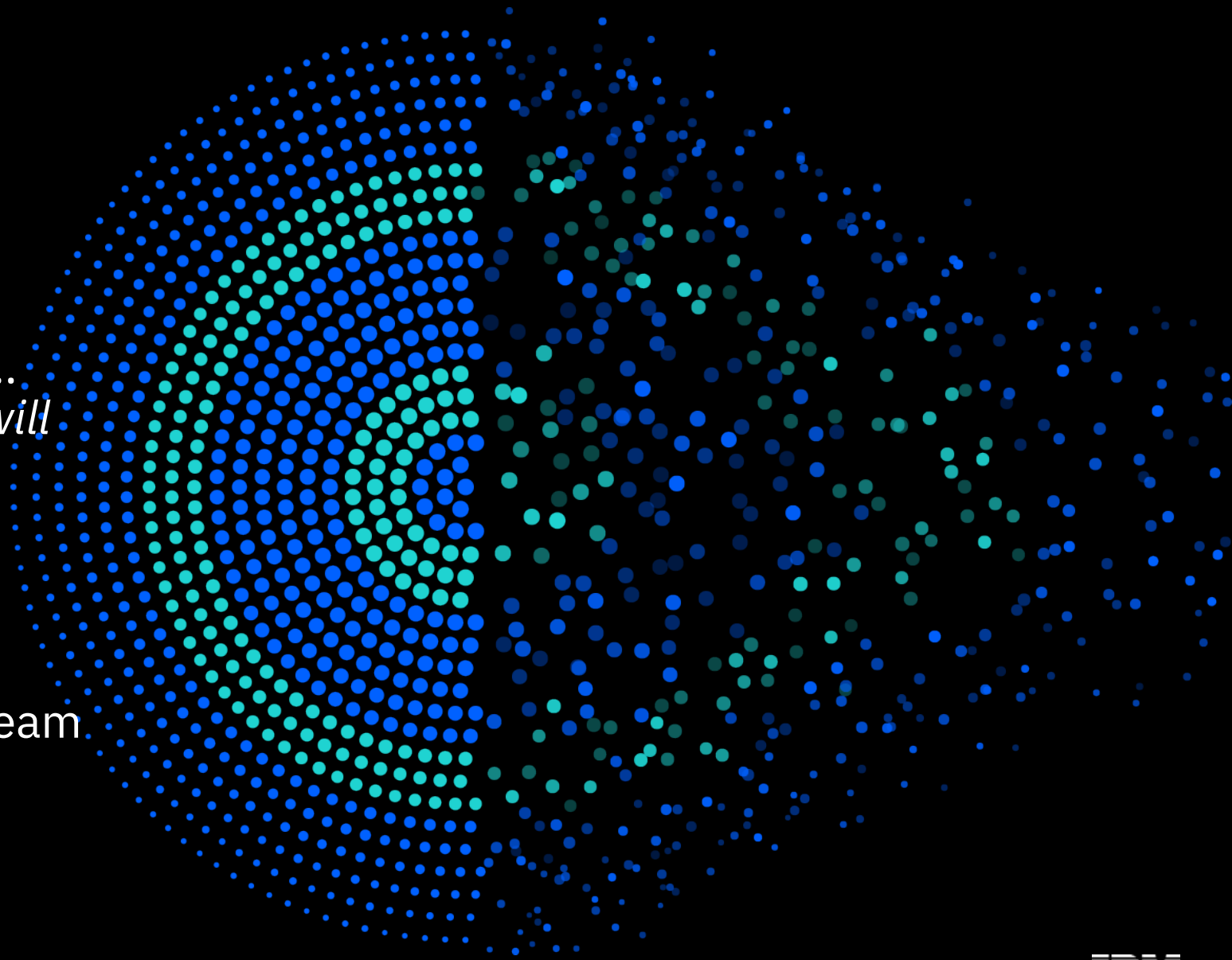
Data and AI

Db2 for z/OS SWAT Team “*Tales*” ....
“Murphy's Law What can go wrong will go wrong”

Anthony Ciabattoni
Executive IT Specialist
Db2 for z/OS Development SWAT Team



IBM

Agenda

- Db2 preventative maintenance considerations
- Db2 Recovery Assets
- Data Integrity Checking
- Local Db2 Recovery
- Continuous Data Protection (CDP) Solution
- System and application performance monitoring
- HiperDispatch
- An update look at Universal Tablespaces
- Questions

Db2 preventative maintenance considerations

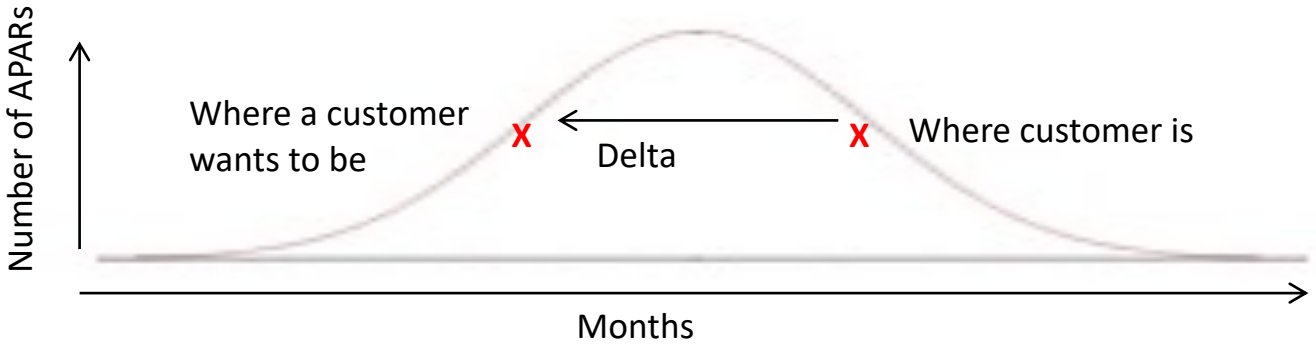
- Introduction
  - The process of applying or not applying Db2 PTF(s) is often-times one of the most important tasks performed by a Db2 professional
    - The result of applying/not applying an APAR can result in …
      - Db2 crash
      - Data corruption
      - Incorrect output (INCORROUT)
      - Etc.
  - Technical resources commonly have developed processes to apply routine maintenance but have not created exception process to apply for critical PEs or HIPER PTF(s)
  - Maintenance frequency, decision processes and exceptions are not effectively communicated to management/executives
    - The end results supports a “blame” culture, the technician is always wrong

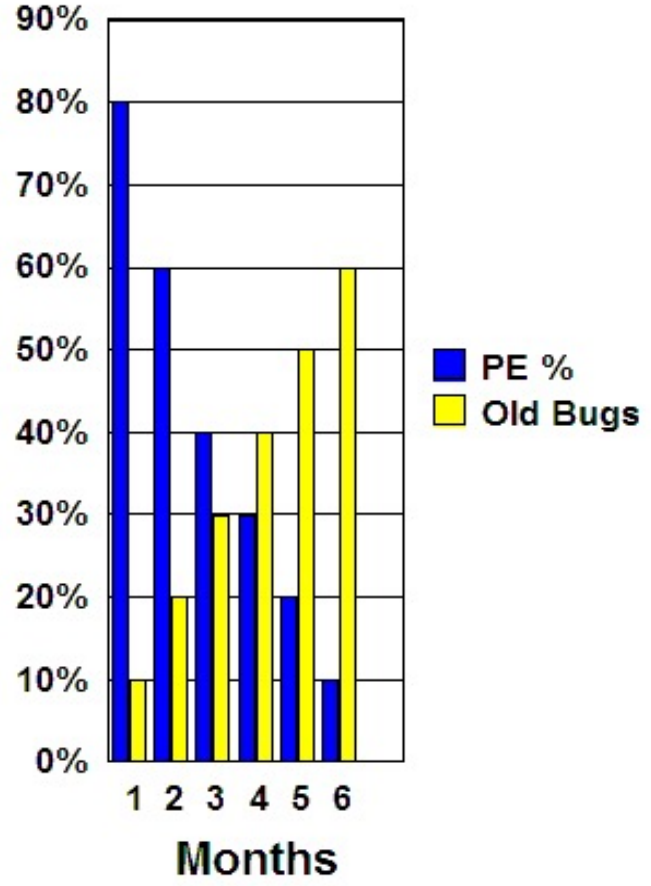
Db2 preventative maintenance considerations …

- Common Problems
  - Too many customers are very back level on preventative service
  - No HIPERs or PE fixes applied since the last preventative service upgrade
  - High profile production incidents could have been avoided by applying missing HIPER(s)
  - Not exploiting Db2 Data Sharing technology to avoid planned outages and remove dependency on change windows
  - ***'Fix-on-failure'*** culture introduces the probability of long prerequisite chain when having to apply emergency corrective service
  - Delay in exploiting new availability functions
  - Delay in applying Db2 serviceability enhancements to prevent outages
  - 'One-size-fits-all' approach across multiple different application environment leads to escalating maintenance costs
  - Static/traditional SMP/E infrastructure which does not allow for flexibility

