DB2 V3 Buffer Pool Tuning Options

A Trip To Tune Town

By Craig S. Mullins

When DB2 V3 is examined by DB2 shops, the area that seems to generate the most excitement from DB2 DBAs is enhanced buffer pool support. Yet many do not realize that a vast array of tuning options are available with DB2 V3 buffer pools. Most everyone knows that instead of four buffer pools (three 4K pools and one 32K pool), there are now 60 (50 4K pools and 10 32K pools). But these buffer pools can now be explicitly tuned and dynamically sized without recycling DB2!

First, examine the basics. DB2 V3 extends the concept of a buffer pool. The definition of a DB2 V3 buffer pool is the combination of a virtual pool and, optionally, an associated hiperpool. Hiperpools are used to store nonupdated pages of data in memory for longer periods of time.

Additionally, with 60 buffer pools instead of four, it will be much easier to isolate performance-critical tablespaces and indexes into their own buffer pool(s). This provides a huge opportunity for tuning those applications that receive the most attention from management.

Consider a situation where one table in a Decision Support System (DSS) application is queried by the head of your department. When performance deteriorates, the phone calls start. If the particular table can be isolated into its own buffer pool, not only can you ensure that other requests are not monopolizing the pool, but it is much easier to tune the buffer pool for the single table in question. This should become much clearer as you examine the tuning options outlined in this article.

Virtual Pool Tuning

DB2 V3 provides two levels of caching for DB2 data. The first level, the virtual pool, caches data as DB2 always did in prior releases. Virtual pools are byte-addressable and reduce I/O by caching data pages in memory.

For DB2 V2.3 and earlier releases, the only buffer pool tuning available was changing the minimum and maximum sizes of an entire buffer pool. DB2 V3 eliminates minimum and maximum sizes, using only a single size. Most shops specified MIN=MAX for buffer pool size, anyway. However, DB2 V3 also provides many tuning options that can be set using the ALTER BUFFER POOL command. These options are described in the following paragraphs.

Virtual Pool Size (VPSIZE)

VPSIZE indicates the size of an individual virtual pool. The value can range from 0 to 400,000 for 4K buffer pools; from 0 to 50,000 for 32K buffer pools. The minimum size of BP0 is 56 because the DB2 catalog tablespaces and indexes are required to use BP0. The ability to dynamically alter the size of a virtual pool enables DBAs to expand and contract virtual pool sizes without stopping DB2. Altering VPSIZE causes the virtual pool to be dynamically resized. If VPSIZE is altered to 0, DB2 issues a quiesce and when all activity is complete, the virtual pool is deleted.

Virtual Pool Sequential Steal Threshold (VPSEQT)

VPSEQT indicates the sequential steal threshold expressed as a percentage of the VPSIZE. This number is the percentage of the virtual pool that can be monopolized by sequential processing, such as sequential prefetch. The value can range from 0 to 100 and the default is 80.

Virtual Pool Parallel Sequential (VPPSEQT)

VPPSEQT indicates the sequential steal threshold for parallel operations expressed as a percentage of the VPSEQT. The value can range from 0 to 100 and the default is 50. By setting VPPSEQT to 0, you can ensure that parallel I/O will not be available for this virtual pool (this is not a recommendation, just information).

Deferred Write (DWQT)

DWQT indicates the deferred write threshold expressed as a percentage of the VPSIZE. It specifies when deferred writes will begin. When the percentage of unavailable pages exceeds the DWQT value, pages will be written to DASD immediately (not deferred, as normal) until the number of available pages reaches 10 percent of (DWQT*VPSIZE). The value can range from 0 to 100 and the default is 50.

Vertical Deferred Write (VDWQT)

VDWQT indicates vertical deferred write threshold expressed as a percentage of the virtual pool size. This can be thought of as the

deferred write threshold for a single data set. When the percentage of pages containing updated data for a single data set exceeds this threshold, immediate writes will begin to occur. The value can range from 0 to 90 and the default is 10. This value should always be less than DWQT!

The sequential steal threshold (80 percent) and the deferred write thresholds (50 percent and 10 percent) were automatic and unchangeable for DB2 V2.3 and earlier. The concept of VPPSEQT did not exist prior to DB2 V3.

