DB2 for z/OS – Ultimate Database for Cloud, Analytics and Mobile

Industry-leading performance, security, scale and reliability

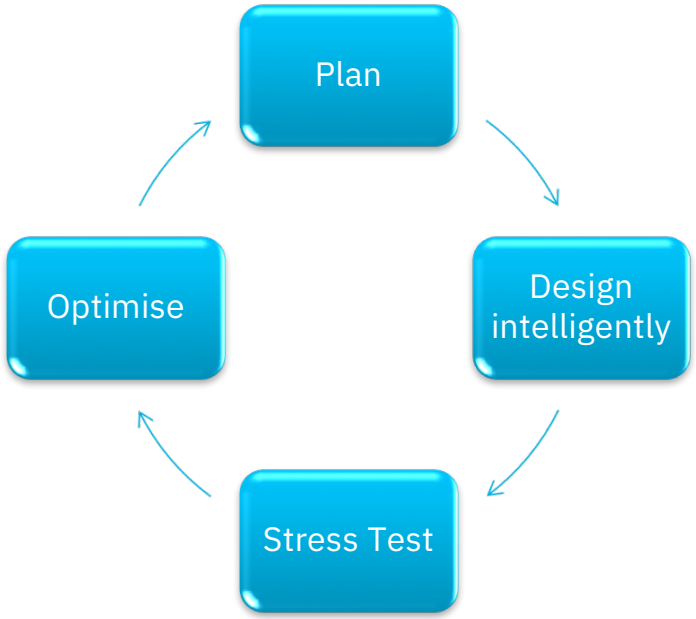
Mass Db2 Application Recovery….

"The Nightmare of a Db2 Professional"

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

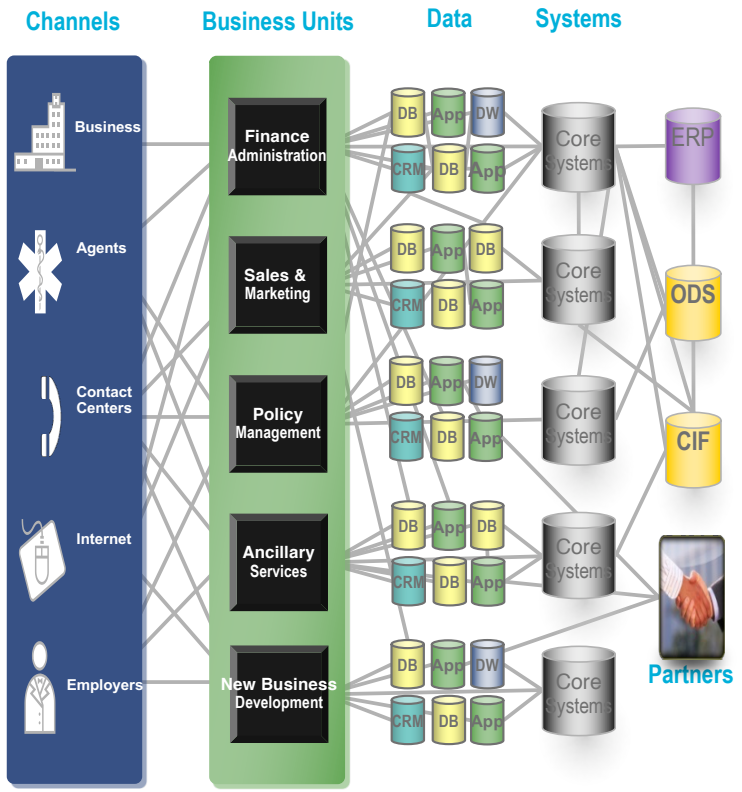
Agenda

- Introduction
- High performance multiple object recovery – common issues
- Plan
  - Logging environment
  - Db2 catalog/directory
  - Image copy backups
- Design intelligently
- Stress test
- Optimize
  - COPY/RECOVER vs. REBUILD INDEX
  - FlashCopy
  - RECOVER … BACKOUT YES
  - Application design



Introduction

- Recovery Execution is Complicated!
  - Consistent recovery point
  - What tables need to be recovered
    - Table priority
    - Application integrity
  - Recovery is normally not practiced or discussed
  - Recover *"Mode Fast"*
  - Immaculate execution is required
  - Recovery assets and procedure are not discussed until they are needed
  - Backup processes directly impacts speed of recovery
- Multi-object/Mass Recovery introduces additional complications and complexities



Introduction

- 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'

Introduction ...

