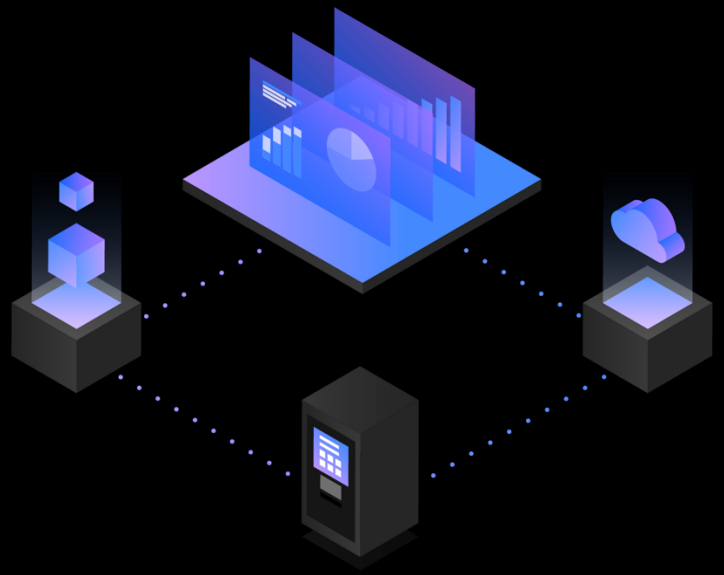
“Show me what you got, Db2!”
(my favorite Db2 for z/OS -DISPLAY commands)

New England Db2 Users Group

January 16, 2025

Robert Catterall, IBM
Principal Db2 for z/OS Technical Specialist



Some up-front information

- To fit the command output lines I wanted to have on a slide, I sometimes removed some lines – when you see a gap between lines, that indicates a place where lines were removed
- My intention is NOT to explain every field in the output of every command (we'd never get beyond -DISPLAY BUFFERPOOL in that case); rather, my aim is to highlight particularly useful fields

Agenda

-DISPLAY BUFFERPOOL

-DISPLAY GROUPBUFFERPOOL

-DISPLAY GROUP

-DISPLAY DDF

-DISPLAY BUFFERPOOL

-DISPLAY BUFFERPOOL(ACTIVE) DETAIL (first part)

```
DSNB401I  -DP01 BUFFERPOOL NAME BP14, BUFFERPOOL ID 14, USE COUNT 102
DSNB402I  -DP01 BUFFER POOL SIZE = 200000 BUFFERS  AUTOSIZE = NO
            VPSIZE MINIMUM  =         0  VPSIZE MAXIMUM  =         0
            ALLOCATED       =    200000  TO BE DELETED   =         0
            IN-USE/UPDATED  =    193686  OVERFLOW ALLOC  =      6400

DSNB406I  -DP01 PGFIX ATTRIBUTE -
            CURRENT = YES
            PENDING = YES
          PAGE STEALING METHOD -
            CURRENT = NONE
            PENDING = NONE
DSNB404I  -DP01 THRESHOLDS -
            VP SEQUENTIAL    =100   SP SEQUENTIAL   =  0
            DEFERRED WRITE   = 50   VERTICAL DEFERRED WRT  = 10,  0
            PARALLEL SEQUENTIAL =50   ASSISTING PARALLEL SEQT=  0
```

Big buffer pools can be a big boost for performance, but don't put too much pressure on the z/OS LPAR's real storage resource – if the LPAR's demand paging rate is < 1 per second, real storage is not over-burdened

PGFIX(YES) is prerequisite for large frames

Default is LRU – NONE means contiguous pool

Default for VPSEQT is 80

- Higher value (90-95) can be good for a pool dedicated to work file table spaces
- VPSEQT(100) can be good for PGSTEAL(NONE) pool in a data sharing environment – can reduce synch reads related to page sets going into, out of GBP dependency
- Lower value might be helpful when synchronous reads are dominant, but don't take it down too far – that could actually increase synchronous read activity

Defaults for DWQT/VDWQT are 30/5 – I generally like to see default values

- Higher values (e.g., 70/40 or 80/50) can be good for a pool dedicated to work file table spaces
- Lower value of VDWQT can be good if number of synchronous writes exceeds number of asynch writes (less relevant in data sharing environment)

-DISPLAY BUFFERPOOL(ACTIVE) DETAIL (second part)

```
DSNB546I  -DP01 PREFERRED FRAME SIZE 1M
        0 BUFFERS USING 1M FRAME SIZE ALLOCATED
DSNB546I  -DP01 PREFERRED FRAME SIZE 1M
        200000 BUFFERS USING 4K FRAME SIZE ALLOCATED
```

Pools with > 1000 GETPAGEs/sec should be backed by large real storage page frames

- 2 GB page frames can boost efficiency, especially for a buffer pool that is at least 20 GB in size

If FRAMESIZE(1M) was specified for this pool, why are none of the buffers allocated in 1 MB frames?

