Db2 13 scalability

Maximum open data sets
DSSIZE 256 GB for SPT01 and SYSLGRNX
Virtual and real storage
DBAT termination

Open data sets limitation in Db2

Motivation

- Conversion from segmented table spaces (deprecated) to UTS (1 table space per data set) or Db2 consolidation rapidly increases number of Db2 data sets and drives a lot of open data sets
- DSMAX limit 200k (practically less due to storage constraints)
- Aggravated by image copy activity, Db2 open data sets can spike
- Could cause Db2 system performance degradation due to increased open/close data set activities

Goal

- Reduce the DFSMS and z/OS below-the-bar memory footprint for open data sets
- Open/close data sets more efficiently

Support more open data sets in Db2

FL 100

– **Db2 12**

- PH09189: Reduce risk of hitting DSMAX & prevent application failure when hitting DSMAX
- PH27493/PH33238: Proactively close data sets that were opened for utility processing & prioritize closing utility-only data sets when DSMAX is hit

– **z/OS 2.5 and Db2 13** (not planned for retrofit to 2.4)

- Dynamic allocation above the bar reduces below the bar footprint of open data sets
  - Allowing for roughly 50% more open data sets with same memory
  - Activated in ALLOC*xx* parmlib member: SWBSTORAGE=ATB
- CPU cost and elapsed time improvements for open & close
  - More-dynamic Db2 open/close processing as you approach the DSMAX limit

– **Db2 13** doubles the DSMAX limit from 200k to 400k

Challenge for Db2 directory tablespaces

64 GB space limit of Db2 directory tablespaces

- DSNDB01.SPT01
- DSNDB01.SYSLGRNX

Reasons for growth include:

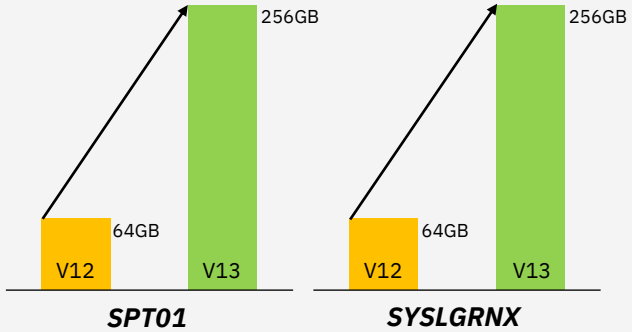
- Significantly increased number of Db2 objects
- Conversion of more non-UTS objects to UTS
- Retained package information due to rebind phase-in for packages (V12R1M505) and new plan management options

