

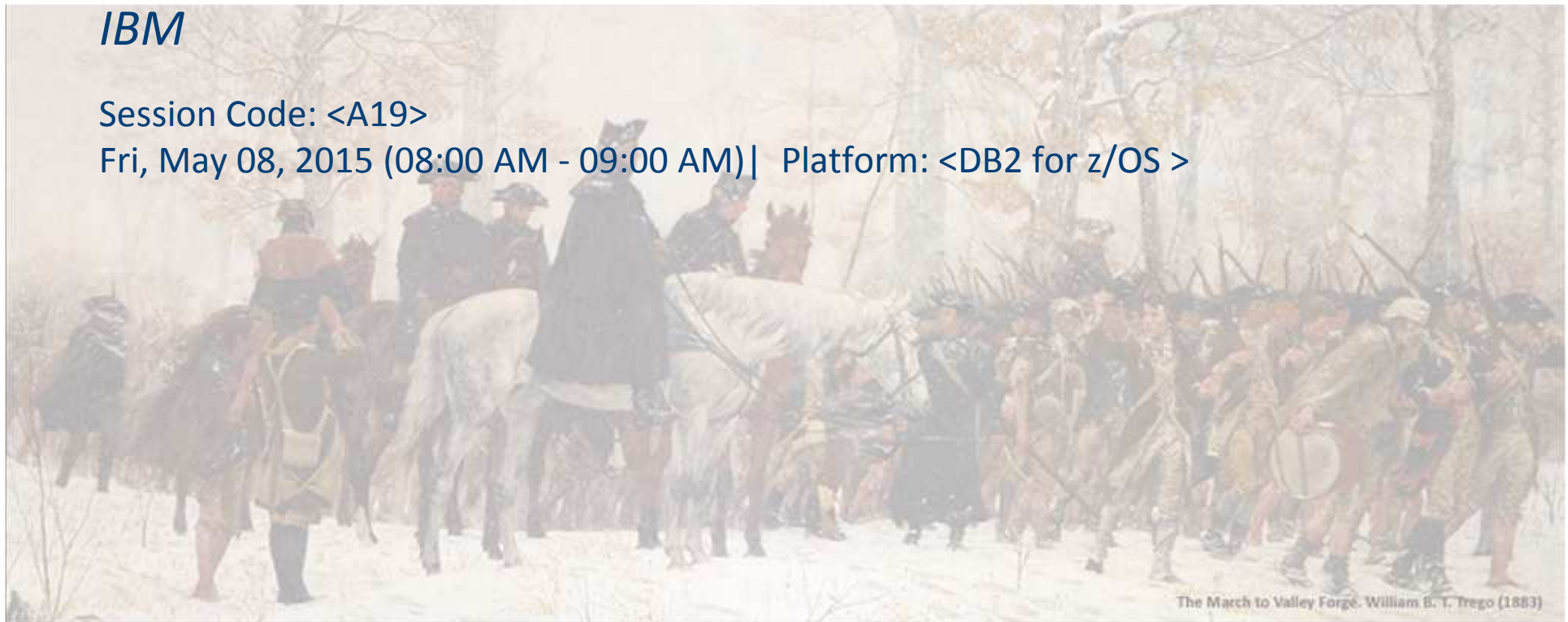
DB2 11 for z/OS : Performance Topics

Akiko Hoshikawa (akiko@us.ibm.com)

IBM

Session Code: <A19>

Fri, May 08, 2015 (08:00 AM - 09:00 AM) | Platform: <DB2 for z/OS >



Disclaimer

© Copyright IBM Corporation 2015. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

The information contained in this presentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided “as is” without warranty of any kind, express or implied. In addition, this information is based on IBM’s current product plans strategy, which are subject to change by IBM without notice. IBM shall not be responsible for damages arising out of the use of, or otherwise related to, this presentation or any other documentation. Nothing contain in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of any agreement or license governing the use of IBM products and/or software.

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.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the pages of the presentation:

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both: DB2, DB2 Connect, DB2 Extenders, Distributed Relational Database Architecture, DRDA, eServer, IBM, IMS, iSeries, MVS, z/OS, zSeries

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

DB2 for z/OS 11 : DB2 Performance Topics



System z Synergy – DB2 and IBM z13

System z Synergy – DB2 and Logging

System z Synergy - Large Memory

DB2 11 Performance - Migration performance

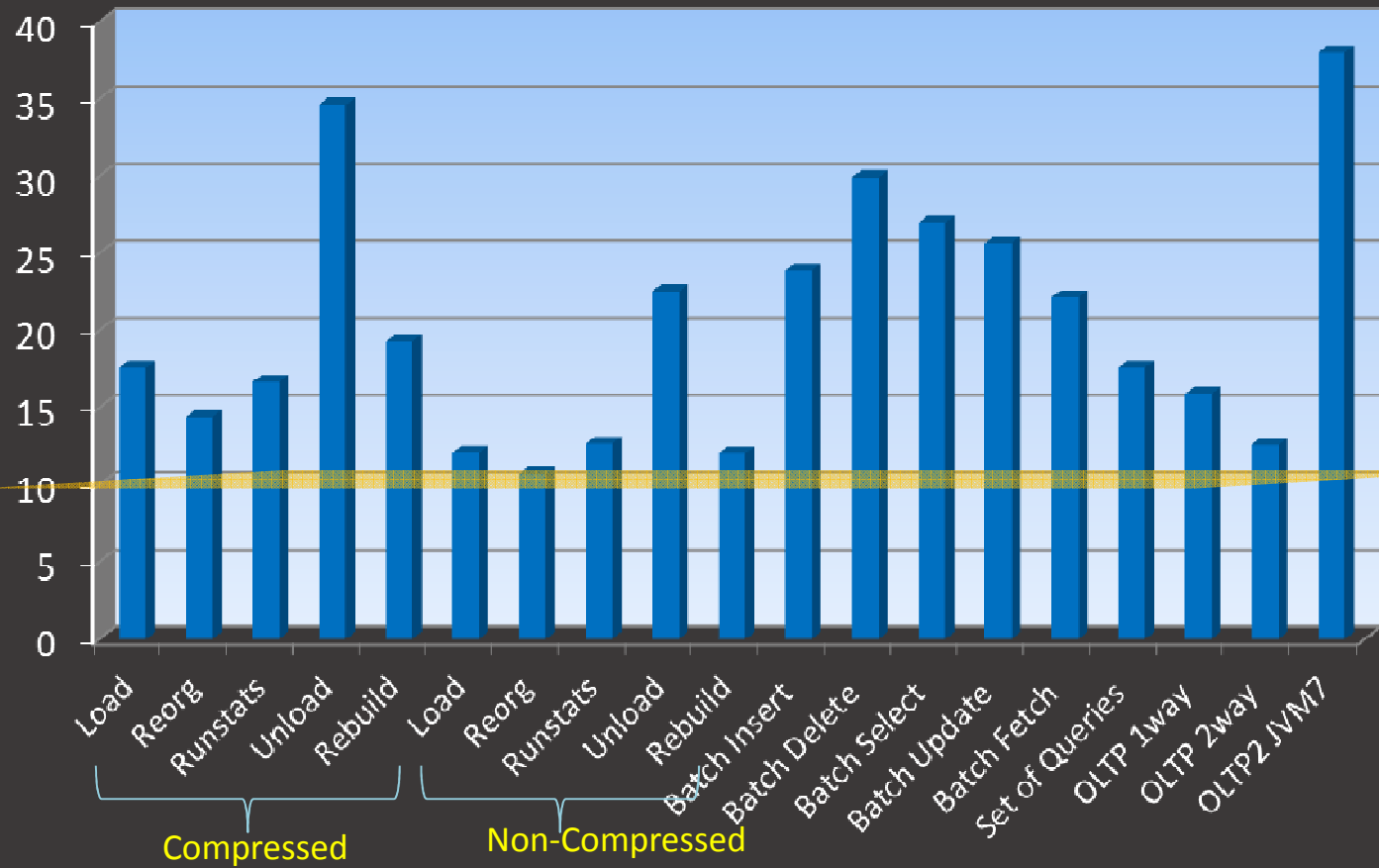
DB2 Cypress Performance

z13 and DB2

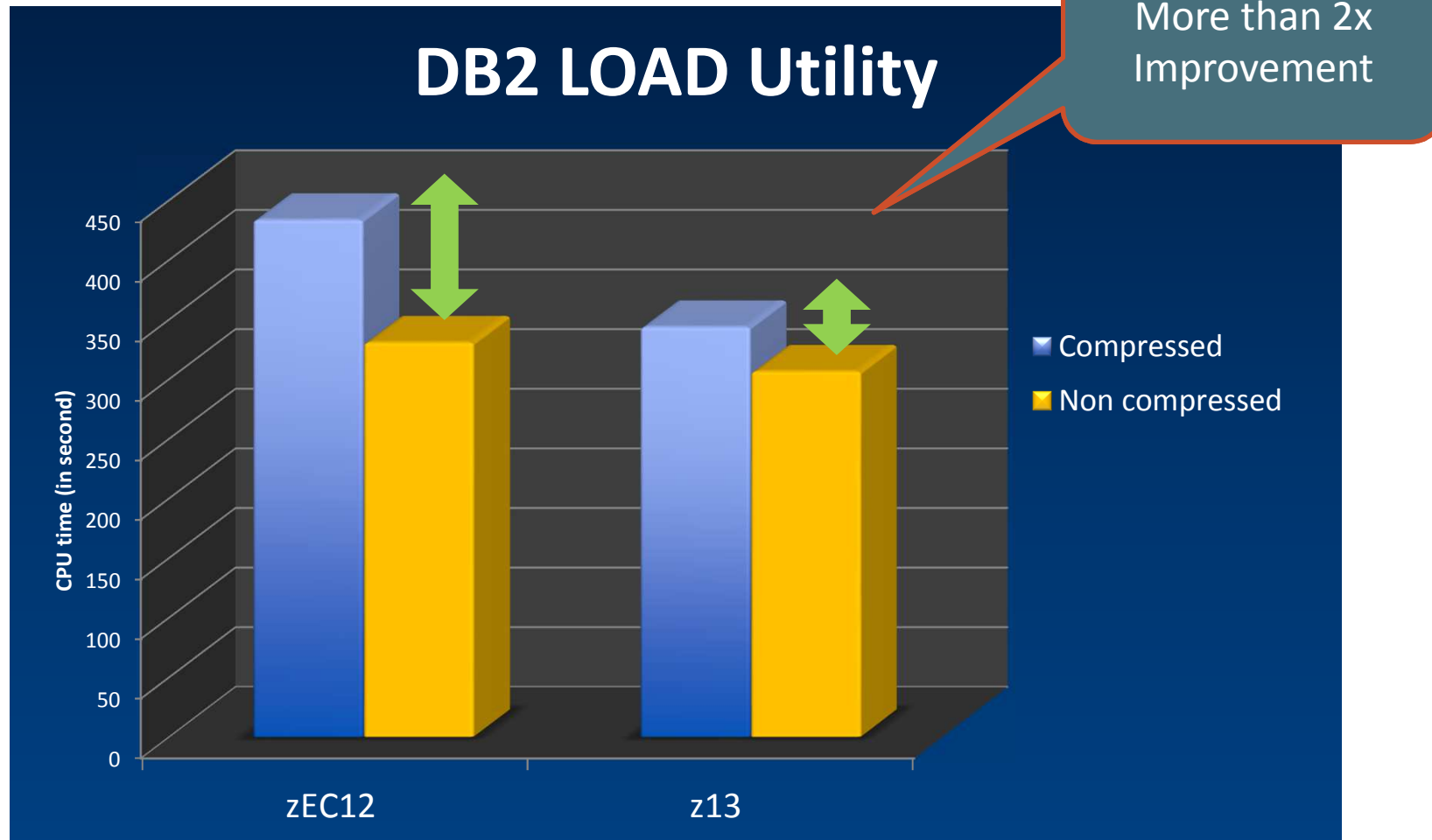


IBM z13 and DB2 z : Compete on Speed to Insight and Action

- DB2 workloads showing 4 to 38% range (mostly better than expected 10% improvement)
- 2.4x Reduction in compression cost in utility