Db2 preventative maintenance considerations …

- Applying preventive maintenance can and will avoid outages
  - Up to 20% of multi-system outages could have been avoided by regularly installing 'critical' (e.g., HIPER and PE fixing) PTFs
- Executing a preventive maintenance process requires an understanding of trade-offs
  - Position on the adoption ‘bell curve’
  - Problems encountered vs. problems avoided
  - Potential for PTF in Error (PE)





Db2 preventative maintenance considerations …

- Recommendations
  - Change Management process should balance the risk of making 'no change' vs. making 'a change'
  - Apply preventative maintenance every 3 months
    - Use RSU instead of PUT to be less aggressive on applying non-HIPER maintenance
      - Use the latest quarterly or quarterly+1 month as the base
    - 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')
    - Early adopters of new releases and/or new functions should be more aggressive about applying preventative service
  - Review Enhanced HOLDDATA on a weekly basis
    - Best practice to implement review weekly regardless rolling a maintenance package
    - Identify and implement any potential operational bypasses
    - Expedite critical fixes to production after 1-2 weeks in test
    - Others can be deferred until the next major or minor maintenance drop

Db2 preventative maintenance considerations …

- Common problems - PEs
  - As a new package is built, PEs (and any blocked maintenance) are systematically taken off
  - If no PE fixers are added to the package, it has the potential to leave the package very back-leveled and/or increase operational exposure to serious problems if a large number of HIPERs or even a small number of very critical ones are blocked
- Recommendations
  - Proactively monitor for PE fixers and new PEs on a weekly basis as you are building the package and during the roll, up to ~1 month before production
  - Apply PE fixers as soon as available + any RSU maintenance that can now be unlocked
  - Need a “judgement” call around remaining PEs that are not yet resolved or have emerged
    - If the PEs are blocking significant HIPER maintenance (large number and/or ‘vicious’ ones) → do not simply take them off, but look at options
      - Wait for the PE fixer, or look for an operational bypass to get the PE on or to avoid hitting the problems covered by the HIPERs
      - Do not hesitate to open PMRs to get additional information to make a well-informed decision

Recovery asset infrastructure

- Introduction
  - Recovery assets
    - Db2 active logs, archive logs, Db2 image copies
- Common Problems
  - Wrapping active log datasets too frequently (< 6 hours)
    - Long running units of work should be monitored and addressed
  - Writing archive logs directly to tape or VTS
  - If a Global or Metro mirror configuration is being used …
    - Active/archive logs at target location needs to be configured to eliminate potential “holes”
  - Image Copies
    - Datasets written or moved to tape or VTS
    - Not taken during LOG NO operations
    - Image copy and archive log datasets do not have date/time in the name convention
  - Pools are not monitored for growth …. e.g., volume increase, Db2 12
  - Retention period for recovery assets are not consistent e.g., archive logs, image copies

Recovery asset infrastructure …

- Risks
  - Accessing Db2 image copies and archive logs that reside on VTS will result in serialization for multi-object recovery operations
    - VTS is backed by DASD but is recognized and treated as a tape device to the operating system
    - Additional storage pools are not pre-allocated to copy VTS objects to DASD prior to recovery
  - Wrapping of the active log pairs too frequently can expose availability concerns
    - Small window of time to react if there are issues with archiving the active logs
    - When Global Mirror relationship exists and archive logs are written to VTS, VTS replication latency can result in “holes” in the Db2 active log at target location
  - Archive log and image copy naming standards
    - Using GDGs for image copy datasets can result in reduced retention period
    - Not using a useful time-oriented naming standards can make it unnecessarily complicated
  - If a single image copy created during a LOG NO is accidentally deleted or damaged the object will become unrecoverable through the LOG NO event

Recovery asset infrastructure …

- Recommendations
  - Increase the size of the active log configuration to always hold at a minimum 6 hours of recovery log data, if possible 24 hours is preferable
    - Standardize on a uniform dataset size of 4G-1 bytes (if needed), Db2 11
    - Add in additional preformatted log pairs (if needed, max 93 pair)
      - Defining 93 active log pairs to Db2 is the maximum and will prevent the ability to dynamically allocate additional logs
    - In Db2 12, can have larger sized active log datasetsDb2 12 allows up to 768GB log data set size
    - This position must be proactively maintained
  - Write both copies of each archive log pair to DASD
    - Keep LOGCOPY1 on DASD for at least 48 hours (must be maintained going forward)
    - Can immediately migrate LOGCOPY2 away to VTS using DFSMShsm
  - Must factor in the increase in log record size after entry into Db2 12 (circa. 10%)
    - May have to add in additional active log dataset pairs
    - May have to increase the size of DASD pool for archive log datasets
  - Set ZPARM TSTAMP=YES to inject the date and time into the name for archive log datasets

Recovery asset infrastructure …

- Recommendations …
  - Take dual inline image copy for LOG NO events (e.g. REORG, LOAD REPLACE)
  - At the first sign of data corruption implement a procedure to stop recovery assets from rolling off
  - Standardize on a nn day e.g., 31 retention period for all recovery assets with a modify of nn days e.g., 28
    - Advanced data recovery to generate REDO SQL
    - Drive all data integrity issues to root cause
  - Consider including date/time and image copy type (full/incremental) in image copy dataset name

Data Integrity Checking

- Introduction
  - CHECK utilities are critical diagnosis tools in the event of data corruption
    - Identify objects that need repairing/recovering and assess the extent of the damage
  - Customers potentially are not set up to run the CHECK utilities non-disruptively
    - Cannot take advantage of FlashCopy unless directing the CHECK utilities to use a pool of volumes outside of Global Mirror for the creation of the shadow objects
    - ZPARM CHECK_FASTREPLICATION = PREFERRED would allow the CHECK SHRLEVEL CHANGE utilities to use ‘standard’ I/O to create the shadow objects, which could result in an elongated interference with the applications
- Risks
  - Interference with updating applications during accidental use of CHECK SHRLEVEL CHANGE utilities
  - Delays in detecting objects with data inconsistencies and the degree thereof

Data Integrity Checking …

- Recommendations
  - As a defensive measure, set ZPARM CHECK_FASTREPLICATION = REQUIRED to prevent accidental use of CHECK utilities with SHRLEVEL CHANGE
  - Exploit dataset-level FlashCopy to run the CHECK utilities non-disruptively
    - Carve out a pool of volumes outside of Global Mirror for temporary shadow copies
      - The volumes could be part of Metro Mirror, but then must set ZPARM FLASHCOPY_PPRC = REQUIRED to use Remote Pair FlashCopy and avoid going out of full duplex on the Metro Mirror pairs
    - Use ZPARM UTIL_TEMP_STORCLAS to specify a storage class mapped to the pool of target volumes outside of Global Mirror (or use the DFSMSdss FlashCopy Batch Protection feature)
    - Note: Dataset FlashCopy will fail if individual dataset should span multiple DASD storage subsystems

Db2 recovery background

- Db2 log-based recovery of *multiple objects* may be required when…
  - Catastrophic DASD subsystem failure and no second copy
  - Plan B for disaster recovery
    - Mirror is damaged/inconsistent
    - Bad Disaster Restart e.g., using stale CF structures in data sharing
  - Data corruption at the local site caused by…
    - 'Bad' application program
    - Operational error
    - Db2, IRLM, z/OS, third-party product code failure
    - CF microcode failure, DASD subsystem microcode failure

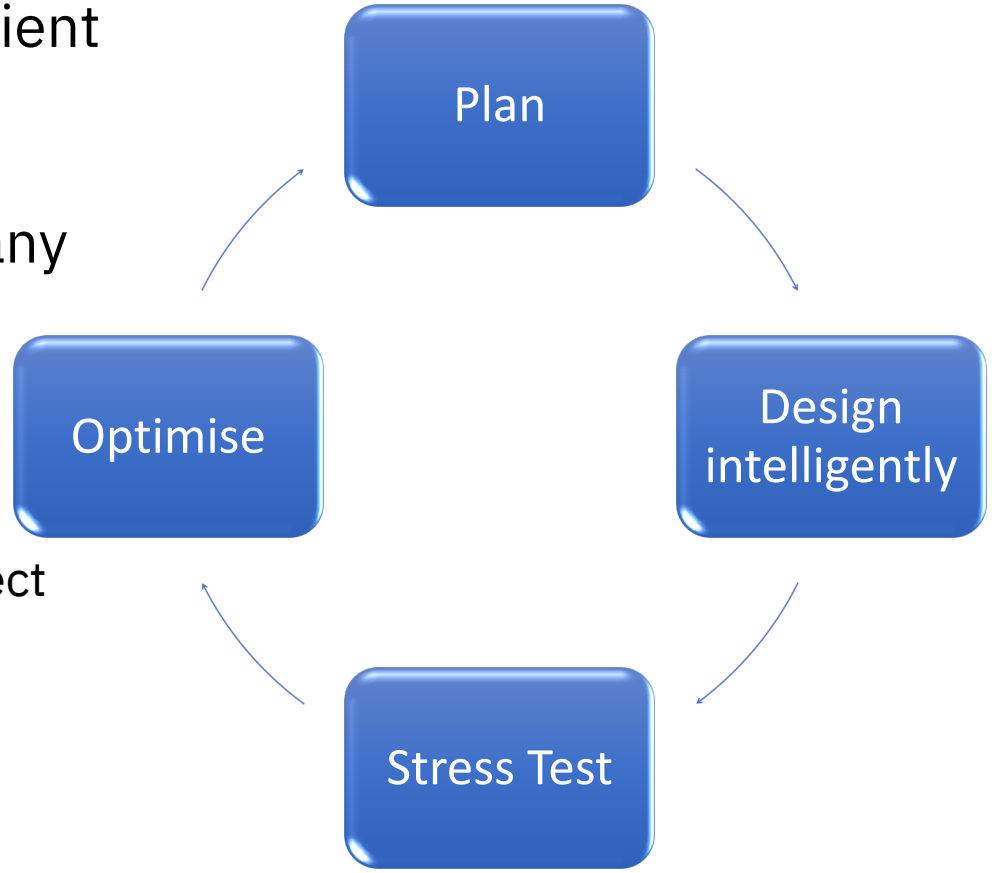
- Scope of the recovery may be more or less extensive
  - One application and all associated objects
  - Part of the system (including a random list of objects across multiple applications)
  - Or, in the worst case, the 'whole world'