Affects production, test, and development environments

- Some customers devote much effort to managing the 64 GB limit

Increased DSSIZE for Db2 directory tablespaces

FL 500

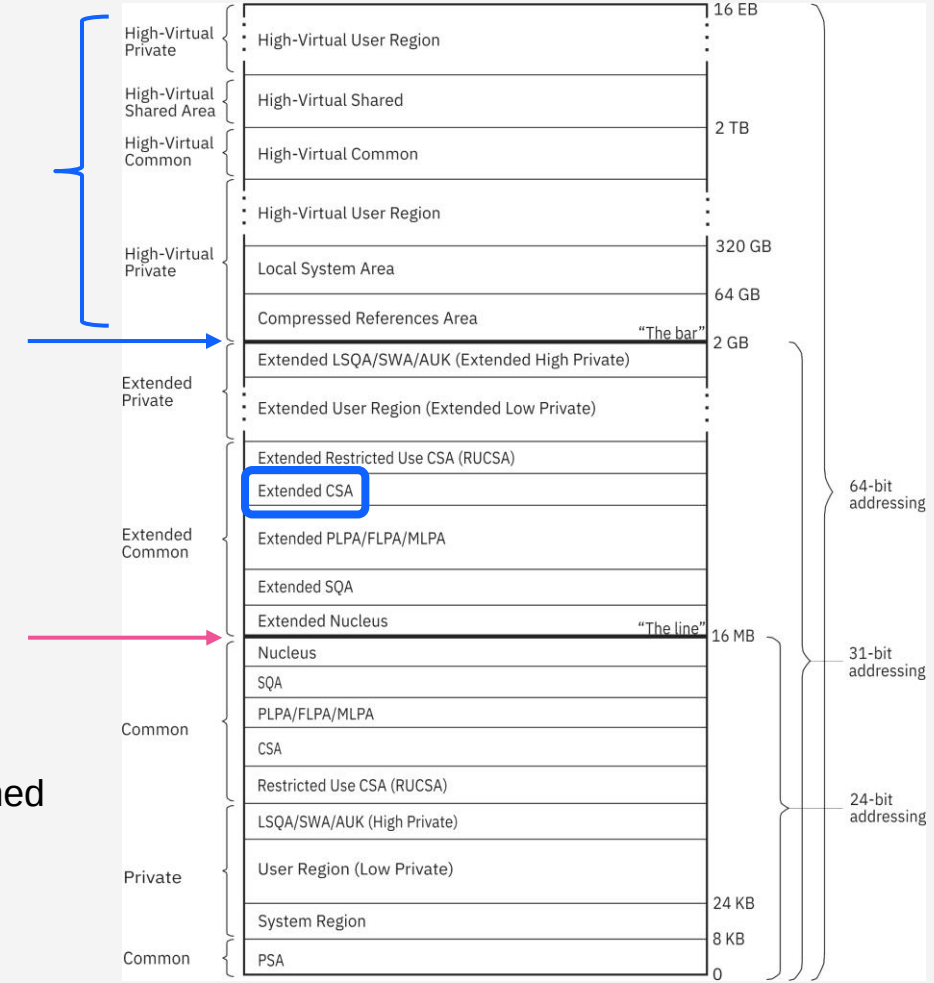


Db2 13 FL 500 or higher

- First REORG with SHRLEVEL CHANGE or SHRLEVEL REFERENCE on table space SPT01 or SYSLGRNX converts DSSIZE to 256 GB
  - No special keyword required.
  - In tables SYSIBM.SYSTABLESPACE and SYSIBM.SYSTABLEPART column DSSIZE updated to 256GB
    - SYSCOPY record will be inserted for the table space to indicate DSSIZE change in the REORG ICTYPE = 'A', STYPE = 'D', TTYPE = ‘64G'.
- Recovery to a point in time (PIT) before the REORG is supported and will revert the size back to 64GB
- If function level is reverted to FL100*, and the tablespace already converted, DSSIZE remains 256GB
  - If tablespace not already converted to 256GB, DSSIZE remains 64GB until the first REORG in FL500

Virtual storage review
(diagram not to scale)

- "high" addressability is "above the bar" (ATB)
- 31-bit addressing limit is 2 GB "bar"
  - Anything lower is "below-the-bar" (BTB)
    - Some sites have severe constraints
      - » Especially ECSA
- 24-bit addressing limit is 16 MB "line"
  - Anything lower is "below the line"
  - CSA (below the line) and ECSA (BTB) defined in PARMLIB(IEASYS*xx*)



