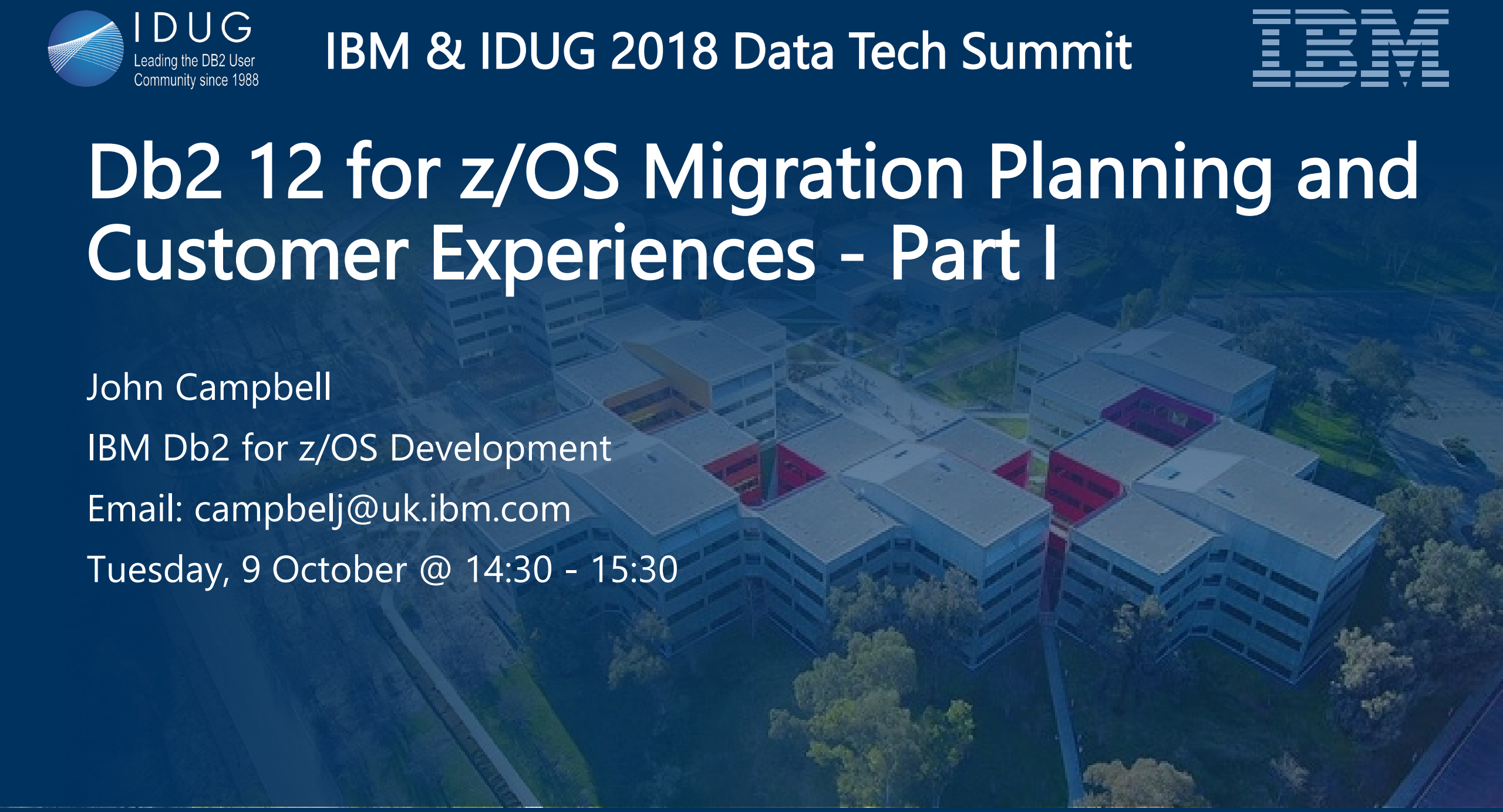
Db2 12 for z/OS Migration Planning and Customer Experiences - Part I



John Campbell

IBM Db2 for z/OS Development

Email: campbelj@uk.ibm.com

Tuesday, 9 October @ 14:30 - 15:30

Objectives

- Share lessons learned, surprises, pitfalls
- Provide hints and tips
- Address some myths
- Provide additional planning information
- Provide usage guidelines and positioning on new enhancements
- Help customers migrate as fast as possible, but safely

Agenda

- Part 1
  - Db2 11 for z/OS prerequisites for migration to Db2 12 for z/OS
  - Db2 12 for z/OS Migration – Quick Hits
  - Maintenance recommendations for early adopters of Db2 12 for z/OS
  - Db2 12 for z/OS Risk Mitigation
  - Understand Continuous Delivery starting with Db2 12 for z/OS
  - Understanding new function levels
  - Db2 12 for z/OS Greatest Hits
  - Fast Un-clustered INSERT
  - RTS enhancements

Agenda …

- Part 2
  - Fast Index Traversal
  - Data dependent vs. numeric based pagination syntax
  - More use of list prefetch
  - Increase in log record size after converting BSDS in Db2 11 and entry to Db2 12
  - Dynamic Plan Stability
  - More granular global commit LSN and global read LSN
  - SQLCODE -109 Issue
  - Enhanced SQL MERGE
  - DRDA Fast Load
  - UTS Relative Page Number (RPN)
  - INSERT Partition
  - Asynchronous CF Lock structure duplexing
  - Setting initial Statistics Profile
- Summary