- 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

- 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
    - 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

High performance multiple object recovery – Common issues ...

- Results: any or all of the following
  - No estimate of elapsed time to complete
  - Elongated elapsed time to complete recovery
  - Performance bottlenecks so that recovery performance does not scale
  - Breakage in procedures
    - Revert to trial and error approach
  - Surprises caused by changing technical configuration
  - Unrecoverable objects

Factors that affect the recovery elapsed time

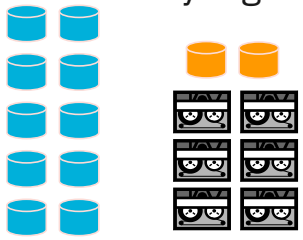
- 'Think time' and preparation of the recovery plan
- Restore
  - Number of pages, number of objects?
  - Backups on tape or DASD? Standard ICs, FCICs, SLBs?
  - Degree of parallelism?
- Log scan
  - Backup frequency
  - Archive logs needed to recover?
  - Archive logs on tape or DASD?
  - Degree of parallelism?
- Log apply
  - Update frequency and update patterns
  - Maximal fast log apply?
- Recover/Rebuild indexes

Logging environment

- Design for availability
  - Keep a minimum of 6 hours of recovery log data in the active log pairs at any time
    - Objective: provide some reaction time in case of archiving problem
    - Adjust number/size of the Db2 active log pairs (limit is 93 log pairs)
    - Db2 10 – Active logs can be added dynamically
      - ✓New -SET LOG NEWLOG option
      - ✓New active log must be IDCAMS defined & preformatted by DSNJLOGF
      - ✓Only a single log dataset at a time
        - » Issue command twice for dual logging
      - ✓No dynamic delete of active logs
- Design for recovery performance
  - Always write archive log COPY1 and COPY2 to DASD, and let DFSMShsm (or equivalent) migrate them away to tape or VTS
    - Eliminate contention on reading the archive logs during recovery
    - Especially important in a data sharing environment
    - Storage needs to be available and processes to recall from tape/VTS prior to recovery

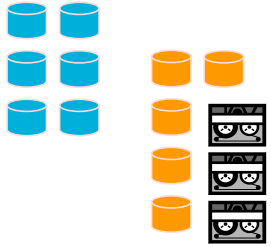
Logging environment ...

- 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
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

Logging environment ...

- Design for resiliency
  - Separate COPY1 and COPY2 of the active log pairs and BSDS across different DASD controllers if possible – across different extent pools (RAID arrays) at the minimum
  - Isolate objects into separate ICF user catalogs
    - Separate out the datasets for each Db2 member into separate ICF catalogs
      - ✓Active logs, archive logs, BSDS for member Db2A away from those for member Db2B
      - ✓Result: an ICF catalog failure would only affect one Db2 member
    - Should also consider further isolation
      - ✓COPY1 of active log, archive log and BSDS into one ICF catalog
      - ✓COPY2 of active log, archive log and BSDS into an alternate ICF catalog
      - ✓Result: an ICF catalog failure would not affect Db2
    - Additional ICF catalog considerations for better performance and resilience
      - ✓Isolate Db2 catalog/directory objects into a separate ICF catalog
      - ✓Use multiple ICF catalogs for the Db2 user objects
      - ✓Separate ICF catalogs for Db2 objects and Db2 image copy backups

Logging environment ...

