Buffer Tuning
Using DB2 10 and 11 for z/OS

Jeffrey Berger
IBM

Session Code: 1414
Date and Time of Presentation | Platform: DB2 for z/OS

Disclaimer

The information contained in this presentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans strategy, which are subject to change by IBM without notice. IBM shall not be responsible for damages arising out of the use of, or otherwise related to, this presentation or any other documentation. Nothing contain in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of any agreement or license governing the use of IBM products and/or software.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

INTERNATIONAL DB2 USERS GROUP
Independent • Not-for-Profit • User Run
Founded 1988

Agenda

- Buffer pool allocation
- Page size selection
- LRU processing and page classification
- Deferred writes
- Critical thresholds
- Long term page fixing
- Free space management

DB2 Buffer Pools

- 4K Page Size: BP0, BP1, ..., BP39
- 8K Page Size: BP8K0, BP8K1, ..., BP8K9
- 16K Page Size: BP16K0, BP16K1, ..., BP16K9
- 32K Page Size: BP32K, BP32K1, ..., BP32K9
- BP0, BP8K0, BP16K0 and BP32K used for DB2 catalog/directory

  Usually small, but it's good to increase the size (especially BP0) while migrating the DB2 catalog to a new DB2 release

Buffer Pool Tuning

- Multiple buffer pools recommended
  - Display BPOOL for online monitoring
    - Data set statistics via –DISPLAY BPOOL LSTATS (IFCID 199)
  - Useful for access path monitoring
- Dynamic tuning
  - Full exploitation of BP tuning parameters for customized tuning
    - ALTER BPOOL is synchronous and effective immediately, except for BP contractio because of wait for updated pages to be written out
- Catalog/directory is in BP0, BP8K0, BP16K0 and BP32K
- Minimum of 4 user BPs: user index (4K) and user data (4K) and work files (4K and 32K)
- Don't fragment your buffer pool space too much

Buffer pool parameters

- Buffer pool size - VPSIZE
- Virtual Pool Sequential Threshold – VPSEQT (default 80%)
- Horizontal Deferred Write Queue Threshold – DWQT (default 30%)
- Vertical Deferred Write Queue Threshold – VDWQT (default 5%)
- Page steal method – PGSTEAL (LRU, FIFO, or NONE) (default LRU)
- To page fix or not to page fix – PGFIX (YES or NO) (default NO)
- Auto size – AUTOSIZE (YES or NO) (default NO)
- FRAMESIZE (4KB or 1MB or 2GB)…. New with DB2 11
- VPSIZEMIN and VPSIZEMAX…. New with DB2 11

- BPs are created in DB2's DBM1 address space, above the 64-bit bar
  - Created at first data set Open
  - Buffer pool deleted when all referenced data sets are closed
- AUTOSIZE (YES)
  - WLM may direct DB2 to increase or decrease VPSIZE by up to 25% based on system needs
  - If DB2 is restarted, the 25% is relative to the size when DB2 is restarted
    - For example, VPSIZE starts out at 100MB and increases to 125MB, then when you restart DB2, VPSIZE is eligible to increase to 156MB.

Buffer pool allocation with DB2 11

- AUTOSIZE (YES) with DB2 11
  - New parameters VPSIZEMIN and VPSIZEMAX control the minimum and maximum value of VPSIZE
    - VPSIZE must be between VPSIZEMIN and VPSIZEMAX
  - Default values are 25%, which is same as V10 behavior
  - VPSIZEMIN(*) and VPSIZEMAX(*) are equivalent to using the default
  - AUTOSIZE(YES) is still required
    - But, if you specify VPSIZEMIN/MAX when AUTOSIZE is NO, DB2 will remember the values in case you later alter AUTOSIZE to NO

Page size selection

- 4K pages usually optimize the buffer hit ratio
- Special considerations
  - The page needs to be large enough to store the max row size
  - DB2 can store at most 255 rows on a page
- When rows are large, a large page helps minimize DASD space consumption
  - On average, each page wastes a half-row of space
    - E.g. If you average 10 rows per page, you waste 5% of the space
- Index considerations
  - A large page size is necessary for index compression
  - A large page size minimizes index splits
  - A large page size reduces the number of index levels
  - A large page size may increase the frequency of deferred write I/Os
- A large page (8K or 16K) provides better sequential performance
- With DB2 10, a large page size helps enable inline LOBs, which may help improve I/O and CPU performance significantly

LOB table space page size considerations

- A page in a LOB table space contains only one LOB (i.e. one row)
- A small page size always provides the most space efficiency, especially when LOBs are small.
- If a LOB fits in one page, then it only takes one I/O to read the LOB
  - Otherwise it takes a minimum of two I/Os. The second I/O will read up to 128KB.
- DB2 10: Special tuning considerations for inline LOBs
  - See “DB2 for z/OS Best Practices website”

