Top Performance Features in DB2 9 for z/OS

Indexing, compression make your applications work faster

David Beulke, Founder/Principal, Pragmatic Solutions, Inc.

Summary: A look at indexing on expressions and index compression in DB2 9 for z/OS

Date: 19 Oct 2009
Level: Introductory

Activity: 2277 views
Comments: 1 (View or add comments)

★★★★★ Average rating (based on 2 votes)

IBM DB2 9 for z/OS is loaded with features that can boost the performance of your applications. In this column, I'll tell you about two that really stand out, but check out the online version of the column for a table of the top 50. Implementing any of these features can help reduce your applications' CPU burden and improve application response time.

After you convert to DB2 9 for z/OS, the first performance feature you should evaluate is one that probably will have the greatest performance impact: indexing on expressions. You can now put an index over a subset of a single column or derivatives of multiple columns, a formula, or a function used within your application. By creating a customized index on an expression, you can change the access path of an application using a tablespace scan, which is a combination of multiple indexes, to a single customized index-a much more efficient method. Because you define the index expression, you can design it to provide direct or superior access for any SQL statements within your applications. You can create an indexed expression on a table of any size, which means that you can use this feature to improve performance on all of your applications.

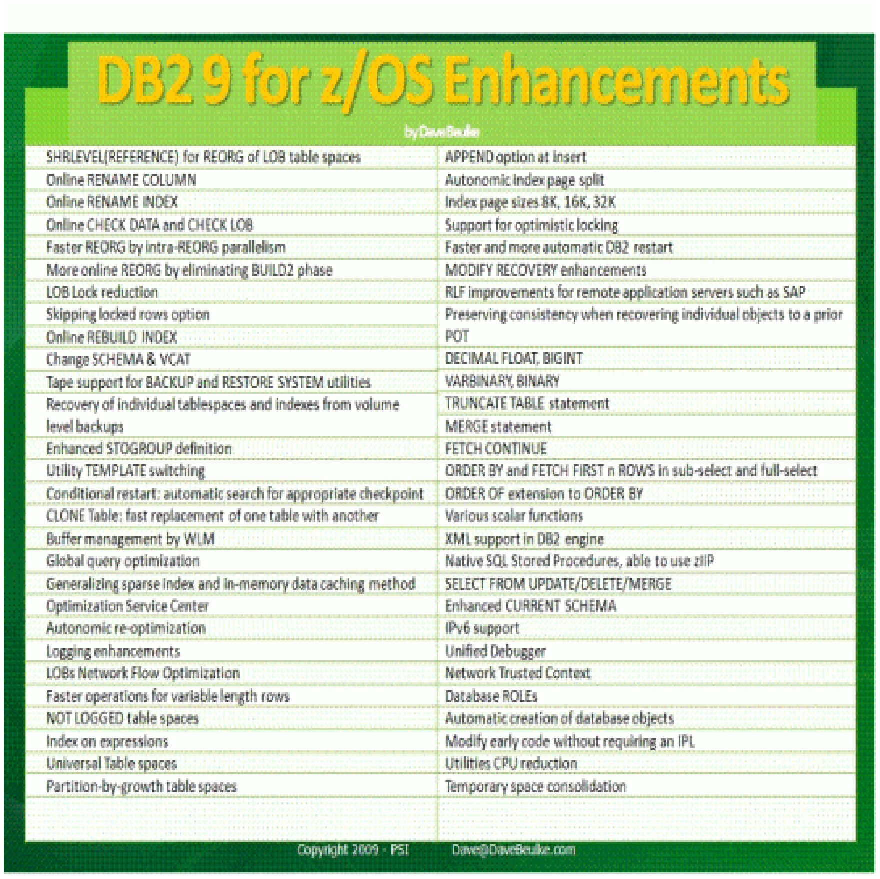
Almost any expression done in an SQL statement can be used to create an expression for an index, and the index expression can use multiple columns or expressions on multiple columns. DATE and other functions used in your application's SQL code are great candidates for indexes on expressions. For example, imagine grabbing the date from the insert timestamp of the row (DATE(IN_TIMESTAMP)). Instead of performing a tablespace scan to get the date portion of a timestamp, an application can access it as an index. This process is not only faster, but it also reduces the load on the CPU.