Db2 11 for z/OS prerequisites for migration to Db2 12 for z/OS

- Ensure catalog consistency
  - REPAIR DBD TEST/DIAGNOSE + CHECK DATA/LOB/INDEX + DSNTESQ + ….
- Run pre-migration check queries and act on the reported findings
  - DSNTIJPM (Db2 12 for z/OS) or DSNTIJPC (APAR PI58254 for Db2 11 for z/OS)
- Apply fallback SPE PTF to all data sharing members
  - APAR PI33871 / II14794
- Make sure Db2 11 for z/OS PTF level is reasonably current especially if exploiting mixed release coexistence with data sharing and all maintenance is applied related to Db2 12 for z/OS migration
  - Use SMP/E Fix categories
    - IBM.Migrate-Fallback.DB2.V12 and
    - IBM.Coexistence.DB2.SYSPLEXDataSharing

Db2 11 for z/OS prerequisites for migration to Db2 12 for z/OS …

- Convert BSDS to 10 byte log RBA before leaving Db2 11 for z/OS NFM
  - For data sharing, convert single member at a time
  - Things to consider before converting the BSDS (DSNJCNVT)
    - Stop the Db2 for z/OS subsystem that owns the subject bootstrap data set
    - Any utility (e.g., RECOVER, REORG) running on alternate Db2 member in data sharing that reads from peer BSDS must be terminated
      - ✓ Special considerations for Data Replication
        - » Stop any data replication process to ensure BSDS is successfully renamed and replaced
        - » Best practice is to stop data replication process first, then stop the Db2 subsystem
    - RACF user ID running DSNJCNVT must have read/write access on the new BSDSs, and read access on the old BSDSs
  - After converting the BSDS, will see increased logging volume (3 <-> 40%)
  - There will be further increase in log record size after entry in Db2 12 for z/OS because of 7-byte RID values and independent of using UTS PBR RPN
  - Need to consider increasing size/number of active log pairs to maintain recommended 6 hours of recovery log data across active log configuration
  - Need to reevaluate the size of the archive log DASD pool to ensure 48 hours' worth of recovery log data can be kept

Db2 11 for z/OS prerequisites for migration to Db2 12 for z/OS …

- Avoid autobind on pre-DB2 10 for z/OS plans and packages under Db2 12 for z/OS
  - Avoid painful lesson related to plans
  - Thread break-in capability delivered in Db2 11 for z/OS may help for packages when performing rolling migration
  - Recommend explicit rebind under Db2 11 for z/OS NFM before leaving for Db2 12 for z/OS
    - Resolve any potential authorization issues that may exist
    - Use plan management for packages to keep a backup copy
- Remember to set ZPARM ABIND=COEXIST if planning to use mixed release coexistence (Db2 11, Db2 12)
- Latest News …
  - New APAR PI87675 - Re-migration autobinds are disabled
    - Both for plans and packages
    - Both for ABIND=COEXIST and ABIND=YES which now behave the same
      - ✓ Will no longer perform re-migration autobinds
    - Complete solution for Db2 11 for z/OS and Db2 12 for z/OS

Db2 11 for z/OS prerequisites for migration to Db2 12 for z/OS …

- FREE inactive package copies (access plan management) before first REBIND under Db2 12 for z/OS
- Upgrade EXPLAIN tables to Db2 12 for z/OS format (should be at least Db2 11 for z/OS version)
  - Can be done in Db2 11 for z/OS NFM with fallback SPE applied
  - Use of sample batch job DSNTIJXA with REXX DSNTXTA can help
- Apply PTFs for APARs PI69589 (Db2 11 for z/OS) & PI69584 (Db2 12 for z/OS)
  - Reduce catalog contention during “online” migration to Db2 12 for z/OS
- Plan for activation of Db2 12 for z/OS EARLY code
  - Activation via IPL or Command –REFRESH DB2,EARLY

Db2 12 for z/OS Migration – Quick Hits

- Minimum OS level lifted from z/OS V1R13 to V2R1
- Minimum hardware level lifted from z10 to z196/z114
- Replication
  - Db2 12 for z/OS (with APAR PI70998) and Db2 11 for z/OS require the Q Capture and Capture programs from IBM InfoSphere Data Replication for Db2 for z/OS Version 10.2.1
  - Q Apply and Apply programs at architecture level 1001 will work with both Db2 11 for z/OS and Db2 12 for z/OS
  - APAR PI70998 for Db2 for z/OS
  - APAR PI66768 for IIDR 10.2.1 Q and SQL
  - APAR PI61562 for CDC

Db2 12 for z/OS Migration – Quick Hits …

- Db2 Connect
  - Any level of Db2 Connect drivers should work with Db2 12 for z/OS, both before and after new function is activated with no behavior change
  - Data server clients and drivers must be at the following levels to exploit Db2 for z/OS function-level application compatibility of Db2 for z/OS FL=V12R1M**501** or greater:
    - IBM® Data Server Driver for JDBC and SQLJ: Versions 3.72 and 4.22, or later
    - Other IBM data server clients and drivers: Db2 for Linux, UNIX, and Windows Version 11.1 Modification 2 Fix Pack 2, or later
  - New ClientApplCompat (ODBC) and clientApplcompat (JDBC) property setting allows you to control the capability of the client when updated drivers ship changes to enable new server capability
    - You might want specific control of driver capability when:
      - ✓ Db2 client driver introduces new behavior currently not controlled by Db2 application compatibility
      - ✓ Change needs to be controlled at the application level to ensure compatibility with new behavior
  - ClientApplCompat/clientApplcompat setting of V12R1M500 is required to access Db2 12 for z/OS Server capability shipped after GA at function levels beyond Db2 12 for z/OS FL=V12R1M**500**

