#IDUGDb2

Insight Into Db2 for z/OS Performance Key Metrics and How To Use Them

John Campbell

IBM Db2 for z/OS Development

Session code: Z08

Friday 13th September, 2019 11:40-12:40

Platform: Db2 for z/OS

IDUG

Leading the Db2 User Community since 1988

Please note

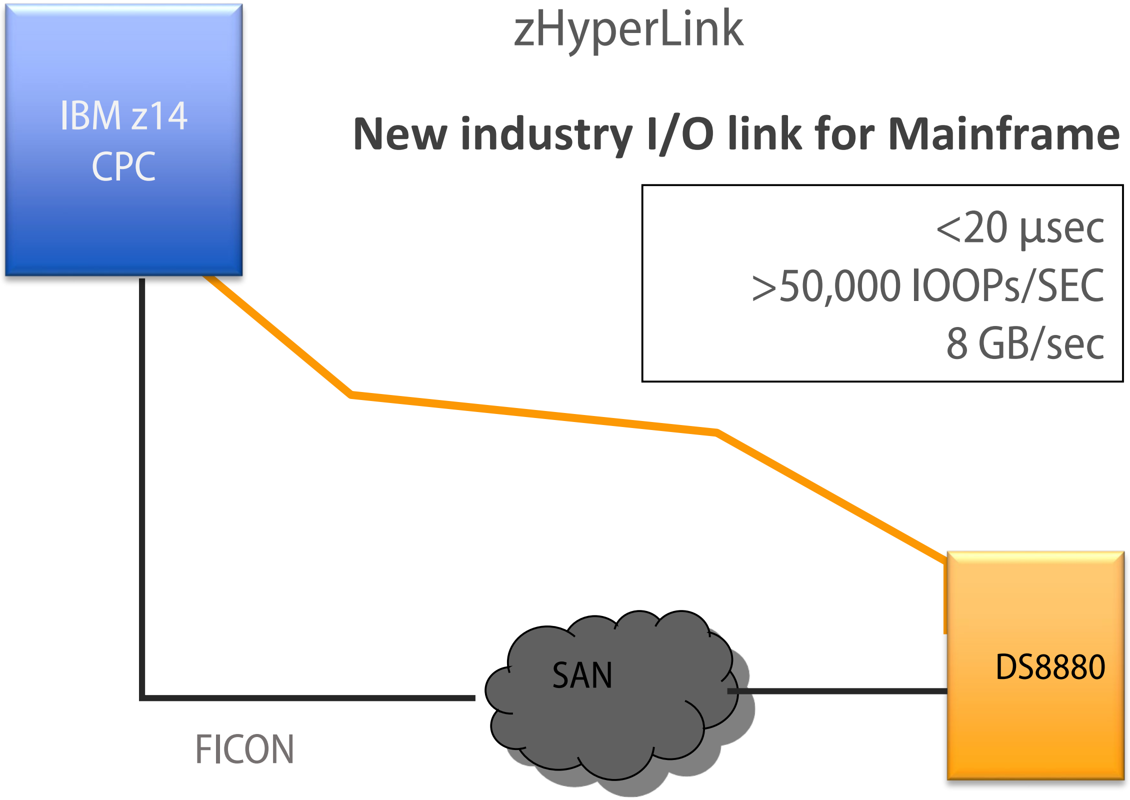
- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice and at IBM's sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing 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.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Agenda

- **Key performance metrics in Db2**
- **Updated metrics and how to interpret them**
  - Class 3
  - EDM pools
  - Dynamic Statement Cache
  - RID pool
  - Buffer pools

How does IBM zHyperLink™ change the game?

- **IBM zHyperLink™ dramatically reduces latency by interconnecting the z14 CPC directly to the I/O Bay of the DS8880**
- **zHyperLink™ is FAST enough the CPU can just wait for the data**
  - No un-dispatch of the running task
  - No CPU queueing delays to resume it
  - No host CPU cache disruption
  - Very small I/O service time
- **Better latency on database I/O intense application or update intensive applications, excluding 100% in-memory (bufferpool) hit cases**



Key Performance Metrics

- CPU
- Memory (Storage) Usage
- Contention

Key Metrics

CPU

- Accounting Class 1 and Class 2
- Statistics: Db2 System address space CPU time

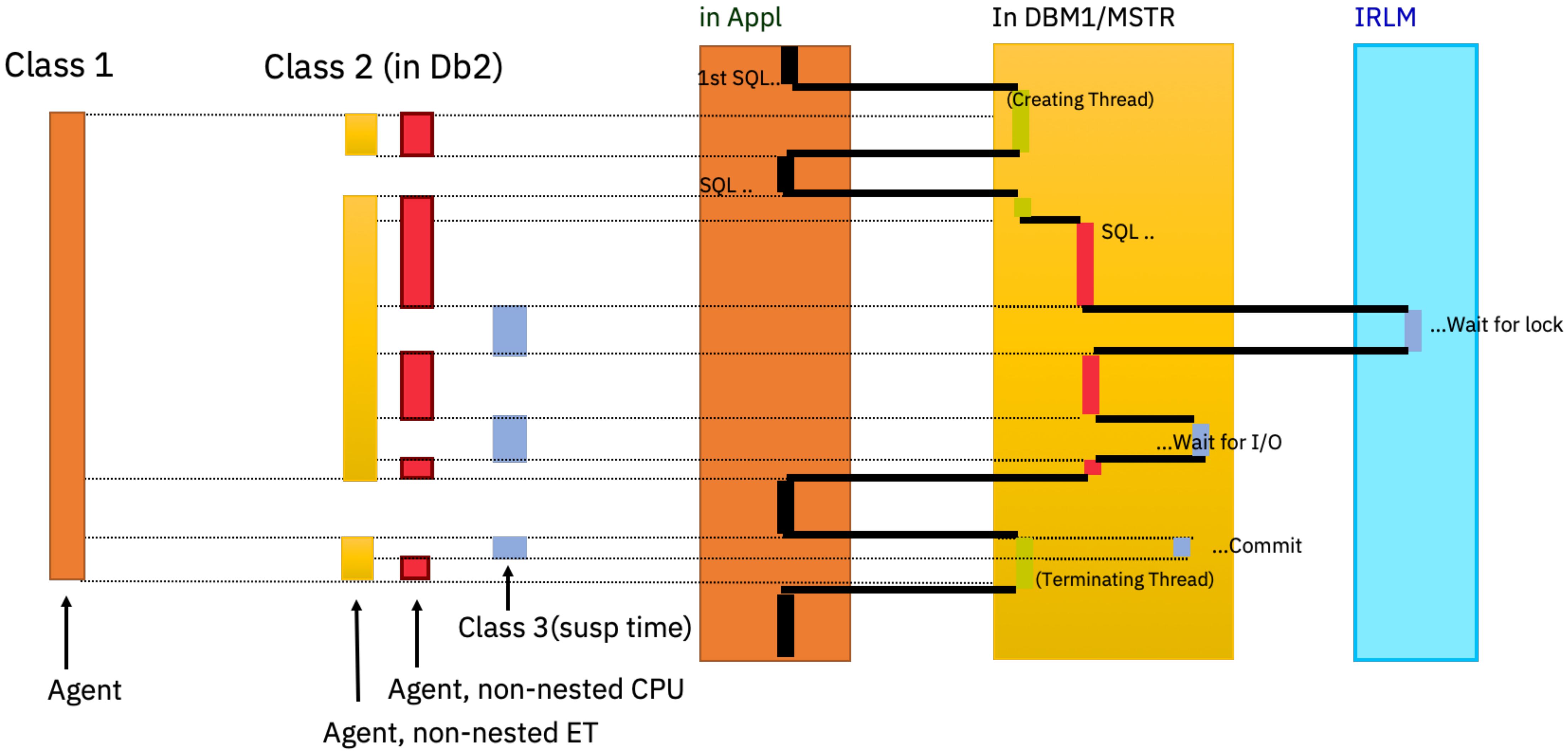
Memory

- Statistics: DBM1, DIST, IRLM Storage Statistics
- Statistics: Buffer pools, EDM pools, RID pools, Sort pools, etc.

