IBM DATABASE MAGAZINE » Powering business information

Quarter 2, 2005 Vol. 10, Issue 2

http://www.dbmag.intelligententerprise.com/story/showArticle.jhtml?articleID=161601910

User Views

By David Buelke and Stuart Litel

Autonomic MQTs

How implementing MQTs can reduce your CPU consumption.

by David Buelke

DB2's autonomic functions and features, particularly those that help utilities automatically understand when it's time to reorganize, check, or move data, are now part of many shops' daily operations.

One of the best new features in DB2 Universal Database (UDB) for z/OS v.8 is the materialized query table (MQT). MQTs originated with DB2 UDB for Linux, Unix, and Windows (first as automated summary tables). An MQT is a table set up from a (usually complex) SQL query result set. Other SQL queries can then use this precomputed, aggregated table to retrieve results. The DB2 optimizer can automatically rewrite SQL queries to access MQTs instead of base tables if it determines that the access paths are more efficient. By rewriting queries to use MQTs, CPU and I/O performance can be improved dramatically.

At one of my client sites, implementing an MQT eliminated more than 750 million I/Os daily from the SQL workload. Some MQT implementations have reduced CPU consumption and improved system performance enough that clients could avoid upgrading their CPU capacities.

Because they're simply tables created from the result sets of SQL queries, MQTs are easy to implement. Although there are some creation restrictions, MQTs quickly help both operational and data warehousing systems. I recently created MQTs for a client that precalculated weekly, monthly, and quarterly totals by doing `SUMs`, `COUNTs`, and `MAX` DB2 SQL functions. Before the MQTs, these various calculations and functions were being done repeatedly, resulting in many scans and sorts in the system workload. With MQTs, the calculations are done once, and then stored in the MQT. If any of the original data is altered, DB2 makes the changes in the MQT to keep it current.

Data administrators can analyze user workloads and create MQTs to satisfy the SQL that repeatedly beats up their systems. Because DB2 will automatically rewrite the query, workloads can be directed to the MQT without recoding the SQL or program. As a result, MQTs are a tremendous help for ad hoc querying or OLAP tool workloads for data warehousing.

Remember that MQTs are tables and require correct space allocations, runstats, and sometimes indexes. Because MQTs are built from base tables' SQL, they could be updated frequently if base tables are modified.

MQT creation parameters can automatically or manually control the data population. The DB2 MQT parameters `DATA INITIALLY DEFERRED` and `MAINTAINED BY USER` allow administrators to control MQT data population. DB2 manuals explain many other MQT parameters that provide flexible configurations.

MQTs provide another option for architects designing and implementing solutions that quickly provide answers to SQL questions. It's another of DB2's autonomic functions that make DBAs' lives easier.

David Beulke is a DB2 consultant, author, and lecturer who specializes in database performance, data warehouses, and Internet applications. He is a past president of the International DB2 Users Group (IDUG).

10-10-25

Tales from Informix's first 25 years.

by Stuart Litel

Over the years, I've noticed that virtually all Informix users will agree with the following statements:

- No database on the market is easier to administer or more reliable than Informix.
- No one ever bought Informix software for the advertising.
- If Informix had sold sushi, it would've called it "raw dead fish."
- In the old days, the number one job of an Informix DBA was to learn how to read the newspaper; today, it's to surf the Web.

A few years back I ran into an old friend who took over a full-time Informix position at a company where I'd done the original consulting. When I started consulting there, the company was doing about $10 million of business, with all the back-office accounting on a homegrown Informix 4GL solution. When it hired a full-time employee, it had grown to a $100 million; today, it's a household name and a $1 billion company.

When I last saw this friend, he told me he managed six employees: one Informix DBA

and five Oracle DBAs. I asked whether the company was changing to Oracle. It turns out the entire company runs on Informix except for one purchased solution that runs on Oracle; that system needs so much database attention that it takes five people to keep it going.

For everyone who smiled at that story, here are more thoughts about why we all love Informix:

- The database can always be online, even when doing a backup.
- You don't have to turn the database off to add, alter, or drop a table.
- If offers two-way bi-directional replication, without triggers or special SQL — the database simply does the replication for you.
- Informix can scale from a single-user mom-and-pop shop implementation to handle millions of transactions daily. (I recently hosted a video presentation about a large Informix implementation, which you can see at www.watchit.com/informix.)

Recently, I met with some users of a different database software. One customer had an Informix-based solution the company wanted to replace with that "O"ther database. The user, an IT director in a world-famous city, told me, "We don't like to use Informix because there's nothing to do, it's for amateurs."

I held my tongue, but I'm letting it go now. I've heard of an Informix system that's been up for more than 1,561 days. How many systems "for amateurs" do you see that have been doing online transaction processing for more than 1,500 days (or even 150 days) without a restart?

Informix (even under IBM) has never had great marketing (the cold, dead fish approach), but it's getting better. The last nine months have been a breath of fresh air from IBM. Now it's time for us to stand up and be counted. We have something special.

I'm currently searching for the following:

1) Any company that gave up its Informix database for technical reasons.

2) Any person or company that doesn't believe Informix will be here for another 25 years. I'm taking bets, and my money is on a 50th anniversary.

3) Ideas about how to convince everyone that Informix can beat any database on equal terms, in speed, reliability, availability, and cost of ownership.

Are you wondering why I called this column "10-10-25"? It's a good year: IBM announced Informix Dynamic Server 10, it's the 10th anniversary of the International Informix Users Group (IIUG), and the 25th anniversary of Informix.

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

Return to Article