Db2 12 for z/OS Migration – Quick Hits …

- Changes to Utilities Suite installation
  - Requires registration in SYS1.PARMLIB(IFAPRDxx)
    - CBPDO is being sunset, and SystemPac is the strategic direction
    - Any separately orderable product using only F or J FMIDs has to be changed to use an E or H base FMID
  - Documented in Db2 Utilities Suite program directory

    PRODUCT OWNER('IBM CORP') NAME('DB2 UTIL SUITE') ID('577-AF4')

    VERSION(12) RELEASE(1) MOD() FEATURENAME('V12R1') STATE(ENABLED)
  - Failure to register Utilities Suite results in utility errors

    DSNU3333I 012 14:35:50.01 DSNUGPRS - THE DB2 UTILITIES SUITE FOR Z/OS HAS NOT BEEN ENABLED

    DSNU3330I 012 14:35:50.09 DSNUGPTS - THE xxxxxxxx UTILITY HAS RESTRICTED FUNCTION

    IT IS PART OF THE DB2 UTILITIES SUITE FOR Z/OS WHICH HAS NOT BEEN ENABLED

Db2 12 for z/OS Migration – Quick Hits …

- REORG MAPPING TABLE format must allow for 7-byte RID values
  - SQL DDL changes
    - Column 'SOURCE_RID' CHAR(5) -> CHAR(7)
    - Column 'TARGET_XRID' CHAR(9) -> CHAR(11)
  - No toleration logic in Db2 11 for z/OS NFM
    - Db2 11 for z/OS NFM REORG running with the Db2 12 for z/OS mapping table format will fail
  - Db2 12 for z/OS REORG in FL=V12R1M100 tolerates Db2 11 for z/OS format mapping table format
  - Db2 12 for z/OS REORG in FL=V12R1M5nn only supports the Db2 12 for z/OS mapping table format
- BIF_COMPATIBILITY system parameter still supported

Db2 12 for z/OS Migration – Quick Hits …

- RACF changes related to IDAA
  - RACF ACCESS(CONTROL) on MVS.VARY.TCPIP.DROP(OPERCMDS)
- HVSHARE should be 510 TB (default)
  - Db2 12 for z/OS requires 1 TB of 64-bit shared (private) storage in z/OS (same as Db2 11 for z/OS)
  - Virtual, not real
  - Monitor use with IFCIDs 217 and 225
- Plan for real memory increase
  - Trend continues … using larger size REAL memory to deliver performance improvements
  - Expect ~ 15% increase
  - Expect up to 30% increase if taking advantage of new in-memory function
    - Largest percentage from use of Fast Traverse Block (FTB) area – 20% increase on allocated VPSIZE
- Consider current zIIP utilization
  - Trend to extend zIIP offload continues
    - REORG and LOAD RELOAD phase
    - SQL query parallelism (child task eligibility 80% -> 100%)

Db2 12 for z/OS Migration – Quick Hits …

- Increased space requirement for RID Pool as a result of RID value increase 5 -> 8-byte value
  - Internally Db2 for z/OS uses a normalized 8-byte RID value to allow for future expansion
  - More RID blocks will be used for the same query because each RIDLIST holds fewer RIDs
  - RID Pool memory usage will be roughly 60% higher (for smaller lists it will be up to 2x higher)
  - May have to increase MAXRBLK (RID Pool size) by up to 60%
  - Data Manager logical limit (RIDMAP/RIDLIST) reduced from 26M (26,602,376) RIDs to 16M (16,625,976) RIDs
  - More RID Pool overflow to workfile is to be expected
- Deprecation of Basic Row Format (BRF)
  - zparm SPRMRRF is now hidden in Db2 12 for z/OS
  - ROWFORMAT keyword option for REORG/LOAD to convert a pageset between BRF/RRF has been removed from the documentation
    - Still supported from a utility syntax perspective
  - New objects created will always be RRF
  - Existing pagesets in BRF will continue to be supported for the time being

Db2 12 for z/OS Migration – Quick Hits …

- Deprecation of Basic Row Format (BRF) - How to recover an object in BRF format
  - For an image copy that is in BRF format, user can either:
    - Run DB2 RECOVER utility using that image copy
      - ✓ At the end of RECOVER completion, it will update the catalog/directory to reflect the state of the restored data
      - ✓ So if the image copy contains BRF data, the tablespace meta data definition will be updated to BRF at the end of RECOVER completion
    - Or use DB2 UNLOAD utility to unload the raw data from that image copy
      - ✓ The UNLOAD utility works perfectly fine still on BRF and RRF pagesets
      - ✓ Then use the LOAD utility to load the data back, into whichever format the target table is
  - Short of providing the instructions for user on how to enable/disable the now hidden RRF zparm, there is no direct user control on creating a BRF table space or partition
    - If there is ever a need to do that, user can create the tablespace or partition as usual, and then run REORG ROWFORMAT BRF to convert the object into BRF
    - ROWFORMAT keyword option on LOAD/REORG is no longer documented, but the option still remains functional as before

Db2 12 for z/OS Migration – Quick Hits …

- Invalidation of prepared SQL statements in dynamic statement cache
  - Prior to Db2 12 for z/OS, RUNSTATS would always invalidate prepared statements dependent on the object that the utility was run against
  - In Db2 12 for z/OS, RUNSTATS by default will not invalidate the prepared statements (incompatible change)
    - Use new INVALIDATECACHE YES option to force the invalidation of prepared statements
  - Invalidation of prepared statements will still occur when
    - RUNSTATS … INVALIDATECACHE YES
    - RUNSTATS after SQL DDL (CREATE/DROP INDEX) and statistics profile updated
    - RUNSTATS … UPDATE(NONE) REPORT(NO)
    - For other utilities, if the object was in an invalid state before the utility began e.g., rebuild pending or reorg pending