Contention

- Accounting Class 3
- Statistics: Locking and data sharing locking, Latch contention

Accounting Class 1,2,3 (Elapsed, CPU and Suspension)



CPU TIME – Application

- **Accounting CPU time: Class 1, Class 2, Class 7**
  - CPU time
  - Part of CPU time but zIIP eligible time
  - zIIP time

  - CPU time includes synchronous CF requests, zHyperLink I/O requests

CPU TIME – System Level (Statistics)

- **Statistics: System address space CPU time**
  - TCB time, SRB time: preemptable, preemptable zIIP time, and non-preemptable SRBs
  - More async task under zIIP preemptable SRBs are added in V11/12
    - V11 Pseudo delete index entry clean up
    - V11 Buffer Pool async allocation (APARs PI84278/PI91719)
    - V12 Fast Traverse Block monitoring
    - V12 Insert Algorithm 2 Pipe management
  - I/O interrupt CPU time (NEW with APAR PI92652)
    - CPU time for read I/O interrupt (including decryption CPU cost)
      - QWSAIIPT: CPU time for I/O interrupt processing for the address space

MEMORY Reporting - Virtual & Real Memory (IFCID 225)

- **DBM1, DDF, IRLM address space report memory usage in IFCID 225**
- **DBM1 Below 2G Virtual (2G less your common memory areas)**
  - Notable consumer: open data set adds approx. 5KB per data set
  - z/OS V2.3 provided some improvement
- **DBM1 Above 2G Virtual**
  - Fuzzy numbers, focus should be on REAL memory monitoring
- **DBM1 Real memory usage**
  - 31 bit and 64 bit real memory usage
  - Considerations to using 64 bit numbers

MEMORY Reporting - Virtual & Real Storage (IFCID 225) …

IFCID 225	Description
QW0225HVPagesInReal	DBM1 REAL in use for 64-bit priv (MB)
QW0225HVAuxSlots	DBM1 AUX in use for 64-bit priv (MB)
QW0225PriStg_Real	DBM1 REAL in use for 64-bit priv w/o BP (MB)
QW0225PriStg_Aux	DBM1 AUX in use for 64-bit priv w/o BP (MB)
QW0225ShrStg_Real	REAL in use for 64-bit shared (MB)
QW0225ShrStg_Aux	AUX in use for 64-bit shared (MB)
QW0225ShrStkStg_Real	REAL in use for 64-bit shared stack (MB)
QW0225ShrStkStg_Aux	AUX in use for 64-bit shared stack (MB)
QW0225ComStg_Real	REAL in use for 64-bit common (MB)
QW0225ComStg_Aux	AUX in use for 64-bit common (MB)
QW0225_REALAVAIL	REALAVAIL (MB) (S)

QW0225xxxStg_Real/Aux: 64bit real storage for threads to be monitored

REALSTORAGE_MANAGEMENT Serviceability

- **ZPARM REALSTORAGE MANAGEMENT = AUTO (default) and ON users**
  - Real frame clean up via z/OS DISCARD operation with KEEPREAL(YES) option
- **KEEPREAL(YES)**
  - Unless paging occurs the real frames are still accounted to Db2 even they are unused, freed and discarded
- **Concern**
  - Inaccurate real memory usage for Db2 address space, difficulty in capacity planning

- **Serviceability to collect accurate stats**
- **Collection is once a day at midnight via z/OS COUNT page operation**
  - Count DISCARDED pages
  - Activation via IFCID 503 on
  - Expect some increase in MSTR TCB time at midnight
- **APARs PI78979 (V12) PI82191 (V11)**
  - Requires z/OS APAR OA50366
- **Not applicable for REALSTORAGE_MANAGEMENT= OFF users**

http://www.ibm.com/support/docview.wss?uid=swg21971496

REAL MEMORY MANAGEMENT – Discarded pages

- **A new flag, QW0225DPGS, indicates that this record contains values for the following new fields in DSNDQW03**
  - QW0225PriStg_dpage
    - Number of 4K pages from the Private Object which are charged to Db2 and are available for Steal, which are Discarded and have not been retouched
  - QW0225ShrStg_dpage
    - Number of 4K pages from the Shared Object which are charged to Db2 and are available for Steal, which are Discarded and not been retouched
  - QW0225ShrStkStg_dpage
    - Number of 4K pages from the Shared Stack Object which are charged to Db2 and are available for Steal, which are Discarded and not been retouched
  - QW0225ComStg_dpage
    - Number of 4K pages from the Common Object which are charged to Db2 and are available for Steal, which are Discarded and not been retouched
- **Accurate real storage usage**
  - QW0225xxxStg_Real + QW0225xxxStg_Aux - QW0225xxxStg_dpage

Recently Updated Metrics

- Accounting Class 3

CLASS 3 SUSPENSIONS	AVERAGE TIME	AV.EVENT	TIME/EVENT
LOCK/LATCH(DB2+IRLM)	0.176721	6.63	0.026648
IRLM LOCK+LATCH	0.000798	3.69	0.000216
DB2 LATCH	0.175922	2.94	0.059741
SYNCHRON. I/O	0.016379	4.17	0.003929
DATABASE I/O	0.000825	0.43	0.001934
DATABASE I/O CACHE HIT (TBD)			
LOG WRITE I/O	0.015553	3.74	0.004157
OTHER READ I/O	0.014771	4.96	0.002976
OTHER WRTE I/O	0.001423	1.07	0.001326
SER.TASK SWTCH	0.000026	0	0.089458
UPDATE COMMIT	0	0	N/C
OPEN/CLOSE	0	0	N/C
SYSLGRNG REC	0	0	N/C
EXT/DEL/DEF	0.000026	0	0.09381
OTHER SERVICE	0	0	0.000231

Contention Reports

V12 zHyperLink Eligible I/Os

V11 includes utility Format Write

CLASS 3 …

CLASS 3 SUSPENSIONS
LOCK/LATCH(DB2+IRLM)
IRLM LOCK+LATCH
DB2 LATCH
SYNCHRON. I/O
DATABASE I/O
DATABASE I/O CACHE HIT (TBD)
LOG WRITE I/O
OTHER READ I/O
OTHER WRTE I/O
SER.TASK SWTCH