The second big-impact performance feature within DB2 9 for z/OS is index compression. Index compression offers the same benefits as DB2 data compression by storing more index entries per index page. The increased number of entries per page allows more keys to be available in the system and buffer, reducing I/O requirements and improving the number of index entries available in the buffer pools.

However, index compression works very differently from data compression: the index entries are compressed only at the physical disk page level, and a compression dictionary is not used. This allows index compression to be available without using a REORG or LOAD utility. The index entries are expanded in the buffer pools, allowing quicker memory traversing of the index structures for data retrieval. For example, compressed index pages may be only 4 K on disk but 8 K, 16 K, or 32 K in the buffer pool.

Index and data compression can save disk space, backup time, and recovery time over the life of your system, and can save CPU time by reducing processing I/O. Because compression does have overhead costs, it may not be appropriate for systems with insert/update loads-but it is ideal for large business intelligence or operational systems where data is mostly being read.

These are only two of many interesting features in DB2 9 for z/OS. For an extensive list of other performance features that can help your applications run faster and more efficiently and help save your company money, take a look at Table 1. These new z/OS features and many other DB2-related topics are also covered at the revamped International DB2 Users Group (IDUG) Web site (www.idug.org), along with more than 450 videos at www.idug.org/ db2-videos.html. Check it out!



DB2 9 for z/OS Enhancements

by Dave Beulke

SHRLEVEL(REFERENCE) for REORG of LOB table spaces	APPEND option at insert
Online RENAME COLUMN	Autonomic index page split
Online RENAME INDEX	Index page sizes 8K, 16K, 32K
Online CHECK DATA and CHECK LOB	Support for optimistic locking
Faster REORG by intra-REORG parallelism	Faster and more automatic DB2 restart
More online REORG by eliminating BUILD2 phase	MODIFY RECOVERY enhancements
LOB Lock reduction	RLF improvements for remote application servers such as SAP
Skipping locked rows option	Preserving consistency when recovering individual objects to a prior POT
Online REBUILD INDEX	DECIMAL FLOAT, BIGINT
Change SCHEMA & VCAT	VARBINARY, BINARY
Tape support for BACKUP and RESTORE SYSTEM utilities	TRUNCATE TABLE statement
Recovery of individual tablespaces and indexes from volume level backups	MERGE statement
Enhanced STOGROUP definition	FETCH CONTINUE
Utility TEMPLATE switching	ORDER BY and FETCH FIRST n ROWS in sub-select and full-select
Conditional restart: automatic search for appropriate checkpoint	ORDER OF extension to ORDER BY
CLONE Table: fast replacement of one table with another	Various scalar functions
Buffer management by WLM	XML support in DB2 engine
Global query optimization	Native SQL Stored Procedures, able to use zIIP
Generalizing sparse index and in-memory data caching method	SELECT FROM UPDATE/DELETE/MERGE
Optimization Service Center	Enhanced CURRENT SCHEMA
Autonomic re-optimization	IPv6 support
Logging enhancements	Unified Debugger
LOBs Network Flow Optimization	Network Trusted Context
Faster operations for variable length rows	Database ROLEs
NOT LOGGED table spaces	Automatic creation of database objects
Index on expressions	Modify early code without requiring an IPL
Universal Table spaces	Utilities CPU reduction
Partition-by-growth table spaces	Temporary space consolidation

Copyright 2009 - PSI Dave@DaveBeulke.com

About the author

David Beulke is former president of IDUG and has more than 22 years experience in architecture, design, and development of high performance DB2 systems across all platforms.