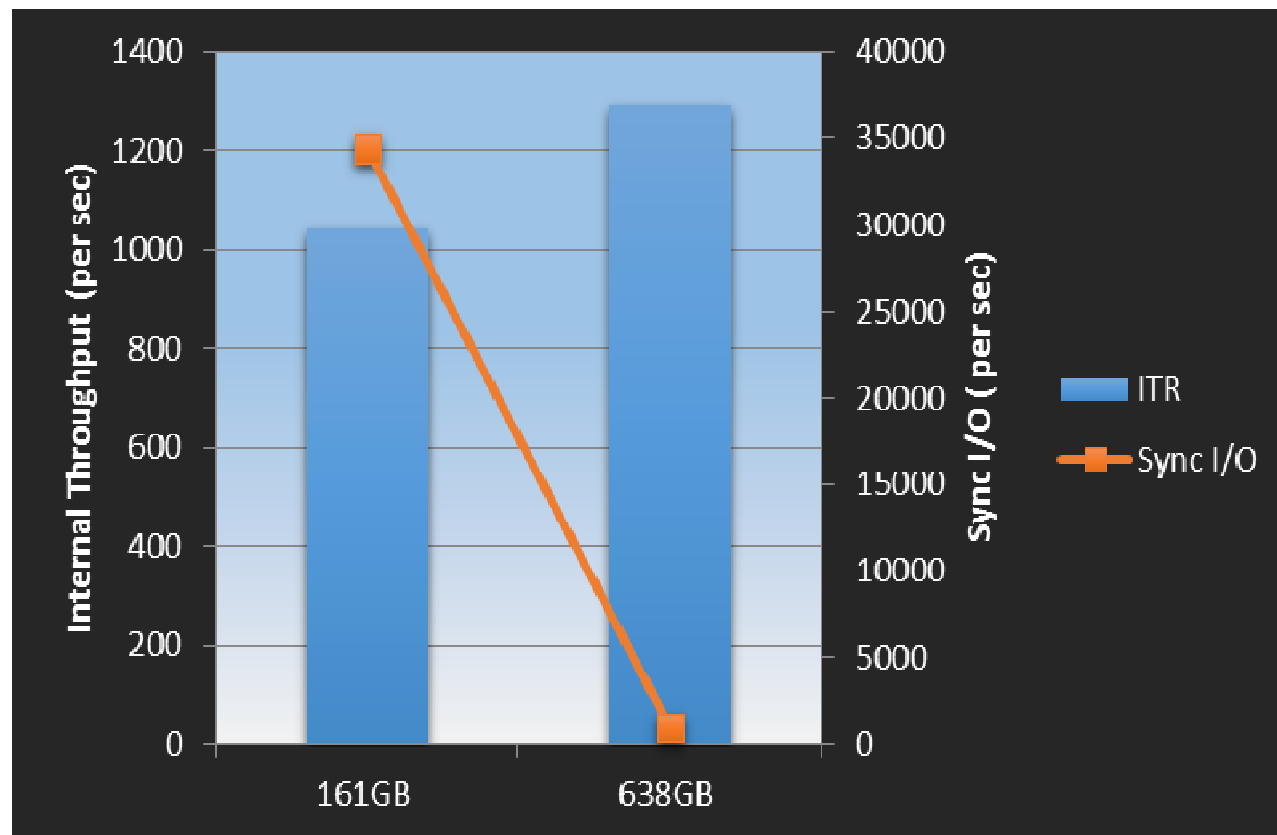


- An Impressive improvement on H/W Compression for DB2 compressed tables

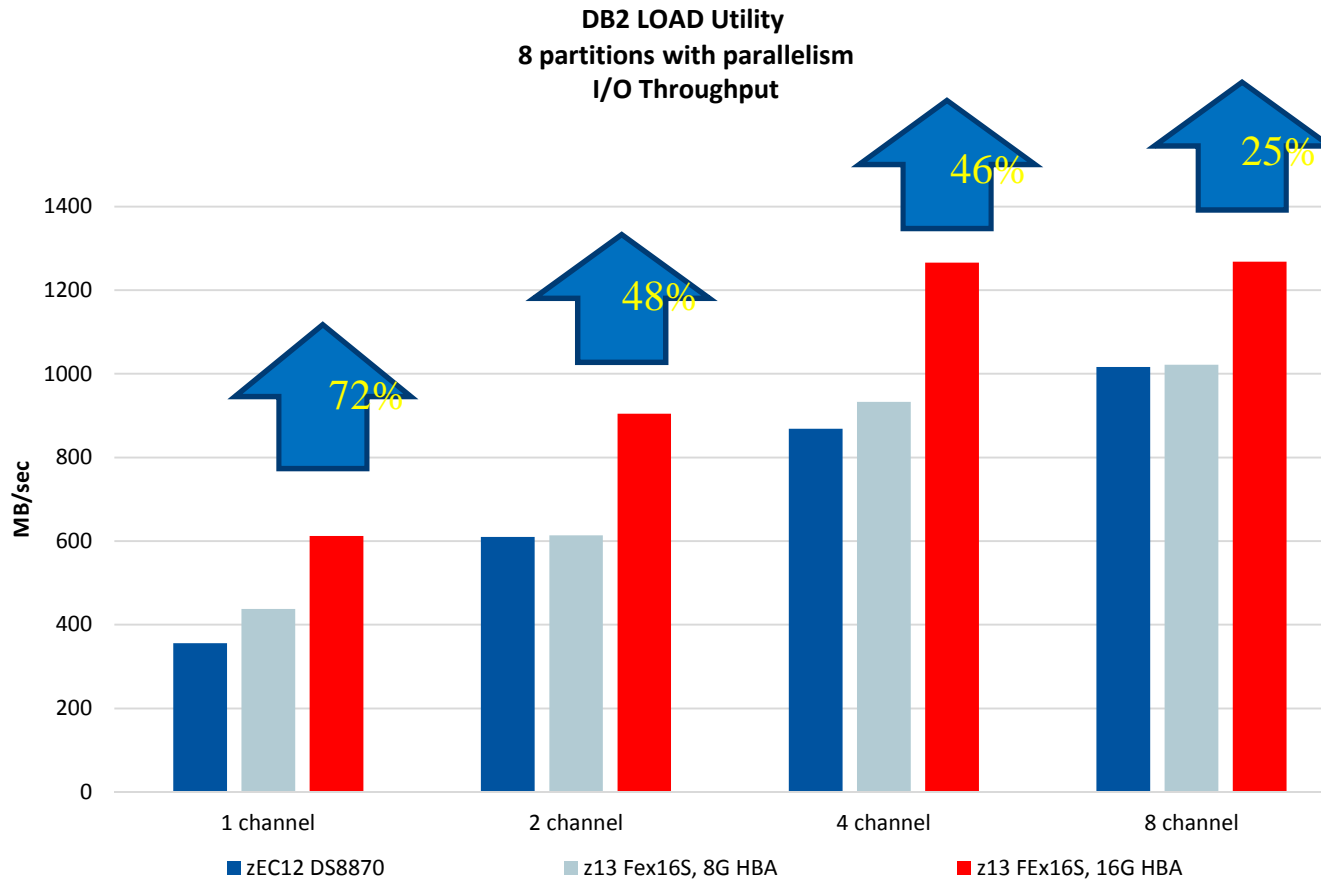


IBM z13 and DB2 z : Up to 4TB Memory

- DB2 workloads can take advantage of larger memory in z13
- Example : Banking transaction workload showing 24% throughout improvement by expanding DB2 buffer pools from 161GB to 638GB



DB2 I/O Throughput Improvement with FEx16S



- ❑ The more constrained the channels are, the more 16 Gb links help.
- ❑ Both the channels and the HBA need to be 16 Gb to get much benefit.

DB2 and zEC12 , z13

- **RoCE (RDMA over Converged Ethernet)**
 - Communications protocol based on Infiniband
 - Available with zEC12 GA2
 - Initial support z/OS to z/OS connection
 - Transparent from DB2
 - Support **up to 300m to 10Gb switch**
 - DDF transaction (DB2z-DB2z) show up to 2x throughput improvement
- **zEDC Compression**
 - Reduction of SMF records (DB2 traces)
 - Transparent from DB2
- **1MB pageable with FLASH Express**
 - DB2 10 (APAR PM85944) and DB2 11
- **1MB and 2GB fixed Pages**
 - Requires LFAREA allocation and DB2 11
 - DB2 11 buffer pools support FRAMESIZE(4K, 1M, 2G)