Prefetch and Utility I/O Performance

- Sequential prefetch will read 256KB per I/O for SQL table scans if VPSEQTxVPSIZE>=160MB
- Utilities will *try to* read or write 512KB per I/O if VPSEQTxVPSIZE>=320MB
  - Unless the storage control unit supports zHPF format writes, z/OS often splits the 512KB into 2 I/Os
  - Use IBM storage for zHPF format writes
    - 50% higher throughput with 4K pages

- Ensure that VPSEQT x VPSIZE >= 320 MB
  - 320 MB is sufficient to do 625 parallel utility 512K read/writes

Optimizing list prefetch

Use IBM DS8700 or DS8800 storage wth zHPF, which uses List Prefetch Optimizer

- http://www.redbooks.ibm.com/abstracts/redp4862.html?Open
  - Up to 3 times faster with spinning disks
  - Up to 8 times faster with solid state disks

Page Classification and LRU Processing

- DB2 has a mechanism to prevent sequentially accessed data from monopolizing the BP and pushing out useful random pages
  - Maintains two chains
    - LRU with all pages (random and sequential)
    - SLRU with only the sequential pages
  - Steals from the LRU chain until VPSEQT is reached, and then steals preferentially from the SLRU chain

- Buffers in a BP are classified as either random or sequential
  - Getpages are classified by the DB2 component that does the Getpages
  - A buffer that is allocated for prefetch always goes on the SLRU chain
  - Prefetched pages are always classified as sequential, but then may be reclassified by the Getpage
    - A buffer is never reclassified from random to sequential, but a random Getpage buffer hit will remove the buffer from the SLRU chain

Page Classification and LRU Processing

DISPLAY BUFFERPOOL(bpid) DETAIL

- Statistics that were added in DB2 11, and in PM70981 for DB2 10
- SEQUENTIAL
  - The length of the SLRU chain
  - DB2 statistics ifcid 2 record contains the minimum and maximum of this statistic during the statistics interval
  - Use can use these statistics to track the mixture of random and sequential buffer pool activity
- VPSEQT HITS
  - Number of times that the length of the SLRU changed and became equal to VPSEQT
- RECLASSIFY
  - The number of times that a random Getpage found the page was on the SLRU chain

DB2 Omegamon Performance: buffer hit ratio

$$\text{Buffer Hit Ratio} = \frac{\#\text{Getpages} - \#\text{Synch I/Os} - \#\text{Asynch Pages Read}}{\#\text{Getpages}}$$

Possible causes of a negative hit ratio

- If a page is prefetched that is “skipped”, the BP hit ratio may be negative, because #Asynch Pages may exceed #Getpages
- This is not a problem, and increasing VPSIZE won’t change a thing

If you observe lots of page skipping, then possibly lower VPSEQT to limit the impact of dynamic prefetch on random I/Os

$$\text{Random Buffer Hit Ratio} = \frac{\#\text{Random Getpages} - \#\text{Random Synch I/Os}}{\#\text{Random Getpages}}$$

- The *random* buffer hit ratio is never negative, unaffected by prefetch activity.

Buffer statistics and buffer classification

- DB2 Accounting data does not distinguish between sequential and random, but DB2 Statistics (ifcid 2) does
  - *Sequential* versus *random s*ynchronous reads
  - *Sequential* versus *random* buffer hit ratios
- The *sequential* buffer hit ratio is meaningless
  - If a sequential Getpage is suspended due to prefetch I/O, it is counted as a sequential synchronous I/O, even though it is not really a synchronous I/O.
  - If the Getpages are themselves misclassified, the random buffer hit ratio is also meaningless
- DB2 9 created a mismatch between how the Getpages were classified and whether or not the buffer was on the SLRU chain subject to VPSEQT
  - V11 eliminates that mismatch

Buffer classification in DB2 10

Getpages classified as random

- Random read
- Random no read (e.g. seq. ins.)
- Dynamic prefetch
- List prefetch
- Sequential format writes (no read)

Getpages classified as sequential

- Table scan sequential prefetch
- LOBs

Buffer classification in DB2 11

Getpages classified as random	Getpages classified as sequential
Random read Random no read (e.g seq. ins.)	Table scan sequential prefetch LOBs Dynamic prefetch List prefetch Sequential format writes (no read)

- DB2 11 yields a more accurate measure of the random buffer hit ratio and the number of truly random synchronous read I/Os
- Sequential inserts that do no reads remain as the most significant exception where buffers are incorrectly classified as random