- Reason: LPAR doesn't have enough 1 MB frames
- Large frames made available in LPAR via LFAREA parameter (IEASYSxx member of PARMLIB)
- Goal: have enough 1 MB / 2 GB frames to fully back pools defined with FRAMESIZE (1M / 2G), without going overboard – important that LPAR have enough 4 KB frames for processes that need them
- Guidance: add up size of all pools that will use 1 MB / 2 GB frames, make the LFAREA value for those frames a little larger than that (maybe 5% larger)
- Note: nothing breaks if a pool defined with FRAMESIZE(1M / 2G) not fully backed by large frames – you're just giving up some CPU efficiency

Note: for a contiguous buffer pool, even when the pool has a high GETPAGE rate, use of 1 MB page frames may not be ideal

- If the objects assigned to the pool are pretty small (e.g. a few tens of pages apiece), 1 MB page frames could lead to a good bit of wasted space, because a given frame can hold pages from one and only one object
- Also, note that 2 GB frames cannot be used for a contiguous pool

-DISPLAY BUFFERPOOL(ACTIVE) DETAIL (third part)

```
DSNB409I  -DP01 INCREMENTAL STATISTICS SINCE 07:22:33 APR 24, 2024
DSNB411I  -DP01 RANDOM GETPAGE    =1622858770
            SYNC READ I/O (R) =236303
            SEQ.   GETPAGE    =672093746
            SYNC READ I/O (S) =5322
            SYNC READ I/O (ZHL) =0
            DMTH HIT          =0
            VPSEQT HIT        =0
1DSNB412I  -DP01 SEQUENTIAL PREFETCH -
              REQUESTS        =58
              PREFETCH I/O    =18374
              PAGES READ      =2347018
 DSNB413I  -DP01 LIST PREFETCH -
              REQUESTS        =147
              PREFETCH I/O    =139
              PAGES READ      =1670
 DSNB414I  -D0P1 DYNAMIC PREFETCH -
              REQUESTS        =637487
              PREFETCH I/O    =42284
              PAGES READ      =425151
```

Time when either a) command was last issued or b) pool was last allocated – whichever occurred most recently

- Tip: issue command, then issue again one hour later – divide counters in output of 2nd issuance of command by 3600 to get per-second figures

Should always be 0 – if not, pool is severely undersized

If you made VPSEQT < 80, and threshold is hit a lot of times, maybe VPSEQT is too low

The most important metric for buffer pool monitoring and tuning is the total read I/O rate for a pool

- That's the sum of the 5 values in the green boxes, divided by the number of seconds since the "incremental statistics" timestamp
- Main goal of buffer pool tuning – drive that rate down
- < 1000 per second good, < 100 per second great, < 10 per second super-great
- Overarching goal: make total read I/O rates as low as they can be, without over-burdening LPAR's real storage resource (if LPAR's demand paging rate < 1 per second, real storage not over-burdened)

-DISPLAY BUFFERPOOL(ACTIVE) DETAIL (fourth part)

```
DSNB415I  -DP01 PREFETCH DISABLED -
             NO BUFFER        =0
             NO READ ENGINE   =0
DSNB420I  -DP01 SYS PAGE UPDATES  =720958
            SYS PAGES WRITTEN =45
            ASYNC WRITE I/O   =9
            SYNC WRITE I/O    =36
            PAGE-INS REQUIRED =0
DSNB421I  -DP01 DWT HIT           =0
             VERTICAL DWT HIT  =1
```

If you made VPSEQT < 80, and either of these values is non-zero, maybe VPSEQT is too low

NO READ ENGINE: a Db2 subsystem has 500 prefetch read engines – if there is a prefetch request and all prefetch engines are busy, prefetch is disabled