SQL locks or IRLM latch

Check if this is IRLM latch wait from locking section. If IRLM latch wait is high & LOB users, make sure to apply APAR PI94553. If IRLM Query requests in stats is high, check the monitoring product use via IFI 150 call

Db2 Latch

Check statistics latch counters to see which latch is contributing this

V12 zHyperLink Eligible I/Os

Sync I/Os which hit the disk cache but did not use zHyperLinks
Possible zHyperLink candidate I/Os in a future

V11 includes utility Format Write

If excessive, check deferred write hit in statistics

CLASS 3 …

CLASS 3 SUSPENSIONS	AVERAGE TIME	AV.EVENT	TIME/EVENT
ARC.LOG(QUIES)	0	0	N/C
LOG READ	0	0	N/C
DRAIN LOCK	0	0	N/C
CLAIM RELEASE	0	0	N/C
PAGE LATCH	0.18965	148.18	0.00128
NOTIFY MSGS	0.000357	1.6	0.000224
GLOBAL CONTENTION	0.009084	6.08	0.001493
COMMIT PH1 WRITE I/O	0	0	N/C
ASYNCH CF REQUESTS	0.000069	0.94	0.000074
TCP/IP LOB XML	0	0	N/C
ACCELERATOR	0	0	N/C
AUTONOMOUS PROCEDURE	0	0	N/C
PQ SYNCHRONIZATION	0	0	N/C
LOB COMPRESSION	0	0	N/C
FAST INSERT PIPE	0.00012	0.25	0.000478

Contention Reports

Time Spent in IDAA

V11 Autonomous Procedure Wait

V11 Parallelism Wait

V12 zEDC Wait (LOB compression)

V12 Insert algorithm Pipe wait

CLASS 3 …

CLASS 3 SUSPENSIONS
PAGE LATCH
PQ SYNCHRONIZATION
LOB COMPRESSION
FAST INSERT PIPE

Data, space map or Index page latch

If from space map, larger page size or IAG2 in V12 may provide relief
If from index page, check if there is split – Log write I/O wait

V11 Parallelism Wait

It was NOT ACCOUNT time in V10
If excessive, possibly parallel task is not cut equally

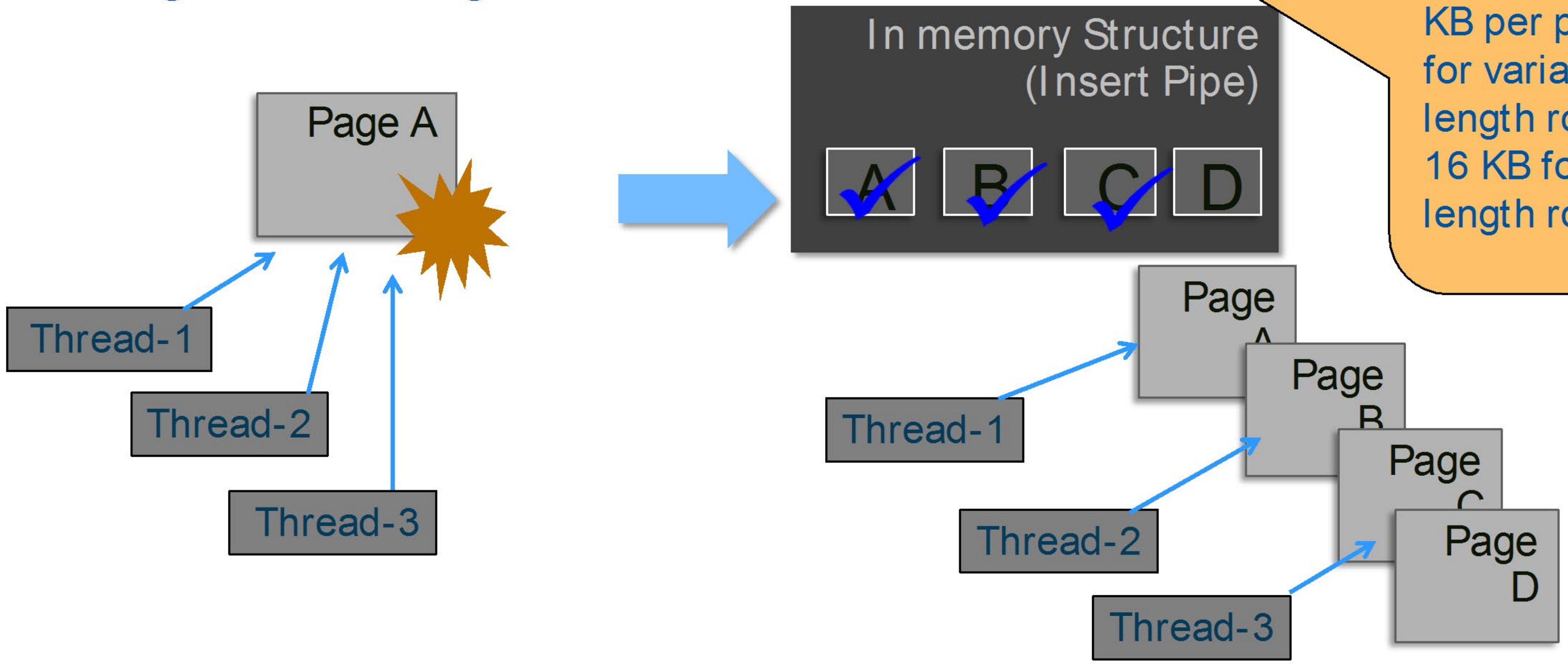
V12 zEDC Wait (LOB compression)

V12 zEDC service time, if excessive check RMF PCIe activity report for possible saturation in zEDC

V12 Insert algorithm Pipe wait

V12 Pipe wait. If excessive check if there is too many data set extension. Also check Db2 latch

Introducing New Insert Algorithm with Member Cluster



Additional 32 KB per partition for variable length rows and 16 KB for fixed length rows

Instrumentation Updates with Insert Algorithm 2 (V12)

- **Class 3 Suspension: Insert Pipe wait (base code)**
- **APAR PI81731 added statistics (IFCID 2) and accounting (IFCID 3) counters for**
  - #rows inserted by IAG1 QXRWSINSRTDAlg1
  - #rows inserted by IAG2 QXRWSINSRTDAlg2

Updated Metrics in Statistics

- EDM Pool and Dynamic Statement Cache

EDM Pools

EDM POOL	V11	V12
PAGES IN DBD POOL (ABOVE)	262.1K	N/A
HELD BY DBD	542	654
STEALABLE PAGES	0	0
FREE PAGES	261.6K	N/A
% PAGES IN USE	0.21%	100%
FAILS DUE TO DBD POOL FULL	0	0
PAGES IN STMT POOL (ABOVE)	262.1K	N/A
HELD BY STATEMENTS	26	29
FREE PAGES	262.1K	N/A
FAILS DUE TO STMT POOL FULL	0	0
PAGES IN SKEL POOL (ABOVE)	2560	N/A
HELD BY SKCT	1	1
HELD BY SKPT	1172	1223
STEALABLE PAGES	1103.66	1192.83
% PAGES IN USE	2.71%	100%
FAILS DUE TO SKEL POOL FULL	0	0