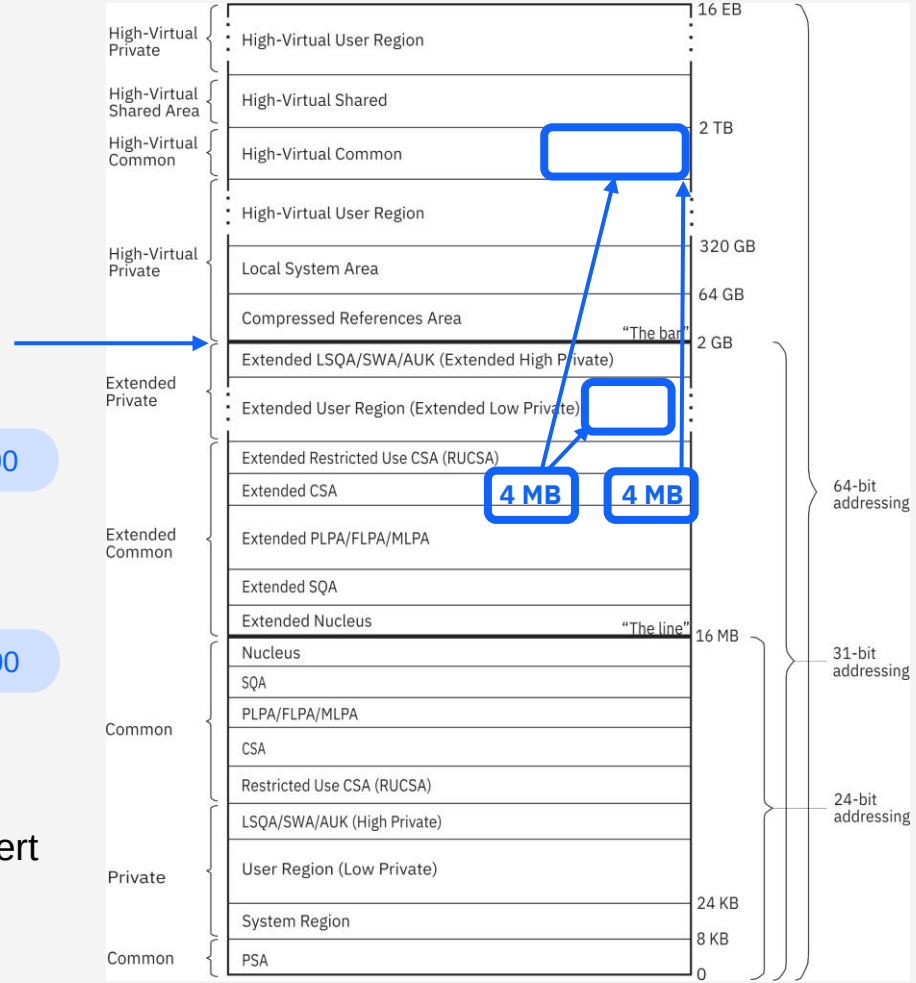
ECSA reduction (1|3)
(diagram not to scale)

Instrumentation facility component (IFC)

– Current: 2 pools in ECSA, each ***4-25*** MB

– Db2 13: most storage moved out of ECSA

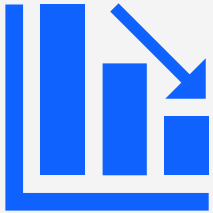
- Step 1: ECSA maximum 25 MB FL 100
- One pool moves to BTB private (**MSTR**) (≤25 MB)
- Step 2: Fixed 8 MB ECSA required to start instrumentation facility interface FL 500
- One pool moves to HVCOMMON (≤ 50 MB)
- New ECSA maximum of 8 MB remains if revert to V13R1M100*



ECSA reduction (2|3)

FL 500

Instrumentation facility component (IFC)



- Result: less ECSA consumption by instrumentation processes, for example:
  - -STA TRACE, -STO TRACE, -DIS TRACE, -MODIFY TRACE
  - Writing Db2 statistics and accounting trace records
  - Monitoring log records with IFCID 306
  - IFI READS requests

ECSA reduction (3|3)

FL 100

Distributed data facility (DDF) ECSA reduction:

- Agent storage improvement: direct effect
- [DBAT termination change: indirect effect]

Agent storage improvement:

- Previously: additional 2 KB for each client using distributed
- Db2 13: 4 KB for each DBAT

Calculate ECSA requirements

- DDF 2.5 MB
- 1 KB for each site in network
  - Servers in communications database (CDB) when this Db2 is a DRDA requester
- 4 KB for each thread (now same as local)
- If SNA, add 1 KB per connection
- See Calculating ECSA in Db2 13 documentation

Reduce agent local below-the-bar (BTB) storage (1|3)

FL 100

Current behavior: dynamic SQL and below the bar (BTB) storage use in DBM1

- PREPARE and EXECUTE IMMEDIATE
  - SQL input statement text and attribute string in agent local BTB storage (BTB)
    - Db2 allocates ***actual*** length of SQL input variable in agent local BTB storage
  - If SQL invokes stored procedure, trigger or user defined function
    - Db2 allocates ***defined*** length of input variable, which could be up to 2 MB
      - » For each nest level