- That could lead to synchronous read activity
- If VPSEQT too low, pages read into a pool via prefetch can get flushed out of the pool soon after being read in, and that means more prefetch requests will lead to prefetch reads, and a high volume of those could overwhelm the prefetch engines

If sync writes outnumber async writes, may want to reduce value of VDWQT

- Caveat: this rule of thumb does not necessarily apply in a data sharing environment, as the write I/O numbers can be very low in that case (reason: vast majority of page externalization may be driven by commit processing, versus hitting deferred write thresholds)

Some people think that a contiguous buffer pool (a PGSTEAL(NONE) pool) is only suitable for read-only tables

- NOT TRUE
- Read vs. update does not matter – good candidate for a contiguous pool is an object (table space or index) that is very frequently accessed and is not too large

-DISPLAY BUFFERPOOL(ACTIVE) DETAIL (fifth part)

```
DSNB416I  -DP01 OVERFLOW RANDOM GETPAGE    =35749610
            OVERFLOW SYNC READ I/O (R) =229316
            OVERFLOW SEQ.   GETPAGE    =18789826
            OVERFLOW SYNC READ I/O (S) =3457
```

For a contiguous buffer pool, you want to see ZERO activity in the pool's overflow area

- To allow for the possibility of buffer stealing, a contiguous pool will have an overflow area
- Size of that area: 10% of pool, but not more than 6400 buffers, not less than 50 buffers
- Any buffer stealing required will happen in the pool's overflow area – this in order to preserve the "contiguous-ness" of the main part of the pool
- Why you want to see zero activity in overflow area: buffers in this area are managed using the FIFO buffer-stealing algorithm – you lose the CPU efficiency benefit of the "direct reads" that happen in the contiguous part of the pool
- How you get to zero activity in the overflow area: ensure that all pages of all objects assigned to the pool will fit in the contiguous part of the pool

-DISPLAY GROUPBUFFERPOOL

-DISPLAY GROUPBUFFERPOOL(*) TYPE(GCONN) GDETAIL(INTERVAL) (first part)

```
DSNB750I  -DPG1 DISPLAY FOR GROUP BUFFER POOL GBP18 FOLLOWS
DSNB755I  -DPG1 DB2 GROUP BUFFER POOL STATUS

             CURRENT DIRECTORY TO DATA RATIO          = 261

             MAX SIZE INDICATED IN POLICY             = 25165824 KB
             DUPLEX INDICATOR IN POLICY               = ENABLED
             CURRENT DUPLEXING MODE                   = DUPLEX

DSNB758I  -DBP1     ALLOCATED SIZE                        =
25165824 KB
```

Default ratio of directory entries to data entries is 10:1

- ALLOWAUTOALT(YES): system can automatically adjust this ratio
- When the ratio gets really high (> 200:1), that's an indication that the GBP is undersized

You ALWAYS want your GBPs to be duplexed

Good for a GBP (or the lock structure) to have a maximum size that is significantly greater (e.g. 50% larger) than its allocated size

- When that is true, the structure can be dynamically enlarged with the z/OS command SET XCF
- If GBP is already at its maximum size, making it larger will require re-define and rebuild
- In sizing structures when Parallel Sysplex has two coupling facility (CF) LPARs, ensure that all primary structures can fit in one of the CF LPARs (in case other one fails or is brought down for maintenance purposes)

-DISPLAY GROUPBUFFERPOOL(*) TYPE(GCONN) GDETAIL(INTERVAL) (second part)

```
DSNB782I  -DPG1 INCREMENTAL GROUP DETAIL STATISTICS SINCE 07:23:14
APR 24, 2024
DSNB784I  -DPG1 GROUP DETAIL STATISTICS
             READS
               DATA RETURNED                          = 894408
DSNB785I  -DPG1     DATA NOT RETURNED
                 DIRECTORY ENTRY EXISTED              = 17385970
                 DIRECTORY ENTRY CREATED              = 38179935
                 DIRECTORY ENTRY NOT CREATED          = 1115818, 0
DSNB786I  -DPG1   WRITES
               CHANGED PAGES                          = 62295064
               CLEAN PAGES                            = 0
               FAILED DUE TO LACK OF STORAGE          = 0
             CHANGED PAGES SNAPSHOT VALUE             = 4871
DSNB787I  -DPG1   RECLAIMS
               FOR DIRECTORY ENTRIES                  = 0
```