V11 Defined in zparm

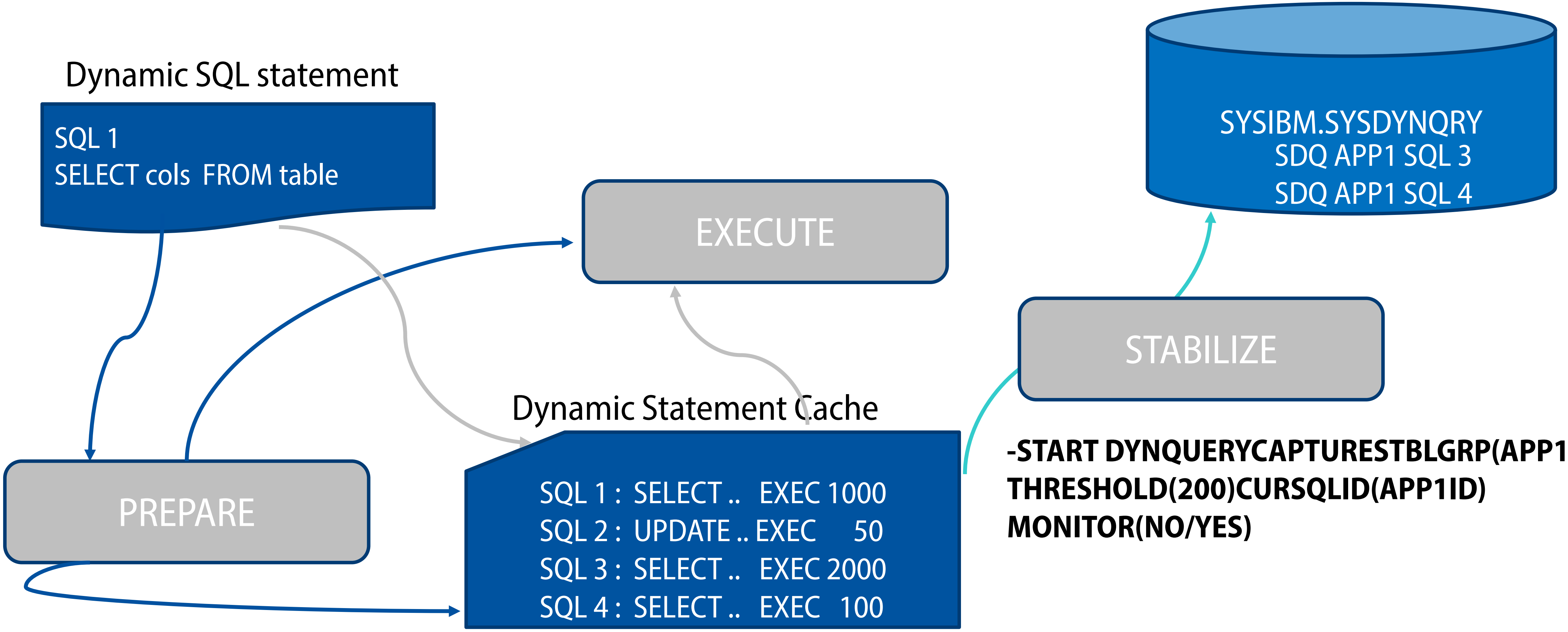
V12 Allocated as needed

V11 or prior release fails when exceeded zparm values

V12 does not fail, but the HELD counter >= zparm value means Db2 frees unused spaces under LRU management *

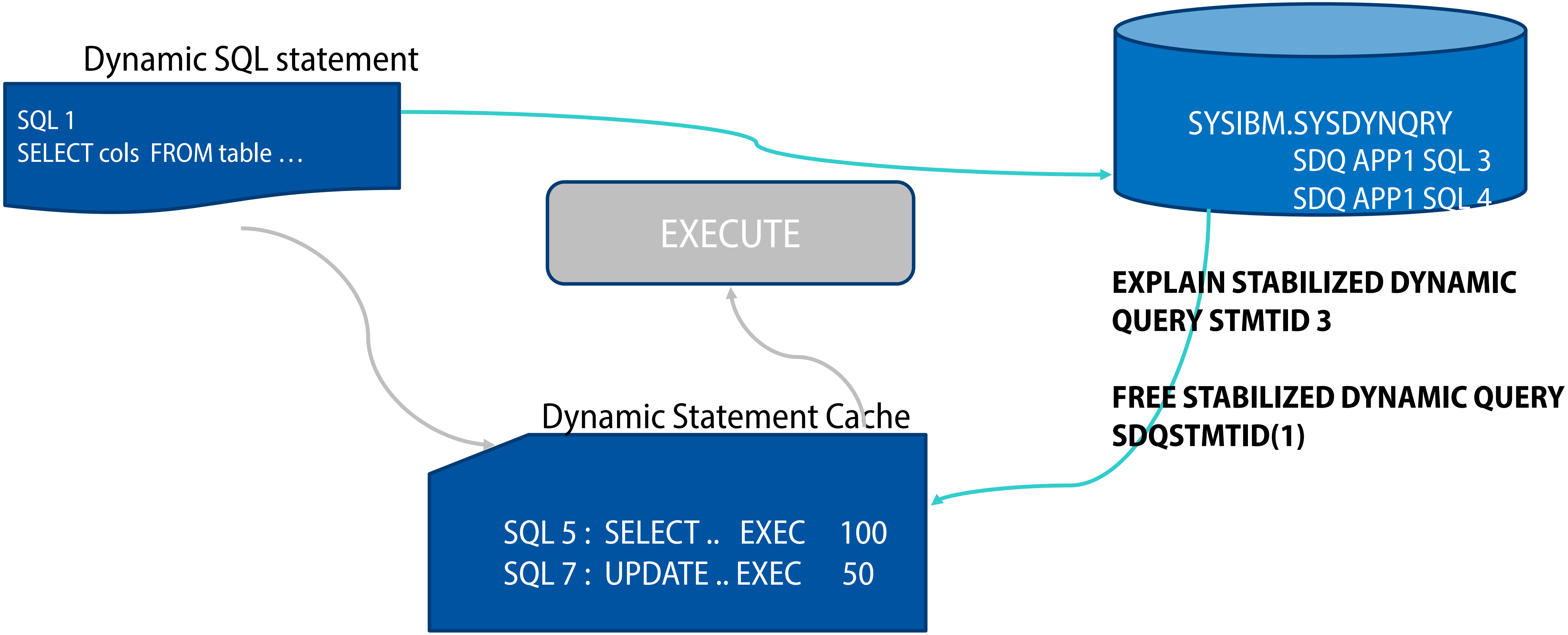
* Note this behavior requires APAR PI91054

Dynamic Statement Stability



-START DYNQUERYCAPTURESTBLGRP(APP1) THRESHOLD(200)CURSQLID(APP1ID) MONITOR(NO/YES)

