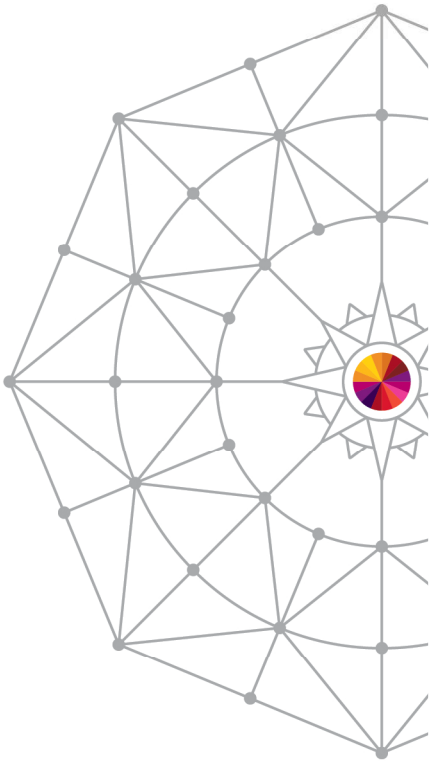
DB2 11 for z/OS Performance Expectations

John Iczkovits
IBM (iczkovit@us.ibm.com)

August 6, 2014
Session number 15440

Test link: www.SHARE.org



#SHAREorg

SHARE in Pittsburgh 2014

Please note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice 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.

Acknowledgment

Measurement data included and described in this presentation are obtained by the following teams

- DB2 for z/OS performance department at the IBM Silicon Valley Laboratory and China Development Laboratory.
- IBM SAP on System z performance group at IBM Poughkeepsie
- IBM Montpellier benchmark center
- DB2 11 ESP customers
- Special thanks to Akiko Hoshikawa at SVL

Agenda

- **DB2 11 Performance Quick Look**
  - ✓ Overview of DB2 11 Performance and Lab. measurements
- **DB2 11 Performance and Expectation**
  - ✓ System Level Performance Improvement
    - ✓General optimization
    - ✓System z Synergy
    - ✓Migration performance
  - ✓ DDF Performance
  - ✓ Query Performance
  - ✓ Application Level Performance
- **REORG Reduction and Consistent Performance**
- **Summary**
  - ✓ Summary on migration performance
  - ✓ ESP customer update

DB2 11 Performance Focus

- **CPU and cost reduction**
- **Scalability enhancements**
- **Focus on customers' pain points**
  - Consistent performance with less REORG
  - Less need of performance tuning

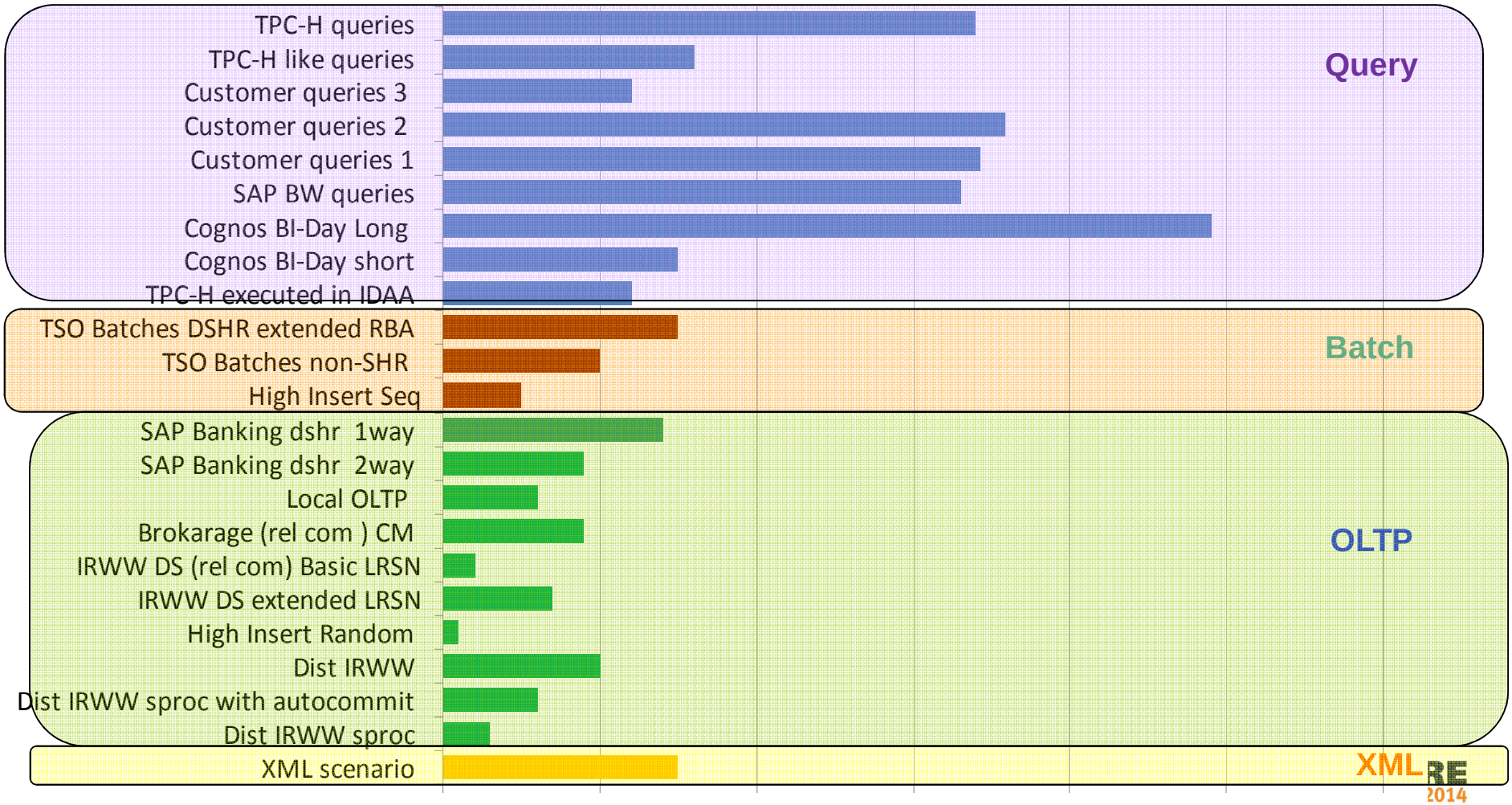
Performance Expectation in DB2 11

Continuing a new trend started in DB2 10, DB2 11 expects,

- **Up to 10% from complex OLTP**
- **Up to 10% from update intensive Batch**
- **Up to 25% from reporting queries without compressed tables**
- **Up to 40% from complex queries with compressed table**

