displays an *antecedent sequence* (21382, 29203) followed by a *consequent sequence* (in this case a single event: 18232), with a *support* rating of 55.7 percent and a *confidence* rating of 93.1 percent. This means that during 2010, this antecedent sequence appeared in 55.7 percent of trains and was followed by the consequent sequence 93.1 percent of the time.

Once the rules have been calculated, the data can be explored using the SPSS Modeler filtering tools (see Figure 3). According to the data, event code 2201 followed event code 2003 in 76.9 percent of the cases and event code 2001 in 84.6 percent of cases. Because event code 2201 indicates a failure that makes the train stop and leads to delays, this information can be used to optimize maintenance planning: if event code 2001 appears, the device related to this code should be inspected as soon as possible to avoid a failure.

Is the train at the station?

As mentioned earlier, the team needed to eliminate data that was recorded while the trains were in a depot for maintenance; the service process triggered sensors in ways that did not reflect the actual behavior of the train systems. The best way to detect events that occurred during maintenance was to use the well-known point-in-polygon algorithm

(see Figures 4 and 5) to calculate whether the train's position at the time an event was recorded corresponded with the location of the maintenance depots. The manufacturer had an existing table with the depot polygons already defined.

Initially, the team thought that the calculation would be performed in a Visual Basic or Java application. But faced with processing 20 million events, the team quickly determined that these solutions would not be fast enough. Instead, they implemented the algorithm in SQL directly in IBM DB2. (Note: IBM DB2 Spatial Extender provides a native function to find whether a given geometry falls within the boundaries of another geometry.)

Implementing point-in-polygon in SQL

To express the point-in-polygon algorithm in SQL, the team first built a pair of functions to transform longitude and latitude coordinates into mathematical X and Y planar coordinates:

► **X (longitude) implemented as GetX SQL**

function: `CASE WHEN LonDeg <= 180 THEN -1 * (LonDeg + (DOUBLE(LonMin) + DOUBLE(LonSec)/100)/60) ELSE (360 - LonDeg - (DOUBLE(LonMin) + DOUBLE(LonSec)/100)/60) END`

► **Y (latitude) implemented as GetY SQL function:**

`CASE WHEN LatDeg <= 90 THEN -1 * (LatDeg + (DOUBLE(LatMin) + DOUBLE(LatSec)/100)/60) ELSE ((LatDeg + (DOUBLE(LatMin) + DOUBLE(LatSec)/100)/60) - 90) END`

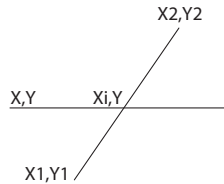
Next, the team built a table of the segments of the polygon. They created a POLYSEG table with the polygon segment coordinates X1,Y1 and X2,Y2, keeping track of the segment

order. They used a `CASE WHEN ... THEN ELSE END` structure to detect whether a given point was actually the first point in the table. If it was, then it became the endpoint for the final segment. Otherwise, the code moved on to the next point:

```
SELECT p1.DepotID,p1.DepotName,p1.CoordinateRank, p2.GPSOrder,
GetX(p1.LonDeg, p1.LonMin,p1.LonSec), GetY(p1.LatDeg, p1.LatMin,p1.LatSec),
FROM POLYGONS p1,POLYGONS p2
WHERE p1. DepotID =p2. DepotID
AND p2.GPSOrder=CASE WHEN p1.CoordinateRank =(SELECT MAX(p3.
CoordinateRank) FROM POLYGONS p3 WHERE p3. DepotID =p1. DepotID)
THEN 1 ELSE p1.DepotID +1 END;
```

To detect the number of intersections to the right from a point that is not aligned with a vertex, start with the formula of the coordinate where the horizontal line intersects a segment (Xi):

$$X_i = X_1 + (Y - Y_1) * (X_2 - X_1) / (Y_2 - Y_1)$$



For a given horizontal line that intersects the segment, use $X \leq X_i$, or:

$$X \leq X_1 + (Y - Y_1) * (X_2 - X_1) / (Y_2 - Y_1)$$

To avoid any division by zero or overflow, transform all division into multiplication and check for the sign (horizontal lines with $Y_2 = Y_1$ do not count in the algorithm):

```
If (Y2-Y1)>0 then X*(Y2-Y1)<=X1*(Y2-Y1)+ (Y-Y1)*(X2-X1)
If (Y2-Y1)<0 then X*(Y2-Y1)>=X1*(Y2-Y1)+ (Y-Y1)*(X2-X1)
```

The resulting `WHERE` clause to count a segment that is intersected is the following:

```
WHERE Y2<>Y1 AND (((Y2-Y1)*(TEMP.Y-Y1)>0 AND TEMP.X*(Y2-
Y1)*(TEMP.Y-Y1)<=X1*(Y2-Y1)+(TEMP.Y-Y1)+X2-X1)
OR ((Y2-Y1)*(TEMP.Y-Y1)<0 AND TEMP.X*(Y2-Y1)*(TEMP.Y-Y1)>=X1*(Y2-
Y1)*(TEMP.Y-Y1)+X2-X1)) AND (TEMP.Y>Y1 OR TEMP.Y>Y2) AND
(TEMP.Y<Y1 OR TEMP.Y<Y2)
```

To deal with the points on the line aligned with a segment vertex, clauses that count only the segments that are lower than the line are necessary:

```
OR (TEMP.X<TEMP.X1 AND TEMP.Y=Y1 AND Y1>Y2)
OR (TEMP.X<TEMP.X2 AND TEMP.Y=Y2 AND Y2>Y1)
```

Removing events from maintenance depots

To actually count the intersections, identify the points that have an odd number of intersections, and then filter out those points using `GROUP BY` and `HAVING` clauses. Notice that `COUNT` is an integer. In an odd number division, it does not retain the decimal part; this is why it can be used to test for an odd number.

```
GROUP BY EVENT.ID, P.DepotName HAVING COUNT(*)-2*(COUNT(*)/2)=1)
```

Finally, the team needed to assemble everything into a table and update the events table with the depot location name or null. The full SQL can be seen online at ibm.com/developerworks/data/library/dmmag/DMMag_2011_Issue3/Train/index.html. The final code updates the 20-million-row table in less than one hour, identifying the events that happen in a maintenance depot. Because this update is performed on a monthly basis, this performance is acceptable. Additional optimization with appropriate indexes and `EXPLAIN` command usage could be used to increase performance.

Keep the trains running

The team demonstrated that IBM SPSS Modeler provides the capability to automate failure predictions in real time by calculating confidences on the fly and generating alerts when needed. In addition, DB2 provides advanced SQL features to help handle complex queries, including geospatial data and complex computations. To see the real-world benefits, consider the money saved by reducing the need for emergency repairs. Or think of the lessons learned that can be used to make the next generation of trains more reliable. Or perhaps, simply contemplate the railway traveler: watching the landscape smoothly unfurl while the train rushes quietly to the next exciting destination. *

Marc Fiammante is an IBM distinguished engineer (directeur technique) and worldwide chief architect of SOA engagements delivered by the IBM SOA Advanced Technology team. He has filed several software domain patents and co-authored the book Service-Oriented Architecture Compass (ibmpressbooks.com/title/0131870025).

Grégory Neuvéglise, predictive analytics solution architect at IBM, has 10 years of experience in implementing business analytics. He specializes in customer relationship optimization and predictive maintenance.



Maximize Data Analytics Performance

Rethink the data warehousing equation and empower data-driven business decisions

Skechers USA Inc. is a global leader in lifestyle footwear with more than US\$2 billion in annual revenue. Retail daily sales and inventory data is loaded each night into the company's data warehouse system. About 15 business analysts use this data to understand retail sales and market trends, and to decide on product marketing and promotions.

With more than a billion rows in fact tables, analytical queries could take anywhere from a few minutes to 45 minutes to complete. Time lost waiting for answers is a productivity loss because it limits the number of business analytical queries being executed. This puts more pressure on DBAs to tune each query.

Skechers isn't the only company facing these hurdles. We live in a data-rich world, and the opportunities that come with unlimited data are epic. Retailers have access to fine-grain demographics, buying history, social media identities, online community involvement, information from public records, and more. Manufacturers record and track supplier metrics in exquisite detail, from performance against delivery deadlines to breakage and discount levels. Law enforcement agencies rely on enormous amounts of data and information, locally, nationally, and internationally. Criminal and scientific data that is fast and easy to access allows law enforcement to nail down airtight cases.

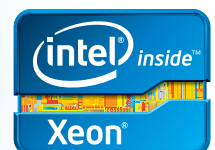
When it's used in a timely manner to drive critical business decisions, data analysis equals competitive advantage. The key phrase is "when it's used in a timely manner."

Companies are finding that as data volumes grow, they overwhelm storage systems and the analytics engines that turn data into game-changing insight. In this article, we'll look at how and why the tremendous potential of unlimited data can become a huge challenge, and how Intel and IBM are partnering to help organizations structure data delivery and data management processes that turn unlimited data into unlimited business value opportunity.

Companies Strive for Rapid Analysis

Trying to keep up with unlimited data puts a huge strain on an entire organization, starting with slow analytics response times. As organizations amass more and more data, even basic queries can take a substantial amount of time. More sophisticated queries—particularly those generated by business analytics applications—take even more time. In extreme cases, the query may return results already out of date.

Businesses and public agencies are devoting growing amounts of their IT budgets to handle these vast quantities of data. Data tools and infrastructures have become quite efficient, but even the most advanced technologies cannot keep pace with unlimited data. Creating schemas, indexing, building cubes and summary tables, and creating partitioning strategies all require extensive planning and human effort. Where a single database administrator (DBA) once managed a centralized data center for the entire company, there now may be many DBAs



managing and tuning data warehouses for just one business unit of a large organization.

And then there is the pure capital cost of the hardware needed to manage growing data. Data warehouses are often large, centralized systems—replacements and upgrades can be expensive, risky, and labor intensive. But with critical decisions dependent on the collation and analysis of diverse data sources, no company can afford to fall behind the warehousing and analytics performance curve.

IBM and Intel In-memory Optimizations

As organizations struggle to maintain query performance in the face of unlimited data, a common bottleneck is the hard-disk I/O throughput. The tipping point has been reached: hard-disk I/O speeds cannot increase quickly enough to keep up with unlimited data, especially as the total amount of storage needed increases. In fact, most strategies intended to accelerate data warehouse queries—including compression, optimization, buffering, and index tuning—are designed to minimize disk I/O.

However, other areas of information technology are advancing rapidly. RAM and solid-state storage become less expensive almost by the day; processor technology continues to make great strides. Processor-to-RAM access speed is increasing at a much higher rate than processor-to-disk access speed. Modern processors are not only faster and less costly than their predecessors, but have been engineered to fully utilize large quantities of RAM and take efficient advantage of larger on-die caches.

That's why Intel and IBM have worked together to develop a cost-effective way for organizations to apply the advanced technologies of the Intel® Xeon® processor E7 family to meet the challenges around data delivery and data management for solution optimization.