Db2 recovery background …

- **Db2 log-based recovery of multiple objects is a very rare event …**

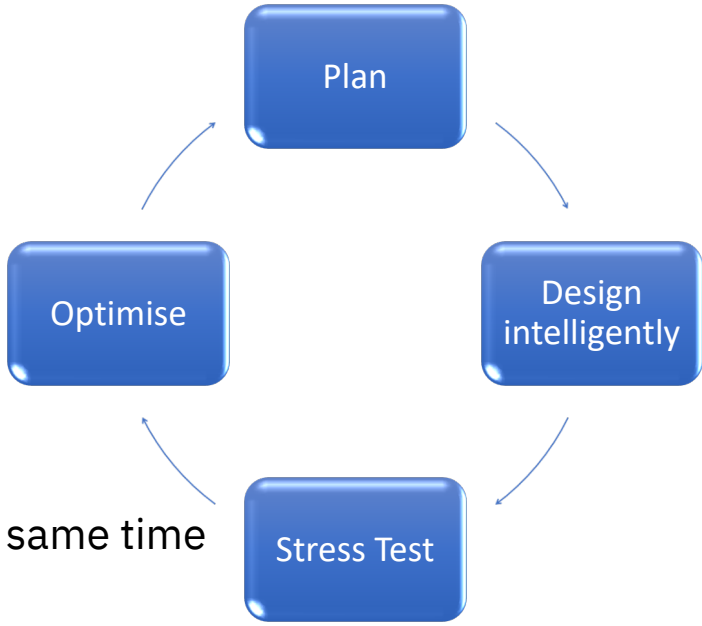
  … but statistically, it is more frequent than a true DR event (flood, fire, etc.)

- Taking regular backups is necessary but far from sufficient for anything beyond minor application recovery
- If not prepared, practiced and optimized, will lead to extended application service downtimes – possibly many hours to several days
  - Things to consider
    - Are my procedures up to date?
    - Configuration changes? Db2 release?
    - Are image copies and recovery jobs created based on object priority?
      - How long will the “recover” take?
    - Are the image copies on DASD, VTS or physical tape?
    - Are all my objects backed up?
    - If not practiced “what do you not know?”



High performance multiple object recovery

- Common issues
  - Lack of planning, intelligent design, optimization, practice & maintenance
  - No prioritized list of application objects and inter-dependencies
    - Limited use of Db2 referential integrity
    - Data dependencies and integrity management are buried in the applications
    - Heavily dependant on application knowledge and support
  - Procedures for taking backups and executing recovery compromised by lack of investment in technical configuration
  - Backup and recovery procedures have not been addressed for years
  - Use of tape including VTS *("Identity Crisis")*
    - Cannot share tape volumes across multiple jobs
    - Relatively small number of read devices
    - Concurrent recall can be a serious bottleneck
    - Even though VTS has a disk cache, it is known to z/OS as tape device
      - Same serialization characteristics as all tape devices
      - A single virtual volume cannot be shared by different jobs or systems at the same time



Db2 logging environment …

- Design for *recovery performance* …
  - Keep at least 48h of recovery log data on DASD

Option #1: Over-configure the active log pairs (number/size)
Write archive log COPY1 and COPY2 to DASD but they can be migrated to tape/VTS at any time

Pros: Optimal log read performance with automatic load balancing for reads between active log COPY1 and COPY2,
Db2 12 increases capacity to 93x768GB
Cons: Maximum capacity in V11 = 93x4GB

Option #2: Keep archive log COPY1 on DASD for 48-72h before migrating it to tape/VTS – archive log COPY2 can be migrated to tape/VTS at any time

Pros: Good log read performance from archive on DASD, potential for less DASD requirements than Option 1

  - Be ready to extend the amount of recovery log beyond what is available on DASD
    - Set BLKSIZE=24576 to optimise reads on DASD
    - Prepare a procedure to copy archive logs from tape or VTS to DASD

Mass application recovery

- Recommendations
  - Agree on a prioritized list of business-critical applications
  - Keep a list of all related data required by these applications
    - Dependencies across application domains
    - Including non-Db2 data
  - Critical information needed during a recovery event
    - Objective: Bring back critical application services as soon as possible
    - Without these lists, either have to wait for the whole world to be recovered, or take a risk in bringing back some of the application services earlier
    - Should not rely exclusively on application expertise
  - At the first sign of potential data corruption develop a process to stop the rolling away of recovery assets

Mass application recovery …

- Recommendations …
  - Build recovery jobs that exploit the capacity of the entire Db2 data sharing group
    - Maximum parallelism in the RESTORE phase
      - For partitioned tablespaces, use parallelism by part
        - LISTDEF utility statement with the PARTLEVEL option will build a list of partitions for an object and automatically handle partitions that are added or pruned
      - Use PARALLEL for parallel processing from image copies on DASD
      - Use PARALLEL(n) TAPEUNITS(n) for image copies stacked on tape
    - Optimal use of fast log apply (FLA)
      - In Db2 10, ZPARM LOGAPSTG has been removed and is set internally to 510MB
      - Schedule up to 51 RECOVER jobs per Db2 subsystem
      - RECOVER a list of objects rather than individual objects
        - But no more than 98 objects per RECOVER job for best results (1 partition = 1 object)
        - 20-30 objects per RECOVER job seems to be optimal for FLA use
        - Single pass of the recovery log for all objects in the list
      - Spread the jobs across all Db2 data sharing members

Continuous Data Protection (CDP) Solution

- CDP solutions provide for potentially much better point-in-time data recovery
  - Taking more frequent backups with thin provisioning to limit the amount of Db2 log apply which leads to faster data recovery
  - Faster and operationally simpler to restore the complete system in either an isolated “forensic” environment or in the primary environment
    - “Forensic” environment = made available to application teams to investigate data corruption/inconsistency and reconcile against the current production data
- Each “safeguarded” copy in the vault is I/O crash consistent
  - Data is not application transaction consistent
- Danger in recovering individual datasets and subset of datasets unless super confident about understanding application and data dependencies
  - Avoid “jagged” edge in terms of time consistency across related applications and objects

Continuous Data Protection (CDP) Solution …

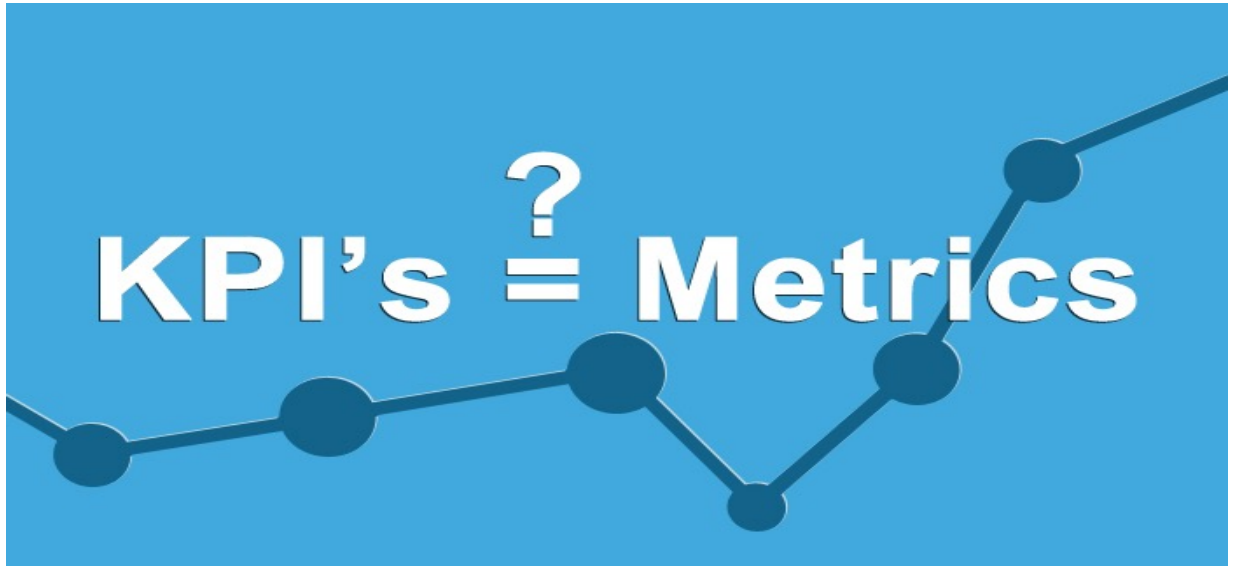
- Db2 crash restart (system) or Db2 RECOVER LOG ONLY (pageset/partition or dataset) is required to make any data extracted from the vault, application transaction consistent
- Must be super confident about understanding application and data dependencies when recovering individual datasets and subset of datasets
- Solutions designed for recovering individual dataset, subset of datasets, whole system and building a “forensic” environment need to be tested, validated and regularly practiced to make sure they are in correct working order
- Taking Db2 image copy backups is still strongly recommended
  - Drive Db2 data recovery of last resort
  - Need to go back beyond 24–48 hours to support problem determination and advanced data recovery
  - Intra-page integrity checking performed when taking daily image copies
  - Db2 image copy backups are still required as input to some IBM and vendor Db2 tools
  - Db2 Image copy backups need to taken for Db2 LOG NO event