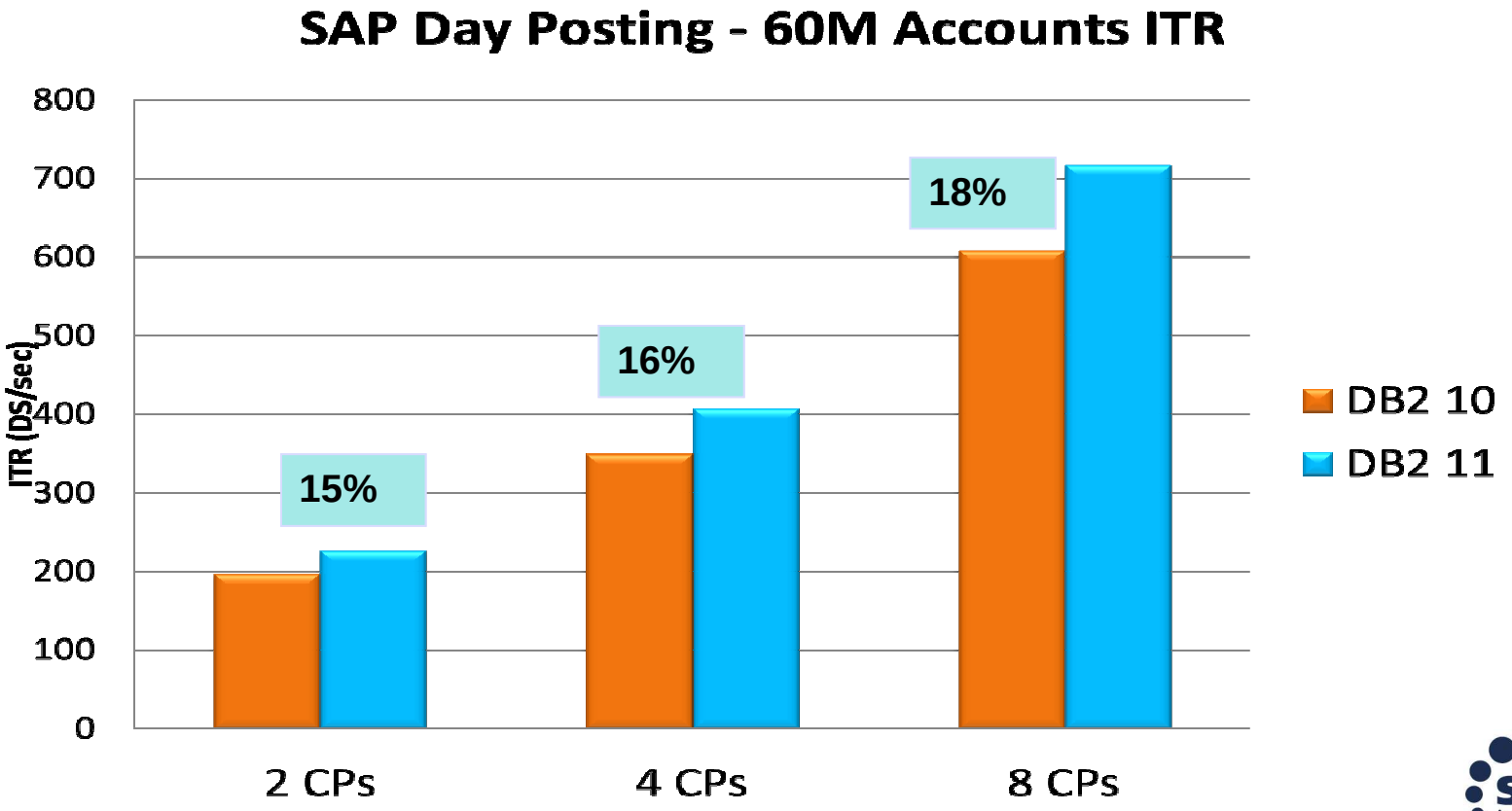
…. Your mileage varies

DB2 11 % CPU Improvement From DB2 10



DB2 11 Scalability and SAP Banking Workload

- SAP Banking Service Day Posting workload (SBS 7.0) with 60 million accounts using 512 partitions
- Up to 18% ITR improvement with DB2 11



What Does "OUT-OF-THE- BOX" Mean?

- **Transparent from applications**
  - No DDL, DML, application changes
  - Examples of NOT "out of the box"
    - V8 Multi Row Fetch for local applications
    - V10 INCLUDE index
    - V11 Not Logged DGTT
- **REBIND is necessary**
  - REBIND with reused access path (APREUSE) can give good improvement
  - REBIND without reusing access path could give you even more
- **Additional saving is possible with some actions…**
  - Converting to extended LRSN in data sharing

Details on DB2 11 Measurements

- **Redbook : DB2 11 for z/OS Performance Topics (SG248222)**
- **Techdoc : DB2 11 for z/OS with SAP® Performance Report (WP102394)**
  - http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102394

Agenda

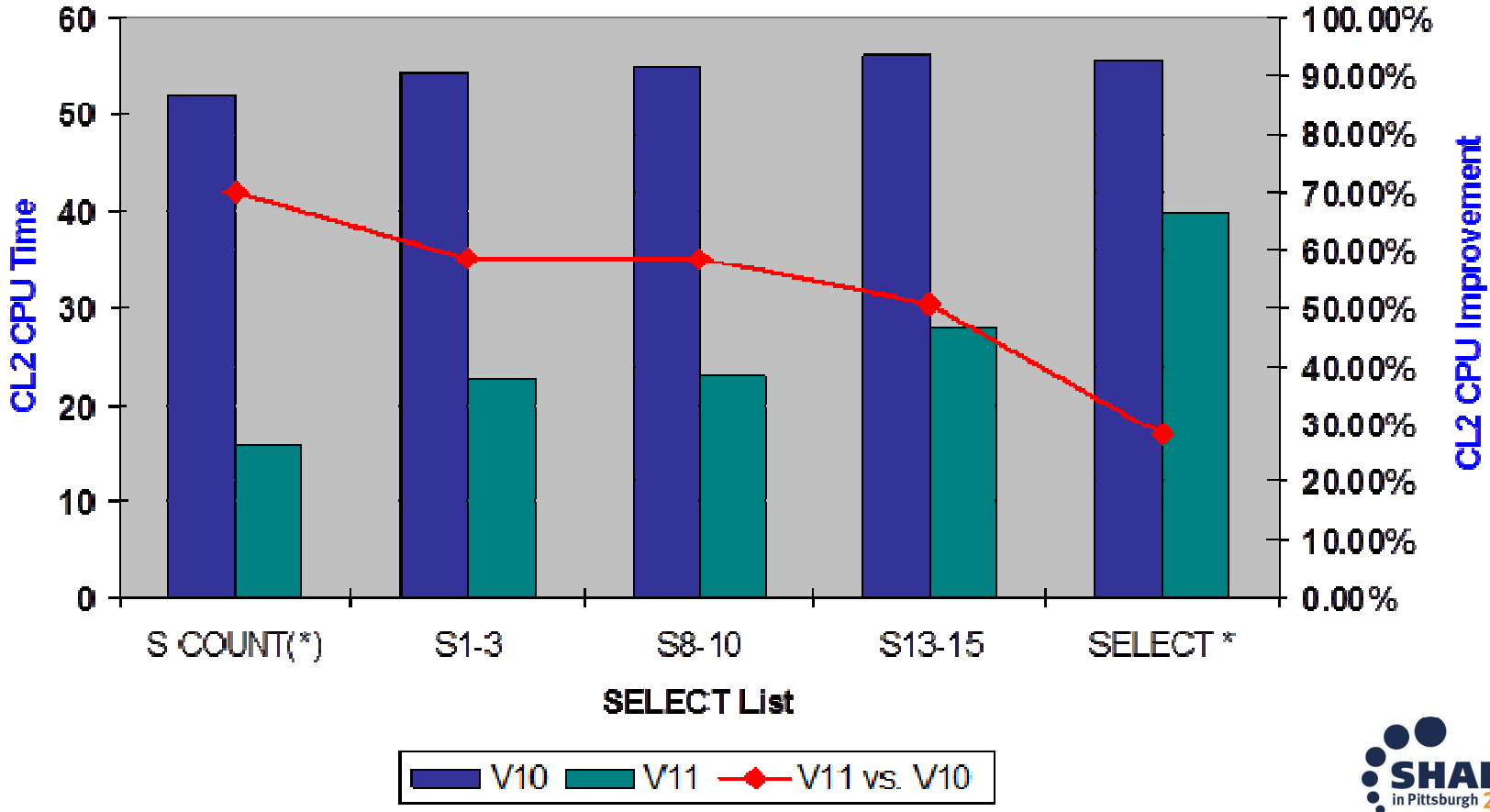
- DB2 11 Performance Quick Look
  - ✓ Overview of DB2 11 Performance and Lab. measurements
- **DB2 11 Performance and Expectation**
  - ✓ System Level Performance Improvement
    - ✓General optimization
    - ✓System z Synergy
    - ✓Migration performance
  - ✓ DDF Performance
  - ✓ Query Performance
  - ✓ Application Level Performance
- REORG Reduction and Consistent Performance
- Summary

Optimization of Decompression

- **Decompression operation can add noticeable CPU overhead**
  - Higher impact with simple queries scanning large number of rows
- **Optimization in decompression process in DB2 11**
  - New decompression routine provides,
    - Decompress only necessary portions of row
    - Efficient processing on dictionary handling
  - Applicable only for table spaces, not for index compression
  - Compatible with existing compression, no user action is required other than REBIND
- **Early prototype evaluation**
  - 8-15% overall CPU reduction in multiple query workloads
    - Up to 70% observed for table space scan
    - Up to 25% observed for index-data access