Maintenance recommendations for early adopters of Db2 12 for z/OS

- **Early adopters of new releases and/or new functions should be more aggressive about applying preventative service**
- Apply preventative maintenance every 3 months
  - Use RSU instead of PUT to be less aggressive on applying non-HIPER maintenance
  - Sample strategy based on two 'major' and two 'minor' releases
    - Refresh of the base every 6 months ('major')
    - Each base upgrade should be based on latest quarterly RSU
      - ✓ Ensure that RSU-only service is installed by adding the SOURCEID (RSU*) option in the supplied APPLY and ACCEPT jobs
    - In addition, two mini packages covering HIPERs and PEs in between ('minor')
- Review Enhanced HOLDDATA on a weekly basis
  - Expedite critical fixes to production after 1-2 weeks in test
  - Others can be deferred until the next major or minor maintenance drop

Different APAR numbers across Db2 11 and Db2 12 for z/OS

- Short term solution was to make the connection visible, bi-directional, either in the SYSROUTED FROM or SYSROUTED TO section of the Outline/Overview of the respective APAR

```
APAR   PI76204   OUTLINE

ENTER   DISPLAY-ITEM   PAGES   ENTER   DISPLAY-ITEM   PAGES
 S     SUMMARY            3     X     SUBMITTER TEXT      5
 E     RESPONDER TEXT     0     P     PIN ITEM            0
 I     INTRSTD PARTIES    1     K     TRACKING            2
 C     CONSTANT           1     Z     SCRATCH-PAD         1
 F     FEEDBACK           0           STRUCTURE           2
 DUP   FIRST DUPLICATE APAR   (OR DUP/NNN)   ORG  ORIGINAL APAR
 PTF   FIRST PTF REQUESTED.   (OR PTF/NNN)
                                          LAST PAGE


PTF RQSTD:
APAR FIXED BY:
DUPLICATES:
SYSROUTED FROM:
SYSROUTED TO: PI76206
APPLICABLE PE-PTF'S:
```

Different APAR numbers across Db2 11 and Db2 12 for z/OS …

- Single APAR solution for a single problem for new APARs has now been delivered (June 2017)
  - Default now is a single APAR for the same problem across Db2 11 for z/OS and Db2 12 for z/OS
  - Assumes the same “basing” – so there will still be some exceptions
  - Will not be applied respectively

Db2 12 for z/OS Risk Mitigation

- Regression testing is critical piece to keep “fires away from production”
  - Test all critical and custom processes, and scale them up
  - Run performance measurements and establish Db2 11 for z/OS baseline for comparison
  - Go / No Go decision for Db2 12 for z/OS migration of production system should be based on positive results from proper testing
    - Be prepared to postpone migration as opposed to forcing in Db2 12 for z/OS
  - Practice migration fallback from Db2 12 for z/OS to Db2 11 for z/OS and back to Db2 12 for z/OS
  - Design fallback strategy and practice it in pre-production environments
- Minimize change and use of new function in and around when Db2 12 for z/OS is first introduced into production
- For production systems, stay on FL=V12R1M100 for at least a month to prove running smoothly
  - Leaves back door open to go back to Db2 11 for z/OS NFM in an emergency
- Make sure very current on preventative service for these specific Db2 12 for z/OS functions
  - Fast Index Traversal
  - Active Log Dataset Size > 4G
  - UTS PBR RPN
  - INSERT ALGORITHM 2 (aka “Fast Insert” or “Smart Insert”) for fast un-clustered insert

Old Db2 for z/OS Strategy for delivering new function

- We deliver most of our new function in a new release ~every 3 years
- Db2 for z/OS is on 3 year cycles, but many of our customers are on 4 year cycles, hence the interest in skip release migrations
- We develop or retrofit a very limited number of new features in the service stream, but only if urgent and generally low risk
- Deployment of new releases is seen as a disruption by our customers
- Many of our customers want new features delivered much faster
- Industry and customer trend is to move away from monolithic code delivery towards continuous delivery model
- IBM is moving towards continuous delivery model
- Time for us in Db2 for z/OS to change

New Db2 for z/OS Strategy for delivering new function

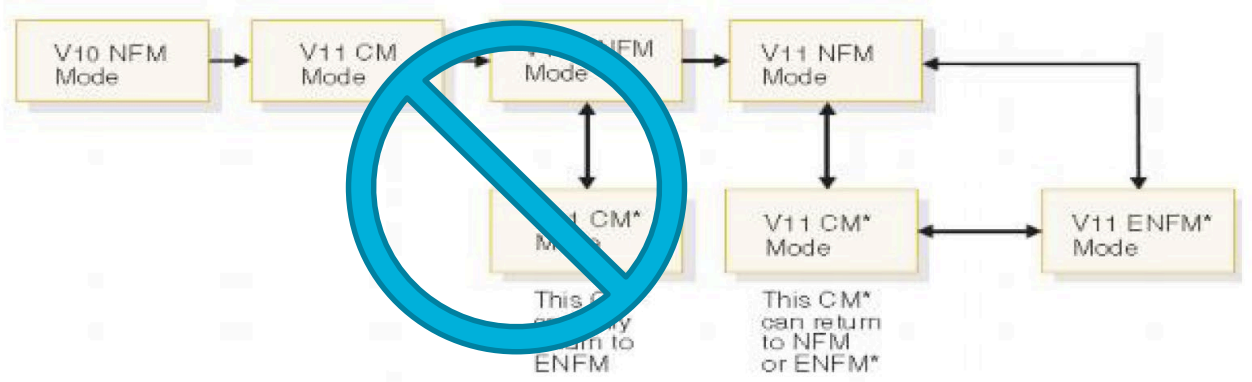
- We are dedicating ourselves to going forward on a continuous delivery model
  - Radical internal changes are required within Db2 for z/OS Development to do this
- Db2 12 for z/OS is the starting point after GA
  - There will be significantly higher volume of continuously delivered items
- Customers will see a single maintenance stream for Db2 12 for z/OS, with the new function delivered into that
  - The function will be designed to be easily consumable
- Point releases or versions will be a very rare exception
  - There are reasons why we might want to have a point release or new version
  - e.g., adopt a new compiler, extend control structures, enable an architecture level set
- Db2 for z/OS Development will have relentless focus on maintaining continuous production level reliability for you in the service stream
- We are dedicated to doing this
  - We will control the input to "the pipe", the size and risk of the items
  - Increased internal focus on function and performance regression testing
  - We will deliver new function when the quality is right, and not based on a promised date for delivery