IDUG DB2 North America Tech Conference
Philadelphia, Pennsylvania | May 2015

 #IDUGDB2

DB2 Logging

Improving Log Write Response Time

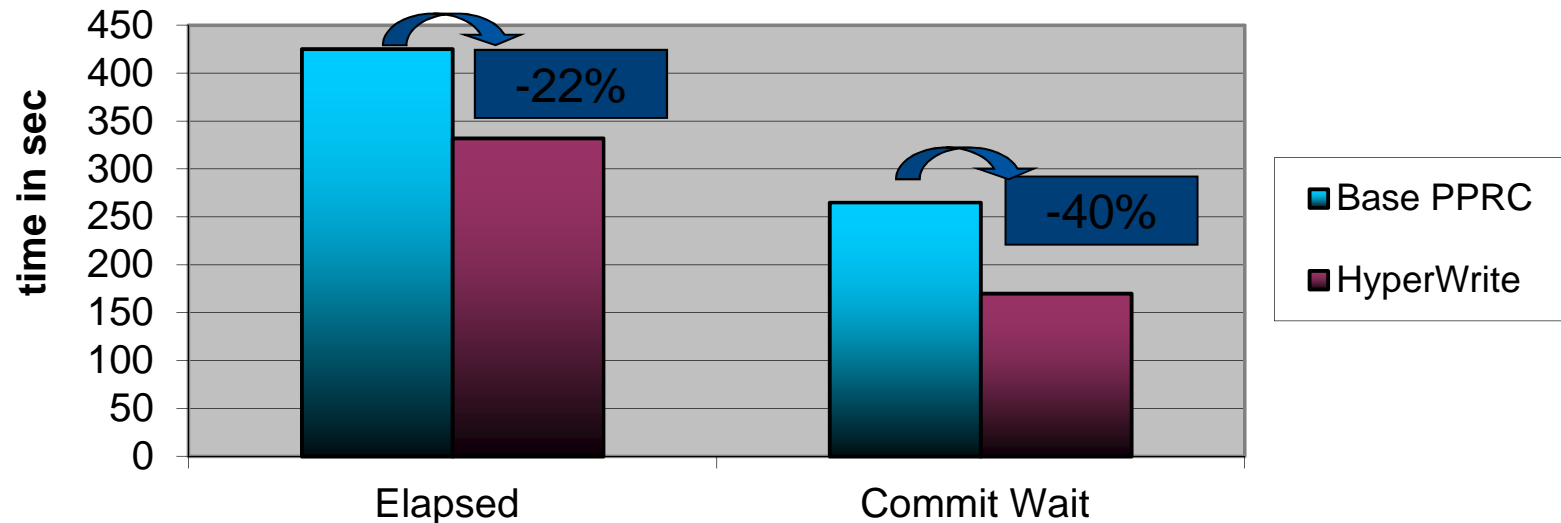
zHyperWrite function for DB2, z/OS and DS8870 with
GDPS or TPC-R HyperSwap

- Designed to help accelerate DB2 Log Writes
 - Improved DB2 transactional latency
 - Log throughput improvement
 - Improved resilience for workload spikes
 - Potential cost savings from workload consolidation
- Response time reduced up to 58%
 - Benefit percentage varies with distance
- Requires,
 - z/OS APAR OA45662+, DFSMS APARs
 - DB2 10 and DB2 11 SPE APAR PI25747
 - Set ZPARM (REMOTE_COPY_SW_ACCEL)
 - **IBM DS8870 Storage Subsystem MCL**
 - R7.4 87.4x.x.x - DS8870 242x model 961



DB2 Elapsed time Reduction with zHyperWrite

HyperWrite Performance in Data Sharing / Multi Threads

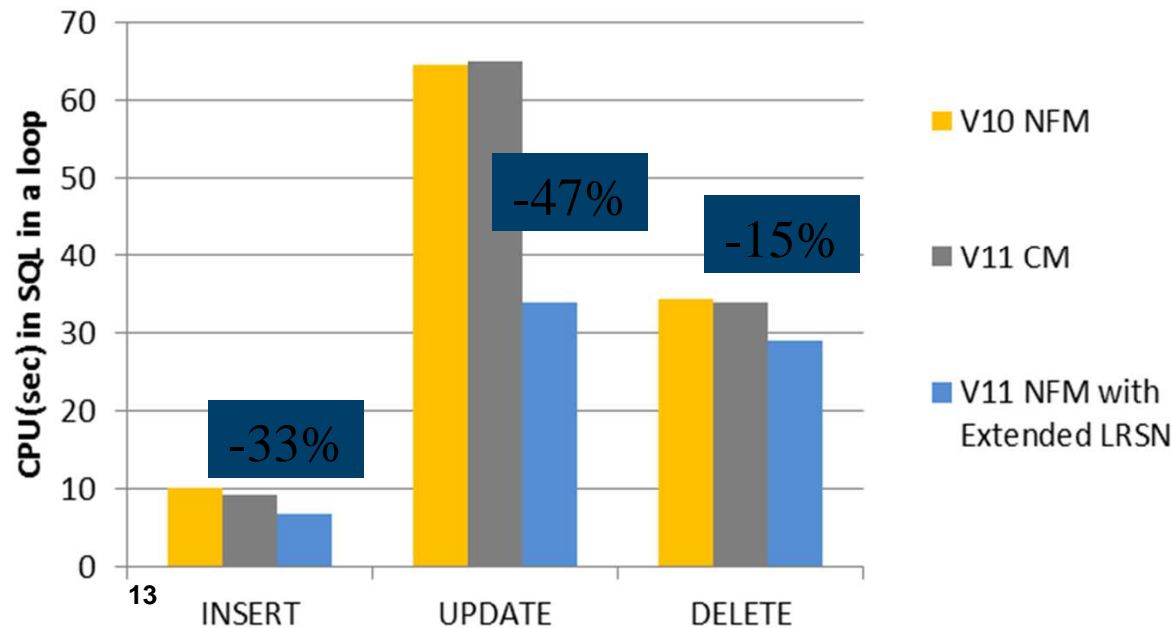


- DB2 concurrent insert batch jobs (commit every 10 insert) in data sharing shows 40% reduction in commit response time and 22% elapsed time reduction
 - Additional CPU in master SRB for triggering additional I/O request
 - DB2 11 zIIP eligible

Eliminate LRSN Spins with Extended LRSN

- Faster processors speed can cause more LRSN duplicates and spins
- Eliminate spin once conversion to extended LRSN (10bytes log records) is done in DB2 11 NFM
 - LRSN avoidance requires both BSDS and objects conversion
 - Recommend to convert BSDS first, then REORG the objects as needed
 - DB2 internally produces 10 bytes logs and convert to 6 bytes in basic log format

CPU time in Seq. Update/Insert/Delete

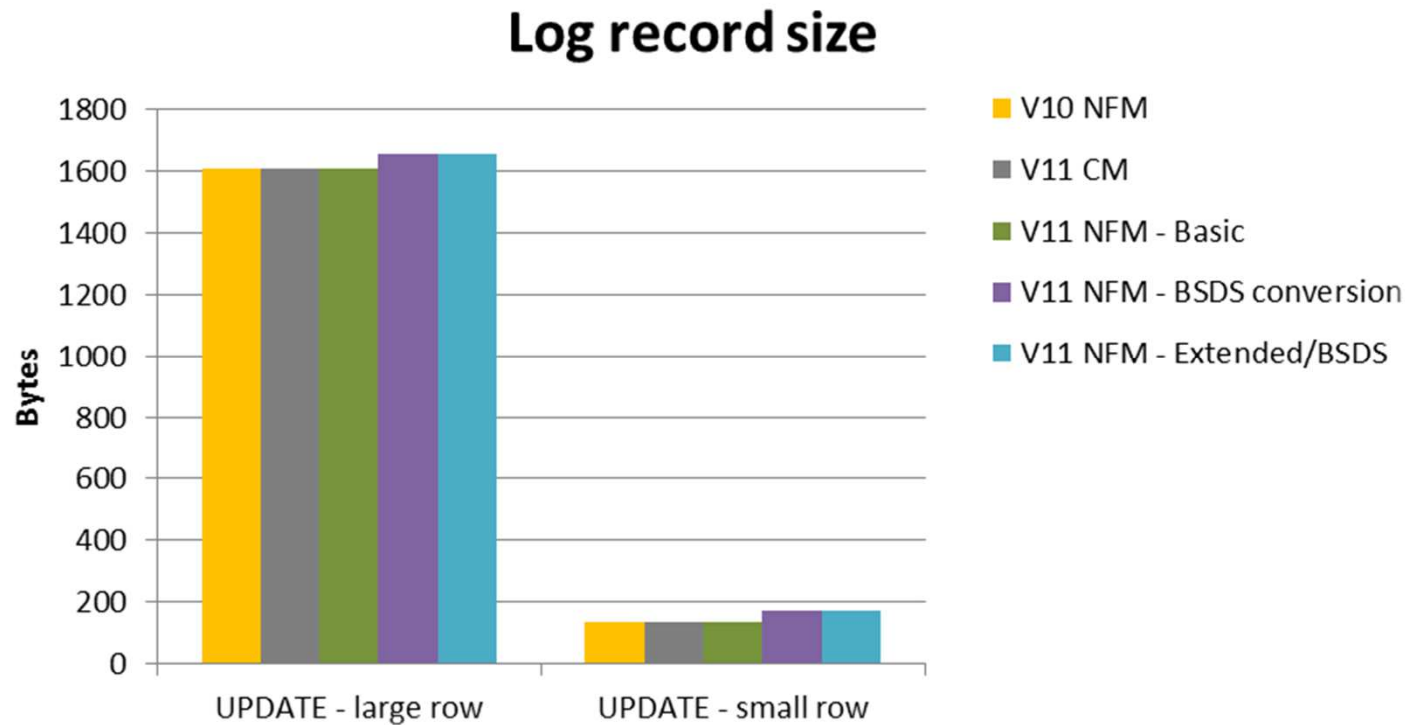


Notes:
LRSN spin can be observed in the service fields in statistics

- QJSTSPNN
- QJSTSPNI

Log Record Size Increase with Extended LRSN/RBA

- Recommend to monitor log I/O performance due to log record size increase
 - 3% to 40% log record size growth observed after BSDS conversion



DB2 for z/OS 11 : DB2 Performance Topics



System z Synergy – DB2 and IBM z13

System z Synergy – DB2 and Logging

System z Synergy - Large Memory

DB2 11 Performance - Migration performance

DB2 Cypress Performance

DB2 and Large Memory

“Memory is cheap or one time charge, CPUs are expensive”
“For every I/O that you save, you avoid the software charge for the CPU that it took to otherwise do that I/O”



Potential DB2 Benefit from Larger Memory

- **DB2 local and group buffer pools**
 - Reduction of elapsed time and CPU time by avoiding I/Os
 - PGSTEAL(NONE) in DB2 10 = In memory data base
 - CPU reduction from PGFIX=YES and large page frames
- **Thread reuse with IMS or CICS applications**
 - Reduction of CPU time by avoiding thread allocation and deallocation
- **Thread reuse and RELEASE(DEALLOCATE)**
 - Reduction of CPU time by avoiding package allocation and parent locks
 - DDF High performance DBATs support with DB2 10
 - Ability to break-in persistent thread with DB2 11
- **Global dynamic statement cache**
 - EDMSTMTC up to 4G with DB2 11, default 110MB
 - Reduction of CPU time by avoiding full prepare
- **Local statement cache**
 - MAXKEEPD up to 200K statements with DB2 11, default 5000
 - Reduction of CPU time by avoiding short prepare
- **In-memory data cache for sparse index**
 - MXDTCACH up to 512MB per thread, default 20MB
 - Reduction of CPU and elapsed time with potentially better access path selection with DB2 11