System and application performance monitoring

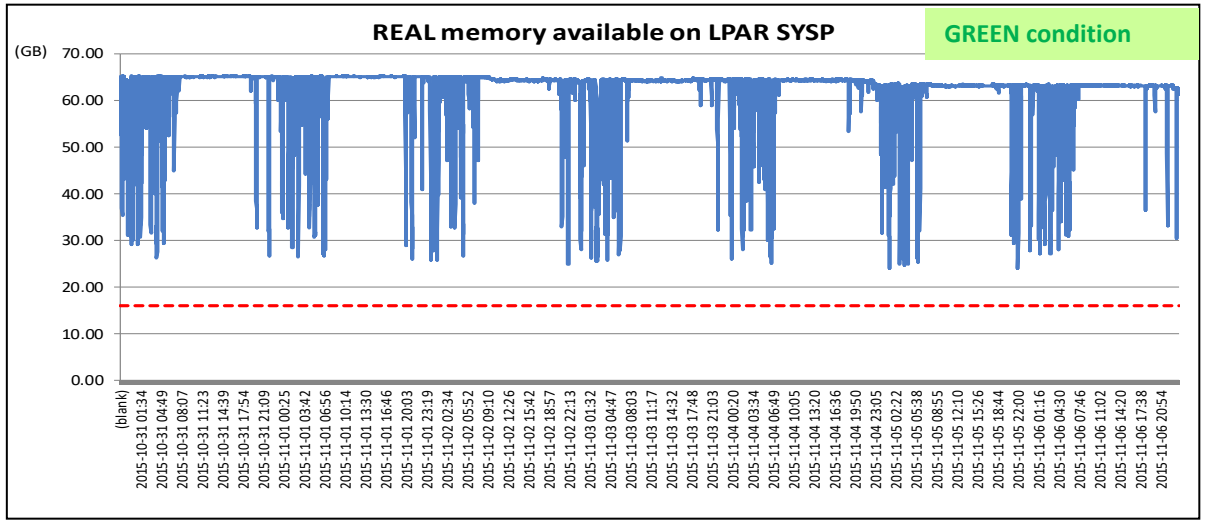
- Common Problems
  - Organizations do not invest the necessary funding, resources in discipline in terms of people resource and specialist skill level
  - Organizations routinely collect Db2 performance data but do not transport the data into consumable graphical display
    - Historically
    - Operational
  - Missing system and application baseline data
    - Key Performance Indicators (KPIs) per application, application type and Db2 group/member often are not identified
  - Organizations do not collect the necessary system (statistics) and application (accounting) to properly measure what is normal activity at a granular level
    - Db2 collects system related data (statistical data) in 1-minute intervals, often the data is summarized
    - Db2 collects application (Db2 accounting data) SMF TYPE 101 at end of unit of work
      - DDF supplies option with ACCUMAC to summarized up to x number of transactions
      - Raw accounting data can be humungous

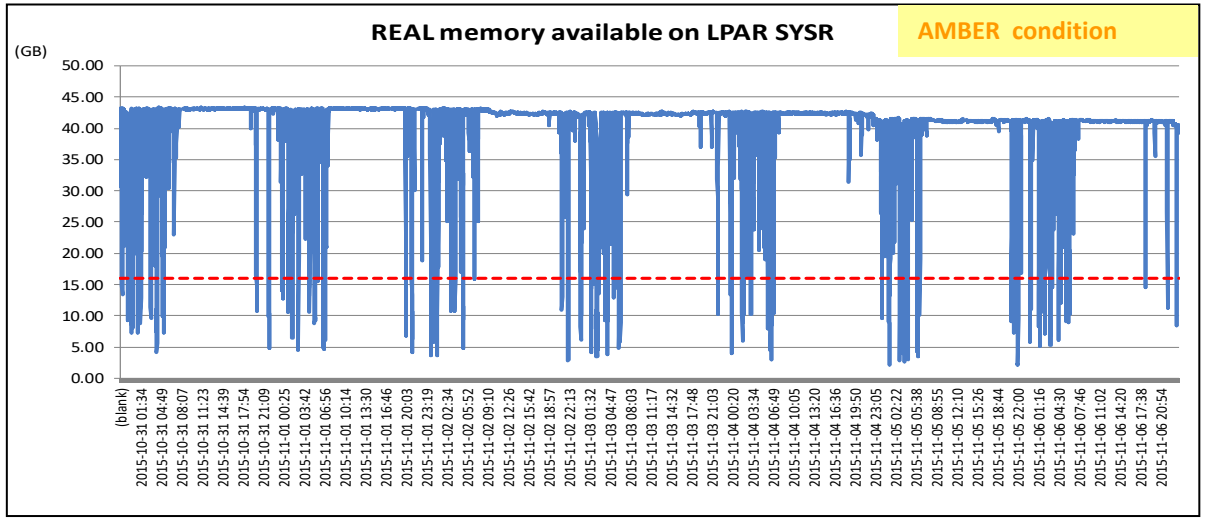
System and application performance monitoring …

- Accounting Key Performance Indicators (KPIs)
  - Number of occurrences
    - Transactions/Second
    - Commits
  - Elapsed Time
    - Elapsed/Transaction
    - Elapsed/Commit
  - CPU Time
    - CPU/Transaction
    - CPU/Commit
  - Additional Indicators
    - DML/Transaction and type – Select, Insert, Update, Delete
    - Getpages/Transaction
    - Average Sync I/O/Transaction
    - Number of select, Insert, update and deletes