Answers at the Pace of Your Digital Investments

IBM® Informix® Warehouse Accelerator helps increase data warehouse performance by exploiting features such as single instruction multiple data (SIMD) and a larger on-chip cache per Intel processor; implementing novel compression and query processing algorithms; and using larger low-cost memory and affordable systems with terabyte memory. The solution is tightly integrated with IBM Informix to accelerate queries transparently to the analytics applications. Data is loaded completely into system memory in a compressed form, using a special columnar scheme developed by IBM. As queries come in, the Informix database passes them to the Informix Warehouse Accelerator, which quickly performs the queries and returns the results.

By compressing and caching the data in-memory, and processing queries on compressed data, Informix Warehouse Accelerator can dramatically increase performance. As processors add more cores and bigger on-chip caches, Informix Warehouse Accelerator is able to capitalize on this trend by running parallel instructions, optimizing algorithms to exploit each on-chip cache. This enables massive scaling and significant performance gains.

With the Informix Warehouse Accelerator query processing engine, you can scan billions of rows of data in seconds. The deep columnar data representation, query processing on compressed data, and innovative algorithms eliminate the need for indexes. Once the basic memory and storage configuration is specified during installation, consistent plans, elimination of disk I/O, fast scans, and joins help eliminate run-time tuning. All data is stored in-memory with just a copy of the in-memory image on a disk. There is no need to plan and create storage spaces for tables and indexes.

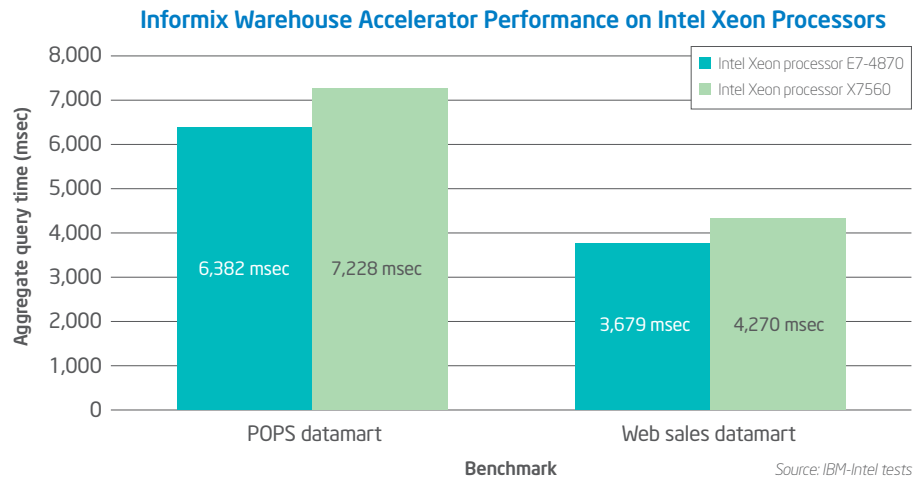
Finally, the accelerator is designed to run on commodity hardware and to easily scale with a growing infrastructure. The accelerator runs on all hardware platforms supported by Informix, and significant data warehouse acceleration can be achieved while minimizing hardware costs.

Informix Warehouse Accelerator is already delivering real-world results. “Complex inventory and sales analysis queries on the enterprise warehouse took anywhere from a few minutes to 45 minutes to run,” says Ashutosh Khunte, manager of data management services for Skechers USA. “When we ran those same queries using Informix Warehouse Accelerator, they finished in 2 to 4 seconds. That means they ran from 60 to 1,400 times as quickly, with an average acceleration factor of more than 450—all without any index or cube building, query tuning, or application changes!”

Informix Warehouse Accelerator Runs Best on the Intel Xeon Processor E7 Family

Transforming traditional disk I/O-based query execution to in-memory execution requires a new level of processor support. The Intel Xeon processor E7 family delivers the high processor performance needed to accelerate even the most demanding warehouse queries, and it supports large quantities of system RAM, enabling organizations to maximize the benefits of the Informix Warehouse Accelerator in-memory technology.

The Intel Xeon processor E7 family is engineered to meet demands for rapidly increasing compute capacity, faster performance, and high availability. It provides up to 10 cores, up to 20 hyper-threads, and 30 MB of last-level cache per processor, all in a four-socket server. The Intel Xeon processor E7 family supports 32 GB dual in-line memory modules (DIMMs), enabling application workloads that require larger memory capacities. Together with Informix Warehouse Accelerator, the Intel Xeon processor E7 family eliminates



the need for disk I/O and dramatically improves the speed of access to complex data sets.

Also, with the Intel Xeon processor E7 family, organizations do not need to introduce additional infrastructure security capabilities. Protection against hardware errors and malicious software attacks is automatically managed by advanced, built-in features. For example, Machine Check Architecture (MCA) enables the CPU to coordinate with the operating system to gracefully recover from memory errors.

Finally, advanced compression capabilities in Informix Warehouse Accelerator enable the Intel Xeon processor E7 family to handle more data in the same memory configurations without sacrificing performance. With Intel® Turbo Boost Technology, processor performance can be enhanced beyond normal operating constraints when thermal, power, and current limits have not been exceeded.

The Intel Xeon processor E7 family—together with in-memory software solutions—demonstrates innovation that can lead to a more efficient performance experience for your data warehousing implementations. Extensive testing by Intel and IBM shows that the Intel Xeon processor E7 family provides a significant performance boost (see Figure 1).

Figure 1: The Intel Xeon processor E7-4870 works with IBM Informix Warehouse Accelerator to deliver increased performance over previous generations of processors for datamarts of widely varying sizes. The proof of performance and scalability (POPS) datamart consists of two fact tables with five dimensions, one with 350 million rows and one with 1 billion rows. The web sales datamart is relatively small, consisting of a single fact table with 7 million rows and 12 dimensions.

Benefit from Better Insight, Exactly When It's Needed

Together, Informix Warehouse Accelerator and the Intel Xeon processor E7 family combine to give organizations the tools they need to analyze unlimited data. Reducing elapsed query times by up to 1,400 times—as Skechers did—means that answers to critical questions are available precisely when they're needed, making knowledge workers more effective and businesses more agile.

For example, marketing teams constantly look for ways to effectively deliver targeted messages to specific audiences. Internet-based marketing has created the expectation of an immediate response to inbound inquiries. As a result, experienced marketing personnel end up devoting excessive time to developing response programs that meet the needs of diverse demographic profiles—yet the resulting programs are often ineffective. System access to large information stores, including huge Internet log files, is crucial for automated, immediate, and personalized marketing communications. Organizations can more fully leverage their investment in analytics applications when the data queries they generate are run and completed at a faster rate.

Making best use of sophisticated analysis and decision-support tools gives organizations a competitive edge. For example, traffic records, criminal records, and other data collected by federal, state, and local governments can help a law enforcement official identify a wanted

criminal during a routine traffic stop. But to do so, the officer must be able to access and query vast stores of complex data in many different repositories, and get accurate, up-to-date results in seconds. Informix Warehouse Accelerator running on Intel Xeon processors can help deliver the appropriate answers in time and in context.

High Performance without the High Price Tag

Despite the need for fast, immediate data access, budget-conscious organizations must balance the benefits of high performance against the resources required to support it. Informix Warehouse Accelerator tips the scale in the right direction. It requires no workload tuning, determines join orders automatically, and requires no application changes—removing rote administration tasks and freeing DBAs to work on higher-value, strategic projects. Also, the in-memory data-handling capabilities of the Intel Xeon processor E7 family help data professionals save time when managing physical schema design.

With Informix Warehouse Accelerator running on the Intel Xeon processor E7 family, organizations can turbocharge their data warehouse performance with scalable, readily available, off-the-shelf hardware. Instead of constantly adding more processing power and more storage, organizations can acquire commodity components designed to work together to overcome performance bottlenecks and make better use of existing infrastructures.

The Intel Xeon processor E7 family is engineered for reliability and security as well as power efficiency. Informix Warehouse Accelerator delivers significant data warehouse performance improvements by fully exploiting the features of these multicore, large-cache processors to eliminate disk I/O.

Organizations need better options to handle unlimited data. Informix Warehouse Accelerator optimized for the Intel Xeon processor E7 family enables knowledge workers to quickly extract business insight from more data using fewer resources. In a time of unlimited data, IBM and Intel created the innovative Informix Warehouse Accelerator to deliver a rich, competitive advantage in today's data-loaded, computing-intensive environment.

Learn More

More about IBM Informix Warehouse Accelerator: www.ibm.com/software/data/informix/warehouse

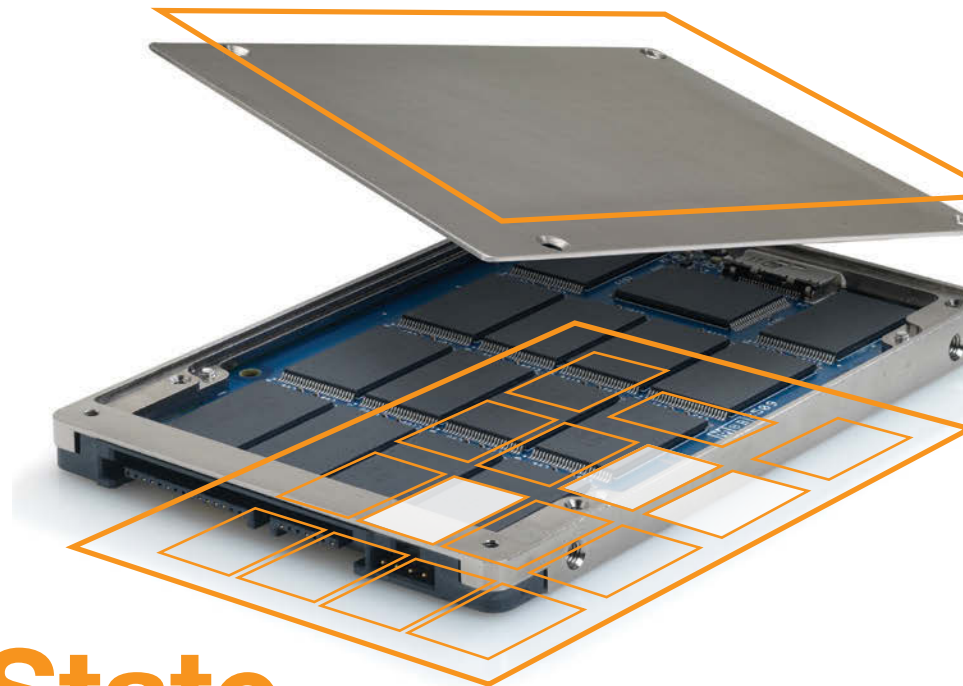
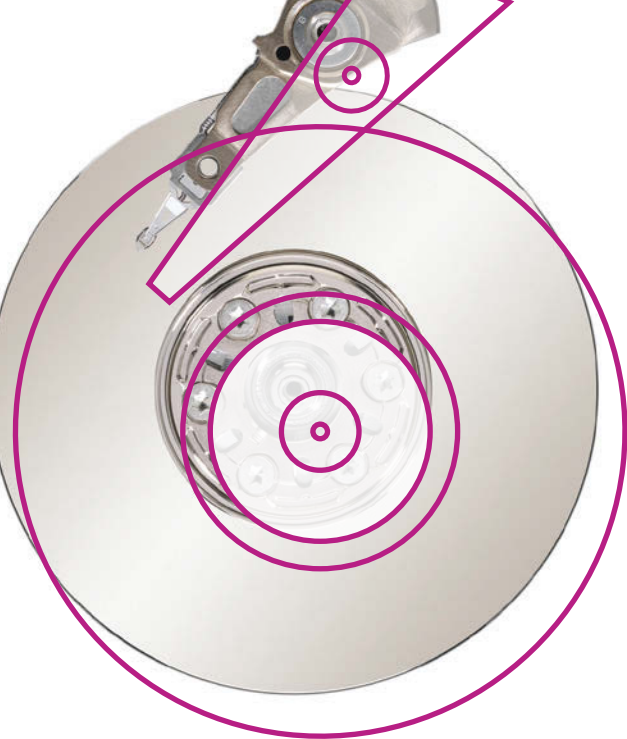
More about the Intel Xeon processor E7 platform: www.intel.com/itcenter/products/xeon

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments might vary significantly. Users of this document should verify the applicable data for their specific environment.