Example of Decompression Improvement

DB2 CPU time scanning compressed table space



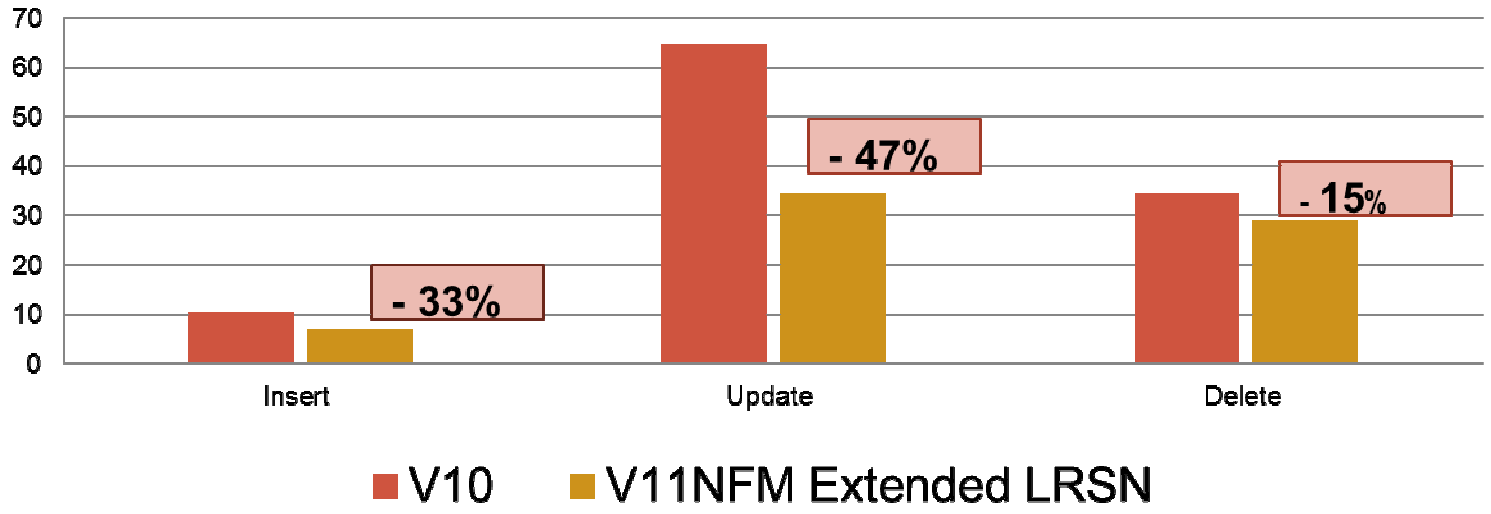
System Level Performance – Log Write

- **Log write buffers (OUTBUFF) in 64 bit common**
  - CPU reduction by removing cross address space operation for logging
    - Output log buffers are now located in 64 bit (high common)
  - Significant CPU reduction in update intensive batch jobs are observed
    - Make sure HVCOMMON in IEASYS can accommodate output log buffers
  - Use 1MB page frames for log buffers if LFAREA is available
    - Log buffers are page fixed since DB2 10
    - Configure additional 1MB LFAREA (z/OS parameter in IEASYSxx) to materialize maximum benefit
- **Reduction in log force write in data sharing**
  - Log write wait in class 3 often impacts elapsed time of insert/delete or index update against indexes with GBP dependent
    - Caused by log force write at index split and delete
  - DB2 11 reduces the number of log force write in insert/delete/update against GBP dependent Indexes
    - No user action

System Level Performance – Extended LRSN/RBA

- **Extended LRSN and LRSN Spin avoidance in Data Sharing**
  - Unique LRSN requires during seq update, delete, some insert in data sharing
  - Once conversion to 10bytes LRSN is done, LRSN spin is eliminated in DB2 11
    - LRSN avoidance requires both BSDS and objects conversion
      - General recommendation if you do not require immediate need of extended RBA/LRSN, is to convert BSDS first, then REORG the object as needed
    - Monitor log I/O performance impact due to log record size increase
      - Up to 30% log record growth observed by conversion

Example of CPU time saving with Extended LRSN



System Level Performance – Log Related

- **Compression dictionary enhancement for availability**
- **IFI Filtering for performance**
  - With filtering support, log read can qualify the objects
  - Additional benefit if the objects are compressed
  - 10-60% CPU reduction in log capture applications is observed
    - Retrofitted to DB2 10 via PM90568
  - Requires replication product upgrade for filtering capability
  - IBM InfoSphere Data Replication Q replication or Change Data Capture require architecture level 10.2.1
    - V10.2.1 provides both DB2 11 extended RBA and filtering support

Release Deallocate Enhancements

- **REL(DEALLOCATE)**
  - Avoid package allocation overhead = less CPU usage than REL(COMMIT)
  - Effective with transactions with frequent commits
- **Concerns**
  - Virtual storage foot print -> DB2 10 31 bit storage relief
  - REBIND, DDL and online REORG cannot break-in with persistent threads using REL(DEALLOC)
  - Accumulation of objects could impact thread footprint and CPU usage
- **DB2 11 change**
  - Allows REBIND/DDL, and online REORG to break in “committed” persistent threads with REL(DEALLOC). APAR PM95929 added local idle thread support
    - Note : Not all of persistent threads can be broken-in
  - Tracks object related resource and lock accumulation and releases them as needed
- **Benefits**
  - Availability improvement and consistently good performance with RELEASE(DEALLOCATE) even large number of objects are touched by the thread
    - Less memory/CPU footprint
  - More aggressive adaptation of RELEASE(DEALLOCATE) may become possible

Agenda

- DB2 11 Performance Quick Look
  - ✓ Overview of DB2 11 Performance and Lab. measurements
- **DB2 11 Performance and Expectation**
  - ✓ System Level Performance Improvement
    - ✓General optimization
    - ✓System z Synergy
    - ✓Migration performance
  - ✓ DDF Performance
  - ✓ Query Performance
  - ✓ Application Level Performance
- REORG Reduction and Consistent Performance
- Summary

Additional zIIP* Usage with DB2 11

- **Asynchronous processing that is executed under enclave SRBs (Service Request Blocks) and that will be "charged" for CPU consumption purposes to an IBM DB2 address space (rather than to user applications), with the exception of P-lock negotiation processing. Such zIIP eligible processing includes:**

  1) Cleanup of pseudo deleted index entries as part of DB2 system task cleanup

  2) Cleanup of XML multi-version documents (available in DB2 10 for z/OS via APAR PM72526)

  3) Log write and log read, Castout write processing

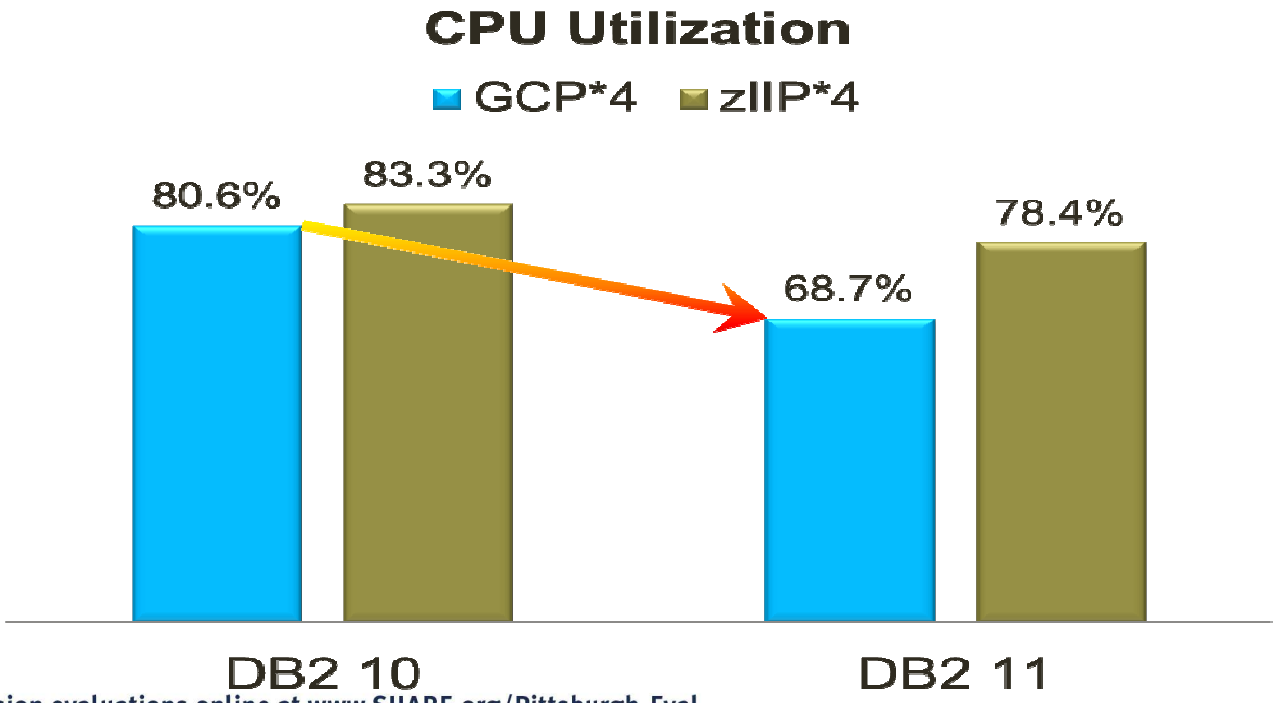
  – This is disabled completely if IIPHONORPRIORITY = NO!