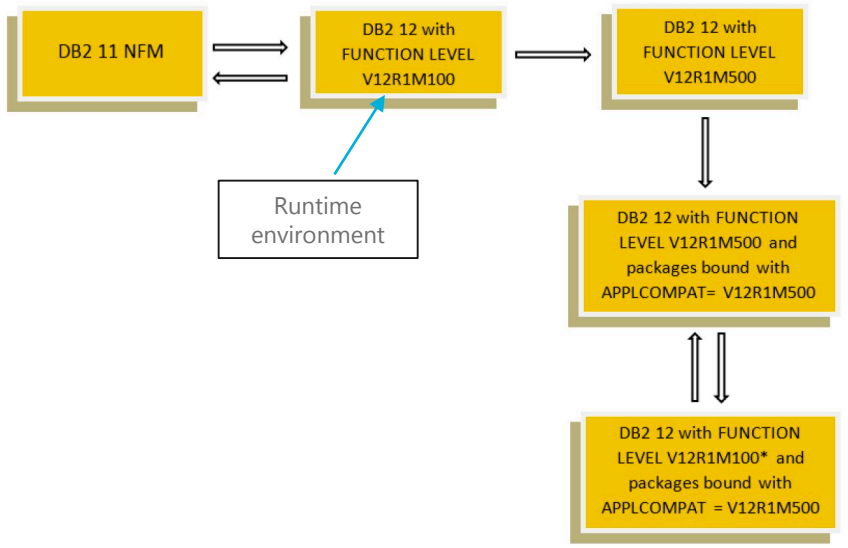
Understand Continuous Delivery starting with Db2 12 for z/OS

- With Continuous Delivery, there is a single delivery mechanism for defect fixes and enhancements
  - PTFs (and collections of PTFs like PUTLEVEL and RSU) → same as today
- With Continuous Delivery, there are four Db2 for z/OS levels
  - Maintenance level (ML) – lifted by applying maintenance
    - Also known as code level - contains defect and new enhancement fixes
    - Most new functions are shipped disabled until the appropriate new function level is activated
  - Catalog level (CL) - vehicle to enable new FL - accumulative (skip level possible)
    - Db2 Catalog changes that are needed for some FLs
  - Function level (FL) – needs to be activated - accumulative (skip level possible)
    - Introduces new Db2 for z/OS features and functionality
    - No impact or change in existing application behaviour
  - APPLCOMPAT level (AC) – set by application - provides an "island of stability" for a given application
    - Determines SQL function level of applications – can increase FL of the application (and fallback)
    - AC must be advanced to exploit new SQL function
    - AC level in BIND/REBIND of package must be <= FL and rules over FL
    - Freezes new SQL syntax even if FL is later moved back to earlier level
- Minimum starting point for Continuous Delivery is Db2 12 for z/OS FL=V12R1M500