Intel, the Intel logo, and Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. IBM, the IBM logo, ibm.com, and Informix are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

* Other product, company, or service names may be trademarks or service marks of others.

Copyright © 2011 Intel Corporation. All rights reserved.



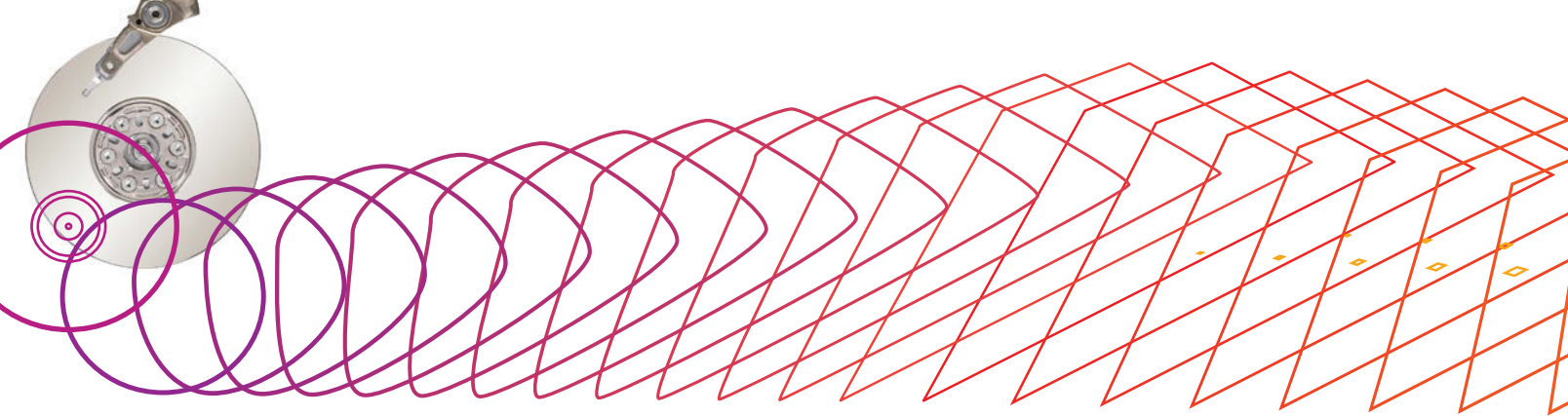
Solid-State Drives: Changing the Data World

By Paul Pendle

Say hello to your new friend

It cannot have escaped your notice, if you have been reading industry articles and press releases, that solid-state drives (SSDs) are now available for the leading enterprise storage arrays. Disk storage companies are pitching these drives as a quantum leap in enterprise storage performance, and they are. The sales literature primarily demonstrates SSD performance advantages over hard disk drives (HDDs) but also includes information about additional benefits, such as power and cooling savings, reliability, cost per I/O, and so on.

Even if you strip away the marketing hype, the numerous performance advantages make a strong argument that SSDs will completely replace Fibre Channel (FC) drives as the primary storage technology in high-end storage systems. Price is likely to be an issue for a while, but ultimately, the end of FC drives is in sight. (Who remembers the arrival of the CD? Although it was the first nail in the coffin of vinyl, vinyl indeed took many years to leave the mainstream.) The end of FC drives might seem like a very ordinary technology transition, but something very out of the ordinary is occurring. It's possible that SSDs will cause you to reevaluate some of the ways that you deal with databases, as the move from spinning disks to solid-state technology will change the rules.



I/O cost and the slowly spinning disk

Clustering. Defragmenting. Reorganizing. Buffering. A vast number of common database tasks and strategies are designed to do one thing only: minimize disk I/O. That's because HDD I/O is usually the most time-consuming part of any database transaction. If we put it in terms of total time costs, HDD I/O is fantastically expensive.

When you move to SSDs, I/O operations become much faster, but most people don't realize how large the difference is. For example, take a single, random 4K-page read. From a 15,000 rpm FC HDD, the average response time is approximately 6.5 ms. When using an SSD in an enterprise-class storage array, you should estimate the same read would take 1 ms. In other words, you might expect an SSD to complete six I/Os in the time it takes an FC HDD to finish one.

But there's more to the story. SSDs can perform operations concurrently; HDDs cannot. The best you can reasonably expect from an FC 15,000 rpm HDD is about 200 random 4K-page reads per second. Using SSDs, read requests can overlap, which gets you more like 5,000 random 4K-page reads per second. A single I/O costs *25 times less* on an SSD than on an HDD.

With SSDs, disk I/O doesn't cost a little less—it costs a lot less. But is the reduction enough to change how you build and manage your databases? In some cases, yes. (Note: This article chiefly addresses online transaction processing [OLTP] access. Sequential access, as exhibited by data warehouse activities, does not produce as great a performance improvement with SSDs as does random access.)

Cluster's last stand?

Database clustering places frequently accessed DB2 rows in the same page and frequently accessed pages close together on disk. A successful clustering strategy can reduce the number of DB2 `GETPAGES` (and also I/Os), especially if many sequential scans use the clustering index. Because the DB2 subsystem reads data in the order in which it needs to process it, clustering helps reduce expensive `SORTs`. Sequential scans are typical in decision support system (DSS) applications but rare with OLTP access patterns.

The notion of adjacency of pages to reduce disk head or arm movement and thus latency during I/O does not apply to SSDs, of course. When data appears to be physically adjacent in DB2 (that is, consecutive pages in the table space), it is very unlikely to be in consecutive cells on the SSD. The data is distributed evenly across the SSD capacity using wear-leveling algorithms. In fact, the exact location on the SSD is somewhat irrelevant, because the latency to retrieve the data is an order of magnitude (and in some cases, two orders of magnitude) less than the latency incurred during a random read function on a spinning disk.

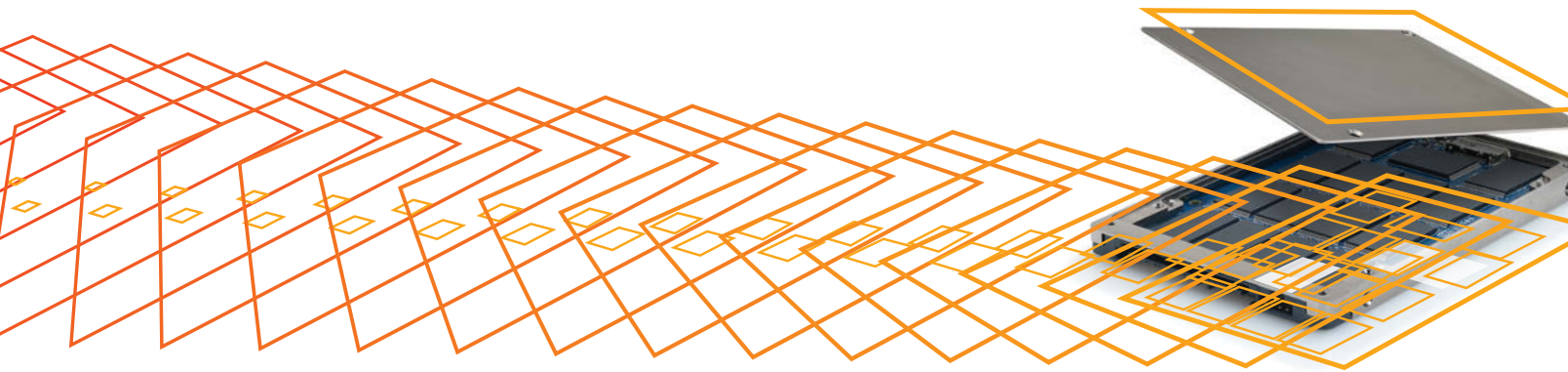
Does clustering (or for that matter, poor clustering) really matter when using SSDs? Consecutive data pages are unlikely to be in adjacent cells due to the wear-leveling algorithms, so do you need to group data on the media? It's an interesting question. If DB2 understands that two rows clustered together are on the same page, requests for both of them might result in a single I/O if the requests are close together in time. This single I/O might not occur if the data is unclustered. That is to say, both of those rows may be on separate pages, resulting in two separate `GETPAGES` and two separate sync I/Os. That raises the question, is the cost of the extra `GETPAGE` (or `GETPAGES`) and related sync I/Os punitive when using SSDs?

Purists might argue that it costs CPU and channel resources to perform the extra `GETPAGES`, and they would be right. However, in the context of reducing the number of `REORGs` that need to be performed and the speed of the SSDs, this extra cost is easily justified.

Embedding free space

When you create a table space, you frequently embed free space so that an application can insert rows in a clustering sequence, and rows can increase in size as they are updated. The extra space can reduce overflows and index page splits. Most folks typically reserve 10 percent of the total table space as free space, but you will often see more with highly volatile applications.

However, reserving free space only trades disk space for time, potentially letting you go longer between `REORGs`. So consider embedding less free space when using SSDs, since the clustering sequence may not be so important.



If you decide to forgo free space in the table space, you might want to consider using `APPEND YES` for the tables in the table space. This option reduces the code path that DB2 must traverse to find a location for the inserted row and also avoids page overflows on `INSERTs`. On the downside, you need to consider concurrency. Multiple threads executing `INSERTs` and competing for locks and latches on the same page can be costly, especially in a data sharing environment (although `MC00` may solve this problem).

Buffer pools

DBAs use buffer pools to keep the most recently used DB2 pages in memory, hoping that the pages will be reused and thus avoiding I/O. And as they say, the best I/O is the

one that does not happen. With SSDs bringing down the performance cost of an I/O, the use of buffer pools is not so critical.

A potential course of action here is to reduce the size of the buffer pools supporting the table spaces resident on SSDs. It is likely that you will have a mixture of HDDs and SSDs, so you can allocate the buffer pool space thus saved to the table spaces on HDDs, which need it more.