- **The DB2 base LOAD, REORG and REBUILD INDEX utility processing of inline statistics collection that DB2 directs to be executed under enclave SRBs (Service Request Blocks)****
- **The DB2 base processing of RUNSTATS utility Column Group Distribution statistics collection that DB2 directs to be executed under enclave SRBs (Service Request Blocks)****
- **The DB2 base LOAD utility index management processing when running LOAD REPLACE that DB2 directs to be executed under enclave SRBs (Service Request Blocks)****

- *** NOTE: This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g, zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the "Authorized Use Table for IBM Machines" provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html ("AUT"). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.**
- **** NOTE: DB2 does not direct all such base utility processing to be executed under enclave SRBs.**

Overall CPU and Cost Reduction

- Up to 81% zIIP-eligible CPU with RUNSTATS COLGROUP
- Up to 40% zIIP-eligible CPU in REORG & LOAD with inline distribution stats
- Essential to check zIIP capacity before DB2 11 migration
  - zIIP help function (IIPHONORPRIORITY=YES, default) should be enabled in case there is a shortage of zIIP capacity
  - Continue to monitor zIIP utilization
- A case study with SAP Banking workload (OLTP-DRDA)
  - Typical CICS or IMS environment not likely to be configured with # GCP = # zIIP



PGFIX (YES) and 1MB Page Frames

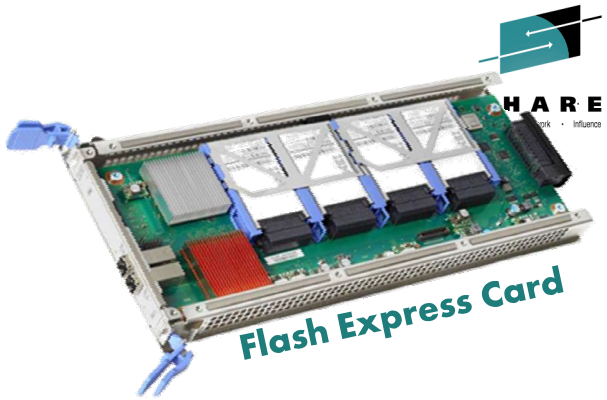
	VP SIZE	Get Page	Sync Read	Pre-Fetch	Write	Hit Ratio	I/O Intensity	GP Intensity	PG Fix	1MB
BP0	3K	138	0	0	0.06	100%	0	5		
BP1	524.3K	1496.3K	0.03	0	589	100%	0	285	Y	Y
BP7	2097K	160.4K	404	0	402	100%	0	8		
BP12	524.3K	93.6K	2101	35300	197	98%	7	18	Y	Y
BP23	2097K	40.9K	9873	2530	433	76%	1	2	Y	

- **I/O intensity = (READ+WRITE) / BPOOL SIZE**
- **GETPAGE intensity = GETPAGE / BPOOL SIZE**
- **PGFIX (YES): recommended for high I/O intensity buffer pools (BP12,23)**
- **1MB page : recommended for high getpage intensity buffer pools**
- **BP1 is the best candidate with PGFIX(YES) with 1MB page frames**
  - DB2 10 : DB2 prefers to allocate 1MB page frames for PGFIX YES buffer pools
  - DB2 11 : User can specify preferred page frame size. 1MB requires PGFIX YES

DB2 and zEnterprise EC12 and BC12

- **2GB Page Frames (fixed)**
  - Requires LFAREA allocation (LFAREA 2G=xx)
  - DB2 11 buffer pools can support FRAMESIZE(4K, 1M, 2G)
    - Potential benefit of further TLB miss
    - A buffer pool needs to be >2GB
    - If you have 3GB buffer pools,
      - 1 of 2GB frame + 1000 of 1MB frames
    - Display buffer pool shows frame size and counters
- **RoCE (RDMA over Converged Ethernet)**
  - Transparent from DB2
  - Communications protocol based on Infiniband
  - Available with zEC12 GA2
    - Initial support z/OS to z/OS connection
  - DDF transaction (DB2z-DB2z) show up to 2x throughput improvement
- **zEDC Express (zEnterprise Data Compression)**
  - SMF compression
- **FLASH Express**

DB2 and zEC12 Flash Express



What is Flash Express :

–Flash SSDs to server side using a standard PCIe IO adapter

–Each card pair provides 1.4 TB storage; Maximum 4 card pairs (5.6 TB)

–Configured as SCM (Storage Class Memory) from HMC and z/OS LPAR

Benefit of Flash express :

1.Use as paging device for faster response time when system level paging are needed

–z/OS Dump service takes advantage of this feature

2.Pageable large pages (PLP or 1MB pageable)

–Similar benefit with 1MB fixed page by reducing processor TLB misses

–PLPs need to be backed by real storages either non LFAREA or LFAREA.

–If PLPs are ever paged out, Flash express can be used as paging device

–Strongly recommend not to page out

DB2 and Flash Express

- **With Flash Express installed and configured (SCM>0)**

  1) DB2 requests buffer pool control blocks (PMBs) to be backed by Pageable Large Page (1MB)

  - PMBs consume only 4-5% of buffer pool allocations but referenced very heavily and no I/Os (no need to fix) -> Good match with PLP
  - Same behavior between DB2 11 and DB2 10 with PM85944
  - If most of buffer pools are using 1M fixed frames, no additional benefit
  - If most of buffer pools are using 4K frames, potential CPU reduction
  - More getpage intensive your operations are, better improvement

  2) DB2 11 requests DB2 code pages to be loaded on PLPs

  - Requires DB2 11 and z/OS 2.1
  - Potential CPU reduction in code execution from large pages

- **Regardless Flash Express installed or not,**

  –Paging with any of DB2 storage pools is NOT recommended

Buffer Pool Management

- **Faster buffer pool allocation**
  - Benefit restart time with users with large buffer pool > 5-10GB
    - DB2 10 - allocate as needed
    - DB2 11 - Virtual allocation with defined size, real allocation as needed except PGFIX buffers
- **Buffer pool re-classification change**
  - DB2 9 and 10
    - Pages read by prefetch stay as “sequential” (subjects of VPSEQT)
  - DB2 11 has better management
    - Prefetched pages can be reclassify as random after random getpage
    - Potential benefit users with mixed access (prefetch + random) to reduce Sync I/Os
- **MRU (Most Recently Used) usage with more Utilities**
  - Goal is to avoid buffer pool contamination by utility operations
  - DB2 9 and 10 : COPY utility
  - DB2 11 adds,
    - UNLOAD, RUNSTATS INDEX/TABLESPACE
    - UNLOAD phase of REORG TS, REBUILD INDEX, CHECK INDEX and DATA

Agenda

- DB2 11 Performance Quick Look
  - ✓ Overview of DB2 11 Performance and Lab. measurements
- **DB2 11 Performance and Expectation**
  - ✓ System Level Performance Improvement
    - ✓General optimization
    - ✓System z Synergy
    - ✓Migration performance
  - ✓ DDF Performance (covered by DDF performance session)
  - ✓ Query Performance (covered by Query performance session)
  - ✓ Application Level Performance
- REORG Reduction and Consistent Performance
- Summary