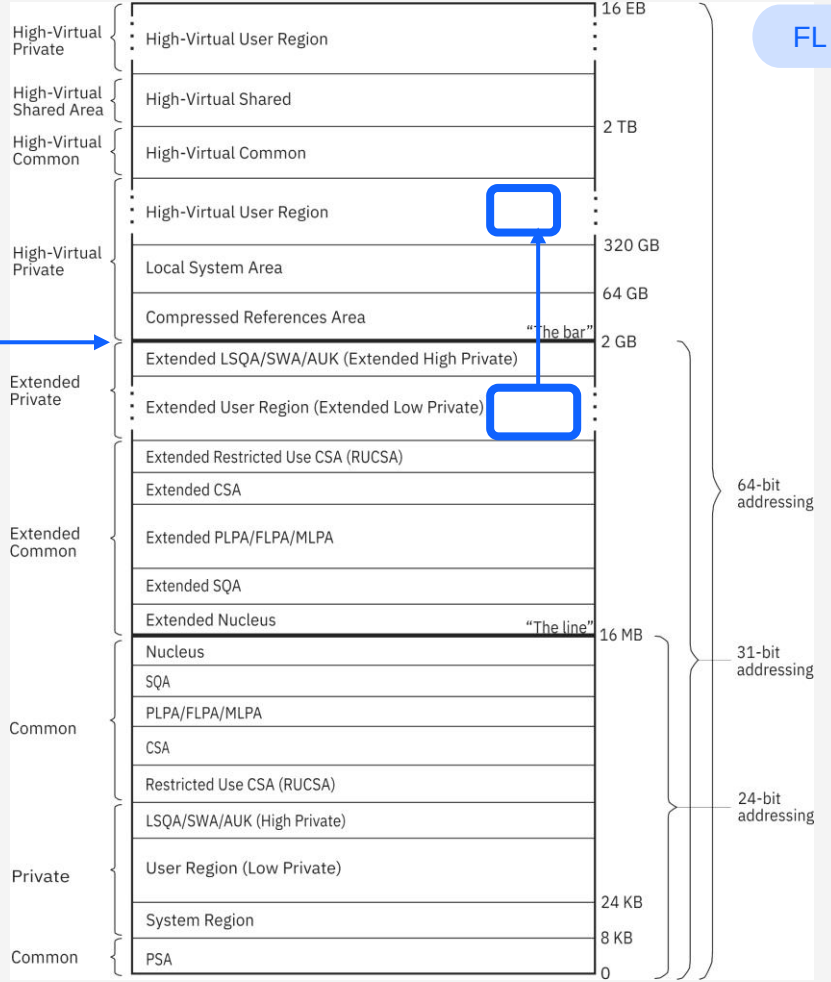
Db2 13 behavior:

- PREPARE and EXECUTE IMMEDIATE
  - SQL statement text and attribute string in agent local above-the-bar (ATB) storage (DBM1)
  - If SQL invokes stored procedure, trigger or user defined function
    - Db2 allocates ***actual*** length of input variable
      - » For each nest level

Reduce agent local below-the-bar (BTB) storage (2|3)
(diagram not to scale)

DBM1 address space

- Db2 12: Agent local BTB for dynamic SQL
  - PREPARE and EXECUTE IMMEDIATE
- Db2 13: Agent local ATB
  - PREPARE and EXECUTE IMMEDIATE
    - Agent local ATB storage consumption possibly less than original agent local BTB