Dynamic Statement Stability



Prepare

- **“Full Prepare”**
  - Most expensive
  - Invokes parser, Access Path Selection
- **“Short Prepare”**
  - Explicit “short prepare” from Global Statement Cache (EDM) - reuse prepared STMT from cache
  - Explicit “short prepare” from Catalog - reuse stabilized STMT from catalog SYSIBM.SYSDYNQRY
  - “Prepare avoidance” possible through KEEPDYNAMIC YES
    - Cheapest - reuse the thread copy of prepared statement
- **“Implicit Prepare” occurs when the thread copy of the prepared SQL statement no longer exists in the local dynamic SQL Cache**
  - Always results in either a “Short Prepare” or “Full Prepare”

STATEMENT CACHE

DYNAMIC SQL STMT	FIELD NAME	QUANTITY
PREPARE REQUESTS	QISEDSG	5201.0K
FULL PREPARES	QISEDSI	296
SHORT PREPARES	DSG-DSI	5200.7K
SHORT PREPARES		5200.7K
BASED ON CACHE	DSG-DSI-DPSF	5200.7K
BASED ON CATALOG	QISEDPSF	0
LOOK-UP IN CATALOG	QISEDPSL	0
CACHE HIT RATIO (%)		99.99
CACHE+CATALOG HIT RATIO (%)		99.99
TOTAL PREPARES		5201.0K
EXPLICIT PREPARES	QXPREP	5201.0K
IMPLICIT PREPARES	QXSTIPRP	0
STABILIZED PREPARES	QXSTSFND	0
PREPARES AVOIDED	QXSTNPRP	0
CACHE LIMIT EXCEEDED	QXSTDEXP	0
PREP STMT PURGED	QXSTDINV	0
LOCAL CACHE HIT RATIO(%)		N/C

V12 Prepared from the stabilised stmt in the catalog

V12 Catalog look up to look for the prepared statement

V12 Prepared from the cache in the catalog, basically same as QISEDPSF

STATEMENT CACHE: Concentrate Statements With Literals

CSWL - STMTS PARSED	QXSTCWLP	# of times Concentrate Statements with literals are parsed
CSWL - LITS REPLACED	QXSTCWLR	# of times literal replacement occurs
CSWL - MATCHES FOUND	QXSTCWLM	# of times the statement matched via literal replacement
CSWL - DUPLS CREATED	QXSTCWLD	# of times duplicate cache entry is created due to matching criteria (reusability)

Updated Metrics – Statistics

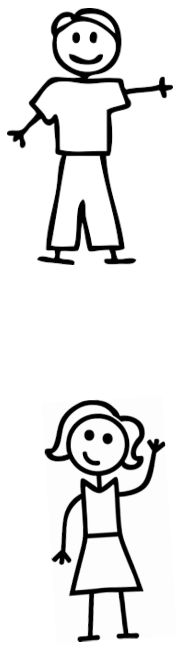
- RID processing

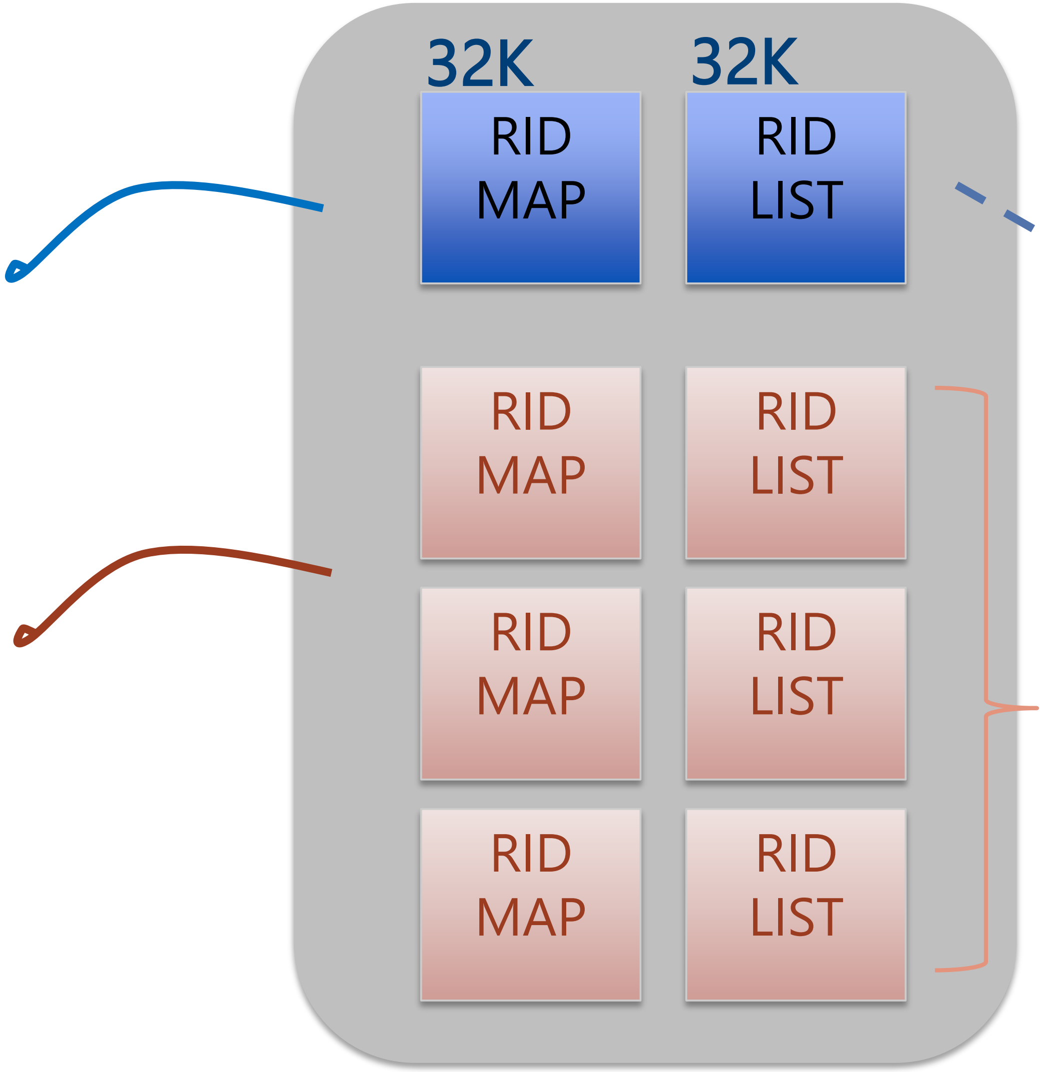
RID Processing

- **RID list is used in:**
  - List prefetch
  - Multiple Index Access
  - Hybrid join
  - Enforcing unique keys for multi row update