REORGs

What does a DBA do today to keep data and indexes organized? Several things, actually, but `REORGs` may be the most significant activity. `REORGs` accomplish a number of goals, many of which are related to disk I/O:

Data Quality Tools for IBM



IP Location



Dedupe



Property



International



Address Verification



Name Parse



Email Validation



Phone Verification



SmartMover



Free Form Parse

Clean your database with tools that make it easy.

Request a free trial at
MelissaData.com/myibm or
 call 1-800-MELISSA (635-4772)

MELISSA DATA[®]
 Your Partner in Data Quality

With SSDs, disk I/O doesn't cost a little less—it costs a lot less. The reduction may be enough to change how you build and manage databases.

- ▶ Place the data in a clustering sequence
- ▶ Recapture lost space due to row deletions
- ▶ Reinsert free space into the table space
- ▶ Reduce leaf levels or pages in indexes
- ▶ Reduce fragmentation in indexes
- ▶ Reduce the number of extents of the table or index space

DB2 does its level best to keep rows close to their optimal location in the clustering sequence. But this is not always possible. The page might lack space for the optimal placement of the row on an `INSERT`, or a row might grow beyond the space allotted for it on an `UPDATE` and so must be moved. And there are many other reasons. Ultimately, over time, the `CLUSTERRATIO` of the data decreases from 100 percent to a lower value. How quickly it decreases depends on the volatility of the data.

You need to monitor the system (ideally in an automated fashion) to determine when a `REORG` needs to be run. All kinds of DB2 catalog statistics describe the table space condition, and IBM suggests thresholds as to when the `REORG` should occur. There are many downsides to `REORG`s:

- ▶ They must be scheduled and monitored.
- ▶ They consume a large amount of I/O and CPU resources.
- ▶ They can reduce concurrency and availability when executed against a live table or index space.
- ▶ They can increase logging during the `REORG` process.
- ▶ They flood wide area networks (WANs) with changed data traffic when disaster recovery replication is used.

The last point may be the most significant if you are replicating your database over a long distance for disaster recovery. The writes generated by a `REORG`, while not transactional writes, are still a necessary evil that must be added to the “real” write workload being transmitted across the link. Because the cost of telecommunications lines is so high, you would hate to fill the pipe with such busywork that could be avoided or at least reduced.

Deploying table spaces on SSDs can reduce the need to `REORG`. This approach can generate a huge savings in management, CPU, I/O, and link bandwidth for remote replication. You just need to understand that the extra work the DB2 subsystem must do to retrieve index or data pages on SSDs is less than the cost of performing I/O on a table space that has been `REORG`ed on HDDs.

IBM trends

Conversations with folks at IBM reveal a viewpoint within the company that DBAs may reduce `REORG` frequency. This notion is independent of SSD implementations and is more a reexamination of why `REORG`s are performed in the first place. For instance: does a large number of extents for a table space really have a measurable negative performance impact? Additionally, `DSNACCOX` with DB2 10 has specific code to reduce the `REORG` requirements for table spaces residing on SSDs.

Think differently about SSDs and DB2

Using SSDs to support a DB2 for z/OS OLTP subsystem can help you achieve the following:

- ▶ Reduce the number of `REORG`s
 - Save expensive MIPS, disk, and channel resources
 - Reduce the costs of remote replication
 - Save personnel time managing the `REORG`s
- ▶ Increase space utilization by embedding less free space
- ▶ Improve buffer pool efficiency by dedicating minimal space to table spaces on SSDs and reallocating the space thus saved to HDD buffer pools

When you consider the purchase of SSDs for your enterprise-class storage array, think beyond speeds and feeds. Think about which tasks SSDs enable you to accomplish differently. ✱

Paul Pendle is a consulting systems engineer for EMC Corporation. He has more than 30 years of experience in databases, hardware, software, and operating systems both from a database administrator's perspective and from a systems programming perspective. Pendle is an IBM Information Champion and has worked with DB2 for z/OS since version 1.2. He can be contacted at paul.pendle@emc.com.

RESOURCES

Ready to Access DB2 for z/OS Data on Solid-State Drives:

www.redbooks.ibm.com/redpapers/pdfs/redp4537.pdf

DB2 10 for z/OS Technical Overview:

www.redbooks.ibm.com/redbooks/pdfs/sg247892.pdf

IBM Easy Tier Enables DS8700 and Storwize V7000 Users to Profit from Solid State Drives (David Hill, Mesabi Group, October 1, 2010):

ibm.com/systems/hk/resources/systems_ap_TSW03065USEN.pdf



By Sasirekha Rameshkumar

GO BACK IN TIME

Get started with
temporal data, a timely
new DB2 feature

For most people, the words “time travel” bring to mind science fiction: H. G. Wells, *Star Trek*, and a certain famous DeLorean automobile. But when it comes to business data, the need to travel back and forth in time is absolutely essential. How else are you going to calculate how much your parts costs have gone up, or automatically give a limited-time discount to a valued customer, or calculate the effect of a retroactive rate change?

Previously, tracking data over time in DB2 for z/OS required lots of manual intervention, but version 10 gives you new abilities to manage multiple versions of data and track changes over time. With simple declarative SQL statements, you can maintain a history of database changes, track effective business dates, and analyze the history of changes.

System time and business time

The basic structure of the temporal features is simple but powerful. DB2 tracks two different types of temporal data: *system time* and *business time*. System time tracks when changes are made to the state of a table, while business time tracks the effective dates of business conditions, such as the terms of an insurance policy or the interest rate of a loan. A table that includes temporal data has a pair of *date-time* columns that indicate the period (the time interval) when the row is valid. When a new row is inserted, DB2 records the system time as the *begin* timestamp. When a row is deleted, the delete time is recorded in the *end* timestamp.

For system time tables, DB2 also helps store historical data in a *history table* associated with the temporal table. When a row in a temporal table is updated, DB2 updates the row and DB2 will automatically move the previous version of the row to a history table.

Creating a temporal table with system time

To create a temporal table with system time, include the `PERIOD SYSTEM_TIME` clause (`sys_start,sys_end`) in the table definition. Also include three additional columns—for example, `sys_start`, `sys_end`, and `trans_startid`—as `TIMESTAMP(12)` in the table definition (see Figure 1).

```
CREATE TABLE policy_info
(policy_id CHAR(10) NOT NULL,
coverage INT NOT NULL,
sys_start TIMESTAMP(12) NOT NULL GENERATED
    ALWAYS AS ROW BEGIN,
sys_end TIMESTAMP(12) NOT NULL GENERATED
    ALWAYS AS ROW END,
trans_startid TIMESTAMP(12) GENERATED ALWAYS AS
    TRANSACTION START ID,
PERIOD SYSTEM_TIME(sys_start,sys_end));
```

Figure 1: Creating a temporal table with system time

Specifying `GENERATED ALWAYS` tells DB2 to automatically generate the temporal values during `INSERT`, `UPDATE`, and `DELETE` operations. Specify the type of each of the three columns by using the following options:

- ▶ **AS ROW BEGIN**—DB2 generates a timestamp value that corresponds to the start time that is associated with the most recent transaction.
- ▶ **AS ROW END**—DB2 assigns the maximum value of `TIMESTAMP` whenever a row is inserted or whenever any column in the row is updated (that is, a new version of the row is created). This column captures the timestamp when the row is deleted from the system-period temporal table.
- ▶ **AS TRANSACTION START ID**—DB2 assigns a unique timestamp value on a per-transaction basis or the null value. DB2 generates the start time column when it executes the first statement that changes the table's data. A transaction-start-ID column is required for a system-period temporal table. If defined as `NOT NULL`, the value corresponds to the start time associated with the most recent transaction (the value of the `AS ROW BEGIN` column). If nullable, it will contain the `NULL` value.

Associating the history table

Although DB2 will automatically store historical versions of rows, you must create a history table to contain the rows and associate it with the system-period temporal table. The history table must have the same number and order of columns as the system-maintained temporal table. The history table must not include a period and must have the same encoding scheme and Coded Character Set Identifier (CCSID) as the temporal table (see Figure 2).

```
CREATE TABLE hist_policy_info
(policy_id CHAR(10) NOT NULL,
coverage INT NOT NULL,
sys_start TIMESTAMP(12) NOT NULL,
sys_end TIMESTAMP(12) NOT NULL,
trans_startid TIMESTAMP(12));
```

Figure 2: Creating a history table

With simple declarative SQL statements, you can maintain a history of database changes, track effective business dates, and analyze the history of changes.

You can enable versioning of the temporal table by using the `ALTER TABLE` command with the `ADD VERSIONING` clause and specifying the history table name in the `USE HISTORY TABLE` clause (see Figure 3).

```
ALTER TABLE policy_info
  ADD VERSIONING USE HISTORY TABLE
    hist_policy_info;
```

Figure 3: Enabling versioning of the temporal table

The `ADD VERSIONING` clause cannot be specified with other clauses on the `ALTER TABLE` statement.

To remove the relationship between the system-period temporal table and history table and archive a disassociated history table, someone with appropriate authority must copy rows to the archiving table and then delete the old rows from the history table. You can use `ALTER TABLE` with the `DROP VERSIONING` clause for the same purpose, but IBM suggests sparse use of `DROP VERSIONING` because it can cause invalidation, which should be avoided if possible.

Data versioning of system-period temporal tables

When inserting a new row to the system-period temporal table, DB2 automatically generates the appropriate `TIMESTAMP` values for its `SYSTEM_TIME` columns and the transaction start time. In the example, the `sys_start` and `trans_startid` have the current system time and the `sys_end` value is set to 9999-12-30-00.00.00.000000000000 (see Figure 4).

When the row is updated, DB2 automatically maintains the old version of the row in the history table and updates the system time values in both tables. DB2 sets the `sys_end` value of the row in the history table and the `sys_start` value in the temporal table to the start time that is associated with the most recent transaction. If the same row is updated again within the same commit scope, then no record is created in the history table. Therefore, DB2 keeps only the original row. If the row is deleted, then DB2 automatically removes the data from the current table and maintains the old version(s) in the history table. DB2 sets the end time of the deleted row in the history table to the start time that is associated with the most recent transaction of the `DELETE` statement.