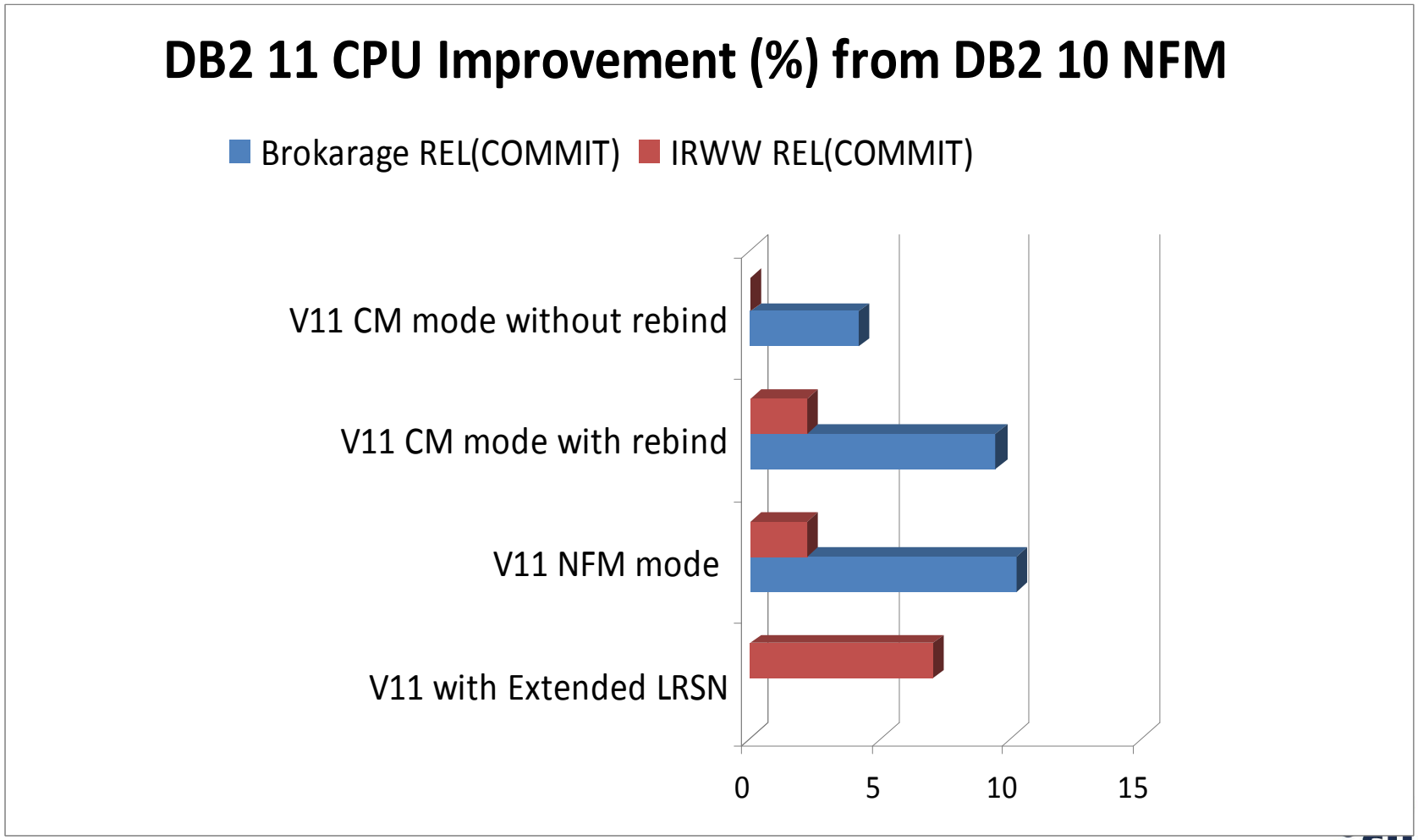
Do I Need to REBIND?

- **YES, strongly recommended (same as DB2 10)**
  - SELECT procedures are disabled for v9, v10 bound static packages until REBIND with DB2 11
    - Impact in the range of 0% to 10%
    - Bypass sproc in QISTCOLS, package names in IFCID 224
  - Structure conversion (so called ‘puffing’) until REBIND with DB2 11
- **REBIND with or without APREUSE**

Which Items are Required REBIND to See Benefit?

V11 System Performance Optimization/Features	Triggering action
General Code optimizations	None
Customized code generation for sort	None
Optimizing the access against the table with large number of partitions	None
Rel(deallocate) optimization	None
DB2 latch reductions (LC14)	None
Buffer Pool enhancements	None
More zIIP processing	None
DDF Sync Receive enhancements	TCP/IP APAR
Data sharing performance improvement	None
Customized code generation for column processing	REBIND (ok with APREUSE)
Decompression Improvement	REBIND (ok with APREUSE)
Xproc above the bar	REBIND (ok with APREUSE)

Example from OLTP measurements



Agenda

- DB2 11 Performance Quick Look
  - ✓ Overview of DB2 11 Performance and Lab. measurements
- **DB2 11 Performance and Expectation**
  - ✓ System Level Performance Improvement
    - ✓General optimization
    - ✓System z Synergy
    - ✓Migration performance
  - ✓ DDF Performance
  - ✓ Query Performance
  - ✓ Application Level Performance
- REORG Reduction and Consistent Performance
- Summary

DDF Performance Summary

- TCP/IP Synchronous Receive
- Implicit Commit for CALL statement
- Continuous Block Fetching Cursors
  - Aka Blasting cursors

TCP/IP Synchronous Receive Support (cont'd)

- Benefit: Reduced network latency and CPU in DIST address space
  - No application changes or binds required
  - DSN6FAC CMTSTAT=INACTIVE only supported
- Performance measurements using the distributed IRWW suite
  - 6% to 10% ITR improvement
  - 4% to 7% CPU improvement
- Performance improvement will not be observed if processing single message transactions

Implicit Commit Support for CALL Statement: Results

- Expectation: Improve single CALL statement transactions in V11 as compared to prior releases
- CLI client performance test results
  - CL.1 Elapsed: Up to -56%
  - CL.1 CPU: Up to -16%
  - CL.2 CPU: No significant change
- JDBC client performance test results
  - CL.1 Elapsed time: Up to -77%
  - CL.1 CPU time: Up to -16%
  - CL.2 CPU: No significant change
- Up to -5% CL.1 CPU for IRWW

Package Based Continuous Block Fetch (cont'd)

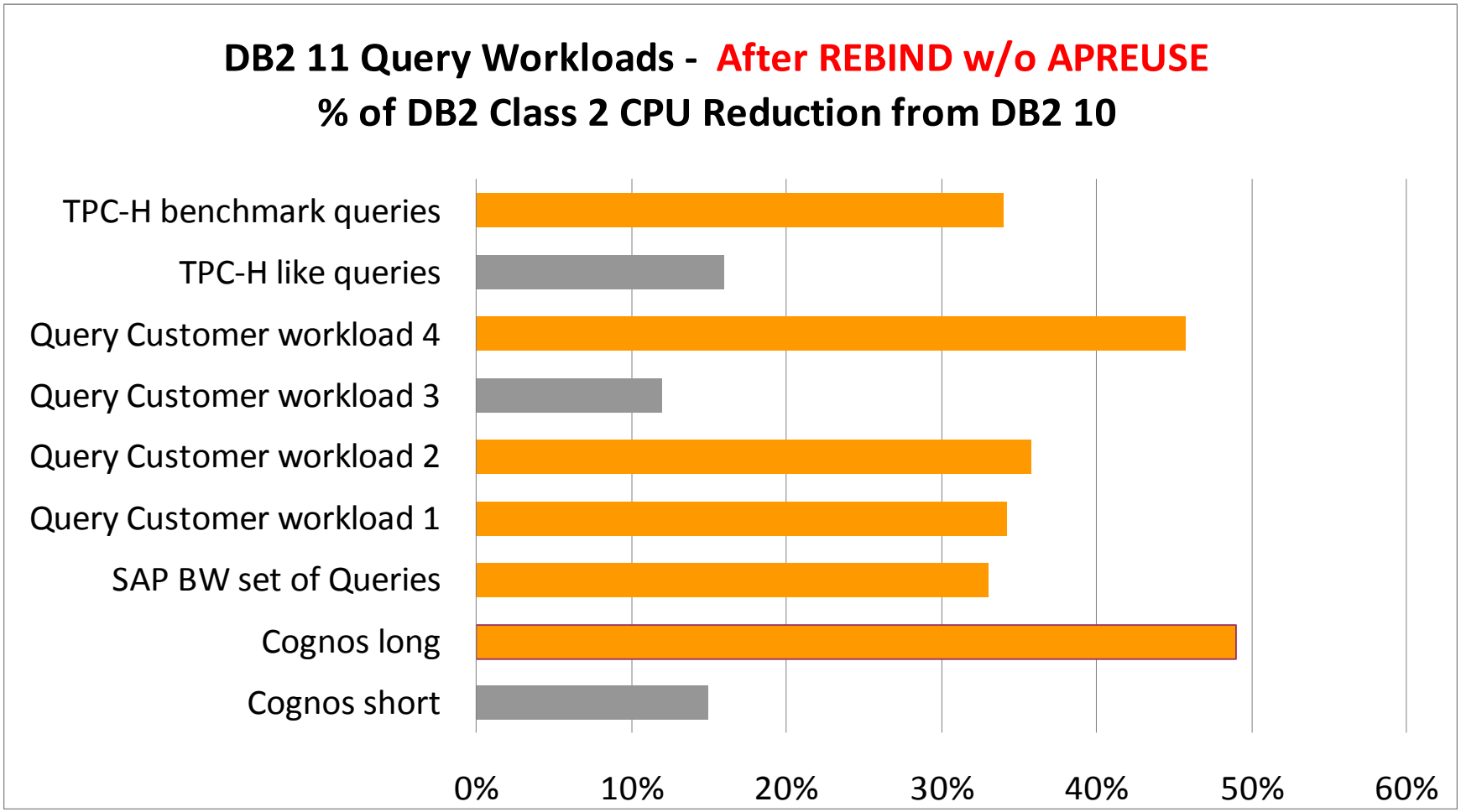
- Application cannot switch between DRDACBF and non-DRDACBF packages if connections exist
- Reviewing impact of query block size on elapsed time
  - Currently 4MB query block size is used
  - Reviewing results with z/OS Communications Server team