This shows time when command was last issued, or when GBP was allocated or reallocated – whichever happened most recently

- Can issue command, then issue it again an hour later, and divide divide counters in output of 2nd issuance of command by 3600 to get per-second values

You always want this value to be zero – generally speaking, when it's > 0 that means the GBP substantially undersized

You want this value to be zero

- Reason: if GBP directory entry reclaimed, system loses its "pointer" to a locally-cached page
- When that happens, buffer in which page is cached locally is marked invalid, and next request for page is going to require a check of GBP and (probably) a disk read I/O – adds overhead

-DISPLAY GROUPBUFFERPOOL(*) TYPE(MCONN) MDETAIL(INTERVAL) (snippet)

```
DSNB771I  -DG1P INCREMENTAL MEMBER DETAIL STATISTICS SINCE 14:08:27
APR  1, 2024
DSNB773I  -DG1P MEMBER DETAIL STATISTICS
             SYNCHRONOUS READS
               DUE TO BUFFER INVALIDATION
                 DATA RETURNED          (A) = 35046
                 DATA NOT RETURNED      (B) = 1457
DSNB774I  -DG1P     DUE TO DATA PAGE NOT IN BUFFER POOL
                 DATA RETURNED              = 52395
                 DATA NOT RETURNED          = 1305422
```

This option provides GBP activity information at member level, versus group level

With these numbers, you can calculate the XI read hit ratio for a GBP (XI is short for cross-invalidation)

- First, see how many sync reads due to XI there were per second for the GBP: (A + B) / (seconds in interval)
- If that is < 20/second, I don't care what XI read hit ratio is for the GBP
- If > 20 sync reads due to XI per second, I care, and XI read hit ratio is A / (A + B)
- If low (especially if < 50%), make GBP bigger if CF LPAR memory is sufficient
- Larger GBP means greater page residency time, and that tends to drive XI read hit ratio higher – I often see a ratio well above 90% for a generously-sized GBP
- Reason this matters: we can get page from GBP way faster than from disk subsystem

The GBP hit ratio for reads due to "page not found in local pool" is typically very low

- That's not something to worry about – ratio is low because there's no reason to expect page in question to be in the GBP

-DISPLAY GROUP

-DISPLAY GROUP (snippet)

Relevant to a Db2 data sharing group and to a standalone Db2 subsystem

Can't activate Db2 function level n unless catalog level is at least n
- Exception: if function level n has no catalog dependencies, catalog must be at least at level of last previous function level that did have catalog dependencies

Could be higher than current level, if system was taken to higher level and then taken back to lower level

Means that with code and catalog levels being what they are, you could activate this function level

```
*** BEGIN DISPLAY OF GROUP(DSNMSG  ) CATALOG LEVEL(V13R1M505)
                  CURRENT FUNCTION LEVEL(V13R1M505)
                  HIGHEST ACTIVATED FUNCTION LEVEL(V13R1M505)
                  HIGHEST POSSIBLE FUNCTION LEVEL(V13R1M505)
                  PROTOCOL LEVEL(2)
                  GROUP ATTACH NAME(DSNM)
-------------------------------------------------------------------
DB2             SUB                          DB2    SYSTEM    IRLM
MEMBER    ID    SYS  CMDPREF   STATUS        LVL    NAME      SUBSYS  IRLMPROC
--------  ---   ---- --------  --------      ------ --------  ----    --------
DB1M        1   DB1M DB1M      ACTIVE        131505 MVS1      IR1M    DB1MIRLM
DB2M        2   DB2M DB2M      ACTIVE        131505 MVS2      IR2M    DB2MIRLM
```

This is the code level – goes up when PTF that delivers functionality of a new level is applied to Db2 load library
- Functionality not available on system until function level is activated