…Page Classification and LRU Processing

- General recommendations
  - Set VPSEQT to 99% for the sort workfile BP, 90% for other workfile usage
  - Set VPSEQT to 0% for data-in-memory BP to avoid the overhead of scheduling the prefetch engines when data is already in BP
    - With DB2 10, alternatively use PGSTEAL=NONE

Lowering VPSEQT with DB2 11

- If you're buffer tuning goal is to avoid synchronous I/Os, as opposed to avoiding prefetch I/O, then….
  - Lower VPSEQT to improve the random Getpage hit ratio
    - Doing so without DB2 11 may yield unpredicatable results
  - However, ensure that VPSEQTxVPSIZE is greater than 320 MB in order to ensure that the utilities use a 512K prefetch quantity and format write quantity. Doing so minimizes the number of utility I/Os.
  - The biggest exposure from lowering VPSEQT is poor list prefetch performance
    - Use IBM DS8000 storage with zHPF to optimize list prefetch I/O
    - Use SSD to further optimize list prefetch I/O

MRU and Utilities

- "Sequential" pages are governed by VPSEQT, whose default is 80%
- DB2 uses MRU for all "format write" Getpages
  - These buffers are eligible to be stolen immediately after they are written
  - Applicable to LOAD, REBUILD INDEX (load phase), REORG (reload phase) and RECOVER
- Prior to DB2 9, DB2 did not use MRU for sequential reads
  - The COPY utility began using MRU in DB2 9
  - DB2 11 adds MRU usage for most of the other utilities:
    - UNLOAD, RUNSTATS, REORG TABLESPACE and INDEX (unload phase), REBUILD INDEX (unload phase), CHECK INDEX and DATA (unload phase)
    - Not covered: LOBs

Sequential detection for data access

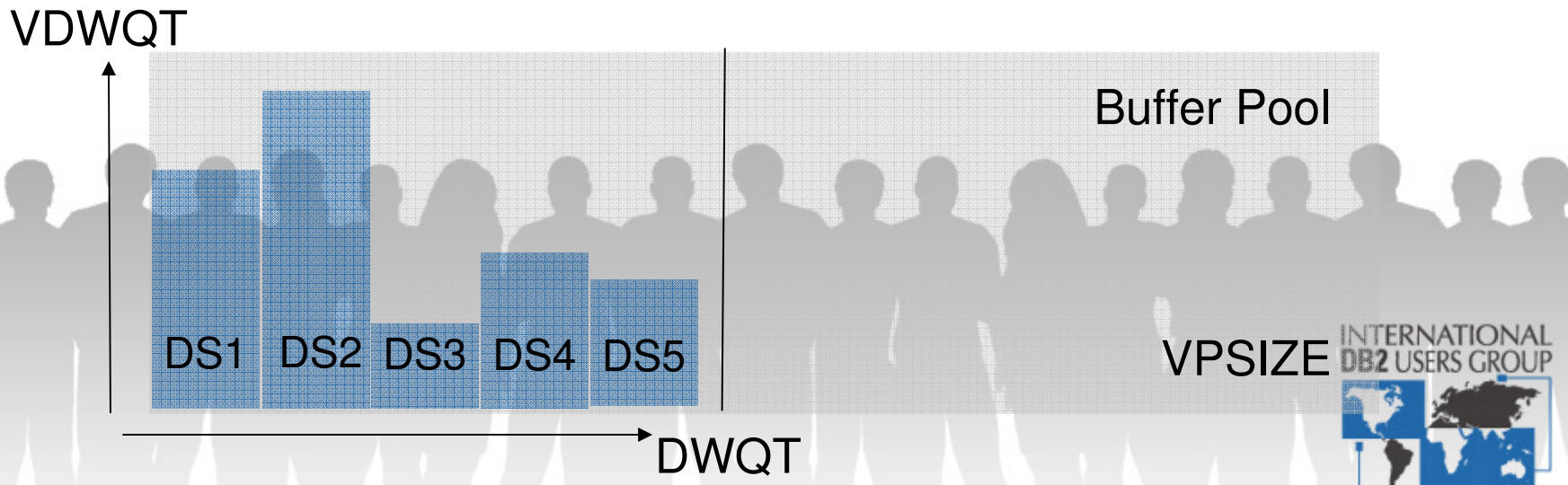
- Prior to DB2 10, dynamic prefetch may be triggered after a minimum of 5 Getpages
- DB2 10: A minimum of 5 Getpages, or 5 rows and 2 Getpages
- DB2 11: A minimum of 5 Getpages, or 5 rows and 3 Getpages