1MB Frames and LFAREA(IEASYSxx)

- Ideal at memory rich environment
- Identify the candidate buffer pools
 - DB2 10 : Existing PGFIX = YES pools
 - DB2 10 and 11 : Buffer pools with high getpage intensity
- Estimating LFAREA

LFAREA = 1.04* (sum of VPSIZE from candidate buffer pools) + 20MB
(+ OUTBUFF size for DB2 11)

- 20MB to accommodate z/OS usage
- 4% for 4K, 2% for 8K, 1% for 16K and 32K page size
- With Java using, additional java heap size needs to be considered
- Related z/OS APARs
 - APAR OA34024 – Documentation on how to select the right LFAREA size
 - Using the DISPLAY VIRTSTOR,LFAREA system command
 - APAR OA41968 – Fixed 1M pages were not be used to satisfy 4K page requests
 - Added the support for INCLUDE1MAFC in the LFAREA parameter (cause the system to take the available fixed 1M frames into account when making paging

How DB2 Steals Pages

- In-memory DBMS have existed for over a decade
- Concepts apply for both row and column store formats
- DB2 for z/OS incorporates extensive in-memory technology and operates almost exclusively on in-memory data

Keeps frequently accessed data in memory using PGSTEAL(LRU)

- Avoids disk I/O: > 90% of data accessed in memory without I/O
- Prefetch mechanisms avoid I/O waits
- Option to pin a table in memory

Writes all data changes (INSERT, UPDATE, DELETE) to memory

- Persistently writes log records to disk by commit time
→ Same behavior as In-Memory Databases

- **PGSTEAL(NONE)**
 - Buffer pool option for In-memory objects
- Extremely efficient memory usage across the cluster

How PGSTEAL(NONE) Works in DB2 10 and 11

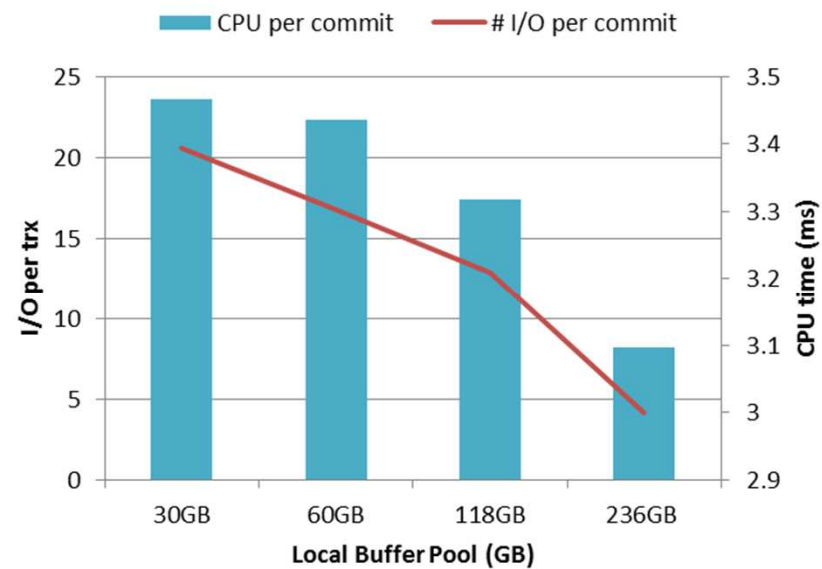
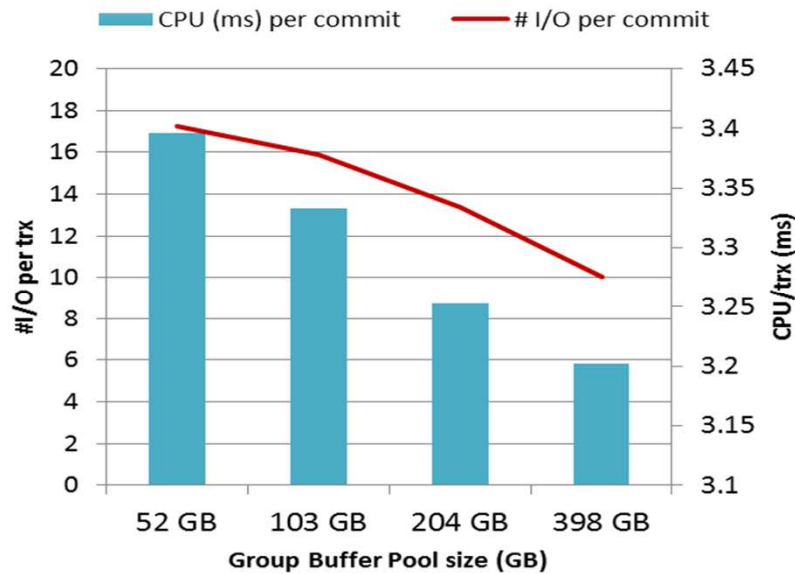
- Ideal **stable objects** which are frequently accessed and can be fit in buffer pools
- Benefit
 - Eliminate I/Os by keeping the objects in memory after first access
 - Reduce page stealing overhead (no maintenance for LRU chain)
 - Disable prefetch
- How it works
 - DB2 preloads the objects (TS, partition, index space) at the first access
 - If a page needs to be stolen, DB2 uses FIFO algorithm
- Recommendations
 - Use for performance sensitive frequently accessed objects without size increase (read only or in-place update)
 - CLOSE(NO)

Local Buffer Pools vs. Group Buffer Pools

- Observations:
 - Local buffer pool - read-only pages and changed pages
 - Group buffer pool with default GBPCACHE CHANGED - changed pages only
 - For the most of workloads, investing local buffer pools likely shows better benefit provided you have enough GBP allocated
 - Need to pay attention for GBP size especially directory reclaim when increasing local buffer pools
 - CPU benefit varies depending on the workload and stress level
- GBPCACHE (ALL)
 - Cache the read and changed pages
 - Less CPU saving if found in GBP instead of LBP
 - Preliminary measurements are looking promising but there is a trade-off

CPU Cost Saving by Reducing DB2 Sync I/Os

- Banking (60M account) workload with 2 way data sharing :
- 11 % response and 6 % CPU reduction from 52 GB GBP to 398 GB for both members with same LBP size (60GB)
- 40% response and 11% CPU reduction from 30GB LBP to 236GB LBP for both members with same reasonable GBP size (60GB)
- Measured 20 to 70 us CPU saving per sync I/O avoided



Benefit and Considerations for Larger Buffer Pools

- Benefit
 - Performance improvement (response time & CPU time) by avoiding I/Os, CF access for read
- Depends on **the size of active workload and access pattern**
 - No benefit with small working set with high hit rate
 - May not benefit with extremely large working set with moderate buffer pool size increase
 - Varies depending on the configuration - CPU utilization, data sharing, etc.
- Try & validate may not work well with customer's workload with high variations
- Available tool requires expensive set of traces and intensive analysis
- Considerations
 - Potential access path changes
 - For static, use plan stability
 - For dynamic, no good way other than using APS CPU modeling
 - Drive more CF structure usage
 - More pages to be registered (need enough directory entries)
 - Need to adjust buffer pool thresholds
 - Lower VPSEQT (Prefetch threshold)

Buffer Pool Simulation (DB2 11 PI22091)

- Simulation provides accurate benefit of increasing buffer pool size from production environment
- ALTER BUFFERPOOL now supports simulation pools
 - To simulate the case of doubling the current 20,000 buffer pools with simulated VPSEQT of 30

```
-ALTER BPOOL(BP1) VPSIZE(20000) SPSIZE(20000) SPSEQT (30)
```

- Simulation is against local buffer pools, not group buffer pools. It supports local buffer pools with GBP dependent objects
- Storage cost for a simulated buffer pool is less than 2% for 4K pages
 - SPSIZE(2000) or 7.8MB of simulated buffer pool will use about 156KB of storage.

Buffer Pool Simulation – Output

- Output from simulation is written in statistics traces and DISPLAY buffer pool output
- OMPE V520 APAR to format statistics (APAR PI28338)

DISPLAY BPOOL DETAIL shows,

```
DSNB431I  -CEA1 SIMULATED BUFFER POOL SIZE = 20000 BUFFERS -
           ALLOCATED           =      20000
           IN-USE               =      20000   HIGH IN-USE           =      20000
           SEQ-IN-USE           =       2229   HIGH SEQ-IN-USE        =       3684
DSNB432I  -CEA1 SIMULATED BUFFER POOL ACTIVITY -
           AVOIDABLE READ I/O -
           SYNC READ I/O (R) =365071
           SYNC READ I/O (S) =5983
           ASYNC READ I/O      =21911
           SYNC GBP READS (R) =89742
           SYNC GBP READS (S) =184
           ASYNC GBP READS     =279
           PAGES MOVED INTO SIMULATED BUFFER POOL =13610872
           TOTAL AVOIDABLE SYNC I/O DELAY =158014 MS
```