-DISPLAY DDF

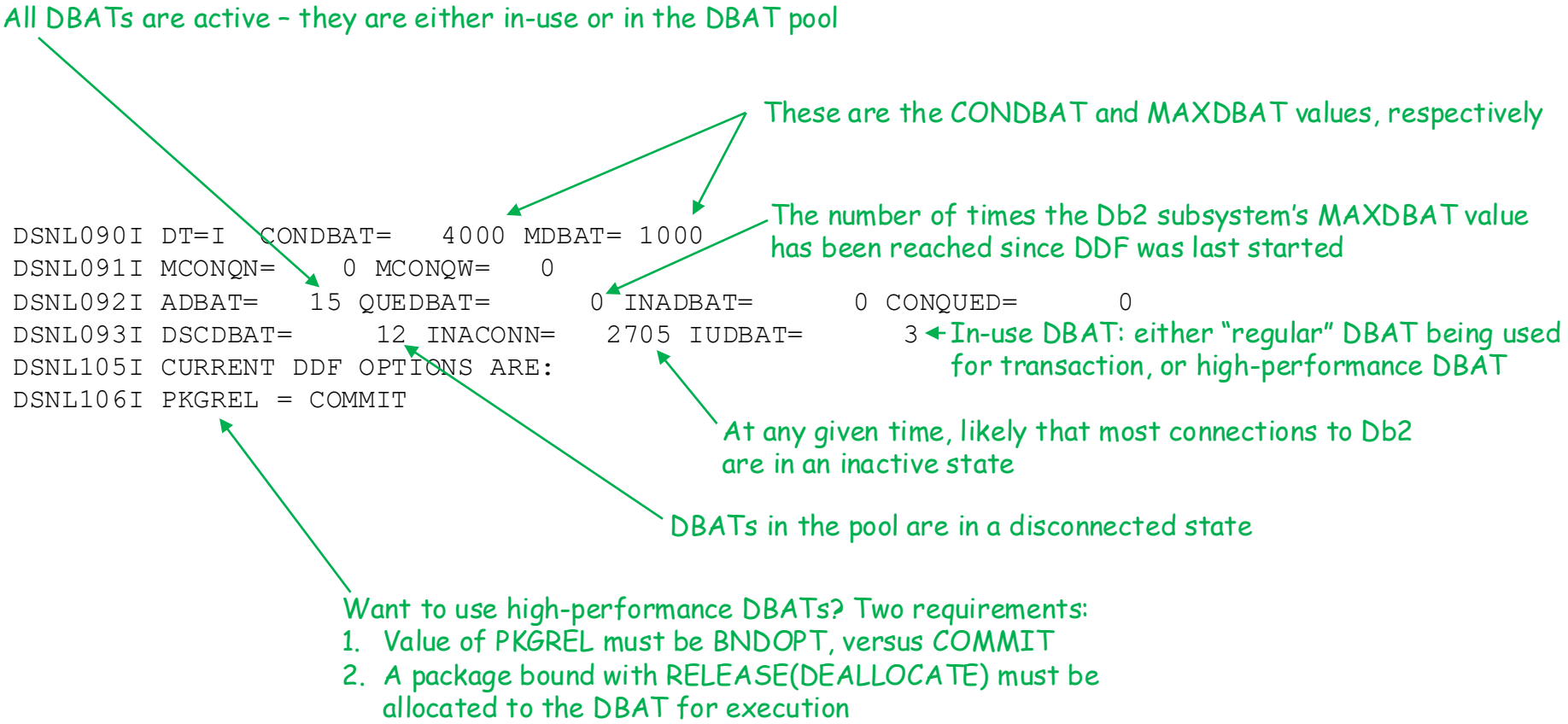
-DISPLAY DDF DETAIL (first part)

```
DSNL080I  -DP01 DSNLTDDF DISPLAY DDF REPORT FOLLOWS:
DSNL081I STATUS=STARTD
DSNL082I LOCATION           LUNAME            GENERICLU
DSNL083I DP01               -NONE             -NONE
DSNL084I TCPPORT=4309  SECPORT=4308  RESPORT=2104  IPNAME=DP01
DSNL085I IPADDR=::1.2.3.4
DSNL086I SQL    DOMAIN=SYSP.BIGCO.COM
DSNL086I RESYNC DOMAIN=SYSP.BIGCO.COM
DSNL087I ALIAS              PORT  SECPORT STATUS
DSNL088I NLDP01             0     0       STATIC
DSNL089I MEMBER IPADDR=::1.2.3.4
```

If client application requests connection to Db2 system via secure SQL listener port, request will be rejected if client does not support AT/TLS encryption (aka SSL encryption)

If DDF transaction left in in-doubt status following a Db2-side or client-side failure, client app will communicate to the Db2 subsystem, via Db2 subsystem's resync port, whether that transaction is to be committed or aborted

-DISPLAY DDF DETAIL (second part)



Robert Catterall

rfcatter@us.ibm.com