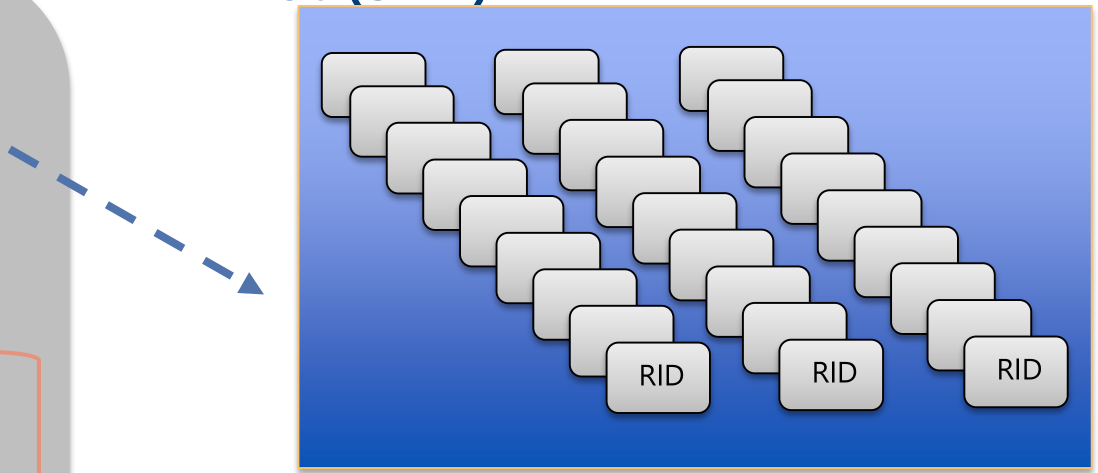
- **RID Pool Related Parameters**
  - MAXRBLK as RID Pool size
    - Default 400MB
  - MAXTEMPS_RID to limit overflow usage in work file per RID process
    - Default NOLIMIT
  - MAXTEMPS to limit MAXTEMPS_RID per agent
    - Default 0 (no limit)

RID Pools





RID List (32K)



V11 6533 RIDs per block
V12 4033 RIDs per block

RID Entries

V11 Up to 26.5 millions
V12 Up to 16.5 millions

RIDPOOL Sizing

- V11: # concurrent RID processing activities * average number of RIDs * 2 * 5 (bytes per RID)
- V12: # concurrent RID processing activities * average number of RIDs * 2 * 8 (bytes per RID)
- Or… QISTRHIG (MAX RID BLOCKS ALLOCATED) * 32K

RID LIST Counters (Since V11)

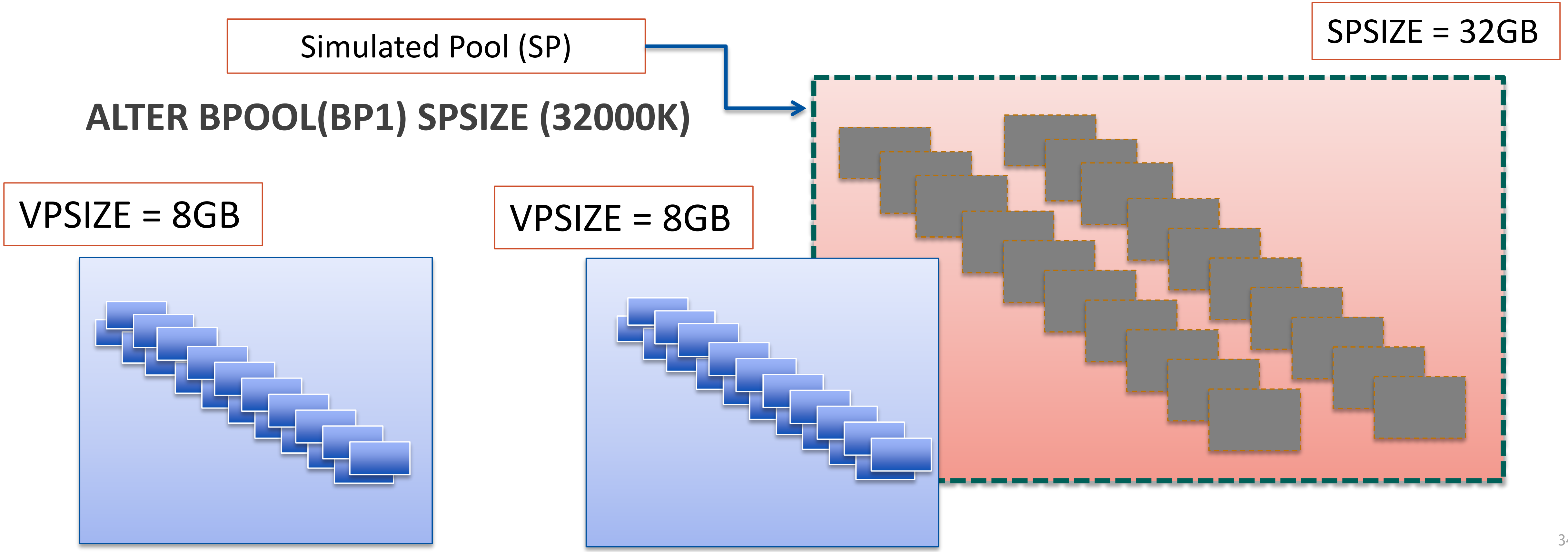
RID LIST PROCESSING	QUANTITY
SUCCESSFUL	50417.6K
NOT USED-NO STORAGE	28
NOT USED-MAX LIMIT	0
MAX RID BLOCKS ALLOCATED	1793
CURRENT RID BLOCKS ALLOCAT.	11.7
MAX RID BLOCKS OVERFLOWED	0
CURRENT RID BLOCKS OVERFL.	0
STORAGE LIMIT EXCEEDED	0
RDS LIMIT EXCEEDED	7
DM LIMIT EXCEEDED	0
PROC.LIMIT EXCEEDED	21
OVERFLOWED-NO STORAGE	21
OVERFLOWED-MAX LIMIT	7
INTERRUPTED (HJ)-NO STORAGE	0
INTERRUPTED (HJ)-MAX LIMIT	0
SKIPPED-INDEX KNOWN	42069

Updated Metrics

- **Buffer Pools**
  - Simulated Pools (V11/V12)
  - Serviceability fields (V11/V12)
  - IFCID 199 (V12)
  - zHyperLinks (V12)
  - Overflow (V12)

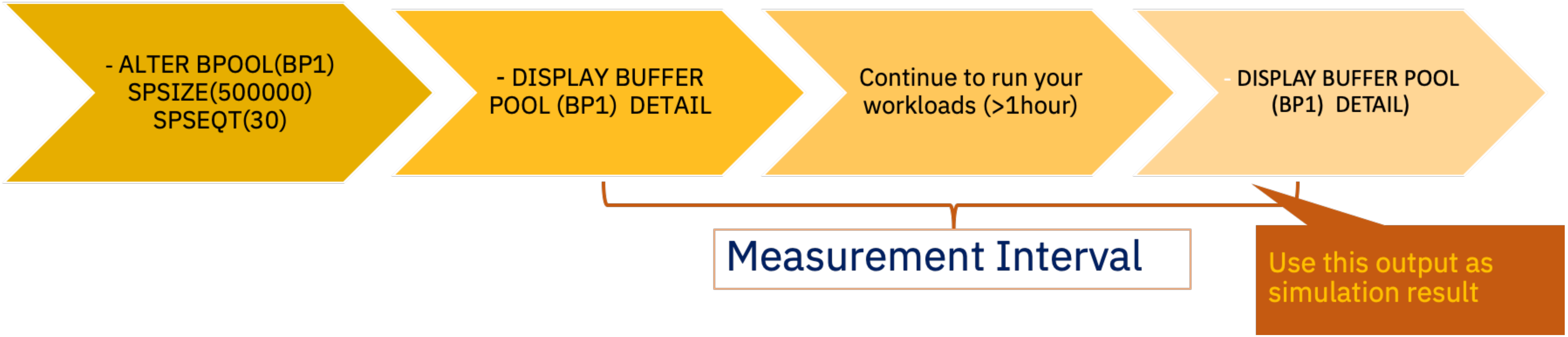
Buffer Pool Simulation (Db2 11 feature introduced with APAR PI22091)