DB2 for z/OS 11 : DB2 Performance Topics



System z Synergy – DB2 and IBM z13

System z Synergy – DB2 and Logging

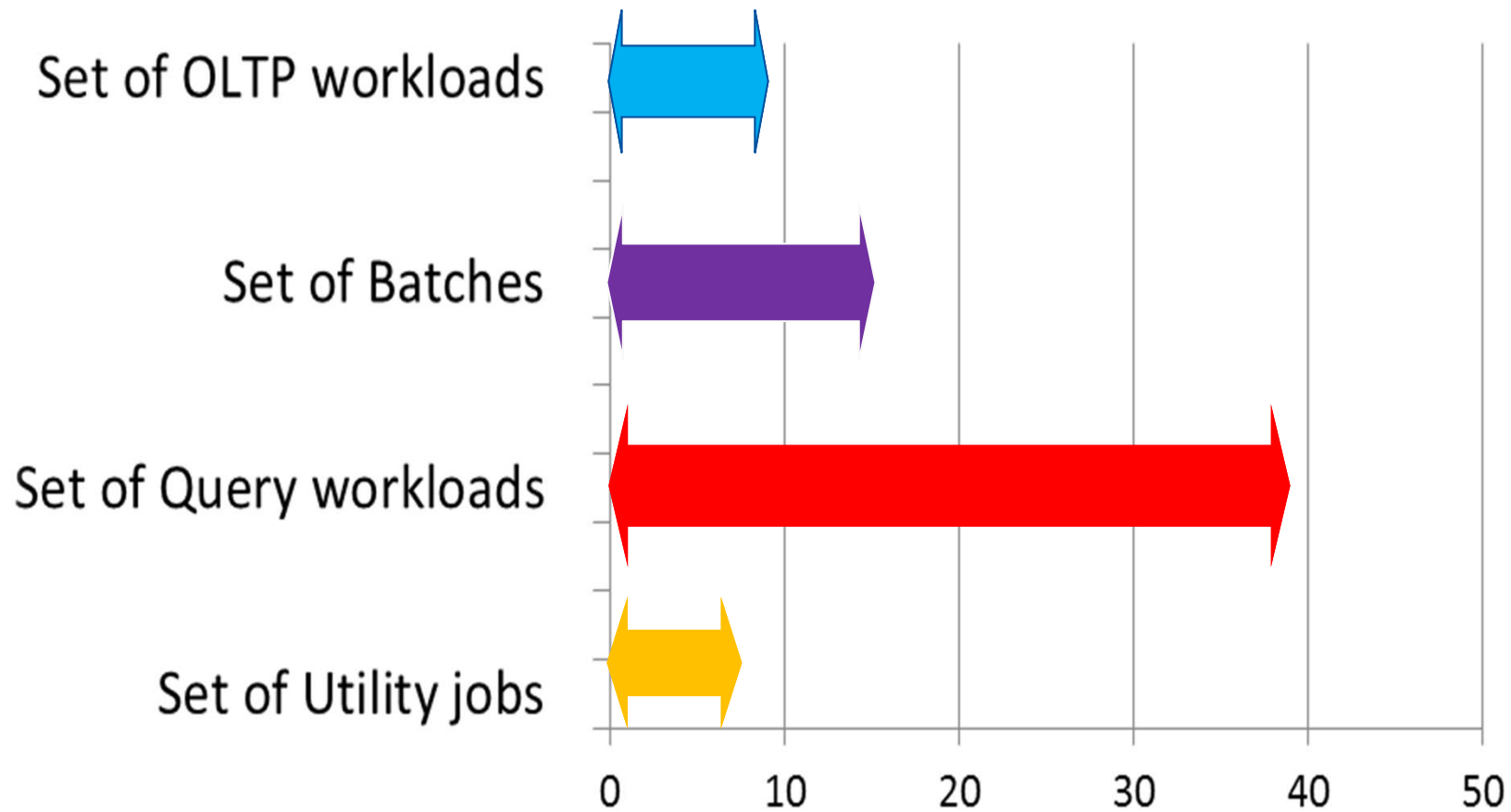
System z Synergy - Large Memory

DB2 11 Performance - Migration & General performance

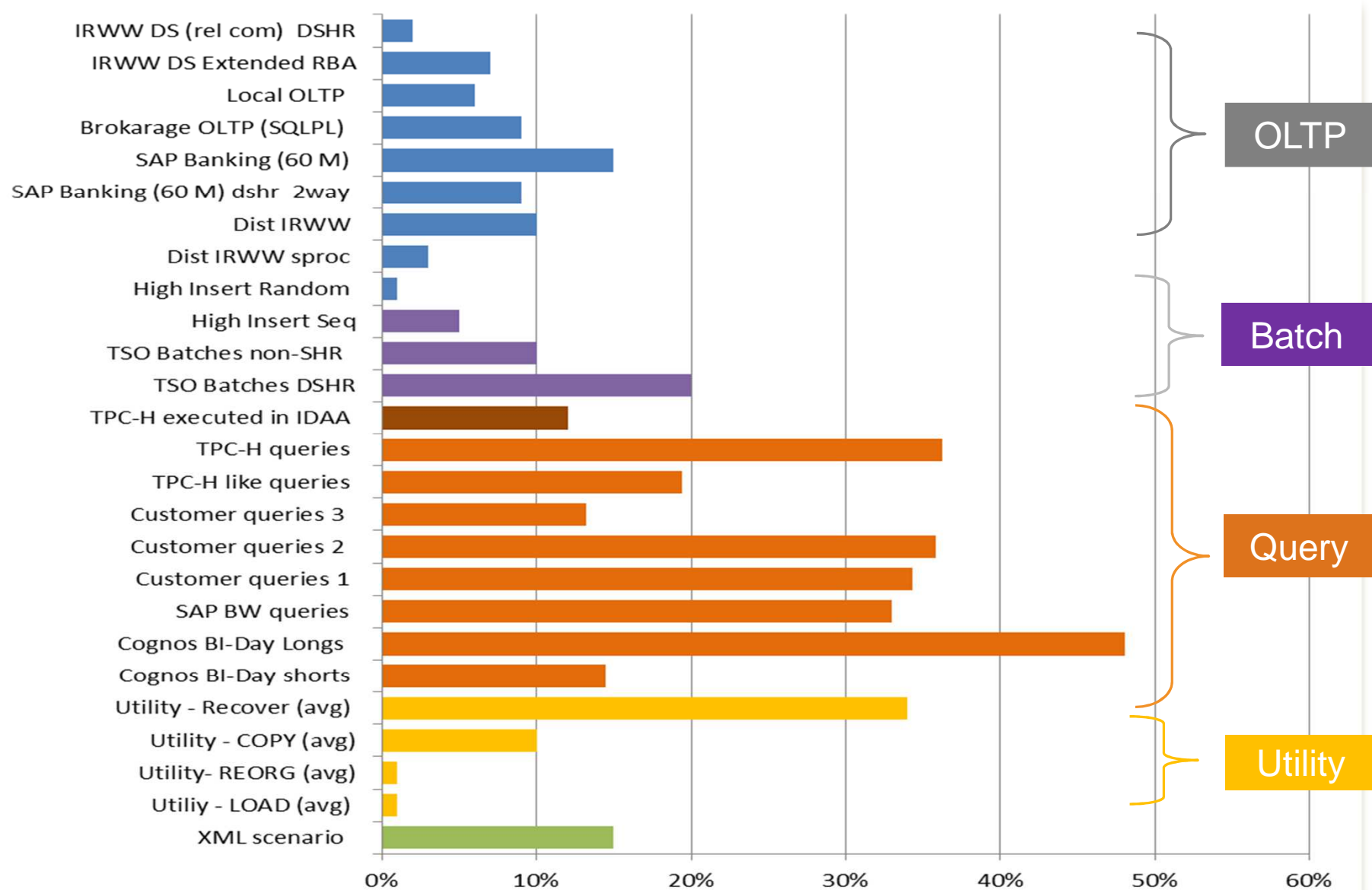
DB2 Cypress Performance

DB2 11 High Level Expectation

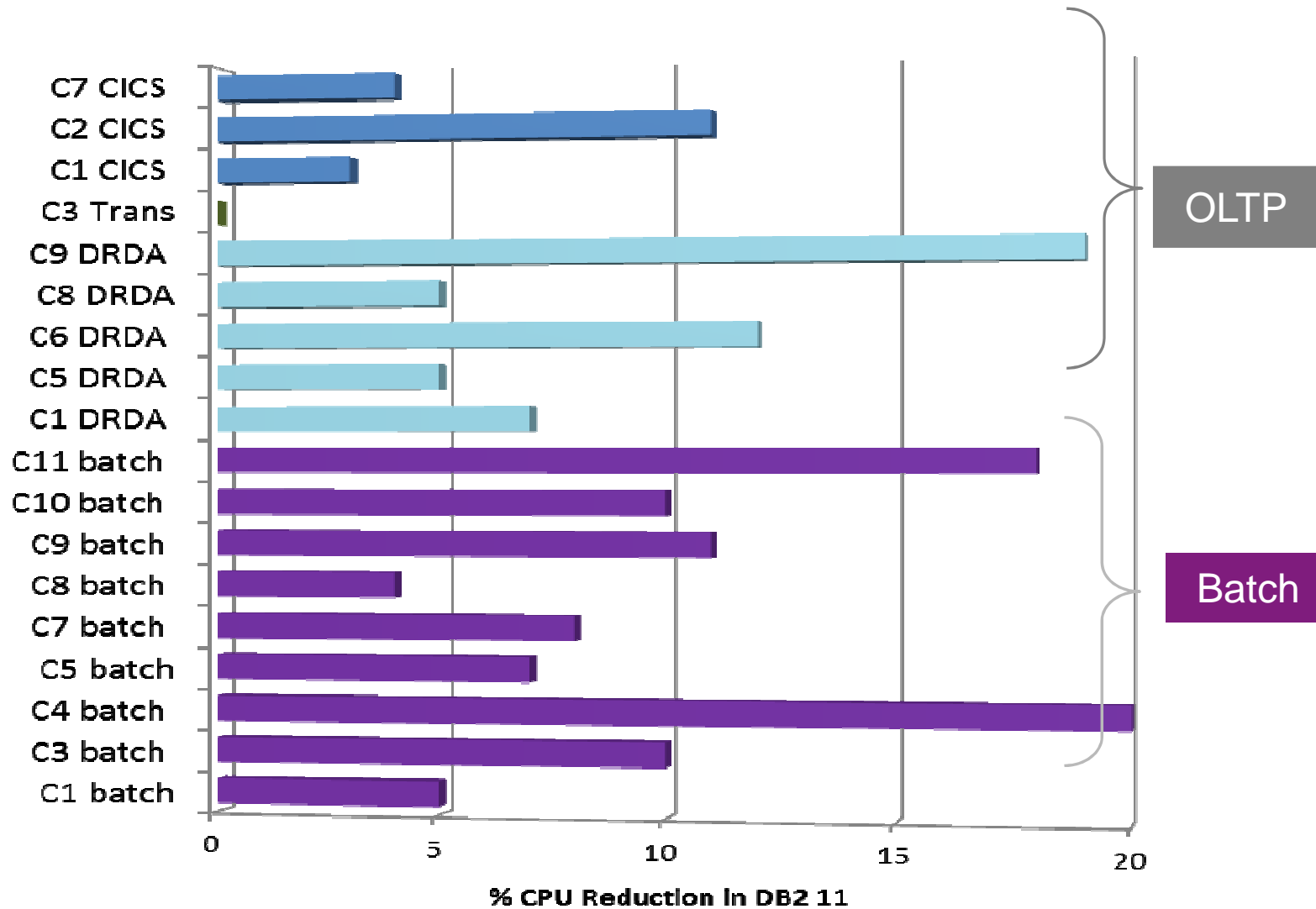
DB2 11 CPU % Typical Range



DB2 11 CPU% Improvement from DB2 10 (IBM)



DB2 11 CPU% Improvement - ESP Customers Results



DB2 11 Performance Sweet Spots

▪ **Batch**

- Write intensive batch (both data sharing and non-data sharing)
 - Potentially more in data sharing configuration with extended LRSN
- With compressed table spaces
- Returning large number of columns
- With a lot of sort

▪ **Online transactions**

- Write intensive transactions
- With large # of partitions (>200-300 partitions) with REL(COMMIT)
- With large buffer pools
- With queries returning a large number of columns (ex. SAP applications)
- Chatty DDF transactions
- With large delete operation (pseudo deleted entries)

▪ **Queries**

- With compressed tables
- With access path improvement
- With a lot of sort
- Accessing multiple partitions through DPSI
- IDAA with large result sets