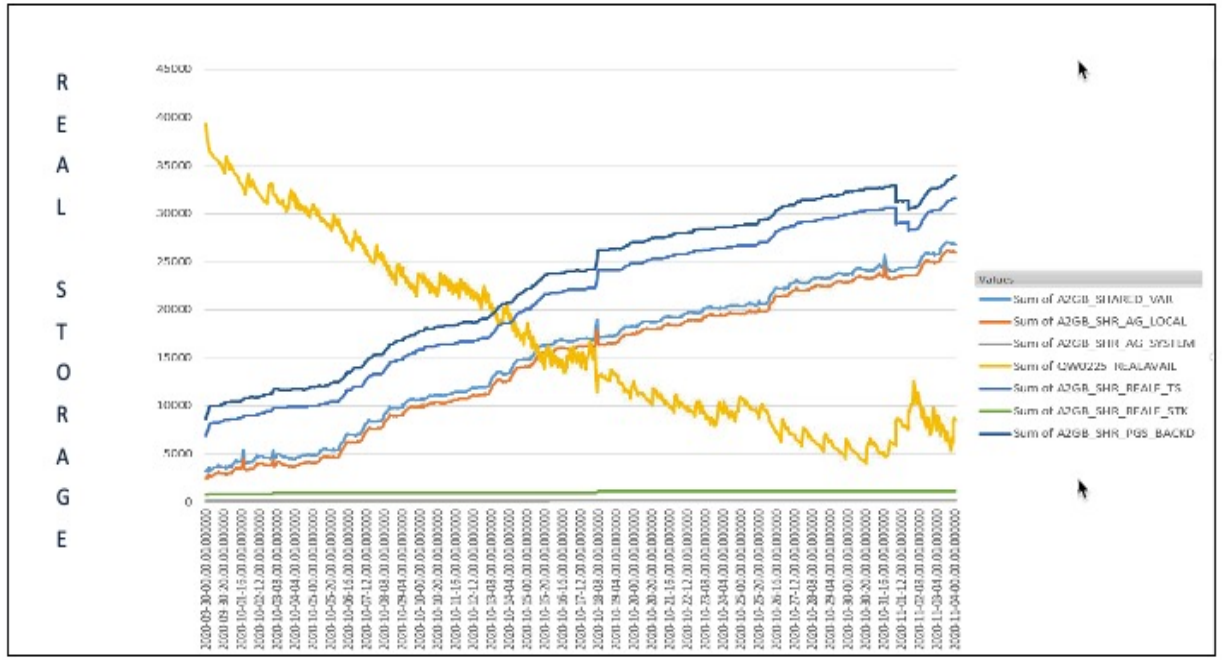


Db2 System Performance Management

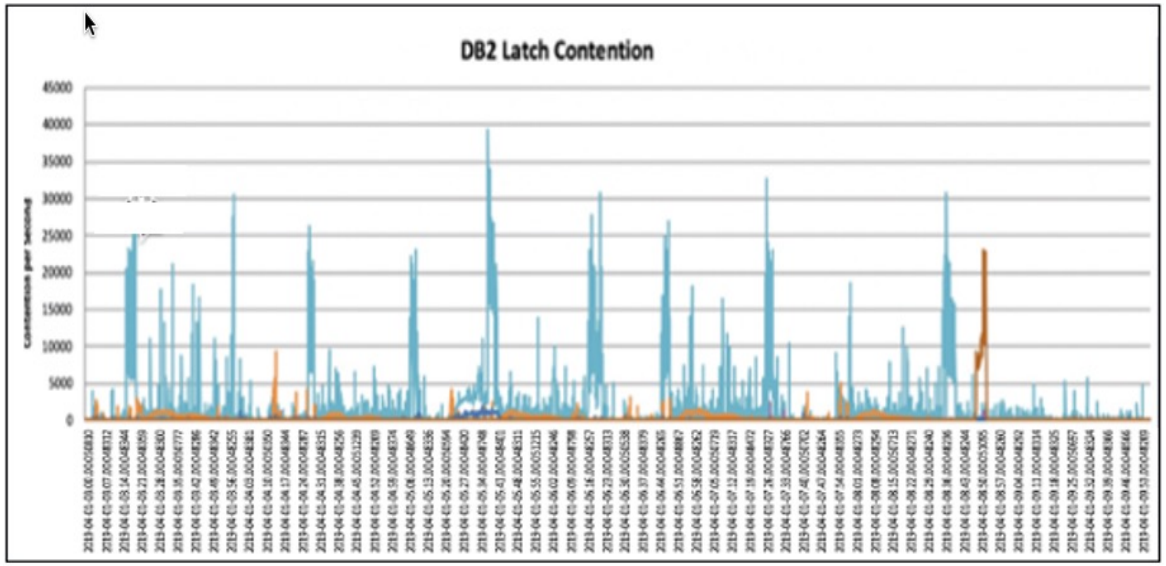




- Graphical/baseline reporting example
  - Real Memory LPAR usage
  - Real Memory storage growth

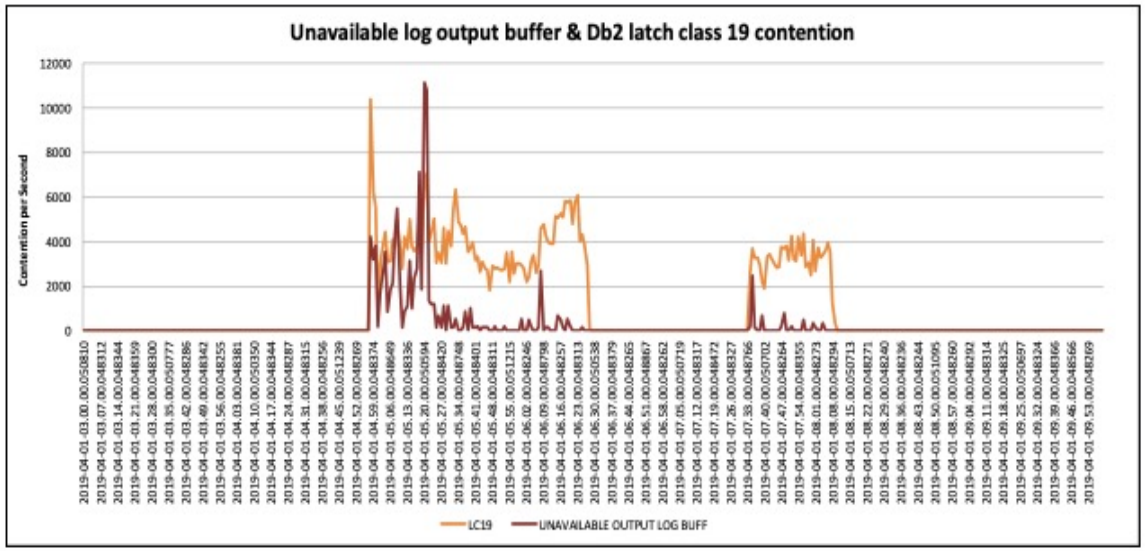


Db2 System Performance Management …



- Risks/Concerns
  - Delays in:
    - Transaction processing and
    - Recovery actions

- Findings
  - Consistent spikes in latch class 19
  - Observed periods of unavailable log output buffer condition
    - Associated elevated Latch Class 19 contention