Delta %	Class 1 Elapsed Time	Class 2 Elapsed Time	Class 1 CPU Time	Class 2 CPU Time
Server	-29.5	-8.3	-20.0	-5.8
Requester	-31.1	-31.1	-13	-13

Agenda

- DB2 11 Performance Quick Look
  - ✓ Overview of DB2 11 Performance and Lab. measurements
- **DB2 11 Performance and Expectation**
  - ✓ System Level Performance Improvement
    - ✓General optimization
    - ✓System z Synergy
    - ✓Migration performance
  - ✓ DDF Performance
  - ✓ Query Performance
  - ✓ Application Level Performance
- REORG Reduction and Consistent Performance
- Summary

CPU Reduction in Query Workloads



- **Most performance improvements are also available with APREUSE**
- **New and improved access path choices may be available without APREUSE**
- **Higher improvement with more complex queries or/and w/compressed tables**

Key Enhancements in Query Area

- Reduced CPU for sequential access, especially for ACCESS_TYPE='R'
- Reduced CPU for processing compressed data pages
- Removed the V10 restrictions on sparse index use and improved performance
- Optimizer cost enhancements
- Index key skipping for DISTINCT, GROUP BY and SET function
- Early out for MIN() or MAX(), inner table in join/sub queries
- Predicate rewrite and pushdown
- More use of in memory work files
- Avoidance of materializing final merge from top sort
- DPSI access across multiple data partitions

Notes: Do I Need to REBIND to Benefit?

V11 Performance Feature	NO REBIND	REBIND w/ APREUSE	REBIND
IX duplicate skipping	N	Y*	Y
Early out enhancements	N	Y*	Y
Stage 2 to Stage 1 predicates	N	Y* (increased MC fails with APREUSE(ERROR), succeeds with APREUSE(WARN))	Y
Predicate pruning	N	Y*	Y
pred. pushdown to DM	N	Y	Y
Select list do-once	N	Y	Y
Sparse Index	N	Y* (limited to existing type T access)	Y
Non-correlated subq using MXDTCACH	N	Y	Y
Non-correlated subq mismatched length	N	Y	Y
Correlated subquery caching	N	Y*	Y
RID overflow to WF (DM set function, list PF)	N	Y	Y
RID processing (HYB join limited to 80% of pool)	Y	Y	Y
DPSI cut on inner	N	N	Y
DPSI merge	Y	Y	Y
Sort performance	Y	Y	Y
In-memory/reusable WF	N	Y	Y
DECFLOAT (XML, implicit cast)	N	Y	Y
Data decompression enhancements	Y	Y	Y

NOTE: Y* indicates that existing access path may be supported with APREUSE, but new and improved access path choices may also be available.

Agenda

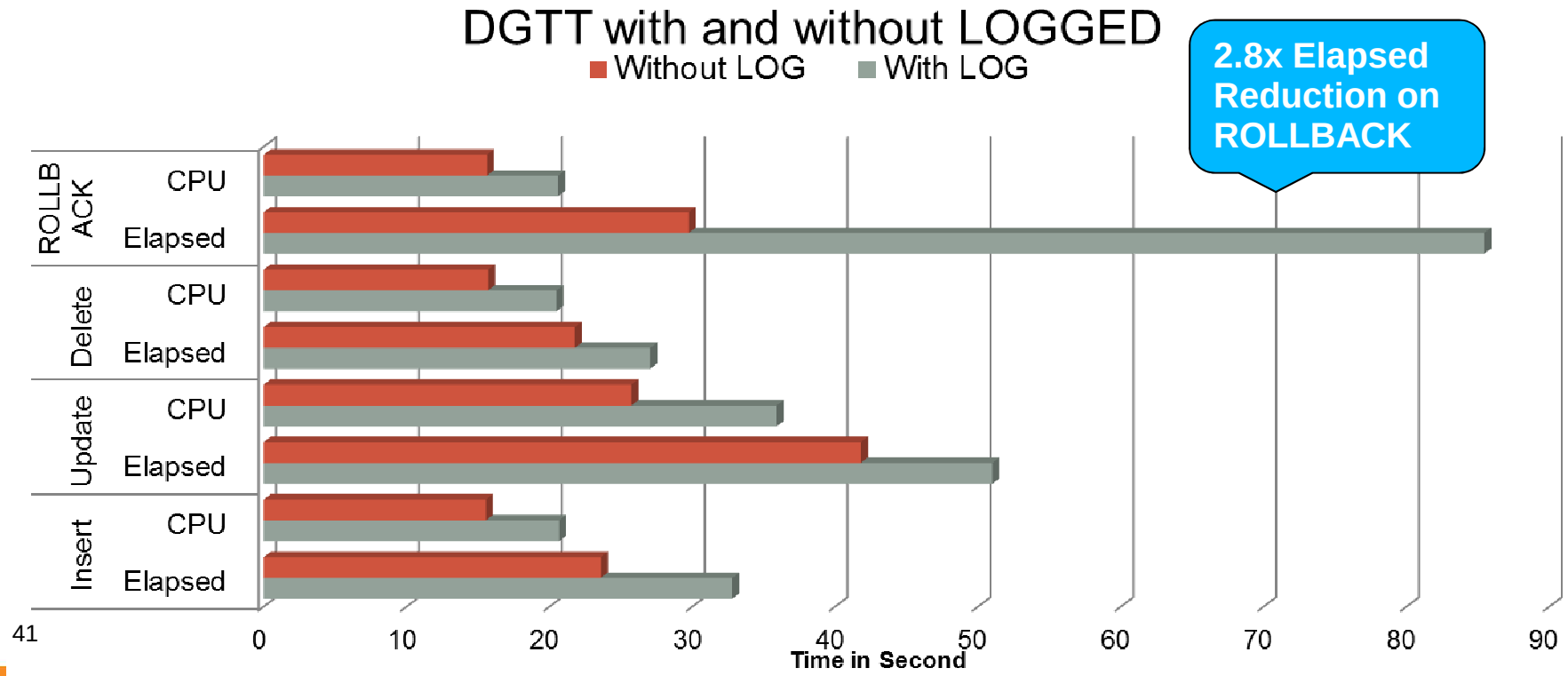
- DB2 11 Performance Quick Look
  - ✓ Overview of DB2 11 Performance and Lab. measurements
- **DB2 11 Performance and Expectation**
  - ✓ System Level Performance Improvement
    - ✓General optimization
    - ✓System z Synergy
    - ✓Migration performance
  - ✓ DDF Performance (covered by DDF performance session)
  - ✓ Query Performance (details are covered by Query performance session)
  - ✓ Application Level Performance
- REORG Reduction and Consistent Performance
- Summary