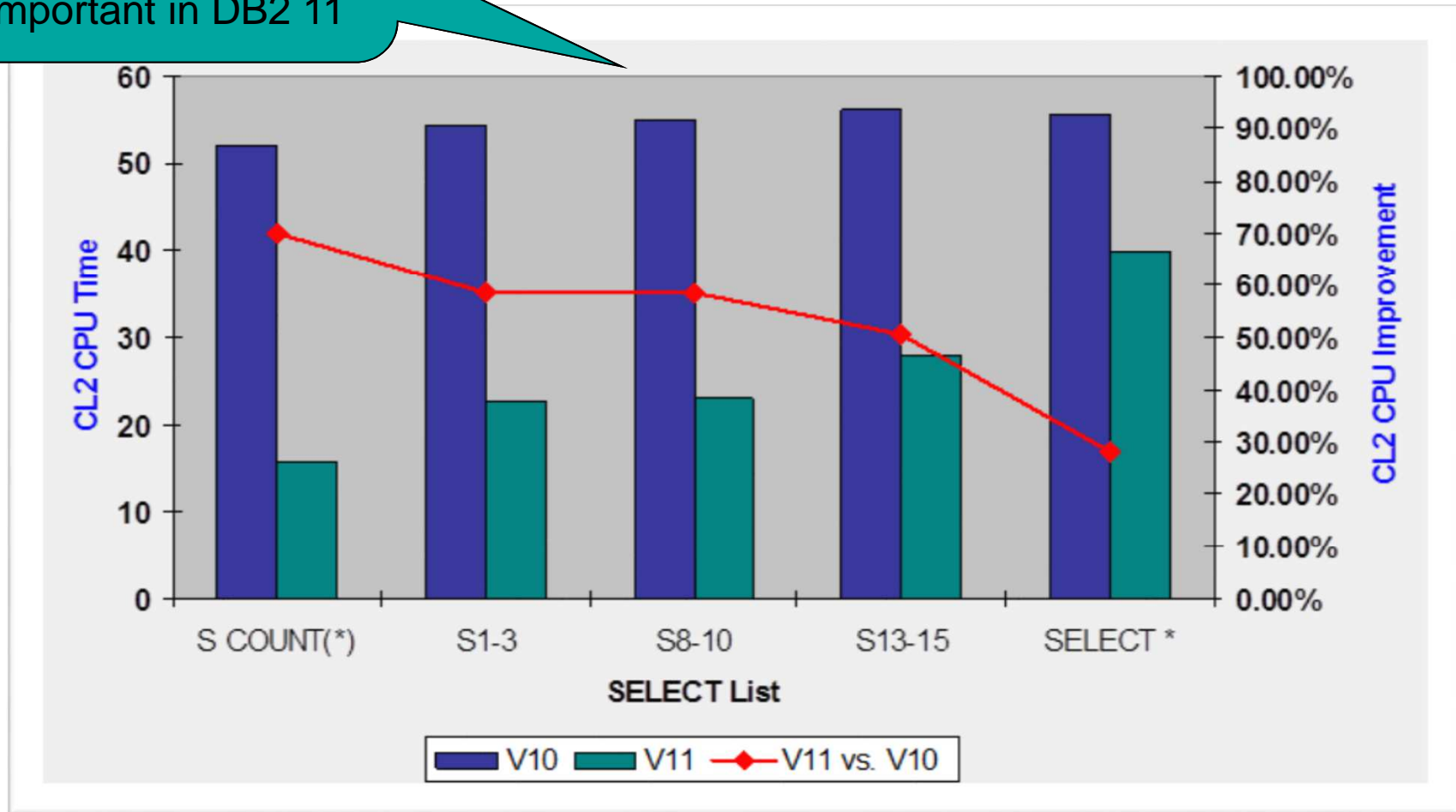
▪ **Cost saving from zIIP eligible address space SRB time**

- DBM1 in data sharing
- MSTR address space for update intensive workloads

Example from Decompression Improvement

Selecting only necessary columns becomes even more important in DB2 11

DB2 CPU time scanning compressed table space



Usability Enhancement with Release Deallocate

- **REL(DEALLOCATE) vs. REL(COMMIT)**

- Deallocate can avoid package allocation, and save CPU (5 to 20%)
 - 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 idol thread support
 - Note : Not all of persistent threads can be broken-in
- Tracks object related resource and lock accumulation and releases them as needed
 - Consistently good performance can be achieved with less memory footprint

Additional zIIP* Usage with DB2 11

- **Asynchronous enclave SRBs** (Service Request Blocks) under DB2 address spaces with the exception of P-lock negotiation processing. Such zIIP eligible processing includes:
 - Log write and log read (MSTR address space)
 - Castout processing (DBM1 address space)
 - Clean up of pseudo deleted index entries, XML multi version documents (DBM1 address space)

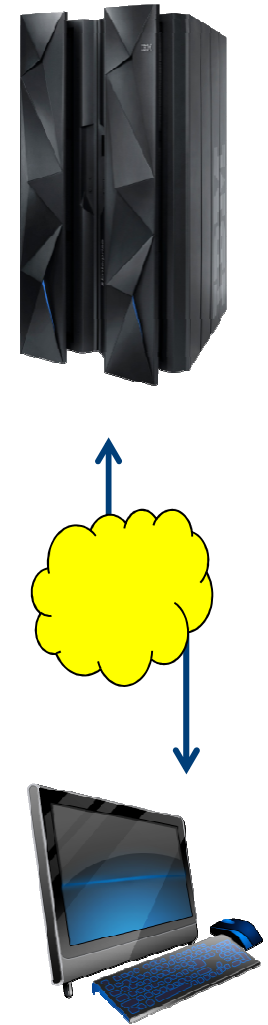
- **Utility**
 - Runstats - Inline statistics, Column group distribution statistics
 - LOAD - Index processing in LOAD REPLACE

Critical to have
enough zIIP capacity

* 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.

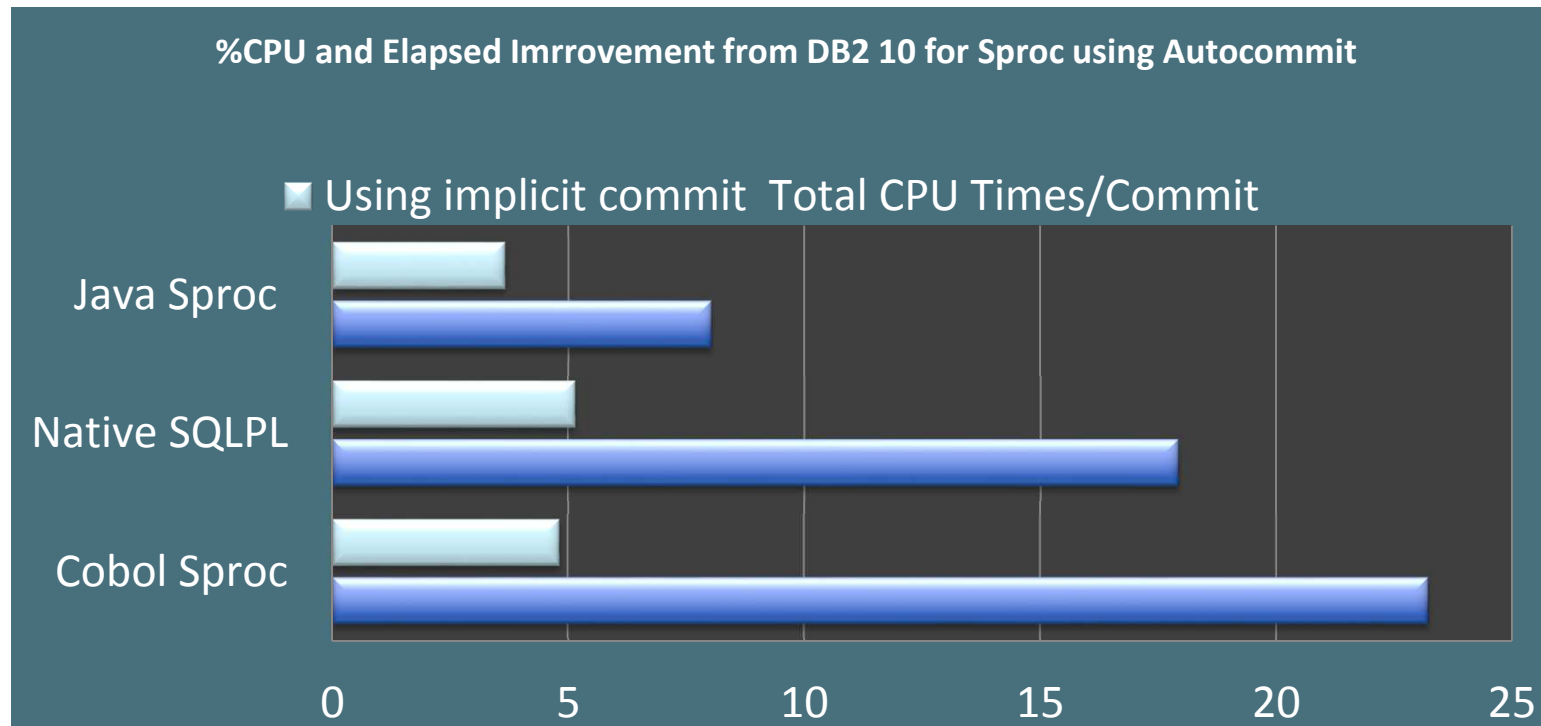
DDF Sync Receive Support

- Today DDF uses TCP/IP asynchronous receive
 - Async service requires expensive SRB switch but needed to detect connection termination
- DB2 11 DDF will exploit synchronous receive
 - z/OS Communications Server provides new Termination Notification via z/OS 1.13 APAR PM80004
 - DDF will be notified of any socket failure, even while in sync receive mode
- Benefit: Reduced network latency and CPU in DIST address space
 - Better benefit for large number of SQL statements (non-fetch) per transaction
 - No user action required
 - Preliminary measurements showed 3 to 7 % CPU reduction and throughput improvement in distributed transactions (average 35 SQL/commit)



Improved Stored Procedures w/Auto Commit

- DB2 11 provides new indicator that allows target server to initiate commit after result sets or cursors closed in initial reply by default
 - Requires support in DB2 V10.5 (or above) IBM Data Server Driver
- Benefits on simple OLTP with Autocommit = true
 - Improved network performance for applications using autocommit
 - Available in CM