- **Benefit of expanding buffer pools varies with data re-reference and the ratio between object size and buffer pool size**
- **Simulation provides accurate benefit of increasing buffer pool size from production environment**



How To Use Simulation – Example

- **Determine simulated pool size (SPSIZE) and simulated sequential prefetch threshold (SPSEQT)**
  - Cost Real memory (64bit above): SPSIZE * 0.02 for 4K pools, 0.01 for 8K, so on
- **Issue ALTER command with SPSIZE, SPSEQT (optional)**
- **Collect statistics data or DISPLAY BUFFER POOL DETAIL command outputs**
  - Recommend to reset SPSIZE(0) when simulating different SPSIZE
  - Recommend to take enough samples (2-3 hours) per SPSIZE



Simulation Output

```
AVOIDABLE READ I/O -
     SYNC  READ I/O  (R)   =25463982
     SYNC  READ I/O  (S)   =81181
     ASYNC READ I/O        =15470503
     SYNC  GBP READS  (R)  =11172099
     SYNC  GBP READS  (S)  =4601
     ASYNC GBP READS       =1181076
PAGES MOVED INTO SIMULATED BUFFER POOL =53668641
TOTAL AVOIDABLE SYNC I/O DELAY =35321543 MILLISECONDS
```

Avoidable synchronous DB I/Os (R/S)

Avoidable prefetched pages

Avoidable GBP requests

Numbers of pages stolen from VPOOL and moved to SPOOL

Avoidable Sync I/O elapsed time in millisec

Avoidable sync I/O per second = Sync I/Os / interval-sec

(25463982 + 81181)/360 = 70958 I/O per sec

Notes: After APAR PH02577

- DSNB432I -D2PA SIMULATED BUFFER POOL ACTIVITY -

```
        AVOIDABLE READ I/O -
             SYNC  READ I/O (R)  =25463982
             SYNC  READ I/O (S)  =81181
             ASYNC READ I/O      =15470503
             SYNC  GBP READS (R) =11172099
             SYNC  GBP READS (S) =4601
             ASYNC GBP READS     =1181076
```

- DSNB432I -D2PA SIMULATED BUFFER POOL ACTIVITY -

```
        AVOIDABLE READ PAGE MISSES -
             SYNC    FROM DASD (R) =25463982
             SYNC    FROM DASD (S) =81181
             ASYNC   FROM DASD     =15470503
             SYNC    FROM GBP (R)  =11172099
             SYNC    FROM GBP (S)  =4601
             ASYNC   FROM GBP      =1181076
```

Buffer Pool SLRU Chains (Serviceability Counters)

BP8 GENERAL	QUANTITY
MIN BUFFERS ON SLRU	86639
MAX BUFFERS ON SLRU	88995
SLRU LENGTH EQUALS VPSEQT	0
GETPAGE REQU RANDOM ON SLRU	62160

BUFFERS ALLOCATED	2097200
VPSEQT	20%
SEQUENTIAL BUFFERS	419440

Validate your VPSEQT using these counters

IFCID 199 Data Set Level Statistics V11

IFCID 199 fields	
QW0199SV	AVERAGE SYNCHRONOUS I/O DELAY (IN MILLISECONDS)
QW0199SX	MAXIMUM SYNCHRONOUS I/O DELAY (IN MILLISECONDS)
QW0199AD	AVERAGE ASYNCHRONOUS I/O DELAY (IN MILLISECONDS)
QW0199AX	MAXIMUM ASYNCHRONOUS I/O DELAY (IN MILLISECONDS)

IFCID 199 Data Set Level Statistics V12

IFCID 199 fields	
QW0199S1	AVERAGE SYNCHRONOUS I/O DELAY (IN MICROSECONDS)
QW0199S2	MAXIMUM SYNCHRONOUS I/O DELAY (IN MICROSECONDS)
QW0199A1	AVERAGE ASYNCHRONOUS I/O DELAY (IN MICROSECONDS)
QW0199A2	MAXIMUM ASYNCHRONOUS I/O DELAY (IN MICROSECONDS)
QW0199Z1	AVERAGE SYNCHRONOUS I/O DELAY (IN MICROSECONDS) with zHyperLink
QW0199Z2	MAXIMUM SYNCHRONOUS I/O DELAY (IN MICROSECONDS) with zHyperLink

DISPLAY BPOOL DETAIL with zHyperLink Read

```
DSNB467I  -CEA2  STATISTICS FOR INDEX SPACE TPCEA100.TEHHIX2
                    DATA SET #:    27 USE COUNT:      0
DSNB453I  -CEA2   VP CACHED PAGES -
                     CURRENT        =   294   MAX             =   354
                     CHANGED        =    47   MAX             =   175
DSNB455I  -CEA2   SYNCHRONOUS I/O DELAYS -
                     AVERAGE DELAY =940   MAXIMUM DELAY   =14370
                     TOTAL PAGES   =797
                     SYNCHRONOUS I/O DELAYS WITH ZHYPERLINK -
                       AVERAGE DELAY =28   MAXIMUM DELAY   =388
                       TOTAL PAGES   =625
DSNB456I  -CEA2   ASYNCHRONOUS I/O DELAYS -
                     AVERAGE DELAY =614   MAXIMUM DELAY   =22271
                     TOTAL PAGES   =700   TOTAL I/O COUNT =486
```

