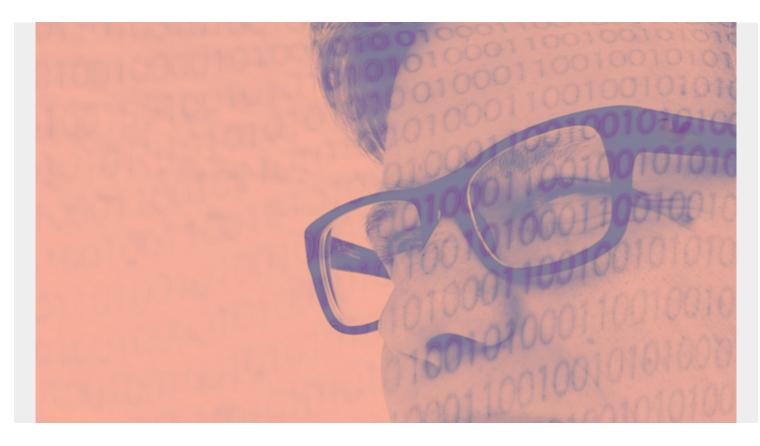# DB2 FOR Z/OS BUFFER POOLS - THE BASICS

Guest post by Lynn Gros, Lead Quality Assurance Representative

The goal of buffer pool management in DB2 for z/OS is to keep application data close at hand and eliminate physical I/Os. This improves overall service levels and reduces CPU cycles and elapsed time. Over the next few weeks, we'll look at several aspects of buffer pool management.

The primary concern is DB2 object placement in the buffer pools to ensure optimum performance based on your priorities, including how many pools you need, what size they will be, and which objects go in which pools. Of course, you need to ensure that buffer pool configurations do not cause an increase in paging and negatively affect overall system performance.

This week, we'll look at some basics.

GETPAGE operations

DB2 objects are physically stored as pages in VSAM data sets. The Buffer Manager services DB2 application data requests with GETPAGE, which requests a specific DB2 page. The DB2 buffer pool is searched and if the page is not found, a synchronous I/O operation occurs to retrieve it. Pages accessed this way are known as random pages.

DB2 pages that retrieved using an asynchronous I/O operation called a *prefetch* are known as sequential pages. Prefetch operations retrieve multiple pages - anticipating an application need for the additional pages. Types of prefetch include sequential, dynamic, and list prefetch.

The retrieved page is placed in the buffer pool associated with the DB2 object being accessed. The

page's life in the buffer pool can be static or dynamic. A page may be accessed a single time and then be flushed from the pool as the buffer is needed for another DB2 process; or that same page may be required again by the same or another process, eliminating the need for an additional I/O operation.

After a page is updated by a DB2 process, it is written back to the underlying VSAM data set asynchronously (normally). Writes are triggered by DB2 checkpoint processing or when certain buffer pool thresholds occur during normal processing. DB2 optimizes the write process to minimize I/O processing.

Buffer pool details

Because DB2 buffer pools are allocated in virtual storage in the DBM1 address space, they are called virtual buffer pools. DB2 virtual buffer pools are defined during DB2 installation (DSNTIP2). Initial specifications are stored in the bootstrap data set (BSDS).

Once a buffer pool is defined, the only way to modify its size is with the DB2 ALTER BUFFERPOOL command. Pools initially defined with VPSIZE (0) can be altered to a positive value and then allocated in the DBM1 address space. You can effectively delete an existing buffer pool by altering the existing size to 0, although any attempts to access a pageset defined to the buffer pool will result in an unavailable buffer pool resource.

DB2 on z/OS allows for 80 buffer pools, including:

BP0 – BP49   50 4K pools

BP8K0 – BP8K9   10 8K pools

BP16K0 – BP16K9   10 16K pools

BP32K – BP32K9   10 32K pools

The following attributes define buffer pool characteristics:

- Virtual Buffer Pool Size (VPSIZE) – Defines the size of the buffer pool in pages.
- Page Stealing Algorithm (PGSTEAL) – Indicates the page stealing algorithm DB2 will use for stealing pages from this pool: LRU (least recently used) or FIFO (first in, first out). When substantial sequential processing leads to unavoidable I/O, set the page steal algorithm to FIFO. Because the pages are unlikely to be re-referenced, there is no need to pay the CPU cost of maintaining a least recently used queue. This can also save some CPU time.
- Page Fix Attribute (PGFIX) – Indicates if pages are fixed in real storage when initially used. Use page fixed buffer pools only if there is sufficient real storage available to support the fixed pools without causing paging. Page fixing is beneficial when there is significant unavoidable I/O. Pages must be fixed to do I/O, so having a fixed pool avoids the need to fix and the un-fix each page repeatedly. This can save as much as 10 percentage points of CPU time. Use page fixing is high update situations where significant amounts of write I/O are occurring.
- Automatic Buffer Pool Adjustment (AUTOSIZE) – Specifies whether this buffer pool uses a DB2 9 feature that allows Workload Manager (WLM) services to automatically change the size of the pool based on I/O delays for random GETPAGES.

Consider modifying these attributes with the ALTER BUFFERPOOL command:

- Sequential Steal Threshold (VPSEQT) – The percentage of pages that can be used for prefetch.

Default is 70%. For pools with mostly sequential access, you can set this value as high as 100. For pools with mostly random access, set this value very low. Review the prefetch failure statistics to determine if applications attempt to use sequential processing against the objects allocated in this pool.

- Deferred Write Threshold (DWQT) – The percentage of the virtual buffer pool pages that can be occupied by unavailable pages (updated or in use) before DB2 begins to schedule asynchronous writes for updated pages. Default is 30%. If the threshold is reached, DB2 will schedule asynchronous write engines to reduce the number of unavailable pages. To lower this threshold, increase the DWQT percentage. Set the threshold lower for buffer pools with predominantly sequential access or lower update activity.
- Vertical Deferred Write Threshold (VDWQT) – Defines virtual buffer pool pages that can be occupied by updated pages from a single page set before DB2 begins to schedule asynchronous writes for that page set. The threshold can be expressed as a percentage of the total pages in the pool, or you can specify an absolute number of pages. Default is 5%.

The following thresholds, which are fixed for all pools, indicate problems that have a severe impact on performance and availability. Things get progressively worse as the percentage of unavailable (in-use or updated) pages increases.

- Sequential Prefetch Threshold (SPTH) – When 90% of the available buffers in the pool are unavailable, prefetch operations are turned off for the pool.
- Data Manager Threshold (DMTH) – When this threshold reaches 95%, GETPAGE requests and releases apply to rows instead of pages. That is, when more than one row is retrieved in a given page, more than one GETPAGE request and release is performed for that page. This has a major CPU impact.
- Immediate Write Threshold (WITH) – This threshold (97.5%) causes synchronous writes of DB2 pages as they are updated by DB2 processes.

The –DISPLAY BUFFERPOOL DB2 command output provides detailed information on buffer pool characteristics as well as incremental performance metrics for each buffer pool. It provides counts on the number of times various thresholds have been reached, and it provides details about the pagesets in the pool.

Next week, we'll look at best practices for configuring buffer pools.