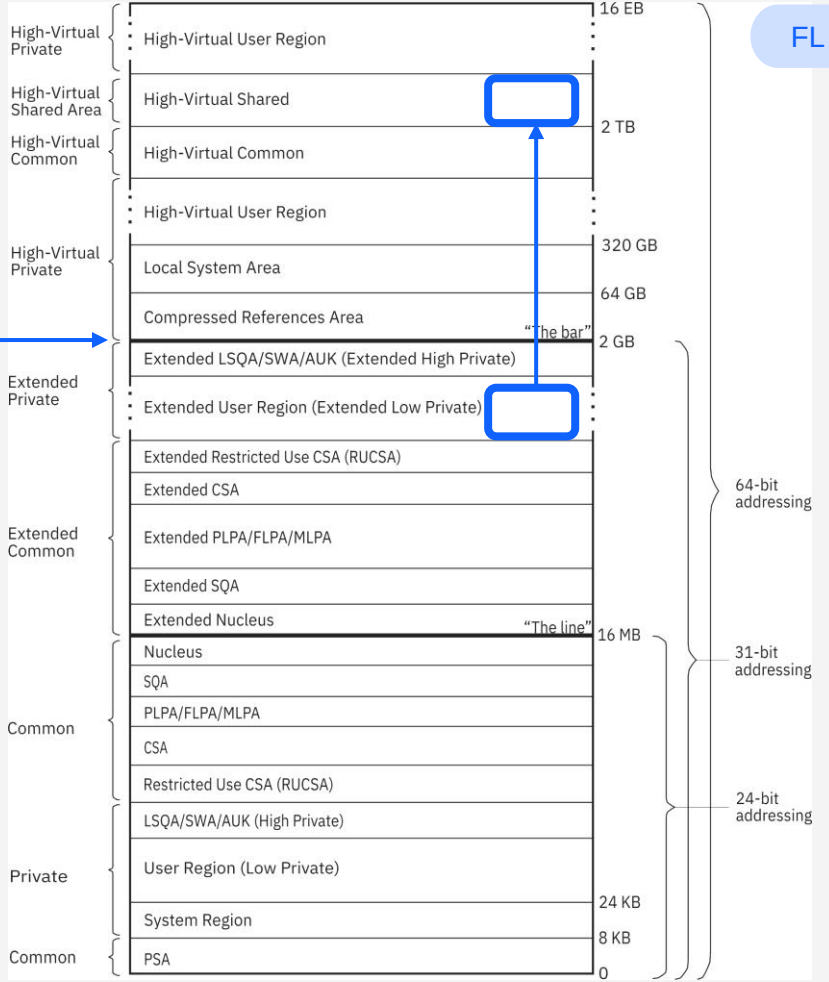


Reduce agent local below-the-bar (BTB) storage (3|3)

(diagram not to scale)

DIST address space

- Db2 12: some control blocks in agent local BTB
  - SQL execution for distributed threads
  - BTB consumption plus cross-memory calls
- Db2 13: these control blocks move to shared ATB
  - Reduce BTB consumption
  - Avoid cross-memory calls



Storage manager contraction: BTB and ECSA

FL 100

Db2 12 behavior

- Subsystem parameter CONTSTOR deprecated
  - Threads no longer contracted
  - BTB thread pools and ECSA can fragment
    - Some sites need to recycle Db2 to resolve fragmented storage pools
      - » Outage for non-data sharing sites

Db2 13 behavior

- System task monitors BTB and ECSA consumption for thresholds
  - If BTB consumption > 64%, automatically begins contraction
  - If ECSA consumption > 85%, automatically begins contraction
  - Contraction ends when consumption drops below threshold
  - Messages:
    - DSNV516I … BEGINNING … CONTRACTION
    - DSNV517I … ENDING … CONTRACTION

Storage manager contraction: memory object contraction

FL 100

Db2 12 behavior

- REALSTORAGE_MANAGEMENT subsystem parameter determines contraction of ATB memory object
  - Db2 issues z/OS service IARV64 REQUEST(DISCARDDATA)
    - At thread deallocation
    - At certain commit intervals
  - z/OS contention possible with large number of concurrent contractions if there is either
    - SQA/ESQA constraint
    - Real storage shortage or system paging

Db2 13 behavior

- REALSTORAGE_MANAGEMENT removed
- IARV64 REQUEST(DISCARDDATA) no longer issued at thread deallocation or certain commit intervals
  - Storage returned to memory object
  - System timer drives contraction of memory object
  - Memory object contraction triggered before paging occurs
    - Db2 13 checks available free frames