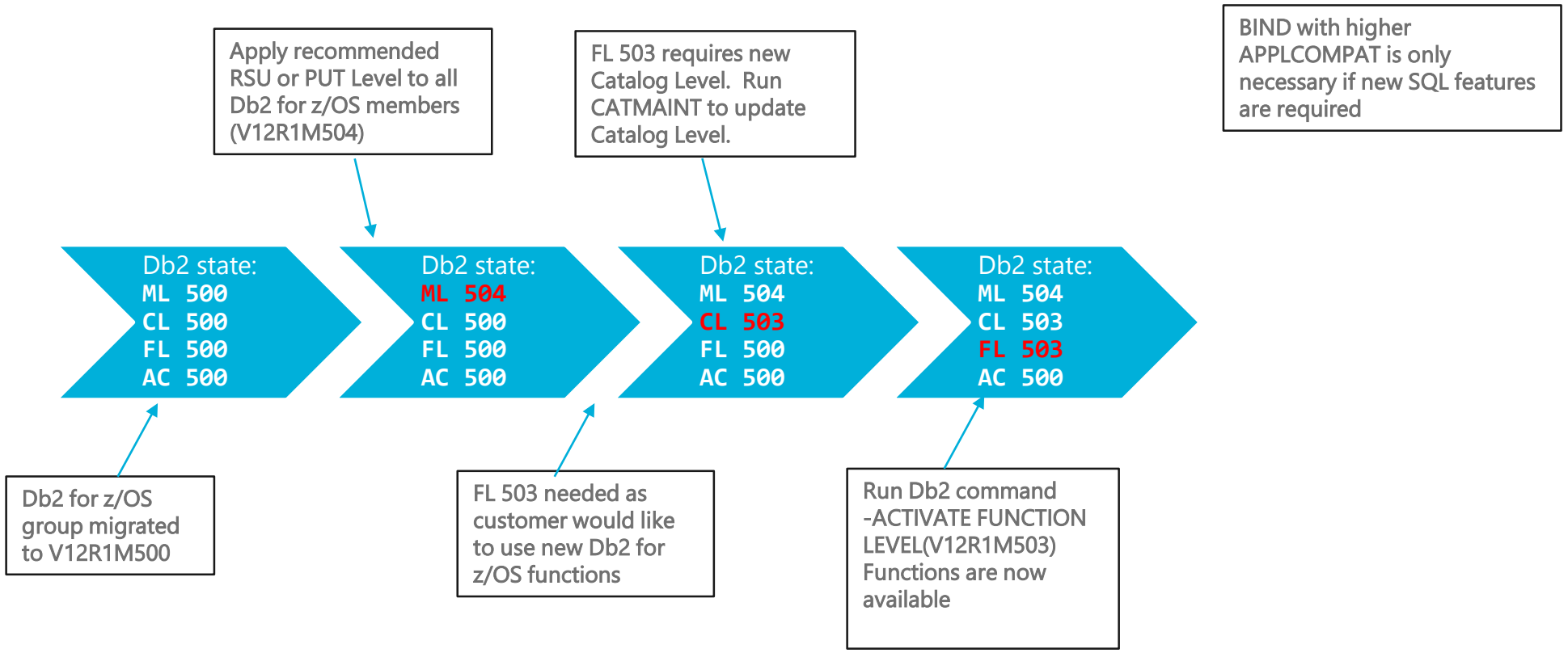
Understanding new function levels

- CM / ENFM / NFM no longer used
- Function Level V12R1M100
  - Similar to CM / BNFA
  - Db2 12 for z/OS engine and catalog / directory
  - DSNTIJTC (CATMAINT) to get there
  - Fallback to Db2 11 for z/OS NFM possible
- Function Level V12RM15nn
  - Similar to NFM /ANFA
  - New functionality available
  - Command –ACTIVATE FUNCTION LEVEL(V12R1M5nn) to get there
  - Fallback to Db2 11 for z/OS NFM no longer possible (PIT recovery would be required)





Example of how to get to a new function level



Change in strategy for APPLCOMPAT

- No need to force the rebind all packages with a new, higher APPLCOMPAT level
- APPLCOMPAT will now have many more versions to support many Function Levels
- Must still rebind a package with a higher APPLCOMPAT level in order to exploit new SQL DML, SQL DDL, SQL DCL, and XML function
  - Applications can only use new SQL if the packages are bound with the necessary and required Application Compatibility (APPLCOMPAT)
    - Packages can only be bound with an APPLCOMPAT less or equal to the current FL

- Still recommended best practice to regularly rebind all packages to
  - Benefit from latest run time performance improvements
  - Gain exposure to new access path selection improvements
  - Benefit from defect fixes
  - Reduce exposure to latent issues seeded previously

Is APPLCOMPAT a 'sticky' option on BIND/REBIND?

- BIND REPLACE does not reuse any bind option from the existing package if the option is not explicitly specified
- SQL statements can be totally different so BIND REPLACE is considered a new bind
- REBIND and BIND COPY are the only subcommands that reuse the existing/source package's options
- This is true in all Db2 for z/OS releases and not just Db2 12 for z/OS

Setting CURRENT APPLICATION COMPATIBILITY special register

- Db2 11 for z/OS
  - Value can be >= APPLCOMPAT level of the executing package but not > current Db2 version
- Db2 12 for z/OS
  - Value has to be <= APPLCOMPAT level of the executing package, independent of the current Db2 Function Level

Function Level Adoption – Best Practices 1/2

- PTFs (RSUs…) are applied that may increase the Maintenance Level (ML) of a Db2 for z/OS subsystem
- After system is stable on maintenance
  - Execute (If Any) Catmaint
    - After execution of Catmaint, the subsystem can only be started with a ML that supports the catalog
  - Activate Function Level (FL)
    - After activating a new FL, the subsystem can only be started with a ML that supports the FL
    - New function not related to SQL DML, DDL and DCL syntax is available
    - REBIND of packages with any APPLCOMPAT would pick up optimizer enhancements
    - Non-stabilized dynamic SQL would pick up optimizer / other non-APPLCOMPAT related enhancements

Function Level Adoption – Best Practices 2/2

- After Function Level is considered stable - allow new application feature rollout
  - REBIND DBA packages to allow new DDL to be utilized
  - REBIND application static packages with higher APPLCOMPAT to exploit SQL DDL/DML new functions/behaviors
  - REBIND dynamic packages with higher APPLCOMPAT to allow new SQL functions to be used
  - REBIND distributed packages (***in separate collection) with higher APPLCOMPAT to allow new SQL functions to be used
    - Switch applications to use new distributed package collection
  - Leverage PLANMGMT extended
    - Use REBIND SWITCH (PREVIOUS) to restore static SQL packages to prior runtime structures
    - Use REBIND SWITCH (PREVIOUS) for dynamic SQL packages would restore prior APPLCOMPAT
    - ***switching to prior collid for distributed dynamic would restore APPLCOMPAT

Db2 12 for z/OS Greatest Hits

- Performance
  - Fast Index Traversal (FTB)
  - Dynamic Plan Stability
  - Granular global commit LSN and read LSN
  - LOB compression
  - DRDA Fast Load
  - REORG (and LOAD) use of statistics profiles
- Application Development
  - Enhanced SQL MERGE
  - SQL pagination syntax LIMIT / OFFSET
- Availability
  - Online ALTER to increase DSSIZE
  - Lifting partition size limit (1 TB)
  - Insert Partition
  - Asynch CF lock Duplexing
- Security
  - TRANSFER OWNERSHIP

Fast Un-clustered INSERT

- Insert workloads are amongst the most prevalent and performance critical
- Performance bottleneck will vary across different insert workloads
  - Index maintenance?
  - Log write I/O?
  - Data space search (space map and page contention, false leads)
  - Format write during dataset extend
  - PPRC disk mirroring
  - Network latency
  - etc
- Common that Index insert time may dominate and mask any insert speed bottleneck on table space