An Introduction To Hiperpools

Hiperpools can be considered extensions to the "regular buffer pools," or virtual pools. Hiperpools provide a second level of data caching. In DB2 V3, when "old" information is targeted to be discarded from the virtual pool, it will first be moved to the hiperpool (if a hiperpool has been defined for that virtual pool).

Only clean pages (read-only pages or those where modified data has already been written back to DASD) will be moved to the hiperpool. No data with "pending" modifications will ever reside in a hiperpool.

Each of the 60 virtual pools can optionally have a hiperpool associated with it. There is a one-to-one relationship between virtual pools and hiperpools. A virtual pool can have one and only one hiperpool associated with it, but it may have none. A hiperpool must have one and only one virtual pool associated with it.

Hiperpools are page addressable, so before data can be accessed by an application, it must be moved from the hiperpool to the virtual pool (which is byte addressable). Hiperpools are backed by expanded storage only, while virtual pools are backed by central storage, expanded storage and possibly DASD if paging occurs. Keeping this information in mind, consider using hiperpools instead of specifying extremely large virtual pools without a hiperpool.

When you specify a virtual pool without a hiperpool, you are letting MVS allocate the virtual pool storage required in both central and expanded memory. If possible, specify a virtual pool that will completely fit in central storage and a hiperpool associated with that virtual pool. The DB2 buffer manager will handle the movement from expanded to central storage and should be more efficient than simply implementing a single large virtual pool. Of course, you will need to monitor the system to ensure that the virtual pool is utilizing central storage in an optimally efficient manner.

To utilize hiperpools, you must be using MVS/ESA 4.3 on an ES/9000-511 or -711 series processor with Asynchronous Data Mover Facility (ADMF).

Hiperpool Tuning

The ALTER BUFFERPOOL command can be used to tune hiperpool options as well. These options are described in the following paragraphs.

CASTOUT

CASTOUT indicates whether hiperpool pages are stealable by MVS. The value can be either YES or NO. Specifying YES allows MVS to discard data in the hiperpool if an expanded storage shortage is encountered. A value of NO prohibits MVS from discarding hiperpool data unless the hiperpool is deleted, MVS hiperspace maintenance occurs, or hiperspace storage is explicitly released.

Hiperpool Size (HPSIZE)

HPSIZE indicates the size of each hiperpool. When the size of a hiperpool is altered, it immediately expands or contracts as specified. The value can range from 0 to 2,097,152 for 4K buffer pools; from 0 to 262,144 for 32K buffer pools. The total of all hiperpools defined cannot exceed 8GB.

Hiperpool Sequential (HPSEQT)

HPSEQT indicates hiperpool sequential steal threshold expressed as a percentage of the HPSIZE. It specifies the percentage of the hiperpool that can be monopolized by sequential processing, such as sequential prefetch. The value of HPSEQT can range from 0 to 100 and the default is 80. If you know the majority of your sequential prefetch requests will never be accessed again, you may want to tune your hiperpools to avoid sequential data. Do this by specifying HPSEQT=0. This ensures that only randomly accessed data will be moved to the hiperpool.

There are no deferred write thresholds for hiperpools because only clean data is stored in the hiperpool, hence pages never need to be written from the hiperpool to DASD.

Monitoring Buffer Pool Status

A DISPLAY BUFFERPOOL command was added for DB2 V3. It can be used to monitor the specifications chosen for each buffer pool. This command provides pertinent information on the sizing, parameters and usage of active and/or inactive buffer pools and hiperpools. It is a good practice to periodically monitor the specifications for active buffer pools to gauge the effectiveness of your buffer pool tuning activities.

Conclusion

With the advanced buffer pool tuning options available to DB2 V3 users, DBAs should have job security well into the 21st century. Just try to explain to your boss that you need to examine the value of VPPSEQT as a percentage of VPSEQT and compare it to the HPSEQT to determine the impact of sequential processing on the buffer pools. Chances are you will be left alone for the rest of the week to play with your buffer pool tuning knobs. Happy tuning! RDJ

ABOUT THE AUTHOR

Craig S. Mullins, a senior education specialist for Platinum Technology, Inc., has 10 years experience in database development. He is the author of DB2 Developer's Guide Second Edition (ISBN 0-672-30512), *Prentice Hall, Carmel, IN. Platinum Technology, Inc., 1815 S. Meyers Rd., Oakbrook Terrace, IL 60181, (800) 442-6861. CompuServe 70410, 237 or Prodigy WHNX44A.*