Note : System Level Performance Features and Actions Required

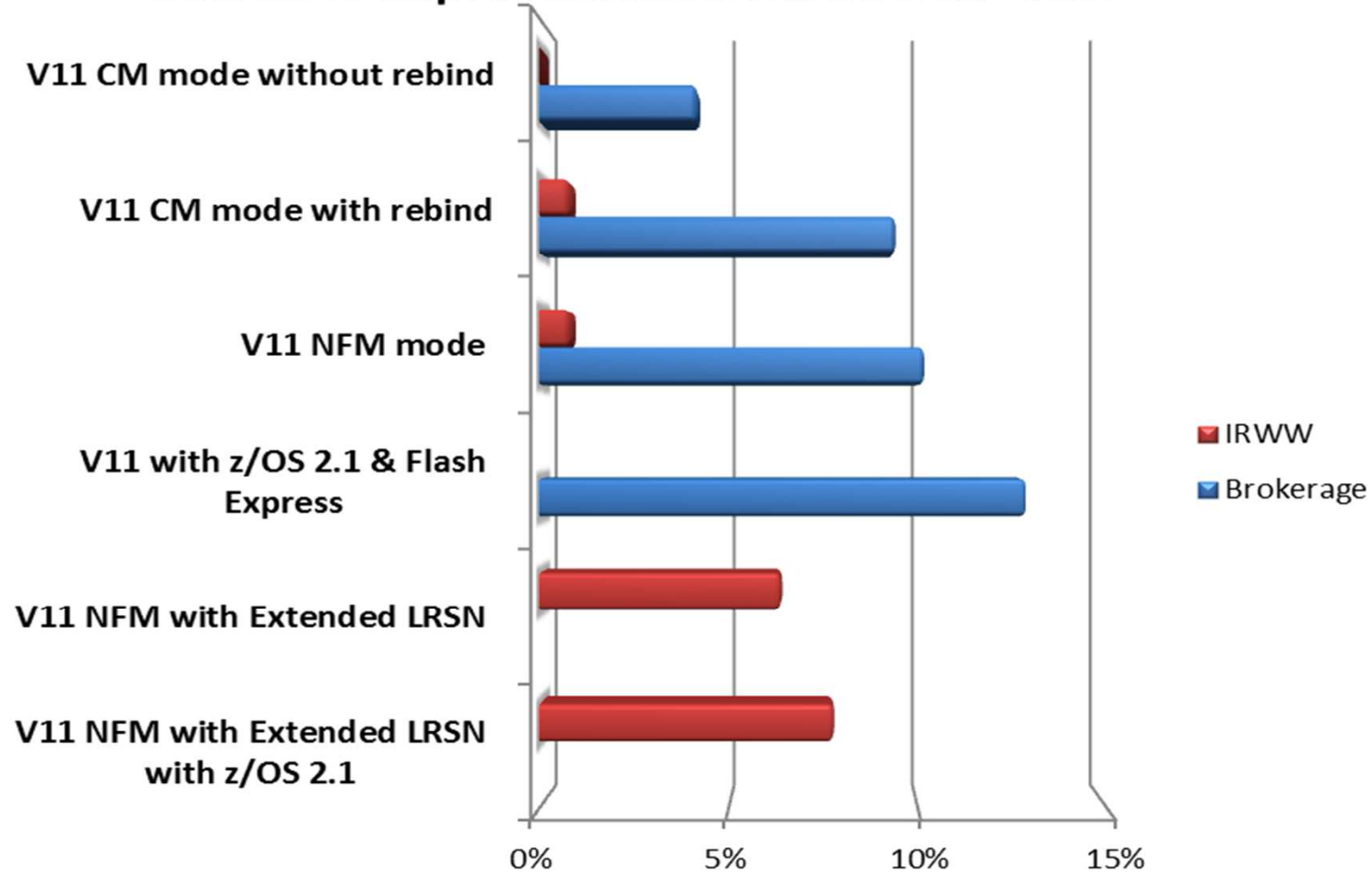
V11 System Performance Optimization/Features	Triggering action
Log Buffer in HVCOMMON	None
General Code optimizations	None
Customized code generation for sort	None
Access against the table with large number of parts	None
Rel(deallocate) optimization	None
DB2 latch reductions (LC14)	None
Buffer Pool enhancements	None
More zIIP processing	None
DDF Sync Receive enhancements	TCP/IP APAR TCP/IP APAR PM80004 / UK92097
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)
LRSN spin avoidance	NFM and Extended RBA
1MB frames for DB2 code	z/OS 2.1 and Flash Express ₃₆

Migration Performance Notes:

- **Catalog Migration (DB2 10->11CM -> NFM) is faster**
 - up to 16x elapsed time improvement from DB2 9->10
- **Rebind is strongly recommended**
 - SELECT procedures are disabled for v9, v10 bound static packages until REBIND
 - 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 to enable many of DB2 11 features even with reused access path (APREUSE)
- **Log format conversion is recommended regardless RBA/LRSN situation**
- **Virtual/Real storage usage observation**
 - DBM1 31 bit virtual – slightly less than DB2 10
 - xProc storages are moved above the bar

Example from OLTP measurements

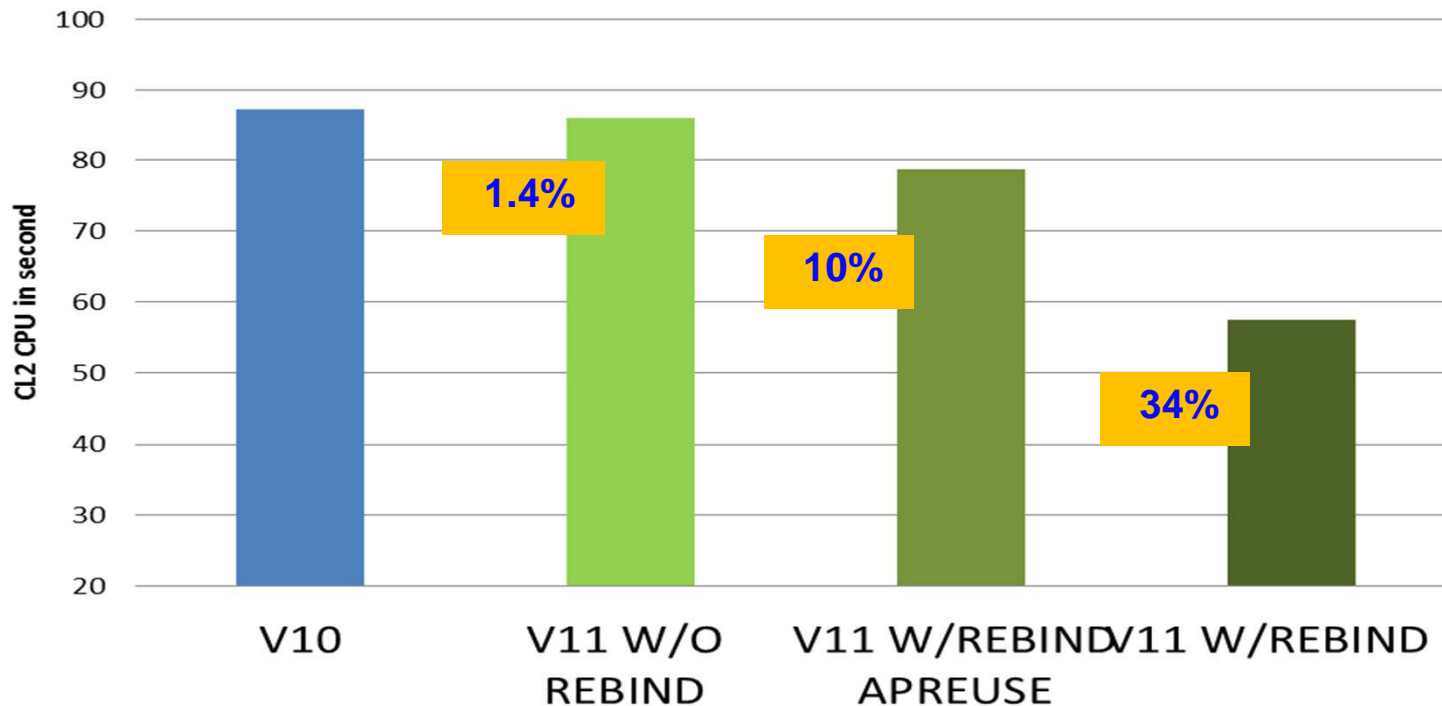
DB2 11 % Improvement from DB2 10 - OLTP



Example : Static Query Workload Migration

- Example of impact of REBIND and APREUSE

Static Queries Avg. CL2 CPU Time



Notes: Key Enhancements in Query Area

Note1 Items in “green” assumes REBIND with access path change (not using APREUSE)
Note2 More details to be covered by T. Purcell’s session

- Reduced CPU for sequential access, especially for ACCESS_TYPE='R'
- Reduced CPU for processing compressed data pages
- Reduced CPU for processing ORDER BY, GROUP BY sort
- More use of in memory work files
- Avoidance of materializing final merge from top sort
- Reusing workfile with correlated subquery using SET functions
- 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
- 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

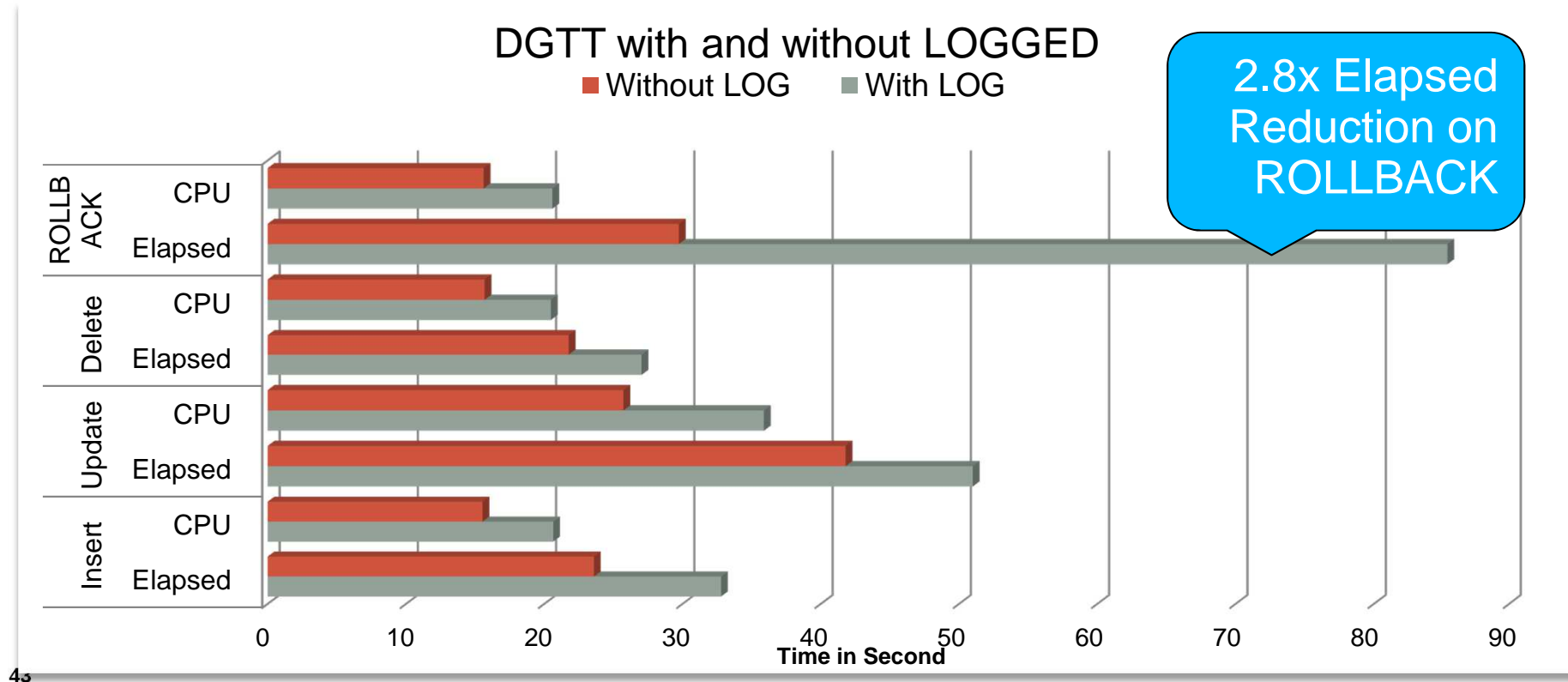
NOTE: Y* indicates that existing access path may be supported with APREUSE, but new and improved access path choices may also be available.