DGTT Improvement

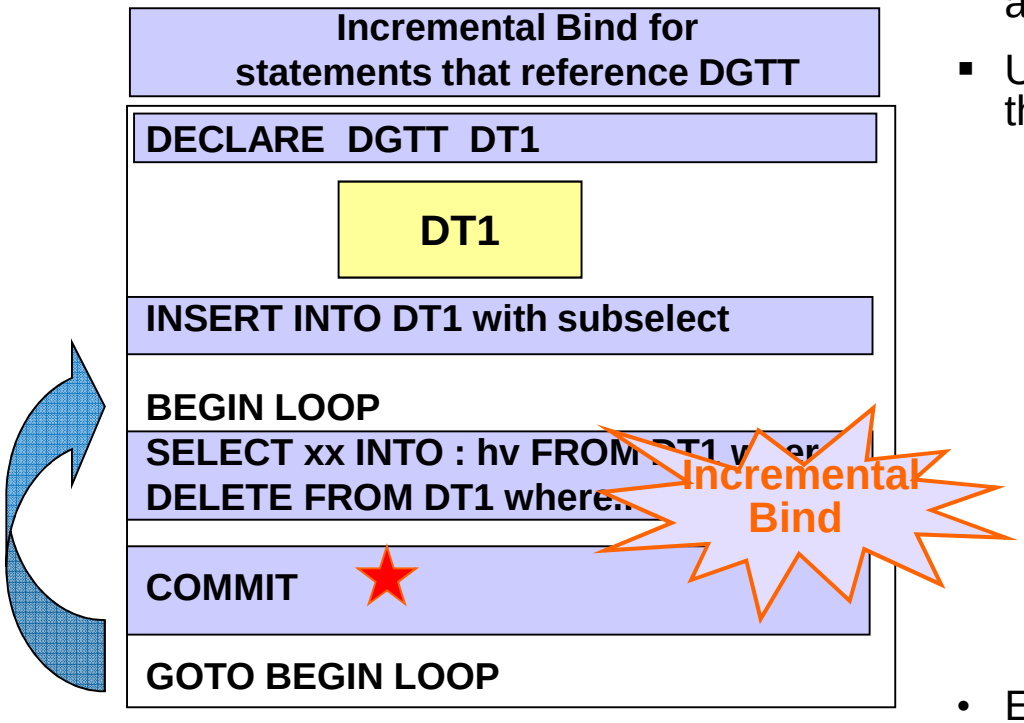
- Pain points in DGTT
  - DGTTs are logged
    - Cost of logging for insert, update, delete
    - Long rollback processes
  - DGTTs that are used frequently with intervening COMMITs result in
    - Repeated incremental binds at execution time for static SQL
    - Repeated full prepares for dynamic SQL

NOT LOGGED DGTTs - NFM

- **Required to use new NOT LOGGED option (default LOGGED)**
  - ON ROLLBACK DELETE ROWS (Default, rows deleted)
  - ON ROLLBACK PRESERVE ROWS (rows preserved)
  - Less CPU from log processing, less log volume
  - Faster rollback/error processing after large insert into DGTTs
  - Application programmers must take ROLLBACK behavior into consideration



Hold DGTT-based Statements Across COMMIT



- Non cursor static and dynamic statements that reference a DGTT are not kept across commit and requires incremental BIND
- Using RELEASE (DEALLOC) DB2 can keep the statements in DB2 11
  - (DEALLOCATE) – DB2 releases resources only when the program terminates
    - Extended to include non-cursor statements that reference DGTTs
      - INSERT, UPDATE, DELETE, MERGE, SELECT
      - Held beyond COMMIT point
      - Cursors already held beyond COMMIT point if WITH HOLD on cursor definition
- Exception: DGTTs defined with ON COMMIT DROP
- Requires REBIND with REL(DEALLOC)
- For Dynamic, application changes to remove PREPARE

  - Non-cursor static or dynamic statements not held across commit
  - Repeated incremental binds (*)
    - Full prepares for dynamic SQL

Exclude Null Index in NFM

```
CREATE INDEX Index_A  ON Table_A (C1,C2,C3)
EXCLUDE NULL KEYS  USING STOGROUP STG1 ;
```

- **Effective on an index with large numbers of NULL value**
  - Support EXCLUDE NULL KEYS in CREATE and ALTER INDEX
  - DB2 will not create an index entry for key columns with the NULL
- **Benefit and consideration**
  - Cost reduction on index maintenance
    - DML Index update,
    - Utility (LOAD, REORG, REBUILD, CHECK IX, RUNSTATS IX)
    - DDL CREATE INDEX
    - Disk storage
  - Potential fetch performance improvement
    - Less index leaf pages and potentially less index level
  - EXCLUDE NULL KEYS index cannot be used whenever a NULL key could satisfy the predicate
- **Preliminary performance look**
  - Up to 38% CPU reduction in high concurrent insert workload

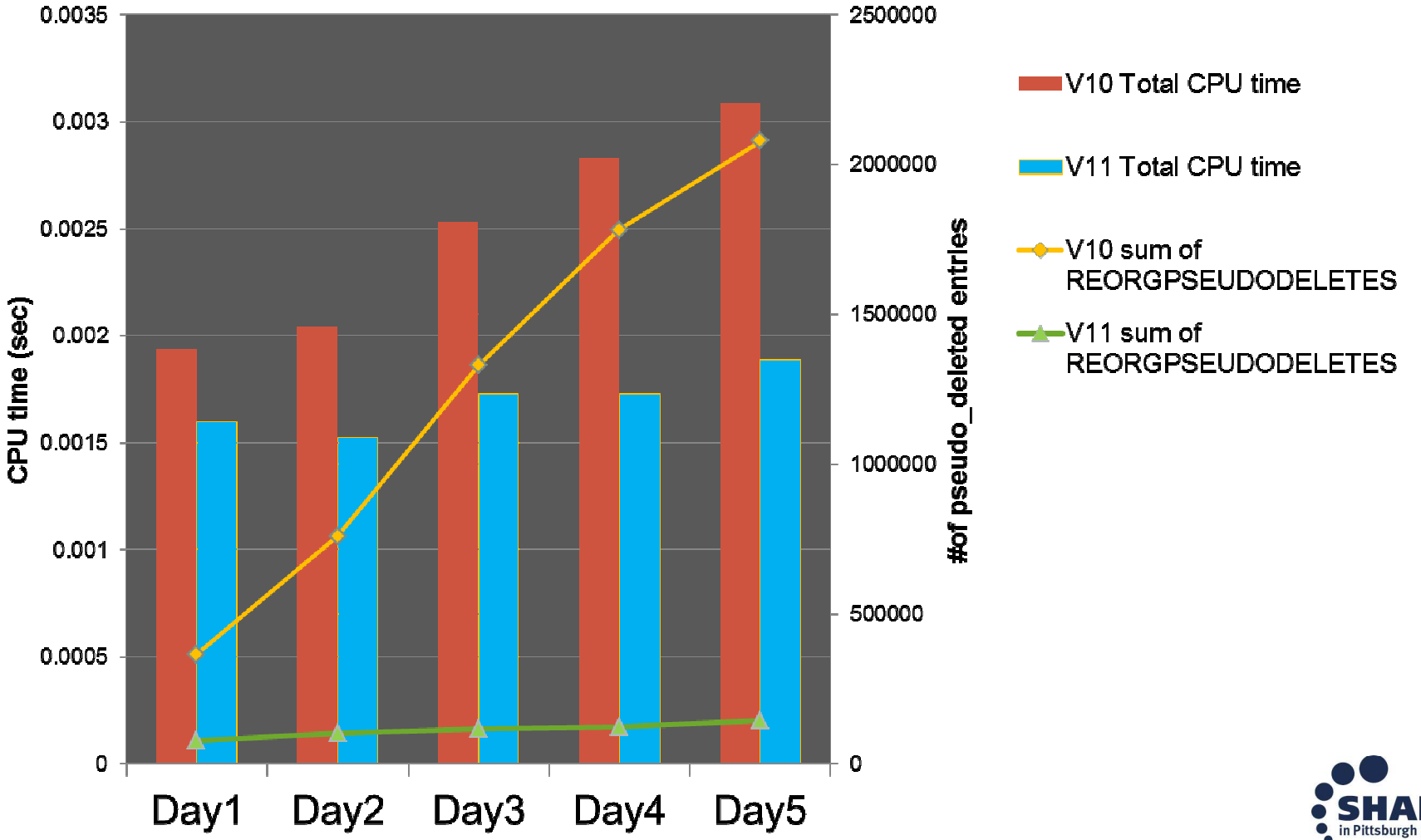
Agenda

- DB2 11 Performance Quick Look
  - ✓ Overview of DB2 11 Performance and Lab. measurements
- DB2 11 Performance and Expectation
  - ✓ System Level Performance Improvement
    - ✓General optimization
    - ✓System z Synergy
    - ✓Migration performance
  - ✓ DDF Performance (covered by DDF performance session)
  - ✓ Query Performance (details are covered by Query performance session)
  - ✓ Application Level Performance
- **REORG Reduction and Consistent Performance**
- Summary

