IBM DATABASE MAGAZINE » Powering business information

Quarter 2, 2006 Vol. 11, Issue 2

http://www.dbmag.intelligententerprise.com/story/showArticle.jhtml?articleID=186700013

User Views

Read: Design for Performance By David Beulke
Read: The Power Of Purchasing By Stuart Litel

Design for Performance

Take the time for proper design, or be prepared to do it again.
By David Beulke

Relational databases and relational set theory hold many performance advantages over other networked and hierarchical databases. Relational set theory and the ability to process multiple rows through one database SQL statement can be very powerful. Unfortunately, I continue to find performance issues created by database design in applications that were migrated to DB2 years ago. Relational database design is extremely important. Without correct design, many application systems continue to process table data a row at a time — and, therefore, fail to leverage DB2's power.

During client performance reviews, I (too often) find applications that are only doing single-row application processing. In such systems, increased transaction volume means that performance hits a wall. These applications open a cursor on one DB2 table, check retrieved values, and then retrieve other database information from another DB2 cursor or `SELECT` statement. This database access method is very inefficient; an "application join" program rewritten with a DB2 join usually saves a tremendous amount of CPU and processing time.

This problem shows up in core business databases whose table structures were never normalized and were certainly not designed for high transaction rates, the Internet age, or expansion into supporting new business opportunities. Normalizing your database design for peak performance is crucial. You must separate the data into the correct number of tables, minimize the number of I/Os for key business transactions, leverage partitioning or `UNION ALL` structures, and define the proper indexes for access and SQL join activity.

Recently, I came across a system in which the database had been converted from a hierarchical architecture. The design seemed to have one DB2 table for each hierarchical table. These converted DB2 structures needed to be joined in order to efficiently retrieve the data for a business transaction, which was costing the system extra CPU and I/Os. I

saw several indexes defined with duplicate entries allowed and many other indexes that each had the same columns defined, but in different orders. Industry best practices suggest four or fewer indexes per table. I deleted the duplicate indexes, leaving only those indexes that were defined with good cardinality and unique rule definitions. Because DB2 rarely uses or references indexes that have cardinality of less than 85 percent, it makes sense to get rid of indexes that are just extra overhead.

When I talked with some of the veteran DBAs and developers at this shop, they told me that the database was designed and converted from the old hierarchical database in a quick four months. Their management didn't take the time to understand relational advantages; they just needed the database to be converted to DB2. This poor design endured for more than two years — until they added additional business critical transactions...and hit that performance wall.

Database design is a very important component of overall performance; making improvements later will be more costly and time consuming than simply designing the database correctly in the first place. One of the DBAs I worked with likened the process of doing a database redesign while the application was in use to changing the design of an airplane's engines while flying across the country.

If you don't have the time to do your database design right, you'll eventually have to do it over. Next time you're rushed through the database design process, remember to keep your notes and ideas handy. In a few months or years, you'll probably have another chance to design the database to meet the business requirements.

You can learn more database design techniques, best practices, and industry standards at International DB2 Users Group conferences and DB2-ListServe.

David Beulk , a past president of IDUG, is a DB2 consultant, author, and lecturer who specializes in database performance, data warehouses, and Internet applications.

The Power Of Purchasing

Support the companies that support Informix.
By Stuart Litel

My first exposure to Informix came when I worked at Sheraton Corp in 1983. My team had to decide on what platform, database, and development language to use to rewrite the worldwide reservation system used in all Sheraton hotels throughout the world.

Early on, we decided to use Unix, and then we had to decide which hardware and database to use. We wrote our own realistic internal benchmarks, which modeled our business, for the hardware and the database. We chose Informix for the database and Informix 4GL as our development language because Informix far outperformed other databases in our test. We didn't pay attention to advertising or what other companies were

using. The performance of the reservation system was what mattered, and that system had to be up and running all the time. As my boss used to say, it was always between 9 a.m. and 5 p.m. somewhere in the world.

I find most benchmarks today are set up with such unrealistic complexity that they don't represent real-life situations. They simply show that the vendor stands behind its product.

If benchmarks are important to you, you'll be glad to know that Baan (SSA Global) recently did perform benchmarks with Informix Dynamic Server 10 (read more at www.iiug.org/news/articles/BaanIV_HPInformix_Insert_20060224.pdf). Baan is one of those independent software vendors (ISVs) that have continued to see Informix's strength. Unfortunately, others (such as PeopleSoft, now owned by Oracle) seem to be leaving their Informix customer base hanging in the wind when it comes to definitive future support.

Informix built its name on the ISV or value-added reseller (VAR) model. Vendors used the Informix database within their applications, and many customers didn't even know which database was under the covers. The Informix product ran so effortlessly and flawlessly that the ISV could drop the database right in and it just ran and ran and ran.

Today, some of these vendors have left the Informix product. Those that have stayed are reaping the rewards of an extremely loyal customer base that is growing, according to IBM insiders, by double digits. IBM's software divisions (Tivoli, Rational, Lotus, WebSphere, and Information Management) have made sure their products work with IDS.

So where have all the ISVs gone? The smart ones have stayed the course and are expanding as they see the benefits of partnering with IBM and using Informix.

I recently reminded PeopleSoft and Informix users that they are the customers, and the customer is always right. So, if ISVs are leaving Informix, let them know you intend to vote with your dollars. If you spend them with Informix ISVs, the ones who have left the fold will be the ones who lose out on the sale.

Stuart Litel is CTO of Kazer Inc. and president of the International Informix Users Group.

Return to Article