Db2 System Performance Management …

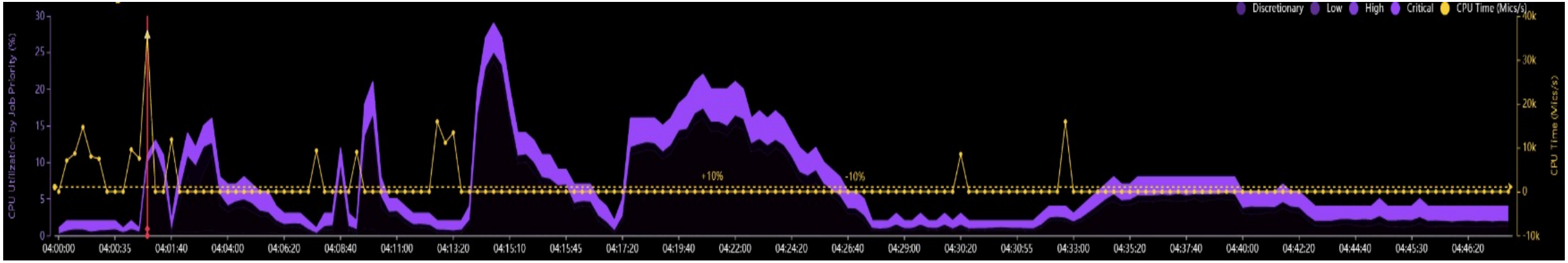
- Exception threshold reporting example

MIN OR MAX ALERTS on KEY INDICATORS

EDM POOL		***THREADS and DBATs***		***DB2 INTERNAL LATCH***		***RID LIST PROCESSING***		***GLOBAL DDF ACTIVITY***	
FAILS - EDM POOL FULL	0.00	NUMBER OF ALLIED THREADS	561.00	LC01	0.00	RID NOT USED-NO STORAGE	0.00	CONN REJECTED-MAX CONNECTED	0.00
FAILS DUE TO RDS POOL FULL	0.00	HIGH WATER MARK CTHREAD	562.00	LC03	0.02	RID NOT USED-MAX LIMIT	0.00	CONN CLOSED - MAX QUEUED	0.00
FAILS DUE TO DBD POOL FULL	0.00	QUEUED AT CREATE THREAD	0.00	LC04	0.00	RID RDS LIMIT EXCEEDED	4.80	CONN CLOSED - MAX WAIT	0.00
FAILS DUE TO SKEL POOL FULL	0.00	CUR ACTIVE AND DISCON DBATS	257.00	LC05	2.67	RID DM LIMIT EXCEEDED	0.05	CUR INACTIVE CONNS (TYPE 2)	5444.00
DBD HR %	99.96	HWM ACTIVE AND DISCON DBATS	408.00	LC06	235.04	RID PROC.LIMIT EXCEEDED	0.00	HWM INACTIVE CONNS (TYPE 2)	5473.00
CT HR %	94.59	CUR ACTIVE DBATS-BND DEALLC	0.00	LC07	0.80	RID OVERFLOWED-NO STORAGE	0.00	HWM TOTL REMOTE CONNECTIONS	5520.00
PT HR %	93.23	HWM ACTIVE DBATS-BND DEALLC	0.00	LC08	0.02	RID OVERFLOWED-MAX LIMIT	0.22	CUR QU INACT CONNS (TYPE 2)	2.00
		DBAT/CONN QUEUED-MAX ACTIVE	0.00	LC09	0.02	RID INTERRUPTED (HJ)-NO STORAGE	0.00	MAX QUEUE TIME	17.292
DYNAMIC STATEMENT CACHE				LC10	33.72	RID INTERRUPTED (HJ)-MAX LIMIT	0.00	AVG QUEUE TIME	0.057
GDSC HR %	14.58	***DBM1 VSTOR AND REAL***		LC11	0.02			HWM QU INACT CONNS (TYPE 2)	205.00
LDSC HR %	-	FULL SYSTEM CONTRACTIONS	0.00	LC12	42.67	***WORKFILES***		CUR ACTIVE AND DISCON DBATS	257.00
CACHE LIMIT EXCEEDED	-	CRITICAL SHORTAGES	0.00	LC13	7.42	WF 4K USED INSTEAD OF 32K TS	0.00	CUR DISCON DBATS NOT IN USE	227.00
PREP STMT PURGED	27.17	ABENDS DUE TO SHORTAGES	0.00	LC14	864.55	WF 32K USED INSTEAD OF 4K TS	0.00	HWM ACTIVE AND DISCON DBATS	408.00
		DBM1 31-VSTOR USED	475.40	LC15	0.00	WF NUMBER OF MAX EXCEEDED	0.00	DBATS CREATED	2.43
OPEN/CLOSE ACTIVITY		MB LEFT BEFORE SC	618.96	LC16	6.53	WF IN-MEMORY NOT CREATED	0.00	DISCON (POOL) DBATS REUSED	11373.85
MAX OPEN DATASETS	33023.00	REAL STORAGE (MB)	190399.95	LC17	0.00	SPARSE IX DISABLED	0.00		
DATASETS CLOSED-THRESH.REACHED	0.00	AUX STORAGE (MB)	54.16	LC18	0.32	SPARSE IX BUILT WF	0.00	***AUTHORIZATION MANAGEMENT***	
CONVERT_RW_TO_RO/minute	2411.00	(S) QW0225_REALAVAIL (MB)	268.80	LC19	9360.52			TOTAL AUTH SUCC RATE	99.93
		(S) QW0225DISC	0.00	LC20	124.93	***MISCELLANEOUS***		PACK CACHE HR %	69.92
LOG ACTIVITY		(S) QW0225_ECSA_CONV (MB)	0.00	LC21	9.53	BYPASS COL	0.00	ROUTINE CACHE HR %	99.37
LOG READS SATISFIED-OUTPUT BUFF	8223.34			LC22	0.00	LOCK TABLE	0.23		
LOG READS SATISFIED-ACTIVE LOG	59799.01	***LOCKING***		LC23	1605.74	LOCK ESCALATION (SHARED)	0.00	***IFC DESTINATION***	
LOG READS SATISFIED-ARCHIVE LOG	0.00	IRLM LATCH % REQUESTS	31.44	LC24	26685.16	LOCK ESCALATION (EXCLUSIVE)	0.00	ACCOUNTING (NOT WRTN)	568.20
LOG READS TAPE VOLUME CONTENTION WAIT	0.00	TIMEOUTS	0.10	LC25	221.47	INCREMENTAL BINDS	269.33	SMF - NOT WRITTEN	0.00
LOG READ DELAYED-UNAVAIL.RESOUR	0.00	DEADLOCKS	0.03	LC26	1.38	AUTO.BIND PACKAGE ATTEMPTS	0.00	GTF - NOT WRITTEN	0.00
UNAVAILABLE OUTPUT LOG BUFF	0.00	UNLOCK/commit	288.49	LC27	3.32	PKLIST SUCCESS RATE	99.89	OP1 - NOT WRITTEN	14.49
OUTPUT LOG BUFFER PAGED IN	0.00			LC28	406.95	SUBSYSTEM ALLIED MEMORY EOT	2.97	OP2 - NOT WRITTEN	0.00
LOG DATA RATE - 1 LOG (MB/sec)	37.80	***DATA SHARING LOCKING***		LC29	547.30	SUBSYSTEM ALLIED MEMORY EOM	0.00	OP3 - NOT WRITTEN	0.00
		GLOBAL CONTENTION %	6.44	LC30	172.44	UNITS OF RECOVERY INDOUBT	0.00	OP4 - NOT WRITTEN	0.00
		FALSE CONTENTION %	3.85	LC31	86.68	UNITS OF REC.INDBT RESOLVED	0.00	OP5 - NOT WRITTEN	0.00
		INCOMPATIBLE RETAINED LOCK	0.00	LC32	988.74	ROLLBACKS %	19.96	OP6 - NOT WRITTEN	0.00
		P-LOCK NEGO % REQUESTS	5.60	LATCH_COU	219.32	CHECKPOINTS/minute	2.00	OP7 - NOT WRITTEN	371085.20
		REJECTED - XES % REQUESTS	10.24					OP8 - NOT WRITTEN	0.00

Db2 System Performance Management …

- Recommendations …
  - Organizations should fund this discipline in terms of people resource and specialist skill level
  - KPIs need to be established and graphical interface or dashboard reporting infrastructure should be established for key metrics or APIs
    - Captured statistical data should be turned around to be easily consumed and analyzed
  - Baseline (normal data) points need to be tracked and adjusted over time
    - Appropriate granularity
      - If SMF data is too excessive consider turning on SMF compression in Db2 ZPARMS (SMFCOMP=YES)
      - Operational problem diagnosis
      - Analytical data for trending purposes



HiperDispatch and efficiency differences between VH, VM, VLs

- HiperDispatch (HD) plays a vital role on the latest CEC models where cache performance has such a big impact
  - PR/SM and z/OS Dispatcher interfaces are establishing an affinity between
    - Units of work and logical CPs
    - Logical CPs and physical CPs
- Impact is the increased likelihood of a unit of work being re-dispatched to the same logical CP and executing on the same or nearby physical CP
  - Optimizes the effectiveness of processor cache at every level, by reducing the frequency of processor cache misses
  - By reducing the distance (into the Nest) required to fetch data

HiperDispatch and efficiency differences between VH, VM, VLs …

- With HiperDispatch active, based on LPAR weights and number of physical CPs, PR/SM will assign logical as
  - Vertical High (VH): 1:1 relationship with physical CP
  - Vertical Medium (VM): 50% share of CP
  - Vertical Low (VL): Low share of physical CP
    - Subject to being ”parked” when not in use
    - Few seconds to “unpark”
    - ~10 second WLM interval

HiperDispatch and efficiency differences between VH, VM, VLs …

- Customer example related to zIIP pool of processors
  - Customer added 2 additional zIIP physical processor to the CEC, but did not see much better zIIP offload
  - # of logical zIIP engines and relative LPAR weights
    - Did not align with documented best practices for HD
    - Did not match with the actual demand from respective LPARs
    - No changes were made
  - Consequences
    - Unnecessary redirect of zIIP eligible workload to the GCPs
    - GCP resource was already constrained during the peak periods
    - Loss of TCO benefit and introducing elapsed time latency for application processes
  - Recommendations for this customer situation
    - Align # of logical zIIP engines and relative LPAR weight with the actual demand for zIIP capacity demand from LPAR
      - Need more dedicated zIIP capacity for the LPAR e.g.
        - Increase weight of LPAR to ensure that at least 3 VH engines are assigned
      - Should reduce zIIP redirect to GCP and increase efficiency of zIIP dispatching

Huffman Compression

- Huffman Compression introduced in Db2 12 Function Level 504
  - Just like Lempel/Ziv, Huffman is a dictionary-driven compression and decompression algorithm
    - Fixed Length vs Huffman compression chosen at ZPARM
  - The entries in the dictionary are sorted by frequency of occurrence with the highest frequency get a shorter bit pattern to identify the entry in the dictionary
  - Prerequisites
    - z14 for the improved CMPSC instruction
    - Function Level 504 of Db2 12
  - No dependency on the Integrated Accelerator for zEDC technology on the z15
  - Seen up to 40% (avg. 20-30%) improvement compared to legacy fixed length compression
  - Wide range of variability in terms of CPU and elapsed time performance (+ / -)
  - Some cases may see CPU and elapsed time reduction e.g., sequential processing

Huffman Compression …

- Huffman Compression introduced in Db2 12 Function Level 509
  - Huffman compression can now be used at a tablespace/partition granularity level
  - DSN1COMP enhancements made for Huffman
  - New columns added to the Db2 Catalog
    - SYSIBM.SYSTABLESPACE.COMPRESS
    - SYSIBM.SYSTABLPART.COMPRESS
    - SYSIBM.SYSTABLEPART.COMPRESSED_USED
  - The use of Huffman compression should be evaluated object by object
- Recommendations
  - Do not implement Huffman compression until the following is in place:
    - Applying regular scheduled drops of preventative service for Db2
    - Db2 provides object level control (Function Level 509)
  - Implement in a controlled and incremental rollout

Universal Tablespace Strategy and Execution

- Universal Tablespaces (UTS) is the current and future direction of the Db2 for z/OS Development Lab
  - Non-UTS tablespaces deprecated in Db2 Function Level 504
  - Non-UTS Tablespaces will be retired in the future as part of a future function level or a future Db2 release
  - Multi-table tablespace conversion to PBG
    - Capabilities will be supplied in future function level, point in time targeting FL508
    - Increase in the total number of open data sets in Db2
- Customers need a regimented cook-book for creating new tablespaces
  - UTS - PBG – Partitioned by Growth is a great replacement for the classic segmented tablespaces
  - UTS - PBR – Partitioned by Range is a natural evolution from moving from classic partitioned
  - UTS - PBR RPN – Partitioned by Range Relative Page Number for supersized UTS PBR
- Customers need a plan to take-action for existing tablespaces
  - Develop a plan and a strategy to migrate from classic segmented & table controlled partitioned non-UTS
  - Develop a plan and a strategy to correct previous incorrect decisions to move to UTS PBG
    - Enable partition level independence and parallelism
    - Eliminate technical debt, protect against CPU burn

