

CA Datacom® Buffer Pool by Area (BPBA)

Joe Lynn, Datacom/DB at Broadcom



Disclaimer

Certain information in this presentation may outline CA's general product direction. This presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. This presentation is based on current information and resource allocations as of July 2021 and **is subject to change or withdrawal by CA at any time without notice. The development, release and timing of any features or functionality described in this presentation remain at CA's sole discretion.**

Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA may make such release available to new licensees in the form of a regularly scheduled major product release. Such release may be made available to licensees of the product who are active subscribers to CA maintenance and support, on a when and if-available basis. The information in this presentation is not deemed to be incorporated into any contract.

Copyright © 2021 CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY. CA assumes no responsibility for the accuracy or completeness of the information. TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. **In no event will CA be liable for any loss or damage, direct or indirect, in connection with this presentation, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if CA is expressly advised in advance of the possibility of such damages.**

Abstract

- Buffer Pool by Area is a generic term for a new feature to Datacom/DB the Multi-User Facility
- It is a significant feature unto itself but is also the first part of a larger feature of the Multi-User Facility Automatic buffer tuning.
- Addressed in this presentation are data area buffering.

Biography

Joe has worked on the Datacom/DB development team since 1975 in many capacities.

Joseph.Lynn@broadcom.com

Agenda

- What's new? Why would it provide value?
- Define the area BPBA
- Dynamic adjustments
- Reporting and Dynamic System Table
- Messages



What's new? Why would it provide value?

General feature, buffer pool for an area

- Provides a new way to have a private buffer pool for a data area
 - Available for when prioritizing a data area outside of the common DATAPOOL.
 - Different than using the Memory Resident Data Facility COVERED
 - Different than using a common alternate data buffer pool with BUFFER_POOL_DEF
 - Provided for every MUF environment without a MUFPLEX Coupling Facility
- New option BUFFER_POOL_DATA with specific choices
 - Single area buffers not within DATAPOOL, or BUFFER_POOL_DEF memory
 - Not 'redundant' with COVERED

Buffer Pool by Area versus COVERED

- Disadvantages with using COVERED being addressed
 - Covered memory is in addition to active blocks existing in other pools (DATAPOOL or BUFFER_POOL_DEF)
 - Active blocks are in other pools, can be few or many
 - Requires memory copy of blocks when written from other pools
 - Memory blocks restored to other pools when deleted as not active enough, then needed
 - With a large DATAPOOL or BUFFER_POOL_DEF individual blocks may be rarely needed.
 - Are designed to be statically sized, not dynamically tuned
 - Does not support pipelined (deferred DASD writes)

Buffer Pool by Area versus BUFFER_POOL_DEF

- Disadvantages with using BUFFER_POOL_DEF being addressed
 - Data pools are built for multiple areas and not for joining with one area Usage and IO information
 - Buffers can be added and deleted but not based upon active work for a data area
 - Not suited for favoring a single data area and automatically adjusting buffers

Buffer Pool by Area versus DATAPPOOL

- Disadvantages with using DATAPPOOL
 - The data buffers are not sized for exactly this one area.
 - Buffers for this one area cannot be prioritized, all DATAPPOOL is Least Recently Used for all areas in DATAPPOOL
 - Restriction of 99,999 maximum data buffers for DATAPPOOL
 - Buffers can be added and deleted but not easy to actively tune to benefit an area.

Buffer Pool by Area benefits

- Advantages over DATAPOOL, BUFFER_POOL_DEF, and COVERED
 - Buffers are sized for the specific data area.
 - Blocks are in memory once and not redundant like COVERED
 - Large quantity of buffers allowed, similar to COVERED and not other pools
 - Allows an important area to have priority and control of memory and so reducing IO
 - Easy to tune either actively by a DBA or automated.
 - Five strategies for defining area sizing.



Define the area BPBA

Define the area

- `BUFFER_POOL_DATA dbid,area-name,BYTES,N`
 - Definition for one area providing the number of bytes for the memory of the buffers
 - 1 to 9 billion using just number or number with 'K', 'M', or 'G', except using 'G' may be up to 19 billion (19G)
- `BUFFER_POOL_DATA dbid,area-name,BLOCKS,N`
 - Definition for one area providing the number of blocks (sized at open) for the memory of the buffers
 - 1 to 9 billion using just number or number with 'K', 'M', or 'G', except using 'G' may be up to 19 billion (19G)
- `BUFFER_POOL_DATA dbid,area-name,INIT,N%`
 - Blocks initialized at dataset open
 - 1%-200%
- `BUFFER_POOL_DATA dbid,area-name,HIGH,N%`
 - Blocks ever used, highest use
 - 1%-200%
- `BUFFER_POOL_DATA dbid,area-name,ACTIVE,N%`
 - Blocks actively in use containing row's at dataset open
 - 1%-200%