Querying system-managed temporal tables

`SELECT` statements without any period specifications apply only to data in the system-period temporal table, providing a performance benefit for applications requiring only current data. DB2 accesses the historical data automatically and transparently only if the `FROM` clause of the query includes the period specifications. The period specification mentions `FOR SYSTEM_TIME` to indicate that the system time is to be used. Functionally, the three supported periodic specifications are the following (also see Figure 4):

- ▶ **AS OF *value***—Query the data as of a certain point in time. Returns the row in which the begin value \leq *value* and the end value $>$ *value*.
- ▶ **FROM *value1* TO *value2***—Query the data from *value1* to *value2*. Returns the rows with the begin value $<$ *value2* and the end value $>$ *value1*. Returns zero rows if *value1* is greater than or equal to *value2*.
- ▶ **BETWEEN *value1* AND *value2***—Query the data between a range of start and end times. Returns the rows with the begin value \leq *value2* and the end value $>$ *value1*. Returns zero rows if *value1* is greater than *value2*.

```
SELECT coverage FROM policy
  FOR SYSTEM_TIME AS OF TIMESTAMP('2010-12-01')
  WHERE id = 1111;

SELECT count(*) FROM policy
  FOR SYSTEM_TIME FROM TIMESTAMP('2011-11-30')
    TO TIMESTAMP('9999-12-31')
  WHERE id = 1111;

SELECT * FROM policy
  FOR SYSTEM_TIME BETWEEN TIMESTAMP('2011-11-30')
    AND TIMESTAMP('9999-12-31')
  WHERE id = 1111;
```

Figure 4: Using the three supported periodic specifications

Creating and using tables with business time

Business time involves tracking when certain business conditions are, were, or will be valid. Unlike system-managed temporal tables, there is no separate history table. You maintain past, present, and future effective dates and their associated data in a single table.

To create a table with business time, use the `PERIOD BUSINESS_TIME` clause (`bus_start`, `bus_end`), and include start (`bus_start`) and end (`bus_end`) columns. They can be date or timestamp data types. An implicit DB2 constraint enforces that `bus_start` values are less than `bus_end` values, ensuring temporal data integrity. To enforce that specified keys are unique with respect to the specific time period, specify the primary key constraint `BUSINESS_TIME WITHOUT OVERLAPS` as the last item with the list of other keys. For example, specifying `PRIMARY KEY(id, BUSINESS_TIME WITHOUT OVERLAPS)` in the policy table would ensure that two versions of the same policy that are valid at the same time cannot exist.

While inserting a row into a table with business time, the user must explicitly specify the start and the end date columns. By default, `UPDATE` statements behave traditionally for tables with business time periods. The period clause with `FOR PORTION OF BUSINESS_TIME FROM value1 TO value2` can be used to apply the update to the specific business time period. The `UPDATE` with period clause implies that all rows between the time period from `value1` to `value2` must be updated.

You can delete data pertaining to specific business time periods by specifying the time range in the `FOR PORTION OF BUSINESS_TIME` clause. If the row to be deleted has data that is not fully contained within the specified time range, DB2 automatically performs the implicit inserts, ensuring the appropriate information is preserved.

`SELECT` statements without any period specifications can be used normally. Temporal queries involving business time are also easy to write—and you need only include `FOR BUSINESS_TIME` in the `FROM` clause. Similar to system-managed temporal tables, three types of queries—involving `AS OF`, `FROM... TO...`, and `BETWEEN... AND...`—are supported. DB2 automatically rewrites the queries with appropriate `WHERE` clause predicates depending on the data types used in the start and end columns of the temporal tables.

For example, multiple rows pertaining to the same policy with different coverage amounts at different time periods can be inserted into the policy table.

ID	Coverage	Bus_start	Bus_end
1111	500000	2010-01-01	2011-01-01
1111	750000	2011-01-01	2012-01-01
1111	900000	2012-01-01	9999-12-31

Trying to insert a row for the same policy with start and end dates as 2011-06-01 and 2012-01-01 would fail, as the `BUSINESS_TIME WITHOUT OVERLAPS` clause ensures that temporal uniqueness is maintained (earlier, achieving this involved significant programming effort).

`UPDATE` with a period clause implies that all rows between time period `value1` to `value2` must be updated and can be used to apply the update to the specific business time period. For example, if you try to update the coverage amount to 600000 for the time period 2010-06-01 and 2011-09-01, the resulting table would be as follows:

ID	Coverage	Bus_start	Bus_end
1111	500000	2010-01-01	2010-06-01
1111	600000	2010-06-01	2011-01-01
1111	600000	2011-01-01	2011-09-01
1111	750000	2011-09-01	2012-01-01
1111	900000	2012-01-01	9999-12-31

Temporal queries made easy

DB2 handles these versioning and temporal queries automatically and transparently with minimal impact on existing applications. IBM's approach is expected to become the standard among databases, and the next edition of SQL standard will include temporal functionality. Now you can handle data version management and effective tracking of business events using this technique instead of application-level triggers and stored procedures. Use temporal tables and save yourself some time! ✱

Sasirekha Rameshkumar is an architecture consultant with more than 18 years of IT experience with mainframes, IT architecture, and performance management. He has published articles in Xephon, DB2, and ISSA Journal. You can read his blog at itknowledgeexchange.techtarget.com/enterprise-IT-tech-trends and e-mail him at rsasirekha@tcs.com.

RESOURCES

DB2 for z/OS Information Center:

publib.boulder.ibm.com/infocenter/dzichelp

A Matter of Time: Temporal Data Management in DB2 for z/OS (IBM Silicon Valley Lab):

public.dhe.ibm.com/software/data/sw-library/db2/papers/A_Matter_of_Time_-_DB2_zOS_Temporal_Tables_-_White_Paper_v1.4.1.pdf

The Rise of the Developer- DBA

Let the database take care of itself. The DBA of the future has more important things to do.

There was a time, not so many years ago, when the worlds of the DB2 DBA and the developer of DB2-accessing applications were fairly distinct. DBAs, true to their acronym, really were database administrators. They created objects (tables, indexes, views, table spaces, and so forth), backed up and (when necessary) recovered data, monitored the organization of tables and indexes (and reorganized these as needed), managed the use of disk resources, and generally looked after the database in a “system” kind of way.

Developers, meanwhile, translated business functionality requirements into application logic, designed user interfaces, and determined—among other things—what data elements had to be stored in DB2 tables to support application functionality. Developers and DBAs came together to create logical database designs and periodically to isolate and resolve data-access-related application performance problems; otherwise, these role-differentiated IT professionals generally tended to their respective responsibilities without a whole lot of intergroup interaction.

That was then...

For quite a while, this mode of operation made sense and worked well. DB2 databases (as was true of databases implemented via competing database management systems [DBMSs]) needed a good bit of attention, and this required DBAs to exercise some pretty highly specialized skills that had little to do with programming (aside from coding SQL queries to get information from the DB2 catalog). Developers did their thing with C or COBOL or Java or whatever, and they knew enough about SQL to get the job done.

Then, some things started happening that ended up increasing—often substantially—the overlap between the DBA and developer worlds. One of these change factors was the drive by IBM to build so-called autonomic capabilities into DB2. That term refers to functionality that enables DB2 to manage itself to a greater and greater extent. Autonomic features introduced on the mainframe platform included DB2-managed data sets, type 2 indexes (do any of you old-timers remember grappling with index page locking?), utility TEMPLATES and LISTDEFS, and sliding-scale secondary space allocation. For DB2 for Linux, UNIX, and Windows (LUW) DBAs, the goodies included automated RUNSTATS, automatic-growth databases, and self-tuning memory management. All of these combined to free up DBAs’ time to do things other than administer databases.

There were DB2 people (I’ll admit: I was one) who actually fretted about the elimination of formerly time-consuming database management activities, worrying that their employers would have less need for them. Other more forward-thinking DBAs moved their focus to where the value was. That is to say, they looked to see how they could use their newly available bandwidth to their organizations’ maximum benefit, and when they did this, these DBAs often reached similar conclusions: accelerating application development is where it’s at. After all, hardware and middleware (like DB2) don’t, by themselves, make any money for a business. *Applications* move products, take care of customers, manage inventory, and uncover new opportunities. Get those apps into production faster, and you are impacting the bottom line in a positive way.



Robert Catterall

(rfcatter@us.ibm.com)

is an IBM DB2 specialist.

Not just the time to help—the tools, too

Another factor in the increased engagement of DB2 DBAs in application development was the ongoing delivery by IBM of DB2 features that reduced the coding burden on application developers. Just a few examples:

- ▶ **Triggers:** A flexible and powerful mechanism for pushing logic into the database level of an application, in the form of user-defined actions that DB2 will take automatically when a program changes data in a table
- ▶ **Stored procedures and user-defined functions:** Convenient, scalable, and secure means of packaging data-access functionality in a very open, reusable form
- ▶ **pureXML:** A breakthrough technology that allows DB2 to recognize the hierarchical structure of XML documents stored in a database and to efficiently navigate that structure for data retrieval and update purposes
- ▶ **Temporal data support** (delivered initially with DB2 10 for z/OS): Another technological leap forward that greatly facilitates data-row versioning (“system time”) and future point-in-time database updates (“business time”)

So, just when DB2 DBAs (thanks to autonomous features) have more time to work with application developers, they are able to bring more value to the table by helping developers to understand and effectively leverage DB2’s programming productivity-boosting capabilities. Not only that, but DBAs are actually getting more into application coding themselves, thanks to DB2’s SQL Procedural Language (SQL PL), which can be used to implement sophisticated logic for stored procedures, user-defined functions (scalar and table), and triggers.

The IT organizational impact

I mentioned previously the forward-thinking DBAs who positioned themselves to deliver more value to their employers by teaming with developers to speed up the creation and deployment of business-impacting applications. Some management types recognized this shift and moved to support it in an

organizational sense. One such supporting action was the introduction of new job roles for application-oriented DB2 DBAs.

At one company for which I did some work a few years ago during my independent consulting days, the new job got the title “procedural DBA.” For that organization, the role was largely focused on the development and implementation of DB2 stored procedures. Other organizations go with a term such as “development DBA.” Whatever the appellation, these roles often draw major interest from established DBAs, who see an opportunity to take their careers forward in new and interesting ways.

For their part, development teams are typically more than just a little pleased to have their efforts augmented by those of a DBA who’s well versed in the application-enabling features of DB2. I’ve already mentioned the potential for development speed-up that can be realized via the coding by DBAs of data-access functionality in SQL PL, but helping to meet or beat delivery dates is just part of the story. There’s more.

Ensuring scalability. The developer-DBA can help to architect an application for maximum scalability on the database side by exploiting such things as static SQL, multi-row (also known as block) `FETCH` and `INSERT`, and DB2’s built-in WebSphere MQ interface (sometimes asynchronous processing is a key part of success in meeting the demands of a high-volume workload).

Satisfying security requirements without hamstringing programmers. DB2 features such as roles and trusted contexts can give developers the data access they need to do their jobs (and end users the privileges they need to use new applications) with the precision, control, and traceability demanded by security managers and auditors.

The reusability ripple effect. When data-access logic is pushed into the DBMS itself, the application development benefits can be wide ranging. For example, a DB2 for z/OS SQL stored procedure developed to support a Java application could just as easily be called

from a CICS transaction program written in COBOL. A trigger defined on a table in a DB2 for LUW-managed database could enforce complex data-domain rules for inserts from programs written in all kinds of languages, from C# to Ruby. SQL user-defined functions can simplify query coding for batch as well as online transactional programs. Get those data-access building blocks built, and let developers concentrate on coding business and user-interface logic.