Real Storage Usage with DB2 11

- Real storage usage
 - Assumption: Same size of buffer pools are allocated between two releases
 - Various IBM workloads show a range of 0 to 15% real storage usage increase from 31 bit and 64 bit (% excluding buffer pools) going from V10 to V11
 - Highly workload dependent
 - SQL statement, access path, release option, commit frequency, concurrency, etc.
- Recommendation
 - Best to use own simulated workloads
 - Having head room for unexpected real storage growth on LPAR level is a good practice
 - Preparing for up to 15% (and with head room up to 30%) increase should be sufficient according to existing measurements and customer sample data

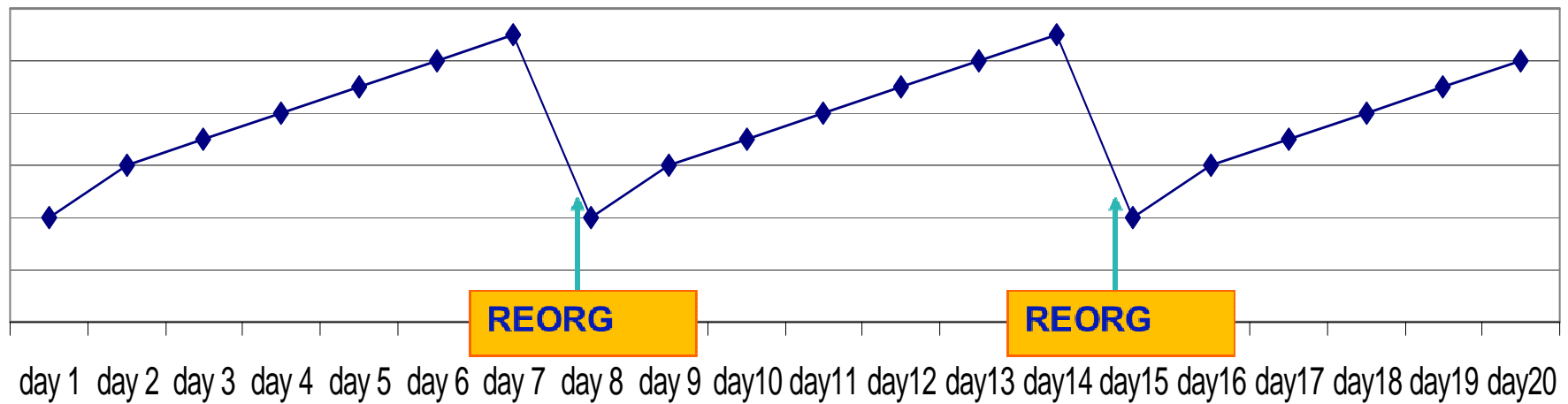
NOT LOGGED DGTTs - NFM

- 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



Consistent Performance with Less REORG

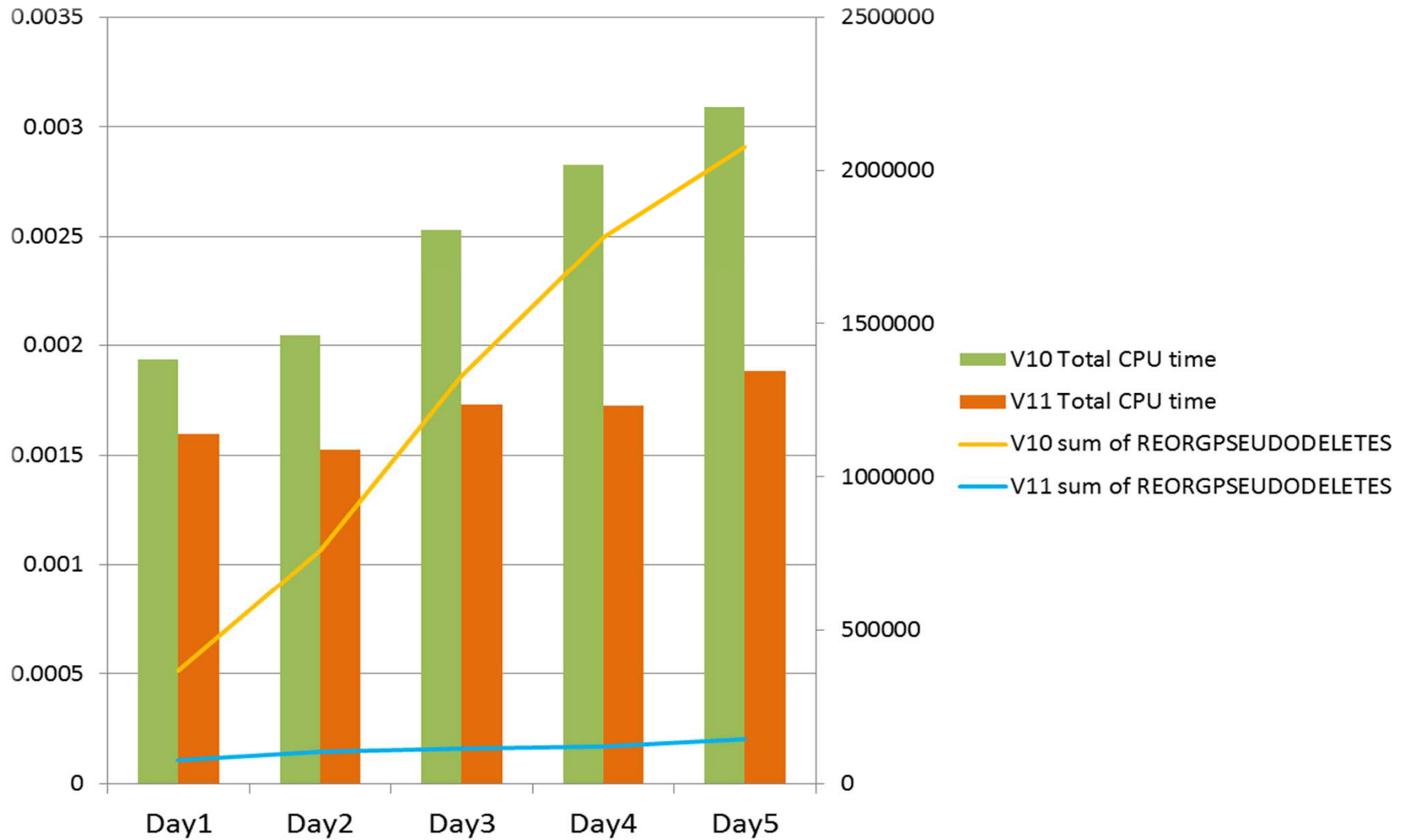
CPU per Transaction



Clean-up of Index Pseudo Deleted Pages and Entries

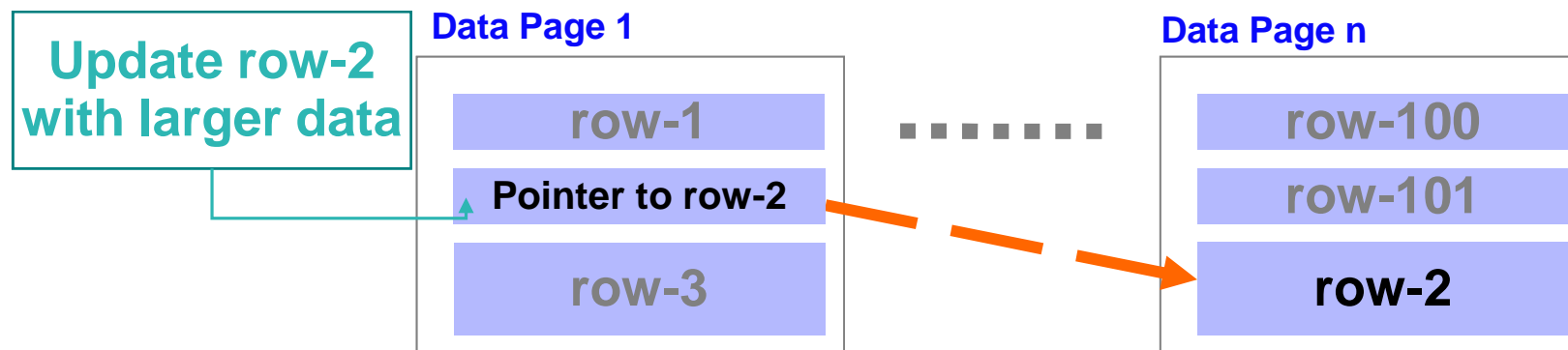
- **Performance impact of index pseudo delete entries**
 - SELECT, FETCH, UPDATE or DELETE through index search could see more getpages and lock requests
 - INSERT/UPDATE/DELETE may see concurrency issue
 - Collisions with committed pseudo-deleted index entries
 - RID reuse by INSERT following DELETE 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

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  
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 “AUTO” 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
 - DB2 to determine the value using the history of UPDATE behavior based on Real Time Statistics
 - Recommendation : Use FOR UPDATE -1 unless you know better
- Originally zparm AUTO option applies only on newly created objects.
 - With PI12400, AUTO becomes applicable for existing objects

Akiko Hoshikawa

IBM

akiko@us.ibm.com



Session : A19

Title : DB2 11 for z/OS Performance Topics

*Please fill out your session
evaluation before leaving!*