Cleanup of Index Pseudo Deleted Pages and Entries

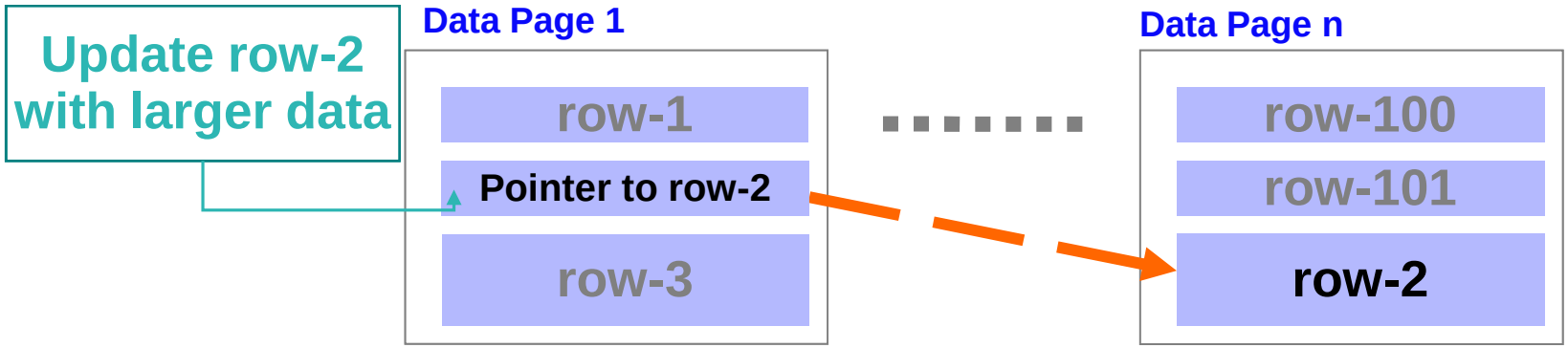
- **Performance impact of index pseudo delete entries**
  - SELECT, FETCH, UPDATE or DELETE through index search could see more getpages and lock requests
  - INSERT/UPDATE/DELETE may see concurrency issue
    - Collisions with committed pseudo-deleted index entries
    - RID reuse by INSERT following DELETE could cause deadlock
  - Frequent execution of REORG INDEX utility to reduce the impact
- **DB2 11 provides automatic cleanup of pseudo deleted entries**
  - Clean up both pseudo empty index pages and pseudo deleted index entries
    - Consistent performance for index access
    - Reducing the need of REORG INDEX utility
- **Potential concerns on automatic clean up**
  - CPU overhead - zIIP eligible under DBM1 address space
  - Disruption to other concurrent threads
    - Control through zparm INDEX_CLEANUP_THREADS (0-128 | 10)
    - Control through SYSIBM.SYSINDEXCLEANUP
  - Recommend to use the default unless you have concerns

Example : IBM WebSphere Portal Workload



Indirect Reference (Overflow Records)

- **Indirect References (overflow records)**
  - Created during UPDATE against variable length rows or compressed rows



- **Impact caused by indirect references**
  - Additional getpages, potentially additional I/Os to the overflow pages
  - Lower clustering
  - REORG TS is necessary to remove indirect references

How To Avoid Overflow ?

- Large enough PCTFREE value
  - Honored by LOAD/REORG
- MAXROWS
  - Honored by LOAD/REORG and Insert

Reduction of Indirect References (NFM)

- **Insert process to reserve the space for subsequent update**
  - CREATE/ALTER TABLESPACE PCTFREE x FOR UPDATE y

```
CREATE TABLESPACE TS1
FREEPAGE 0
PCTFREE 20 FOR UPDATE 10
```

- **PCTFREE x FOR UPDATE y**
  x = % of free space to leave in each data page by LOAD or REORG
  y = % of free space to leave in each data page by INSERT, LOAD or REORG
  INSERT will preserve y% while REORG will preserve (x+y) %
- **System parameter PCTFREE_UPD (PERCENT FREE FOR UPDATE)**
  - System default 0 for FOR UPDATE value when it is not specified in DDL
  - If not specified, same behavior as DB2 10
- **Autonomic option FOR UPDATE -1 or PCTFREE_UPD -1 and AUTO**
  - With AUTO, DB2 uses real-time statistics values to automatically calculate the percentage of free space that is to be used by update operations. This value is equivalent to specifying PCTFREE FOR UPDATE -1 in the CREATE TABLESPACE or ALTER TABLESPACE statement.
  - Recommendation : Use FOR UPDATE -1 unless you know better

Agenda

- DB2 11 Performance Quick Look
  - ✓ Overview of DB2 11 Performance and Lab. measurements
- DB2 11 Performance and Expectation
  - ✓ System Level Performance Improvement
    - ✓General optimization
    - ✓System z Synergy
    - ✓Migration performance
  - ✓ DDF Performance (covered by DDF performance session)
  - ✓ Query Performance (details are covered by Query performance session)
  - ✓ Application Level Performance
- REORG Reduction and Consistent Performance
- **Summary**

DB2 11 Performance Sweet Spots

- **Write Intensive Batch**
  - Both data sharing and non-data sharing, but especially in data sharing configuration with extended LRSN
- **Online transactions**
  - Write intensive transactions
  - With large # of partitions (>200 partitions) with REL(COMMIT)
  - With large buffer pools
  - With queries returning a large number of columns
  - Chatty DDF transactions
  - With large delete operation (pseudo deleted entries)
- **Queries**
  - With compressed tables
  - With access path improvement
  - Sort intensive workload
  - Accessing multiple partitions through DPSI
  - IDAA with large result sets
- **Cost saving from zIIP eligible address space SRB time**
  - DBM1 in data sharing
  - MSTR address space for update intensive workloads

ESP Performance Evaluations

- Summary of the data analysis with comparable workloads and after rebind for static workloads :
  - DRDA workload 0 to 20% CPU reduction
  - CICS workload 3 to 18% CPU reduction
    - ✓ 18% includes 4% CPU reduction with NFM extended RBA/LRSN
  - Batch workload 3 to 20% CPU reduction
  - No obvious real/virtual storage increase
- No major performance issues reported
  - Minor performance issues found so far
    - 2 customers concern on RBA conversion on directory
      - The elongated conversion is due REORG of large LOBs on SPT01. Catalog/directory conversion can be done outside of migration process. Larger buffer pools for BP0 can reduce the elapsed time.
    - 1 customer exposed OQCR/DB2 issue – under investigation
    - 1 customer exposed Access path issue
    - 1 customer raised the concern on GBP write around - investigating
    - 2 customer exposed monitoring issues – resolved

Summary - DB2 11 Performance Focus

- **CPU / cost reduction and synergy with system z**
  - ✓ Internal optimization with more “procedures”
  - ✓ Decompression improvement
  - ✓ Various query performance improvement
  - ✓ More zIIP exploitations
- **Scalability enhancements**
  - ✓ Reduce the cost of large partitions
  - ✓ Latch contention reduction
  - ✓ LRSN spin avoidance
  - ✓ Buffer Pool improvement
- **Focus on customers’ pain points**
  - ✓ Log force write reduction
  - ✓ Pseudo delete clean up
  - ✓ Overflow avoidance
  - ✓ Release deallocate improvement
  - ✓ DGTT improvement


Thank
You