Universal Tablespace Strategy and Execution …

- Primary driver for the developing UTS PBG tablespace was the removal of the 64GB limit for classic segmented tablespace & avoid the disruptive migration to classic partitioned tablespace
- Some considerations
  - All indexes are going to be NPIs
  - Limited partition independence for utilities (REORG, LOAD)
  - Partitioning not used for query parallelism
  - Degraded insert performance (free space search) as the number of partitions grow
  - REORG Considerations
    - REORG PART will fail for a full UTS PBG partition if FREEPAGE or PCTFREE are non-zero
    - Setting system parameter REORG_DROP_PBG_PARTS = ENABLE could lead to operational issues if the number of PARTs are pruned back
      - No point-in-time recovery prior to the REORG that prunes partitions
      - Cannot use DSN1COPY to move data between Db2 systems
- Should not be using UTS PBG as the design default for all tables (with large number of partitions)

Universal Tablespace Strategy and Execution …

- General recommendations for use of UTS PBG tablespace
  - Only use UTS PBG tablespace as the alternative and replacement for classic segmented tablespace
  - A table greater than ~64GB in size should be created as a UTS PBR tablespace
  - Good reasons to limit number of partitions - should have as few partitions as possible - ideally only 1
  - DSSIZE and SEGSIZE should be consistent with the target size of the object e.g.
    - Small size object: DSSIZE = 2GB and SEGSIZE = 4
    - Medium size object: DSSIZE = 4GB and SEGSIZE = 32
    - Large size object: DSSIZE = 64GB and SEGSIZE = 64
  - REORG at the table space level unless do not have sufficient DASD space for sort
  - Setting system parameter REORG_DROP_PBG_PARTS = DISABLE?
    - If required to prune back the number of partitions
      - Use online system parameter to temporarily enable for controlled use
    - Better still, in Db2 12, use the DROP_PART YES option of REORG

Hidden ROWID support to partition

- ROWID can be used as a partitioning column
- Application impact if ROWID cannot be hidden
  - APARs to support to define a hidden ROWID
    - PI76972, PI77310, PI77302 (Db2 12)
    - PI77718, PI77719, PI77360 (Db2 11)
- Benefits
  - Allows table to be partitioned where no natural partitioning key exists or candidate partitioning keys do not provide a good spread across partitions
  - Transparent to the application
  - Improved insert throughput
    - Less lock/latch contention on index and data

```
CREATE TABLE  PRDA.ZJSCNTP0
      (  CLIENT    VARGRAPHIC(3) NOT NULL,
         WI_ID     VARGRAPHIC(12) NOT NULL,
         LENGTH    SMALLINT,
         DATA      VARCHAR(1000),
         ROW_ID    ROWID NOT NULL
         IMPLICITLY  HIDDEN generated always
      )  PARTITION BY (ROW_ID)
          (PARTITION 1 ENDING AT (X'0FFF'),
           PARTITION 2 ENDING AT (X'1FFF'),
           PARTITION 3 ENDING AT (X'2FFF'),
           PARTITION 4 ENDING AT (X'3FFF'),
           :
           PARTITION 14 ENDING AT (X'DFFF'),
           PARTITION 15 ENDING AT (X'EFFF'),
           PARTITION 16 ENDING AT (MAXVALUE))
```

Migration of Multi-Table TS to PBG UTS (FL 508, PH29392)

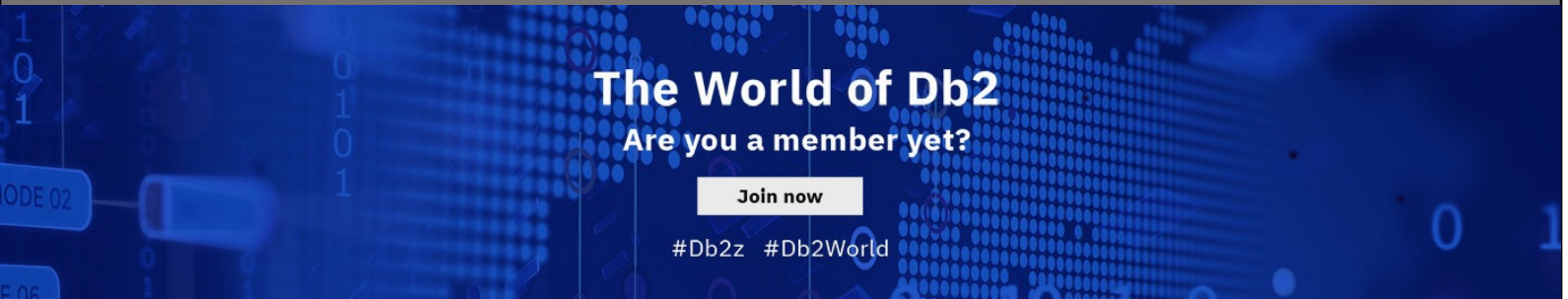
- V12R1M508 enables the ability to migrate multi-table tablespaces to a PBG UTS via a pending Db2 alter and a materializing online Db2 REORG:
  - Create new target PBG UTS target tablespaces
    - New target tablespace needs to be created in advance
    - CREATE TABLESPACE NEWTS1 IN DB1 MAXPARTIONS 1 DEFINE NO;
    - CREATE TABLESPACE NEWTS2 IN DB1 MAXPARTIONS 1 DEFINE NO;
      - DEFINE NO
      - MAXPARTITIONS 1
      - LOGGED/NOT LOGGEWD attribute needs to match source TS
      - CCSID values need to match source TS
  - Alter existing multi-table tablespace
    - ALTER TABLESPACE DB1.TS1 MOVE TABLE TB1 to TABLESPACE DB1.NEWTS1;
    - ALTER TABLESPACE DB1.TS1 MOVE TABLE TB2 to TABLESPACE DB1.NEWTS2;
  - Reorg existing ”alter pending” tablespace (all tables moved)
    - For each table moved:
      - Linear increase time required to perform ALTER statement & REORG SWITCH
  - Drop “old” tablespace that previously was the multi-table, tablespace
    - Only after all tables have been moved to target and not tables exist in tablespace

Db2 12 Migration resources

- **John Campbell's webcasts**
  http://ibm.biz/Db212TechnicalRoadshow
  - Db2 12 Technical Overview Parts 1 & 2
  - V12 Migration Planning and experiences
- **Db2 Master Class**
  - Held twice a year in June and September
  - EMEA - June 21, 2021 (9:00am London)
  - United States - September 27, 2021 (8:30am EST)
  - Held as virtual classes in 2021 because of COVID-19

  https://ibm.biz/Db2ZMasterClass2021

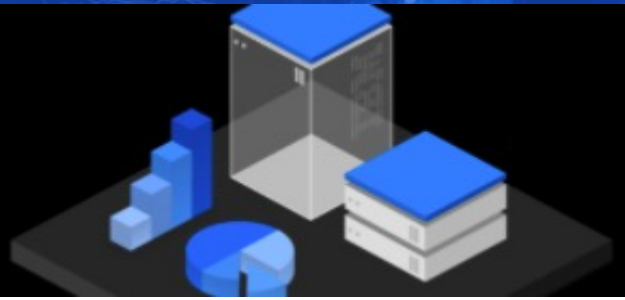
- **Join the World of Db2 for additional webcasts and materials**
  - www.worldofdb2.com



"It's all about robustness"

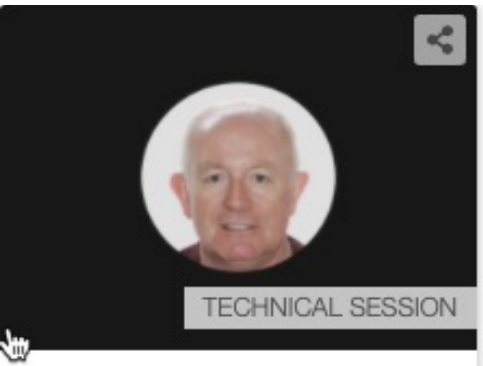
IBM Db2 for z/OS

An AI infused enterprise data server for Hybrid Cloud and business critical transactions



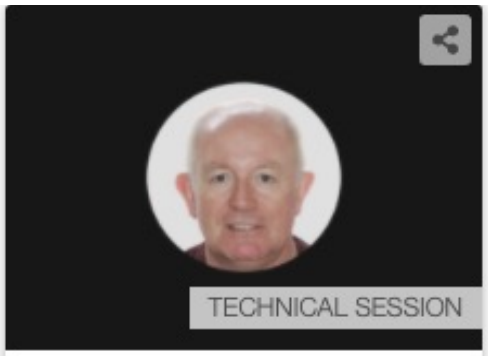
Get a head start for 2021

IBM



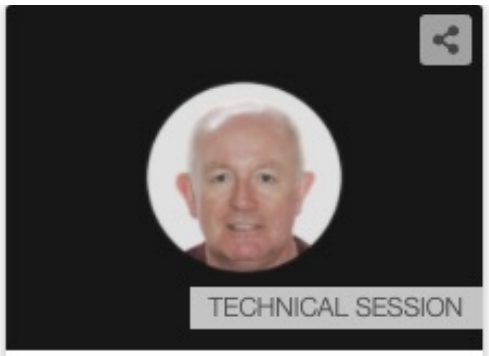
On Demand

Session 7. Primer on Insert Algorithm 2 with Db2 12



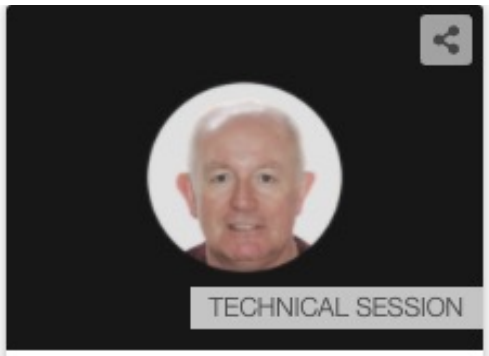
On Demand

Session 6. Database Housekeeping - recommended best practice for generating REORGs