So, developer-DBAs can do a lot to enhance the effectiveness of application development teams. What’s not always clear is where application-enabling DBAs should be placed with respect to the programmers. Should they be “embedded” with development groups, so that their desks are in close physical proximity to those of programmers? That might seem like an obviously good choice, but if it results in developer-DBAs being spread out among physically dispersed application development teams, DBAs may not get together themselves. Esprit de corps and cross-pollination (idea sharing) could suffer. On the other hand, keeping the DBAs physically together might lessen their positive impact on application development if face-to-face time with programmers is thereby reduced. In truth, both of these DBA-placement approaches—embedded and centralized—have been used successfully by different organizations. Go with what works in your environment.

A role with a future

The trends I described in this article—new autonomous and new application-oriented features arriving with each new release of DB2 (on all platforms)—are virtually certain to continue. It’s what IT leaders at DB2-using organizations want: “Let DB2 manage itself, and let my DB2 experts apply their talents to the development of applications that save my company money, grow revenue, and enable us to achieve separation from our competitors.” So, if you’re thinking that developer-DBA is a horse you can ride for a long time, I’d say you’re right. Saddle up. ✱

Automatic Storage Databases

Let DB2 decide where your table spaces live

Table spaces have long been used to provide a layer of indirection between database objects (tables, indexes, views, and so forth) and the storage containers (directories, files, or raw logical volumes/disk partitions) where data physically resides. And DBAs have often been responsible for creating any table spaces needed, monitoring table space storage consumption, and providing additional capacity to table spaces in danger of running out of space. However, beginning with DB2 8.2.2, DBAs have been able to offload a significant amount of this work to the DB2 database manager by creating what are known as “automatic storage” databases.

In this column, I’ll introduce you to the concept of automatic storage and I’ll show you how to construct a simple automatic storage database. I’ll also show you how to use a feature introduced in DB2 9.7 to convert an existing “traditional” database into one that uses automatic storage.

How database storage has traditionally been managed

DB2 supports two very different types of table spaces: system managed space (SMS) and database managed space (DMS). With SMS table spaces, directories are used as storage containers and the server operating system’s file manager is responsible for controlling how the space is utilized; DB2 creates files in the directories for objects that have been assigned to the table space and the file system

is responsible for managing their growth. With DMS table spaces, files or raw logical volumes/disk partitions are used as storage containers instead, and DB2 is responsible for controlling how the space in those containers is employed.

When a database is first created, three table spaces are created by default. These table spaces are as follows:

- ▶ A large table space named SYSCATSPACE, which is used to store the system catalog tables and views associated with the database
- ▶ A large table space named USERSPACE1, which is used to store all user-defined objects along with table data, index data, large object (LOB) data, and long value data
- ▶ A system temporary table space named TEMPSPACE1, which is used as a temporary storage area for certain operations

By default, these table spaces are SMS table spaces that use separate subdirectories on a single file system as storage for the database. (Additional table spaces can be created by executing the `CREATE TABLESPACE` statement.) Figure 1 shows the underlying storage for a database containing the three default table spaces, plus a DMS table space named MY_SPACE1 that uses two different files (residing on two separate file systems) for its storage.



Roger E. Sanders

(roger_e_sanders@yahoo.com), a senior consultant corporate systems engineer at EMC Corporation, is the author of 21 books on DB2 for Linux, UNIX, and Windows and a recipient of the 2011 IBM Information Champion award. His latest book is titled *From Idea to Print: How to Write a Technical Article or Book and Get It Published*.

Special thanks to Robin Grosman, DB2 development manager, buffer pool services and data movement utilities, for providing feedback on the information presented in this article.

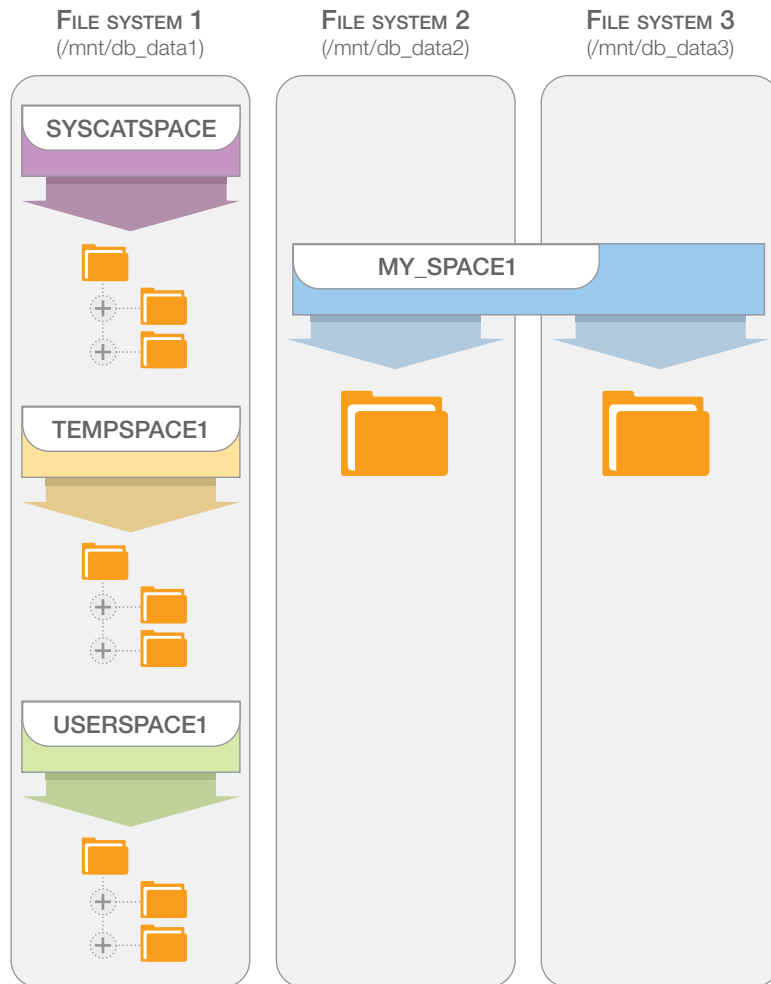


Figure 1: Storage configuration for a traditional DB2 database

When a table space spans multiple containers, data is written in a round-robin fashion—in groups of pages called *extents*—to each container assigned to that table space. This approach helps balance data across all containers used. Thus, the DMS table space MY_SPACE1 shown in Figure 1 uses two file containers since this is the way such a table space would most likely be created.

How automatic storage works

Instead of associating storage containers with individual SMS and DMS table spaces, you associate storage paths with the database as a whole when you create an automatic storage

database. Containers for the default table spaces, as well as any other table spaces that are defined, are then automatically created *on each storage path used*. The table spaces themselves take on the characteristics of auto-resizing DMS file table spaces (for permanent data) or SMS table spaces (for temporary data). And as the database grows, the DB2 database manager automatically extends the appropriate containers (or creates new ones) to meet the database's storage needs. Figure 2 shows the underlying storage for an automatic storage database containing the three default table spaces and one user-defined one named MY_SPACE1.

Creating an automatic storage database

A database can be configured to use automatic storage when it is first created, and with DB2 9.1 and later, new databases are created using automatic storage by default. The storage paths that are to be associated with the database are provided as part of the `CREATE DATABASE` command used to create the database; the syntax for the simplest form of this command looks something like this:

```
CREATE [DATABASE | DB] [DatabaseName]
<AUTOMATIC STORAGE YES>
<ON [StoragePath ,...]>
<DBPATH [DBPath]>>
```

where:

- ▶ **DatabaseName** identifies the unique name that is to be assigned to the database to be created.
- ▶ **StoragePath** identifies one or more storage paths that are to be used to hold table space containers for each table space defined in the database.
- ▶ **DBPath** identifies the location where the database metadata and transaction logs associated with the database are to be physically stored. (If this parameter is not specified, these files will be stored in the first storage path specified in the *StoragePath* parameter.)

Thus, if you wanted to create a database named MY_DB that uses automatic storage, stripes its data across three file systems, and stores its metadata and transaction logs on the first file system specified, you would execute a command that looks like this:

```
CREATE DB my_db
AUTOMATIC STORAGE YES
ON /mnt/db_data1, /mnt/db_data2,
/mnt/db_data3
```

And if you want to create a table space named MY_SPACE1 in this database, simply execute a `CREATE TABLESPACE` statement that looks like this:

```
CREATE TABLESPACE my_space1
```

The resulting table space will have a container on each storage path used by the database. If a DMS table space is created, 32 MB of storage space will be allocated for the table space initially; to determine the size of each container used, divide this number by the number of file systems the database spans.

Converting an existing database to an automatic storage database

Beginning with DB2 9.7, you can modify an existing database—even a database that was not created as an automatic storage database—to use automatic storage simply by executing the `ALTER DATABASE` statement. The syntax for this statement is:

```
ALTER DATABASE [DatabaseName]
[ADD | DROP] STORAGE ON [StoragePath ,...]
```

where:

- ▶ **DatabaseName** identifies the name of the database that is to be altered.
- ▶ **StoragePath** identifies one or more storage paths that are to be used to hold table space data for each table space defined in the database—if storage paths are to be added. If storage paths are to be dropped, this parameter identifies one or more storage paths that are to be removed from the collection of storage paths that are used to hold data for automatic storage table spaces.

Thus, to modify an existing database named `TEST_DB` so that it uses automatic storage, you would execute an `ALTER DATABASE` command that looks like this:

```
ALTER DATABASE test_db
ADD STORAGE ON /mnt/db_data1,
/mnt/db_data2
```

When executed, this statement has the effect of both adding two new storage paths to the existing database, as well as enabling the database for automatic storage; any future table spaces created will use automatic

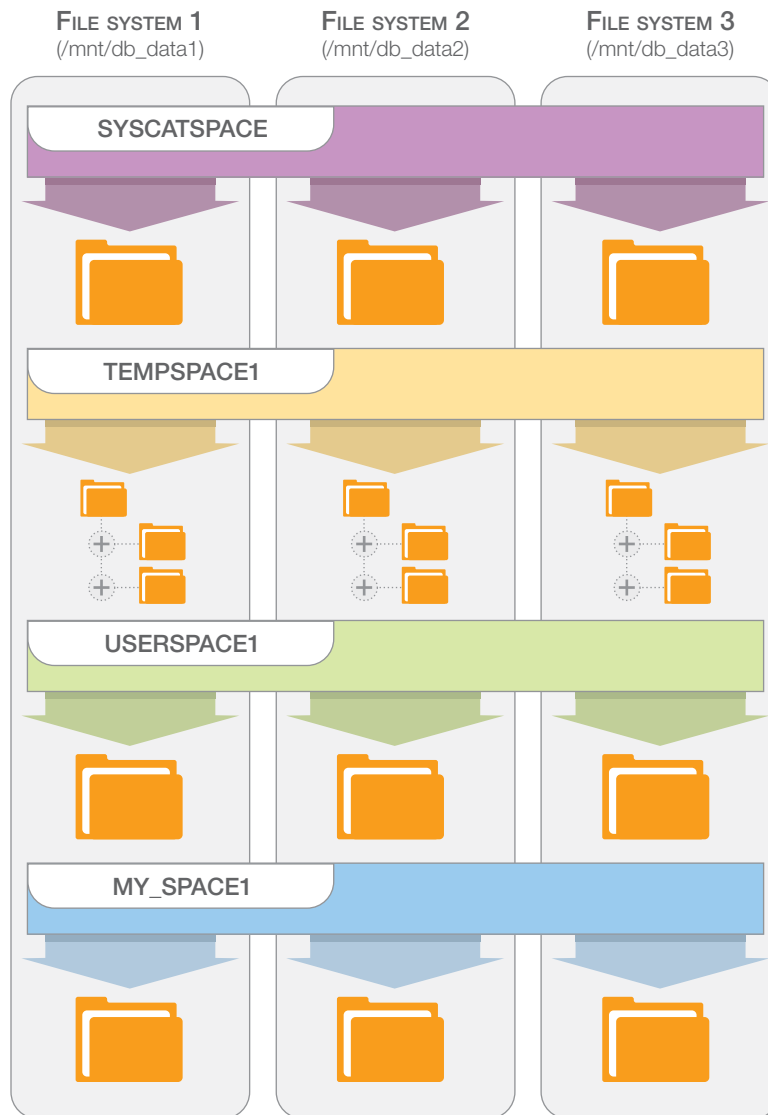


Figure 2: Storage configuration for an automatic storage database

storage by default. However, existing nonautomatic storage table spaces will not automatically be converted. Instead, you must use the `ALTER TABLESPACE... MANAGED BY AUTOMATIC STORAGE` statement to make the conversions yourself. Only DMS table spaces can be converted; the conversion of SMS table spaces is not allowed.

