

FROM THE BUFFER POOL:

To Partition or Not to Partition, That is the Question.

DB2 for OS/390 offers three different types of tablespaces—simple, segmented, and partitioned. Each one is useful in different situations. Until recently, segmented tablespaces were used most of the time. Except under rare circumstances simple tablespaces are almost obsolete and partitioned tablespaces are only used for large amounts of data. However, you should consider using partitioned tablespaces more frequently.

Tech Tip: Use simple tablespaces only when you need to mix data from different tables on one page. Use simple tablespaces infrequently—programmatically, close monitoring, and diligent administration are required to maintain the data efficiently. Otherwise, use either segmented or partitioned tablespaces.

Partitions Possible Per Tablespace

DB2 permits 1 to 254 partitions per tablespace; from 1 to 64 partitions for non-LARGE tablespaces and those created prior to DB2 V5. For more information on LARGE tablespaces consult the previous installment of PLATINUM "From The Bufferpool" or read it on the Web at www.platinum.com/dbtips/tps_os01.htm.

For LARGE partitioned tablespaces, each partition can be a maximum of 4GB. For non-LARGE partitioned tablespaces, the number of partitions influences the maximum size of the data set partition as follows:

Number of Partitions	Maximum Data Set Size
1 to 16	4 GB
17 to 32	2 GB
33 to 64	1 GB

Deciding to use a partitioned tablespace is not as simple as merely calculating the size of the table. Application-level details, such as data contention, performance requirements, and the volume of updates to columns in the partitioning index must be considered.

In general, partitioned tablespaces are better than alternative methods of spreading data over multiple volumes. Avoid implementing several smaller tablespaces, each containing a subset of the total amount of data. In this scenario separate tables are built, each with the same data characteristics, and then each is placed into separate smaller tablespaces. This is a poor database design because it is too complex and introduces unneeded denormalization and SQL operations to access the data as a logical whole.

Partitioning Pros and Cons

Partitioned tablespaces are the most efficient for large data storage requirements. The largest size a non-partitioned tablespace can become is 64GB. With partitioning, DB2 V5 can accommodate tablespaces of up to 1TB.

DB2 partitioning offers many other benefits, including:

DB2 creates a separate compression dictionary for each tablespace partition.

Partitioned tablespaces can reduce utility processing time and decrease contention because of partition independence.

Volume, free space, and other characteristics can be specified at the partition level.

Using partitioned tablespaces can improve data availability.

When data is partitioned by location, partitions can be made available while others are being reorganized.

Partitioning can also improve recoverability.

Tablespace scans on partitioned tablespaces can skip partitions that are excluded based on the query predicates. This is known as limited partition scanning.

Use of partitioned tablespaces encourages parallelism. Query I/O, CPU, and Sysplex parallelism enable multiple engines to access different partitions in parallel, usually resulting in reduced elapsed time.

Tech Tip: What is query parallelism? Starting with DB2 V3 multiple read tasks can be used to satisfy a single SELECT statement. Multiple, concurrent read engines can reduce overall elapsed time for an individual query. This is called query I/O parallelism and is useful for I/O-bound queries.

DB2 V4 improves parallelism by taking advantage of microprocessor CPU engines for DB2 queries. When query CP parallelism is engaged, each concurrent read engine utilizes its own portion of the CPU. This aids CPU-bound queries.

DB2 V5 improves parallelism even further with query Sysplex parallelism enabling DB2 to spread the work for a single query across multiple DB2 subsystems in a data sharing group. This further aids intensive, CPU-bound queries.

The potential disadvantages of partitioned tablespaces include:

Only one table can be defined within a partitioned tablespace.

The entire length of the key for the partitioning index cannot exceed 40 bytes.

The columns of the partitioning index cannot be updated. To change a value in one of these columns, you must delete the row and then reinsert it with the new values.

The range of key values for which data will be inserted into the table must be known and stable before you create the partitioning index. A range of values must be hard-coded into the partitioning index definition to evenly distribute the data throughout the partitions.

After defining the partitioning method, you cannot change the number of partitions easily. Individual partitions cannot be deleted or redefined. To drop the index that defines the partitioning, you must drop the table to which the index applies.

PLATINUM Partition Expert

PLATINUM Partition Expert simplifies partition maintenance and increases data availability by smoothing out, rolling off, and optionally archiving data stored in partitioned tablespaces without having to drop and recreate tablespaces. It eliminates troublesome hot spots. Partition Expert allows you to avert a full partition situation and the service interruption that comes with it. Partition Expert redistributes data and automates cyclical maintenance without the need for an outage of the entire tablespace. Partition Expert also speeds and simplifies maintenance even further, including the ability to perform tablespace and index reorganizations.

Call PLATINUM technology today for information on Partition Expert, the only product that automates DB2 tablespace partitioning.

Craig Mullins, author of *DB2 Developer's Guide*, is a frequent contributor to industry publications, and is a vice president at PLATINUM technology, inc.

PLATINUM
TECHNOLOGY

www.platinum.com
800-890-7528 x20217

To get insightful tips on how to manage your databases better, visit our DB tips site at www.platinum.com/dbtips