DBAT termination process changes (1|5)

FL 100

Current behavior

- DDF workload spikes drive high demand for connection and thread resources
- Brief spikes can correspond to spikes of concurrent termination activity
  - Includes releasing real storage back to z/OS
    - Real storage manager processing
    - Potential spin lock bottleneck
  - Db2 12 PH36114: Db2 checks for ‘not used’ DBATs more frequently and limits number of concurrent terminations for ‘not used’ DBATs

Db2 13 behavior

- Improve termination process
  - Reduce frequency and number of DBAT terminations
  - Reduce number of concurrent terminations
- Reduce thrashing, smooth out spikes
- Entire section assumes subsystem parameter CMTSTAT (commit status) is INACTIVE

DBAT termination process changes (2|5)

Types of DBATs

- Normal, disconnected pooled DBATs
  - At commit: DBAT pooled, connection marked inactive
    - DBAT can be used for other connections, but some cost for disconnect/reconnect process
- KeepDynamicRefresh (KDR) DBATs
  - Distributed applications with packages bound KEEPDYNAMIC(YES)
    - Client specifies KDR processing with either Sysplex workload balancing or seamless failover
    - DBAT not pooled: benefit of reuse of dynamic SQL statement(s) already prepared
- High performance (HIPERF) DBATs
  - If any package touched during DBAT execution was bound with RELEASE(DEALLOCATE)
    - And -MODIFY DDF PKGREL = BNDOPT
  - Then DBAT not pooled: remains active, avoiding disconnect/reconnect process

DBAT termination process changes (3|5)

FL 100

Reduce frequency and number of terminations

Db2 12 DBAT termination behavior

- Normal, disconnected pooled DBATs
  - After DBAT reused 200 times
  - If not used in POOLINAC seconds, if > 0
- KDR DBATs, at clean transaction boundary
  - If KDR DBAT has existed > 1 hour
  - If KDR DBAT has not been used for 20 minutes
- HIPERF DBATs
  - After DBAT reused 200 times
  - If no new request in POOLINAC seconds [0 does not disable; means 120]

Db2 13 DBAT termination behavior

- Normal, disconnected pooled DBATs
  - After DBAT reused 500 times
  - If Db2 detects excessive BTB or ECSA usage
- HIPERF DBATs
  - After DBAT reused 500 times
  - If Db2 detects excessive BTB or ECSA usage

DBAT termination process changes (4|5)

FL 100

Reduce number of concurrent terminations

Db2 12 DBAT termination behavior

- Normal, disconnected pooled DBATs
  - After DBAT reused 200 times
  - If not used in POOLINAC seconds, if > 0
- KDR DBATs, at clean transaction boundary
  - If KDR DBAT has existed > 1 hour
  - If KDR DBAT has not been used for 20 minutes
- HIPERF DBATs
  - After DBAT reused 200 times
  - If no new request in POOLINAC seconds [0 does not disable; means 120]

Db2 13 DBAT termination behavior

- Normal, disconnected pooled DBATs (not used)
  - If POOLINAC = 0, not terminated for inactivity
  - If POOLINAC > 0, after POOLINAC + random time
- KDR DBATs, at clean transaction boundary
  - If KDR DBAT has existed > 1hour + random time
  - If KDR DBAT has not been used for 20 minutes + random time
- HIPERF DBATs (no new requests)
  - If POOLINAC > 0, after POOLINAC + random time
  - If POOLINAC = 0, after 120 seconds + random time

DBAT termination process changes (5|5)

FL 100

Statistics record changes

- Global DDF Activity: new and changed counters
  - Number of times DBAT created
  - Current number of DBATs active due to KEEPDYNAMIC YES
  - Maximum number of DBATs active due to KEEPDYNAMIC YES
  - Number of times DBAT terminated since DDF started
  - Current KDR DBATs
  - Maximum KDR DBATs
  - Number of DBAT terminations
- Included in IFCID 001 (part of Statistics Class 1)