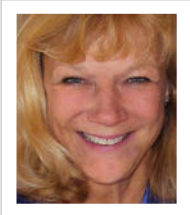
Finally, there's an important caveat you need to remember: once a database has been created or converted to use automatic storage, it cannot be converted to a nonautomatic storage database.

Conclusion

The primary reason for the introduction of the automatic storage model was to simplify the management of database storage, while retaining the performance characteristics typically found when DMS table spaces are used. (DMS table spaces tend to be faster, in terms of input/output operations per second or IOPS, than SMS table spaces.) There may be situations where a DBA must retain full control over how the storage for a particular table space is provided, but in most cases, databases will benefit from the use of automatic storage. *

PLANS + COLLECTIONS + VERSIONS + PACKAGES + DBRMs = **Confusion**

Revisiting program preparation on DB2



Bonnie Baker

(bkbaker@bonniebaker.com) specializes in teaching onsite classes for corporations, agencies, and DB2 user groups. She is an IBM DB2 Gold Consultant, an IBM Information Champion, a five-time winner of the International DB2 Users Group (IDUG) Best Speaker award, and a member of the IDUG Speakers' Hall of Fame. She is best known for her ability to demystify complex concepts through analogies and war stories.

ong ago, I wrote a three-part column to explain the program preparation process on the DB2 for z/OS platform. I received more e-mail referencing that column than all of my other columns put together. Now I'd like to revive that popular series, which is no longer available in the magazine archives. So, with some changes to reflect DB2 10, I am going to share that same information in a condensed column, using the basic preparation process for a COBOL batch program as my basis. This much-wanted information will be back on the Internet, and I will have a link for those who ask.

Write the program

When we write a program that needs data from DB2 tables, we embed SQL in the program at the appropriate execution spot and surround it with delimiters. The delimiters for COBOL are `EXEC SQL` and `END-EXEC`. Between pairs of delimiters, we also code `INCLUDE member-name` (which is similar to coding `COPY copybook-name`) to specify each member we want brought into our program from the INCLUDE library. The INCLUDE library holds DCLGENs. We include DCLGENs for each referenced DB2 table because a DCLGEN consists of an image of a DB2 table (the way it looks in DB2-speak) and a COBOL working storage structure into which rows can be read and from which rows can be written.

With the advent of multi-row processing, many development groups now have two DCLGENs for each table, one standard DCLGEN and one multi-row DCLGEN in which each column occurs approximately 100 times.

Prepare the program

Once the program is written, you must put it through a process so that it can run and so that it can connect to DB2 whenever it needs to. The first step of this process is to `PRECOMPILE` your COBOL source code. Both your program source code in your COBOL SOURCELIB and your DCLGENs in your INCLUDE library are input to the DB2 precompiler.

The precompiler does three things. To do these three things it must read your program top to bottom in one pass, looking for your SQL delimiters. If it finds an `INCLUDE member-name` between the delimiters, it copies the member into your program. If it finds SQL, it does a modest syntax check. Why modest? Because it has only the table description from the DCLGEN as a reference. You see, the precompiler is not connected to DB2, which means that running `PRECOMPILE` does not add any DB2 overhead. The third thing that the precompiler does seems contrary to reason: You have very carefully embedded SQL into your COBOL program—the precompiler just as carefully removes it. The precompiler replaces any executable SQL that was in your source with a COBOL `CALL` to a COBOL-DB2 interface module. Non-executable SQL statements (like your `DECLARE CURSOR`s and the `DECLARE` portion of the DCLGENs) are simply commented out.

At the end of the `PRECOMPILE` step, you have two entities—fraternal twins. One twin is the modified COBOL source code with its `INCLUDE`d members and its calls to DB2, but no SQL. The other twin is a database request module (DBRM) containing all of that SQL

that was removed from your program. To make matters more complicated, the precompiler gives these fraternal twins the same name, the name of the program, and they go down separate paths outside of DB2's control.

Eventually, when Twin 1, the COBOL program, runs, it will need to find and identify Twin 2. Therefore, the precompiler "tattoos" each twin with a modified time-stamp. This "tattoo" will be used at run time for the COBOL twin (the searcher) to identify the SQL twin (who is waiting to be found) to make sure that both twins came out of the same **PRECOMPILE** step.

Twin 1: From SOURCE to LOAD

To create a load module, the newly generated, modified COBOL source code must go through the standard COBOL **COMPILE** and **LINK EDIT**. One key module that must be **LINK EDIT**ed into the program is the COBOL-DB2-TSO batch interface module. The resulting load module (with its "tattoo") is put into a LOAD library where it waits to be executed.

Twin 2: From DBRM to PACKAGE

The SQL in its DBRM now must go through a process similar to **COMPILE** to produce run-time code. In DB2, the "compile" process is called **BIND**, and the "load module" is called a **PACKAGE**. **BIND** is to your DBRM what **COMPILE** is to your COBOL. **BIND** must take your source code (the SQL) and turn it into executable code (a **PACKAGE**).

To convert your SQL into executable run-time instructions, **BIND** attaches to DB2 and does three things. It first verifies that you are authorized to do a **BIND**. Second, it checks the syntax of your SQL. We know that a syntax check may have been done at **PRECOMPILE** time, but that only happened if we **INCLUDED** a **DCLGEN**. Also, that syntax check was only as accurate as the **DCLGEN**. This **BIND**-time check is much more thorough and sophisticated because it will use data from the DB2 catalog. Third, it will perform the most important step: optimization. During this step the DB2 Optimizer will turn our SQL into executable code.

The DB2 Optimizer creates run-time instructions for each SQL statement. It will evaluate your SQL, asking questions like: What columns did you **SELECT**? What table(s) do you want to read? What conditions are in your **WHERE** clause? It will check the DB2 catalog for useful statistical information and to see what indexes are available. Using this information, the DB2 Optimizer will determine the cost of using each index. It will also check the sizes of your **RIDPOOL** and your **BUFFERPOOL**, find out if you used **BIND** parameters like **DEGREE ANY** or **ISOLATION UR**, and much, much more. The DB2 Optimizer considers all of this information and for each possible path assigns a cost in terms of I/O, CPU, **RID SORT**, and **DATA SORT** overhead. Then it will either pick the cheapest access path or (as of DB2 10) a slightly more costly but less risky path. (More on that in a future column.)

Once the path is chosen, the run-time code can be generated and put into a **PACKAGE**. So let's review the process so far: First, all of the SQL for a single program becomes a single **DBRM**. That **DBRM** becomes a **PACKAGE** (our DB2 load module). And all of these entities have the same name, and that very important "tattoo" has been copied into each.

A copy of the **DBRM** is stored in the DB2 catalog (our DB2 "SOURCELIB") for documentation and future **REBINDS** (a topic left for another column). The **PACKAGE** is written to the **DIRECTORY** (our DB2 "LOADLIB") where it sits and waits for its twin, the program's load module, to find it.

COLLECTIONS

Many **PACKAGEs** have characteristics in common. Maybe one group of **PACKAGEs** is used by the payroll application. Maybe another group is very popular and called by every program regardless of application area. Maybe they share a special, not so common **BIND** parameter like **OPTHINT** or **REOPT(ALWAYS)**. DB2 has anticipated this situation by providing a high-level grouping name for your **PACKAGEs**. This is called a **COLLECTION**. You specify which **COLLECTION** you would like each **PACKAGE** to be part of by using the **INTO collection_name** clause in your **BIND** statement.

The reunion

How do our twins reconnect? Often programs call other programs, which then call even more programs. The **PACKAGEs** for all of these programs may have been bound into the same **COLLECTION** or different **COLLECTIONs**. To tell our load module (Twin 1) where to look for the **PACKAGE** (Twin 2), we give it a list of **COLLECTIONs** that it is allowed to search, by creating a **PLAN**. To create the **PLAN**, we perform another **BIND**. This **BIND** (**BIND PLAN**) specifies the search chain. It lists each **COLLECTION** that is a possible location for the **PACKAGE(s)** that will be needed by our **EXECUTE PROGRAM** statement in our **JCL**. By providing this **PLAN**, we tell our COBOL load module exactly where to search for its **PACKAGE**. And at run time, at the first **CALL** to DB2, Twin 1 will search for Twin 2 in each **COLLECTION** named in the **PLAN** until it finds its twin—with the same name and the same "tattoo." If it does not find its twin, the program will receive a **-805 SQLCODE**. A **-805** means, "I searched for my twin in every **COLLECTION** that you gave me in the **PLAN** list, and I could not find him."

Multiple VERSIONS?

Sometimes we would like to keep multiple **VERSIONs** of the same **PACKAGE**. For example, in a fallback situation, it would be nice to have both the current **VERSION** and the prior **VERSION** of a **PACKAGE** available. If we have both, for quick fallback, we could just point to the old load module in our **JCL**. Maybe we want one **VERSION** bound with **ISOLATION UR** and another bound with **ISOLATION CS**. If we have both, we could run with **UR** during read-only windows when all of the rows are committed and run with **CS** when maintenance is being performed and we need to be protected from "dirty" data.

Confusion eliminated

There are infinite possibilities for setting up your **PLANs**, **COLLECTIONs**, **VERSIONs**, and **PACKAGEs**. I hope this column has not only clarified the confusion but has also helped you see those possibilities. ✱

Informix Warehouse Accelerator

Shockingly fast performance



Lester Knutsen

(lester@advancedatools.com)
is president of Advanced DataTools Corporation, an IBM Informix consulting and training partner specializing in data warehouse development, database design, performance tuning, and Informix training and support. He is president of the Washington, D.C. Area Informix User Group, a founding member of IIUG, an IBM Gold Consultant, and an IBM Data Champion.