Define the area

- Common at area open for all Buffer Pool by Area definitions
 - Block counts less than 5 will be rounded up to provide minimum 5 blocks. (Not recommended)
 - Large memory size over the MUF maximum (2g default or X_COVERED_MAX_SIZE) rounded down
 - Block counts more than blocks initialized at the time of the dataset open rounded down
- Currently 'gated' to allow maximum 5 data areas
 - Major new facility, learning period
 - Does use 31-bit memory for each buffer as a 'directory' for managing



Dynamic adjustments

Tuning after the area is open

- Tuning commands may be used by any console-like API
- BUFFER_POOL_NEW_SIZE DATA,dbid,area-name,class,new-size
 - Example: BUFFER_POOL_NEW_SIZE DATA,15,DDD,ACTIVE,50%
 - Example: BUFFER_POOL_NEW_SIZE DATA,2,DD1,INIT,110%
 - Example: BUFFER_POOL_NEW_SIZE DATA,100,ABC,BYTES,4G
- BUFFER_POOL_RESIZE 1-5000
 - May be a single DBID or a range of DBID's
 - Available with open areas using percentages, INIT/HIGH/ACTIVE
 - Each area in the DBID(s) buffers compared to the current INIT/HIGH/ACTIVE
 - Buffers are added or removed as needed to reach the set percent
 - Example: DB01421I - BUFFER_POOL_DATA RESIZE 3318 AR3 ACTIVE FROM 24 TO 209 BLOCKS



Reporting and Dynamic System Table

Reporting

- MUF EOJ report (ALL_INFO_REPORT PXXDST)

DBID	AREA	TYPE	REQ	1-BFR	INIT'D	-----POOL-----		
			SIZE-CL	SIZE	BLOCKS	SIZE	BLOCKS	PCT
2	DD1	DATA	200%-AC	4096	42,120	37M	9,270	22
15	DDD	DATA	75%-AC	4096	48,240	8148K	2,037	4
3100	AR1	DATA	50%-IN	4096	24	96K	24	100
3120	AR3	DATA	50%-IN	4096	1,440	2880K	720	50
3310	AR3	DATA	10M-BY	32768	900	10M	320	35
3310	AR4	DATA	10M-BY	32768	900	10M	320	35
3312	AR1	DATA	75%-HU	32128	11,025	13M	403	3
3318	AR1	DATA	1M-BY	28000	180	1M	37	20

Reporting

- MUF EOJ report (ALL_INFO_REPORT PXXDST)

```
---HIGH--- --ACTIVE-- -REMOVED--  
BLOCKS PCT BLOCKS PCT BLOCKS PCT  
4,730 11 4,635 11 0 0  
2,796 5 2,716 5 0 0  
2 8 2 8 0 0  
1,311 91 1,011 70 28,091 91  
315 35 281 31 75,386 71  
485 53 466 51 90,256 75  
538 4 526 4 32,965 42  
2 1 2 1 19,885 76
```

Reporting

- MUF EOJ report (ALL_INFO_REPORT PXXDST)

	USED
USED 1	5+ %
734	97.8
662	95.1
25	99.8
30,540	66.5
105,858	76.5
120,228	82.7
77,095	82.2
26,022	12.6
4,667	67.7

Dynamic System Table

- MUF_BFR_POOL_AREA
 - DBID
 - AREA_NAME
 - TYPE (DATA)
 - REQ_SIZE size requested
 - CLASS (BYTES/BLOCKS/INIT/HIGH/ACTIVE)
 - BUFFER_SIZE
 - BLOCKS_INIT at time of query blocks initialized
 - POOL_SIZE memory pool size (buffer size * blocks in pool)
 - POOL_BLOCKS number of blocks in the pool
 - POOL_PCT percent of pool blocks to blocks initialized
 - HIGH_BLOCKS at time of query highest blocks used
 - HIGH_PCT percent of high blocks to blocks initialized
 - ACTIVE_BLOCKS at time of query actively blocks used
 - ACTIVE_PCT at time of query current active blocks
 - REMOVED_BLOCKS blocks removed not as oldest
 - REMOVED_PCT percent of blocks removed to reads
 - BUFFER_USED_1 (2,3,4,5) blocks used once, twice, etc.
 - GENERATION



Messages

Message: DB01420 opened

- DB01420I - BUFFER_POOL_DATA OPENED 3318 AR2 BLOCKS-5 B_SIZE-28000 CL-ACTIVE
 - Provides the number of blocks allocated for the area at the dataset open
 - Provides the buffer size of each block for the area
 - Provides the class being used, BYTES/BLOCKS/INIT/HIGH/ACTIVE
- DB01421I - BUFFER_POOL_DATA new_size/resize 2 DD1 ACTIVE FROM 2,303 TO 2,282 BLOCKS
 - Provides the adjustment from command NEW_SIZE
 - Provides the class of BYTES/BLOCKS/INIT/HIGH/ACTIVE
 - Provides the number of blocks at start of command and number after command

Questions & Answers





Thank You