Fast Un-clustered INSERT …

- Officially referred to as “Insert Algorithm 2 (IAG2)”
- Sometimes referred to as “Smart Insert” or even “Fast Insert”
- Potentially delivers significant improvement for un-clustered inserts (e.g., journal table pattern) where both
  - Heavy concurrent insert activity (many concurrent threads)
  - Space search and false leads on data is the constraint on overall insert throughput
- Applies to any UTS table space defined with MEMBER CLUSTER
  - Applies to both tables defined as APPEND YES or NO
- Implemented advanced new insert algorithm to streamline space search and space utilisation
  - Eliminates page contention and false leads
  - Default is to use the new fast insert algorithm for qualifying table spaces
    - DEFAULT_INSERT_ALGORITHM system parameter can change the default
    - INSERT ALGORITHM table space attribute can override system parameter
- It is NOT a replacement for the existing insert algorithm (IAG1)!

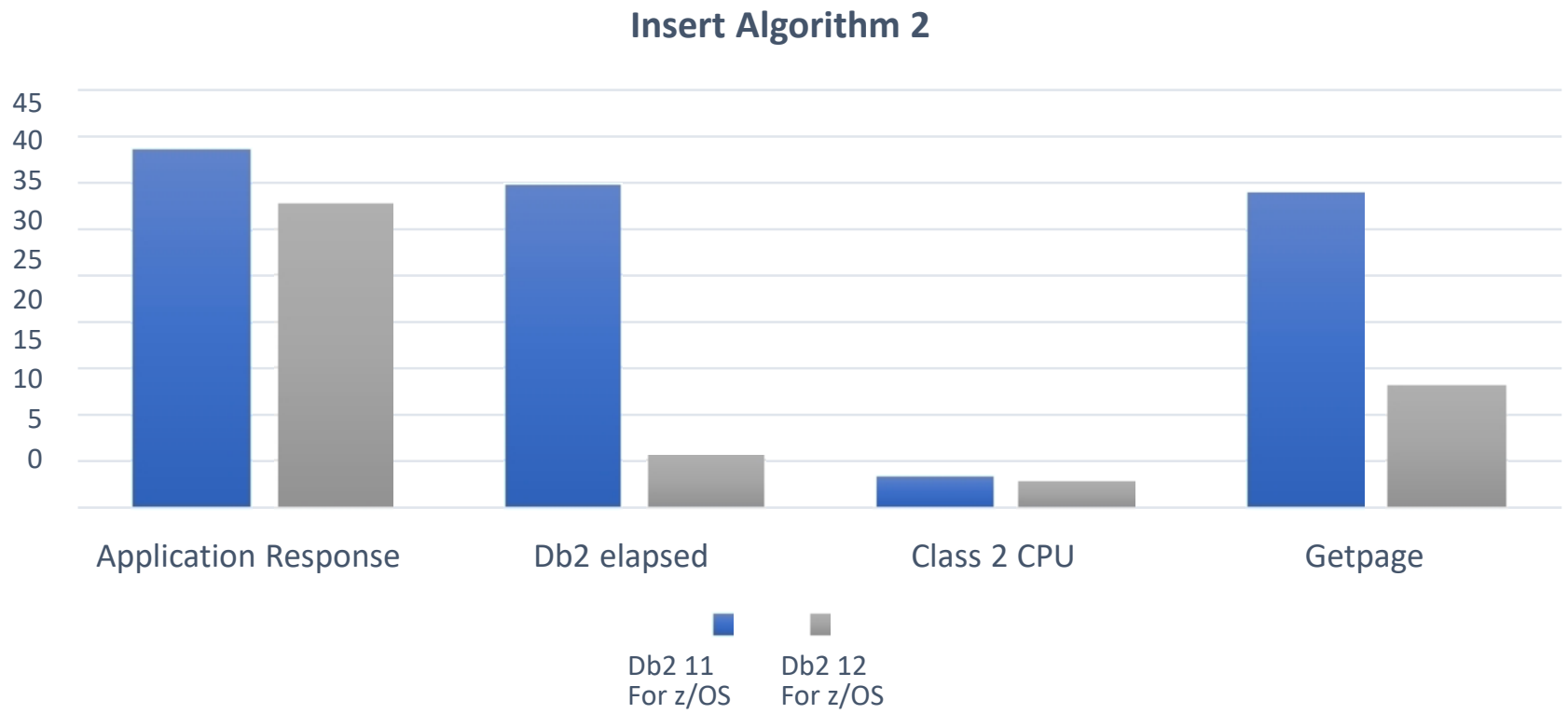
Fast Un-clustered INSERT …

- Your mileage will vary
  - Many insert workloads will see no improvement and is to be expected
  - Some specific insert workloads may see significant improvement
- Will shift the bottleneck to the next constraining factor
- LOAD SHRLEVEL CHANGE can also use Fast Un-clustered INSERT
- Fast Un-clustered INSERT will not be used when lock escalation occurs or use of SQL LOCK TABLE
- Available after new function activation (FL=V12R1M5nn)