At the end of March, IBM announced the new Informix Warehouse Accelerator (IWA) for Informix 11.70.xC2. I have been beta testing this accelerator for a while and would like to share some of my findings with you. IWA is an exciting breakthrough—combining new data warehouse technology with traditional Informix relational database servers—that results in very fast performance. My benchmark run of 16 queries went from 9 hours and 40 minutes to just over 15 minutes using this new technology. The chart in Figure 1 shows the results of my benchmark testing.

IWA technology loads the records from a data warehouse star schema into memory in a compressed format, and organizes the data using new columnar storage technology. The records are not stored as rows but as columns of data that can be very quickly accessed using this new technology. When an application sends a query to Informix, the Informix optimizer decides whether the query should be sent to IWA. When the Informix server gets the data back from IWA, it returns the results to your application—only much faster than before (Figure 2). Your applications do not need to change or be recompiled to benefit from the speed of IWA. My favorite business intelligence query tools continue to work unchanged with Informix: Cognos, Hyperion, Pentaho—they all just work. All ODBC, JDBC, or ESQL/C applications connect to Informix and do not even need to know that they are connecting to IWA.

From one hour to nine minutes

Based on my testing, one-hour queries run in two to three minutes with IWA. Disk access is the slowest part of a data warehouse query, and when you need to read millions of records to get a result set for a query, the performance of your disks determines how fast you can get data back. The best performance gains with IWA are in table scans of large fact tables, where disk activity is normally the limiting factor.

Over the years, I have built up a midsize data warehouse star schema that I use for testing new releases of Informix, and I used it for this benchmark. It contains a fact table of 630 million records, a customer dimension of 2.5 million customers, and three small dimension tables for organization, programs, and commodities. It uses real data on tracking government payments to farmers, which is available to the public by request.

The 16 queries in the benchmark were based on real application query requests I have seen from business users. Some seem simple, like the first one in the chart in Figure 1. The objective is to find the top 1,000 farmers that received payments. This query requires reading all 630 million records, sorting and summing the payments by farmer, and returning 1,000 records. This query can take hours to run on big servers with expensive storage systems.

Running this query without IWA on Informix took more than an hour to read,

Run	Query	Type of query	IDS only	IDS & IWA	Improvement multiple	Rows returned
1	Query12	Scan - Select First 1000	105m7.730	9m48.071	10.7261	1000
2	Query13	Index - by Dim	0m0.116	0m11.605	0.0100	9
3	Query14	Index - by Dim	0m0.055	0m11.971	0.0046	9
4	Query15	Index - by Dim	0m8.345	0m4.469	1.8673	9
5	Query0	Scan - Select First 1000	106m44.474	0m20.946	305.7612	1000
6	Query10	Index - by Dim	0m4.817	0m6.062	0.7946	1
7	Query11	Index - by Dim	0m3.430	0m6.068	0.5653	1
8	Query1	Scan - Select First 100	63m33.083	1m13.501	51.8780	100
9	Query2	Scan - Select First 100	65m17.075	1m29.518	43.7574	100
10	Query3	Scan - Sum by Dim	87m47.842	0m17.207	306.1453	3107
11	Query4	Scan - Select First 100	53m57.169	0m14.162	228.5813	100
12	Query5	Index - by Dim	0m0.049	0m12.692	0.0039	0
13	Query6	Scan - Select First 1000	1m48.886	0m9.495	11.4677	1000
14	Query7	Scan - by Dim w Group by	0m15.855	0m9.171	1.7288	26937
15	Query8	Scan - by Dim w Group by	0m14.846	0m2.347	6.3255	28358
16	Query9	Scan - by Dim w Group by	79m14.253	0m42.011	113.1669	523499
Total in hours			9 hr 40 min	15 min 32 sec		

Benchmark configuration: • IBM System x Server • 256 GB RAM • Informix and IWA on the same machine

Figure 1: These test results compare the performance of Informix with and without Informix Warehouse Accelerator.

gather, sort, and sum all the data. The data set was too large to fit into Informix’s memory buffers. Running this query with IWA took nine minutes, and we were able to reduce the size of the Informix memory buffers. Both of these tests were from a cold start with no data in the memory buffers. Running the same query a second time on Informix alone did not speed things up much because all the data would not fit in memory and the buffers were continually thrashing. In contrast, the same query run a second time with IWA finished in less than a minute.

When Informix takes over

For some queries, IWA does not improve performance. As the chart in Figure 1 shows, most queries that returned nine or fewer records and can use an index are faster without IWA. Most were very fast queries, and Informix returned the results in less than a second. IWA does not handle all types of queries yet, such as correlated sub-queries or queries using temp tables. One of my tests used a query to select records into a temp table and then a second query to further refine the results from the temp tables. IWA could send the results from the first query into a temp table, and that was very fast. But since the temp tables had not been defined and realized in IWA, Informix handled the second query as a normal query. Any query that cannot be optimized by IWA is optimized and handled by Informix without users even knowing it, except that they will notice the slower response time.

A database must be designed using a dimensional model star schema to work with IWA, and the tables must be defined and loaded into IWA to take advantage of its performance. Load times were very fast, comparable to creating some of the summary tables I traditionally build in a data warehouse. In fact, I see IWA replacing the need for many of the summary tables I design in a data warehouse.

I have been building data warehouses with Informix for more than 18 years, and it has been getting better and faster with each new release. The Informix Warehouse Accelerator really raises the performance level that we can get and expect from a data warehouse. It is also exciting to see IBM invest so heavily in Informix and its role as a data warehouse engine. *

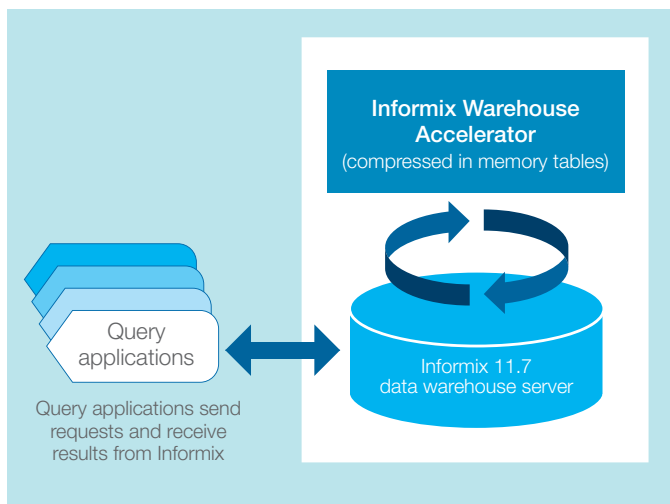


Figure 2: Informix manages the Informix Warehouse Accelerator so that applications do not need to be aware of its presence.



Smarter is...

Saving Minds, Saving Lives

Columbia University Medical Center's Neuro-ICU uses IBM InfoSphere Streams software to isolate early warning signs of a damaging and often deadly stroke complication.

Imagine that you're a doctor. Your patient has suffered a type of stroke called subarachnoid hemorrhage (SAH). He's currently stable, but a terrifying danger looms ahead. In roughly one-third of SAH victims, the blood vessels suddenly constrict, slowing the flow of oxygen-rich blood to the brain and causing a condition known as *delayed ischemia*. When that happens, there's a 50 percent chance that your patient will suffer irreparable damage or death.¹

Physicians currently monitor for delayed ischemia by performing an hourly neurological exam. For example, nurses ask questions of patients each hour. If they answer one hour but the next they slur their speech, this often indicates a change in their condition. But further diagnostic tests would have to follow, and the tests available today have proven largely unreliable. Worse, if the physician waits until these symptoms start before beginning treatment, it is too late for fully half of those patients, who go on to have ischemic strokes, according to Dr. J. Michael Schmidt, director of clinical

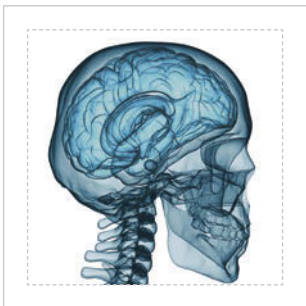
neuro-monitoring and informatics at the Neuro-ICU of the Columbia University College of Physicians and Surgeons.

Physicians in the Neuro-ICU believe indicators of delayed ischemia appear somewhere in 200 treatment-related data points 24 to 48 hours before the complication's devastating arrival. But no human mind can process data from that many variables.

Instead, IBM researcher Nagui Halim and Dr. Schmidt are using IBM InfoSphere Streams to analyze floods of prerecorded clinical data, including CT scan blood findings, electroencephalogram outputs, blood pressure readings, and blood oxygen levels taken from a pool of more than 300 past patients.

"The main feature we are looking at is reductions in heart rate variability that relate to inflammation, which is part of the pathogenesis of delayed ischemia," says Dr. Schmidt. "We are seeing enough indicators in the heart rate signal to make us think this is a promising pursuit."

The team hopes to have some preliminary data models this summer. The next step would be to validate the models against data from new patients. If the data and models prove to be solid, physicians could soon be able to detect when a patient is at high risk of delayed ischemia. Ultimately, analyzing these streams of data could save a mind, or a life. ✱



David Geer is an Ohio-based technology journalist.

¹ S. Dorhout Mees, G. Rinkel, V. Feigin, et al., "Calcium antagonists for aneurysmal subarachnoid haemorrhage," *Cochrane Database of Systematic Reviews* 2007, Issue 3: CD000277, doi: 10.1002/14651858.CD000277.pub3.

In the time it takes to
 browse through this magazine, you could
download, install, and see remarkable breakthrough
 performance **results** from our software

We're the world's fastest route to DB2 LUW productivity and performance. In fact, you can download, install, and actually see results in minutes instead of days, weeks, or months.

Only DBI offers all these reasons to be your smart choice for performance monitoring and tuning tools:

DBI	IBM OPM*	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Patented SQL Workload Cost Analysis
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Tuning Results in 5 Mouse Clicks or Less
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Automatically Tracks Database Changes
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Trend Charts Plot Change Events on Graphs
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Identifies SQL driving I/O to a Table
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Summarizes Programs, Users, and their SQL Costs
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Shows All Users that ran a particular SQL

Contact DBI to experience our results-oriented Proof of Concept process and discover why **Smarter People choose Smarter Tools** for DB2 performance and cost optimization. Act now and receive a free white paper by Scott Hayes, IBM DB2 GOLD Consultant, which discusses critical performance measurements and provides complimentary SQL commands.

www.DBISoftware.com/smarter

Plug and play with IBM's Netezza. Or plug and wait with Oracle Exadata.

IBM's Netezza data warehouse appliance makes analytics fast and easy.

- Easy to install: it's up and running in 24 hours.
- Easy to pay for: it's as low as 1/6 to 1/2 the cost of Oracle Exadata.
- Easy to love: 4 out of 5 trial customers keep it for good.

ibm.com/facts