PGSTEAL

- LRU – Least recently used
- FIFO – First in first out (i.e. MRU)
  - Avoids CPU cost of maintaining LRU chain
- DB2 10 introduces PGSTEAL(NONE)
  - "In-memory" buffer pool
  - Similar to setting VPSEQT(0) in that it turns off prefetch
  - Also avoids LRU management cost, like PGSTEAL(FIFO)
    - But does not disable query parallelism.
  - DB2 will sequentially load data sets when they are opened
  - Enables the buffer pool to be loaded more quickly after a DB2 restart, without doing any random synchronous I/Os
  - Consider setting VDWQT and DWQT very high to minimize write I/O
  - If the buffer pool is too small, performance will be unpredictable

Deferred Writes

- VDWQT (Vertical Deferred Write Queue Threshold) based on the data set level as a % of VPSIZE or number of buffers
  - DB2 schedules a write for up to 128 pages, sorts them in sequence, and writes them out in at least 4 I/Os. A page distance of 180 pages is applied to each I/O to avoid high page latch contention, since buffers are latched during I/O.
- DWQT (horizontal Deferred Write Queue) at the BP level



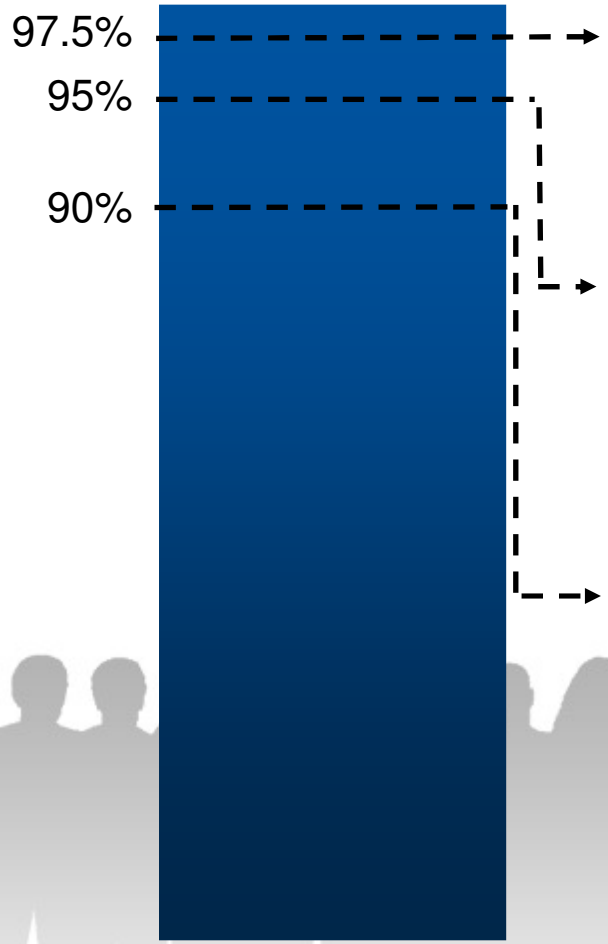
Deferred Writes…

- Setting VDWQT to 0 or 1 is good if the probability of re-writing pages is low
  - DB2 waits for up to 40 changed pages for 4K BP (24 for 8K, 16 for 16K, 12 for 32K) and writes out 32 pages for 4K BP (16 for 8K, 8 for 16K and 4 for 32K)
- Setting VDWQT and DWQT to 90 is good for objects that reside entirely in the buffer pool and are updated frequently
- In other cases, set VDWQT and DWQT low enough to achieve a "trickle" write effect in between successive system checkpoints
  - But… Setting VDWQT and DWQT too low may result in writing the same page out many times, short deferred write I/Os, and increased DBM1 SRB CPU resource consumption
    - If you want to set VDWQT in pages, do not specify anything below 128
- Prefer hitting VDWQT instead of DWQT to increase the number of pages per I/O and reduce the number of I/Os
- Prefer hitting DWQT instead of VDWQT if you want to spread the I/Os over more data disks

Checkpoint interval

- Periodically, the checkpoint interval flushes out all of the buffer pools to DASD, causing a burst of I/Os
- DB2 restart has to read the logs since the start of the prior checkpoint interval and apply those log records
- Reducing the checkpoint interval will help avoid the burstiness of the writes, and reduce the restart time for DB2, but it may cause more write I/O

Critical BP Thresholds



Immediate Write Threshold

Checked at page update

>>After update, synchronous write

Data Management Threshold

Checked at page read or update

>>Getpage can be issued for each row sequentially scanned on the same page – potential large CPU time increase