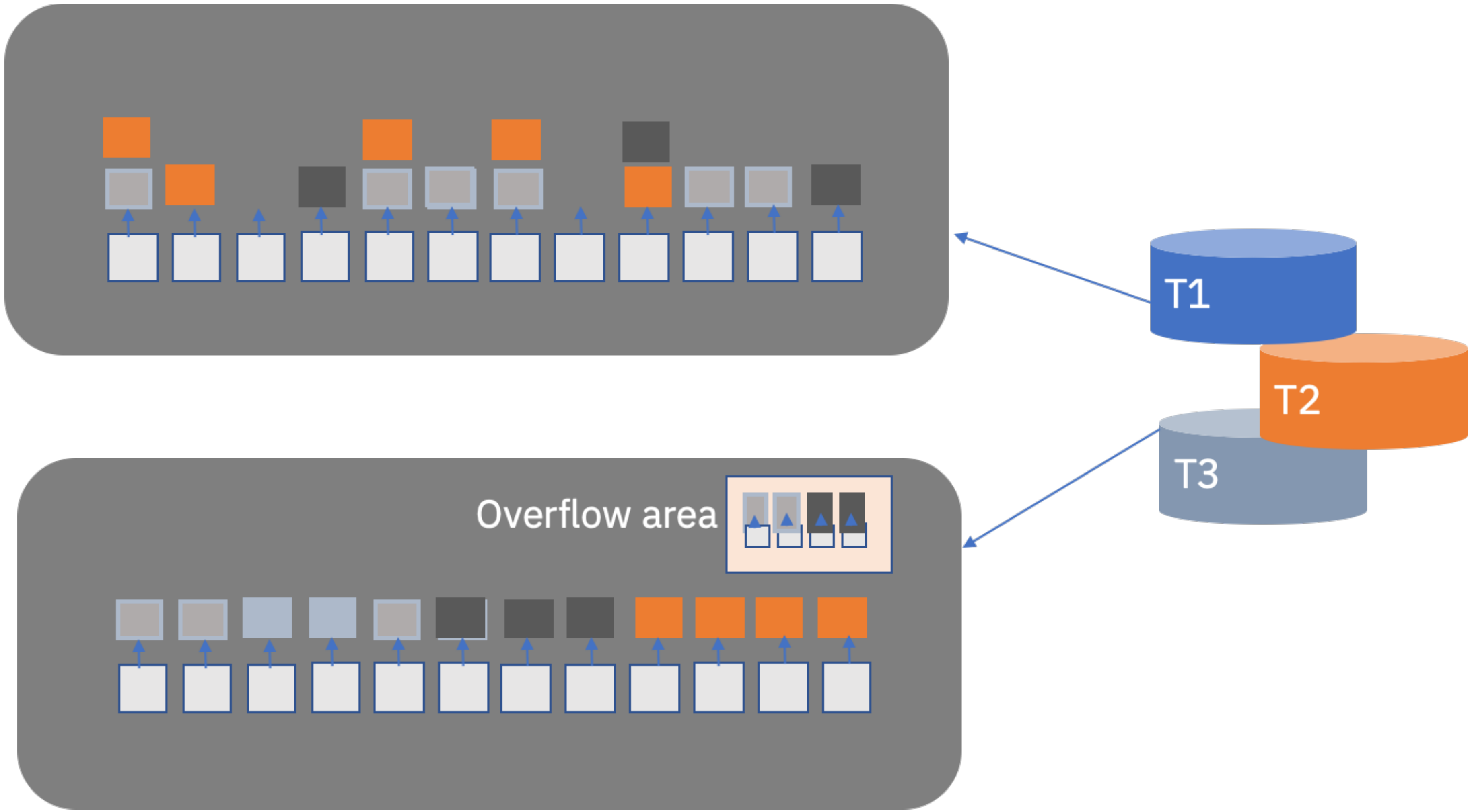
Updated PGSTEAL(NONE): Contiguous BUFFER POOL

Db2 11
PGSTEAL(NONE)

Maintains Hash and LRU chains

Db2 12
PGSTEAL(NONE)

No more chain maintenance except in overflow area

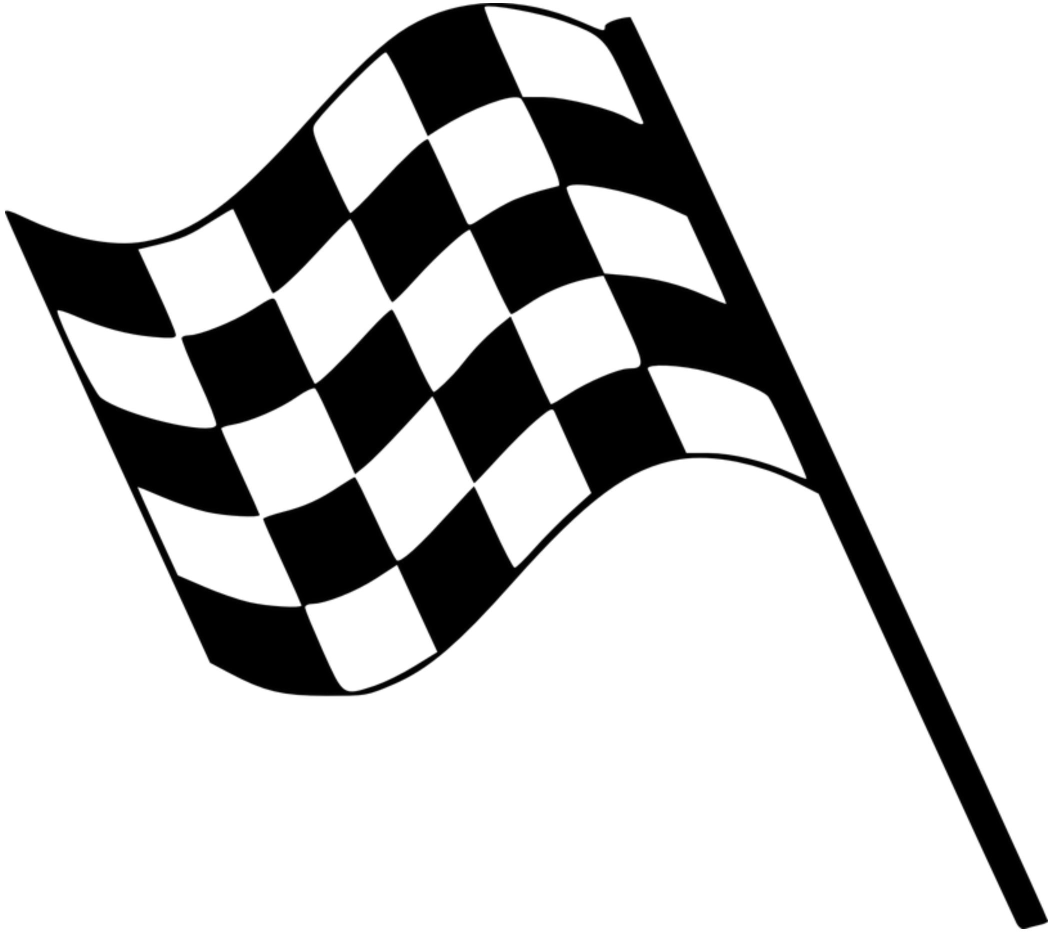


Bpool READ

BP1 READ	QUANTITY
GETPAGE REQUEST	526.5M
GETPAGE REQS-SEQUENTIAL	16584.9K
IN-MEM OVFL SEQ REQS	**0**
GETPAGE REQS-RANDOM	509.9M
IN-MEM OVFL RND REQS	**0**
SYNCHRONOUS READS	14
SYNC READS-SEQUENTIAL	0
IN-MEM OVFL SEQ READS	**0**
SYNC READS-RANDOM	14
IN-MEM OVFL RND READS	**0**

V12 Contiguous Buffer Pool Overflow activities, Ideally all 0, otherwise, check DSNB604I

Questions?



Please fill out your session evaluation before leaving!

John Campbell

IBM Db2 for z/OS Development

campbelj@uk.ibm.com

Session code: Z08

IDUG

Leading the Db2 User Community since 1988