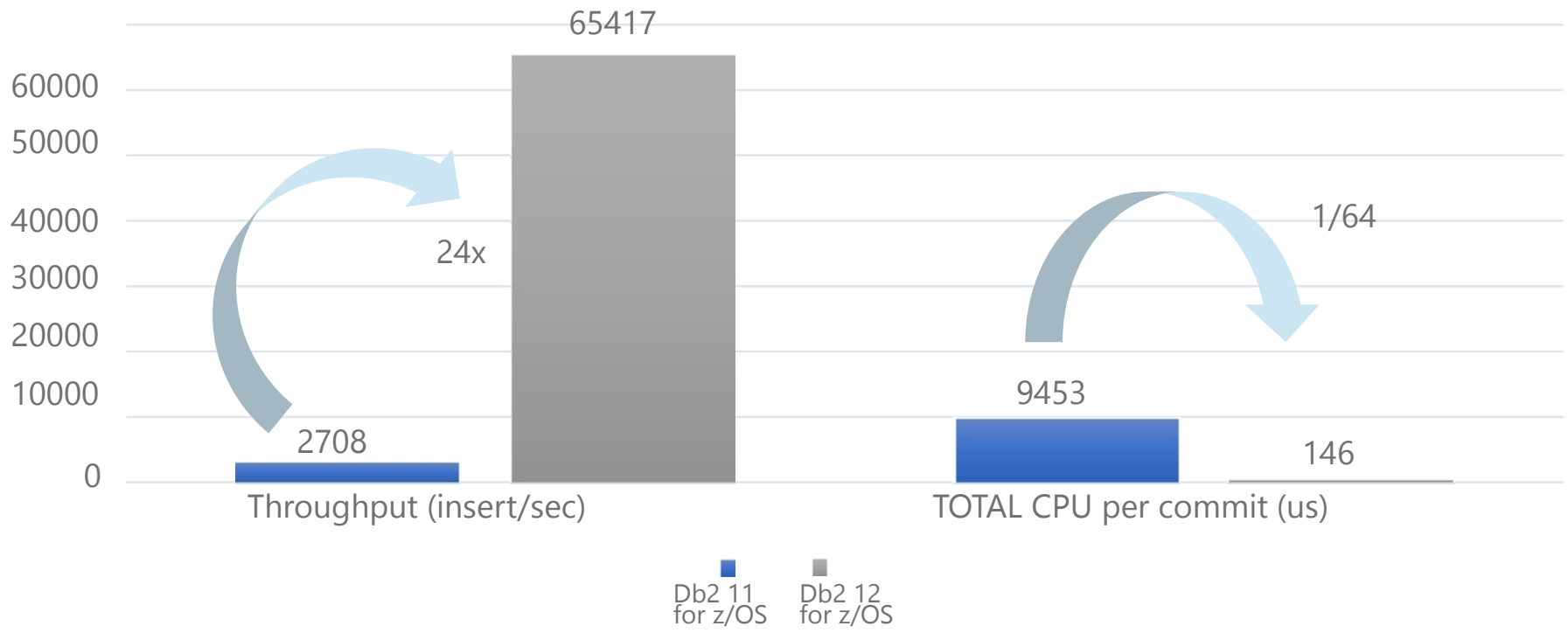
Fast Un-clustered INSERT …

- Recommended preventative maintenance (APARs) for robustness and serviceability as at October 2017
  - PI83083
  - PI83519
  - PI85653
  - PI85018
  - PI74870
  - PI75781
  - PI67997
  - PI77866
  - PI68022
  - PI80532
  - PI77866
  - PI81731
- APAR PH02052 (implements automatic re-enablement with retry logic) about to be closed out
- Current point-in-time recommendation
  - Change system wide default - set system parameter DEFAULT_INSERT_ALGORITHM = 1 (old basic insert algorithm)
  - One size probably does not fit all tablespaces
  - Not much difference/improvement for short running transactions
  - Use INSERT ALGORITHM 2 (new fast insert algorithm) selectively at individual table space level to override system wide default

Fast Un-clustered INSERT – Shifting The Bottleneck …



Fast Un-clustered INSERT - Db2 11 for z/OS PMR Recreate …



UTS PBG with MEMBER CLUSTER, RLL, with 400 bytes per row, one index,
800 concurrent threads, 10 insert per commit

RTS enhancements

- New messages DSNT535I and DSNT536I e.g.,

  DSNT535I =D2E1 DSNIRTST 2 ATTEMPTS TO EXTERNALIZE IN-MEMORY STATISTICS TO REAL-TIME
  STATISTICS TABLES FAILED DURING THE PAST 30 MINUTES 'BECAUSE A RESOURCE WAS UNAVAILABLE:
  TYPE 00000304 NAME DSNDB06 .SYSTSISS.X'0000650D'.X'07'

- New column GETPAGES added to both SYSIBM.SYSTABLESPACESTATS & SYSIBM.SYSINDEXSPACESTATS
  - Very valuable
  - Records number of getpage requests since release migration, last REORG, last LOAD REPLACE or since object creation
  - Do not rely on the value whilst running in mixed release coexistence
- Temporal (system-period data versioning)
  - Requires FL=V12R1M5nn
  - SQL DDL changes performed by CATMAINT
  - Activated by ALTER TABLE … ADD VERSIONING clause -> SYSIBM.SYSTABLESPACESTATS & SYSIBM.SYSINDEXSPACESTATS
  - No indexes provided – must RYO to speed up your SQL queries
  - MAXPART 1 is ‘hard wired’ for history tables
  - Must develop procedures for cleanup of history tables and associated housekeeping

Summary

- Share lessons learned, surprises, pitfalls
- Provide hints and tips
- Address some myths
- Provide additional planning information
- Provide usage guidelines and positioning on new enhancements
- Help customers migrate as fast as possible, but safely

Top DB2z Social Media Channels

#DB2z

- Join the World of DB2 www.worldofdb2.com
- Follow @IBMDB2 on Twitter https://twitter.com/IBMDB2
- Join DB2z LinkedIn Group
- https://www.youtube.com/user/IBMDB2forzOS
- DB2z on Facebook
  - https://www.facebook.com/IBMDB2forzOS/


thank
you!