Sequential Prefetch Threshold

Checked before and during prefetch

90% of buffers not available for steal, or running out of sequential buffers (VPSEQT with 80% default)

>>Disables Prefetch (PREF DISABLED – NOBUFFER)

Causes of poor write performance

- Problems with remote replication
  - E.g. Network contention
  - E.g. A poor control unit implementation can cause reads to queue behind the writes
- Poor local disk performance
  - E.g. RAID 5 rank is overloaded, 10K RPM HDDs
- Poor CF performance
  - E.g. CF links are overloaded
- Low write buffer residency time
  - Caused by VDWQT set too low?
  - Insufficient memory

Long-Term Page Fix for BPs with Frequent I/Os

- DB2 BPs have always been strongly recommended to be backed up 100% by real storage to avoid paging

In a steady-state, PAGE-IN for READ/WRITE < 1% of pages read/written

- Given that there is no paging, might as well page fix each buffer just once to avoid the repetitive CPU cost of page fix and free for each and every I/O
  - Option: ALTER BPOOL(name) PGFIX(YES|NO)
  - Requires the BP to go through reallocation before it becomes operative
    - A DB2 restart is necessary to change PGFIX for BP0, BP8K0, etc.
  - Up to 4% reduction in overall IRWW transaction CPU time

1 MB and 2 GB Page Frames…..

- 1 MB page frames requires z10
  - 1 MB frames with PGFIX(NO) requires Flash Express in zEC12
  - The advantage of 1 MB pages is to improve TLB performance. Best for buffer pools with high Getpage rates.
    - Save 1-4% CPU
  - Specify LFAREA in IEASYSxx
- 2 GB page frames requires zEC12, PGFIX(YES)
- New in DB2 11: Buffer pool FRAMESIZE parameter

- FRAMESIZE(4K | 1MB | 2GB)
  - Example 1 with FRAMESIZE(2GB):

    PREFERRED FRAME SIZE 2G

    1 BUFFERS USING 2G FRAME SIZE ALLOCATED
  - Example 2 with FRAMESIZE(2GB):

    PREFERRED FRAME SIZE 2G

    0 BUFFERS USING 2G FRAME SIZE ALLOCATED

    PREFERRED FRAME SIZE 2G

    256 BUFFERS USING 1M FRAME SIZE ALLOCATED

    PREFERRED FRAME SIZE 2G

    245 BUFFERS USING 4K FRAME SIZE ALLOCATED
- If VPSIZE is less than 2 GB, DB2 will not use 2 GB frames
- If VPSIZE is greater than 2 GB, DB2 will use a mixture of 2 GB and 1MB frames. Some rounding up to a 2 GB frame may occur.

FRAMESIZE(1MB) with Flash Express

- With Flash Express, DB2 can use 1 MB pages with PGFIX(NO)
- IBM continues to recommend PGFIX(YES)
  - The advantage of 1 MB pages is to improve TLB performance. Best for buffer pools with high Getpage rates.
  - The advantage of PGFIX(YES) is to reduce the CPU cost of I/O. Best for buffer pools with high I/O rates.
- IBM continues to recommend that you not oversize your DB2 buffer pools such that it would cause paging

References

- DB2 9 for z/OS: Buffer Pool Monitoring and Tuning, Mike Bracey
  - http://www.redbooks.ibm.com/redpapers/pdfs/redp4604.pdf
- DB2 z/OS Buffer Pool Management by Mike Bracey
  - http://www.gseukdb2.org.uk/techinfo.htm
- *DB2 for OS/390 Capacity Planning*, SG24-2244-00
  - Appendix C documents a technique for calculating the reread ratio using IBM tools.
  - http://www.redbooks.ibm.com/redbooks/pdfs/sg242244.pdf
- *DB2 Version 9.1 for z/OS Performance Monitoring and Tuning Guide,* SC18-9851-05
- *DB2 9 for z/OS Performance Topics*, SG24-7473
- *DB2 Version 9.1 for z/OS Command Reference,* SC18-9844-03
- *Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS V4R1 Report Reference,*
  - SC18-9984-02
- *DB2 10 for z/OS Technical Overview*, SG24-7892
- *DB2 10 for z/OS Performance Topics*, SG24-7942
- DB2 for z/OS Best Practices, webcast http://www.ibm.com/developerworks/data/bestpractices/db2zos/
  - Best Practices for DB2 for z/OS Local and Group Bufferpools, John Campbell
  - Best Practices for DB2 for z/OS Inline LOBs (Large Objects), Jeffrey Berger

Jeffrey Berger
IBM
bergerja@us.ibm.com

Session 1414

Buffer Tuning Using DB2 10 and 11 for z/OS