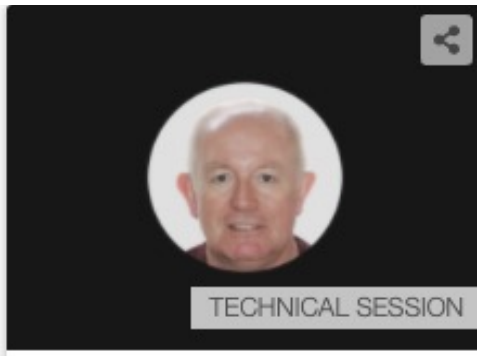
On Demand

Session 5. Point In Time Recovery in Db2 12 with RECOVER and SCOPE UPDATED



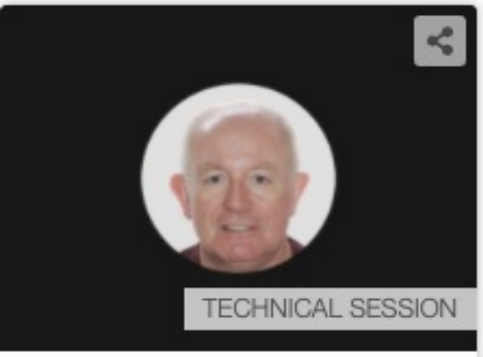
On Demand

Session 4. Saving CPU: Using thread reuse and BIND option RELEASE DEALLOCATE



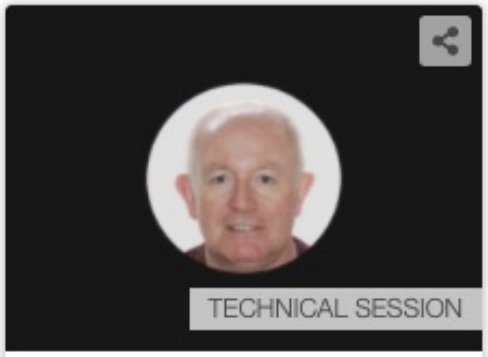
On Demand

Session 3. Latest news on in-memory performance optimization (FTB) in Db2 12 for z/OS



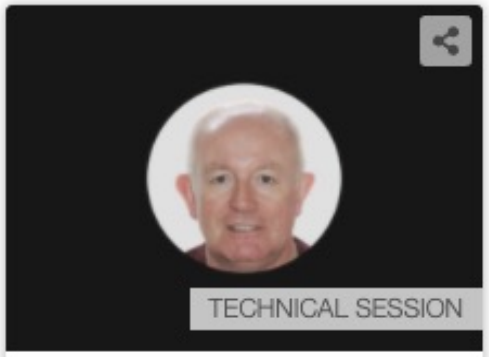
On Demand

Session 2. Dos and Dont's about Continuous Delivery



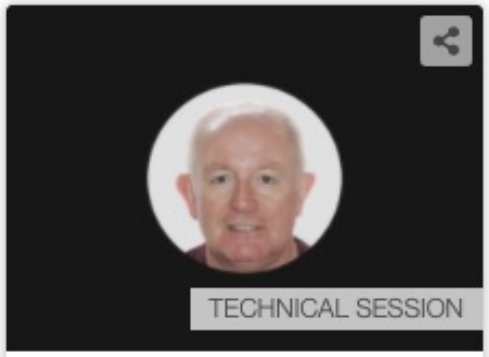
On Demand

Session 1. Planning your Db2 12 for z/OS migration strategy



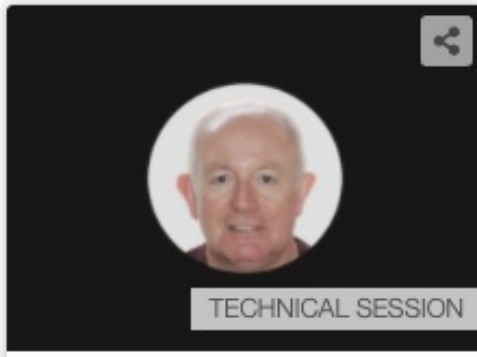
On Demand

Db2 for z.OS: Hot Topics and Best Practices with John Campbell PART 2



On Demand

Db2 for z/OS: Hot Topics and Best Practices with John Campbell PART 1



On Demand

Db2 for z/OS: Trends & Directions and Latest Updates

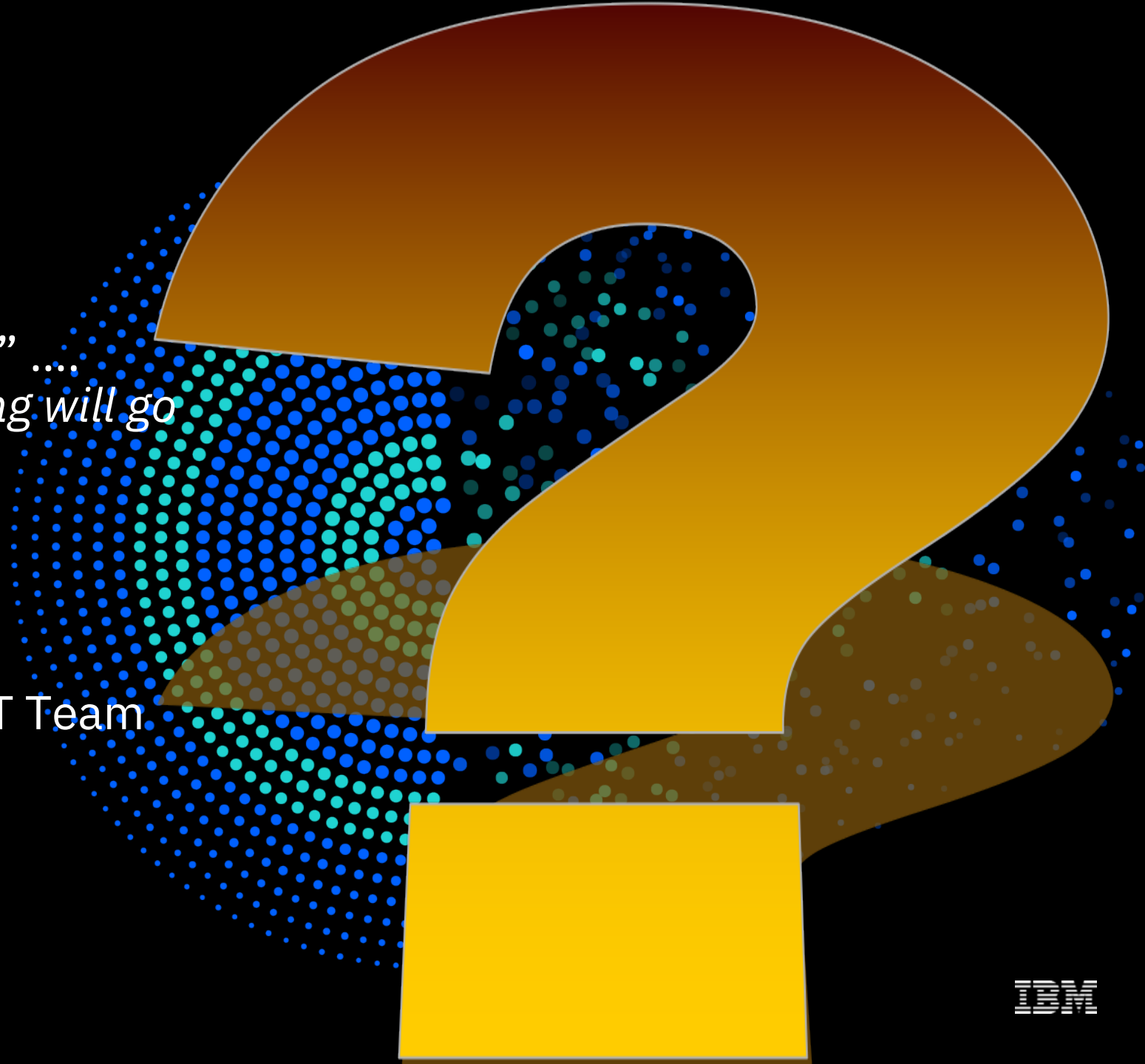
Thank You

IBM®

Data and AI

Db2 for z/OS SWAT Team “*Tales*” ….
“Murphy's Law What can go wrong will go wrong”

Anthony Ciabattoni
Executive IT Specialist
Db2 for z/OS Development SWAT Team
aciabattoni@ibm.com



IBM