- Design for serviceability
  - Retain archive log data for 30 days
  - At first sign of logical data corruption, stop the deletion of image copies and archive log datasets
  - Keep the maximum number of archive logs in the BSDS
    - Set zparm MAXARCH=10000

Db2 catalog/directory

- Take frequent FICs of the Db2 catalog/directory
  - At the very minimum daily – best is several times a day using SHRLEVEL(CHANGE)
  - Keep a copy on DASD to speed up recovery
- Prepare and maintain a JCL to recover the Db2 catalog/directory
  - Db2 10 and above – Db2-managed objects simplify procedures
- Db2 11 enhancements for faster Db2 catalog/directory recovery
  - CM - Enable SYSLGRNX recording for
    - DSNDB01.SCT02, DSNDB01.SPT01, DSNDB01.SYSSPUXA, DSNDB01.SYSSPUXB
    - Indexes over the above table spaces
  - NFM - RECOVER utility uses the SYSLGRNX records to selectively read and apply the log records for ranges of updates

Db2 catalog/directory ...

- Periodically test your recover procedure of the Db2 catalog/directory
  - Correctness
  - Recovery timings
    - Influence the number of copies per 24 hours
- Periodically check the integrity of the Db2 catalog/directory
  - e.g. using a cloned copy of the Db2 catalog/directory into an auxiliary Db2 subsystem
  - See next slide for recommended checks
- Periodically reorganize the Db2 catalog/directory
  - Outside of release migration
  - Most importantly, SYSLGRNX should be reorganized at least every quarter
    - Can be run as SHRLEVEL(CHANGE) at a time of low activity
    - Will speed up online REORG, MODIFY RECOVERY, RECOVER, GRECP/LPL recovery

Db2 catalog/directory ...

- Series of tests that should be run on a regular basis to flush out any latent inconsistency in the Db2 catalog
  - SQL queries from migration job DSNTESQ
    - Should always return zero rows
  - REPAIR DBD TEST or DIAGNOSE
  - Basic RUNSTATS on all objects
  - CHECK INDEX on all indexes
  - For catalog objects with LOB columns:
    - CHECK LOB
    - CHECK INDEX on AUX index
    - CHECK DATA on base tablespace using SCOPE AUXONLY AUXERROR REPORT

Image copy backups

- Always take dual image copies as part of REORG and LOAD REPLACE (LOG NO events)
- Use as much DASD as possible for optimal recovery
  - If DASD space is an issue
    - Use template switching to write image copies for small objects to DASD and manage by DFSMShsm
      - ✓Objective: Allow fast restore and take pressure off the VTS in case of mass recovery

```
TEMPLATE LRG DSN &DB..&TS..D&DA..T&TI. UNIT=TAPE
TEMPLATE SML DSN &DB..&TS..D&DA..T&TI. UNIT=SYSALLDA LIMIT(20 CYL,LRG)
COPY TABLESPACE SMALL.TS COPYDDN(SML)
COPY TABLESPACE LARGE.TS COPYDDN(LRG)
```

    - Use Db2 data compression for table spaces → COPY does not decompress data
- Consider shortening the full image copy (FIC) cycle time (<= 24 hours) for Db2 Catalog and Directory and most critical application data
  - Objective: Reduce log apply time
  - Implement a smart image copy process

Image copy backups

- Consider use of incremental image copy (IIC)
  - Keep the IIC on DASD – otherwise, perform regular MERGECOPY in background
  - Additional considerations
    - If no pages have been updated since the last image copy
      - ✓Db2 9/10
        - » An empty copy data set is always created
        - » No SYSCOPY record is inserted for that image copy data set
      - ✓Db2 11 – COPY … FULL NO and COPY … CHANGELIMIT now check RTS
        - » Incremental copy will not allocate the copy dataset if RTS shows no pages changed
        - » No dummy SYSCOPY record is inserted
    - Objects defined with TRACKMOD NO
      - ✓Can still use COPY … FULL NO to create an IIC, but COPY will do a complete TS scan
      - ✓Cannot use COPY … CHANGELIMIT
    - Db2 11 – CHANGELIMIT option on COPY is deprecated
      - ✓Alternative is to use DSNACCOX to drive the COPY jobs

Image copy backups ...

- Recommend NOT to use GDGs for image copy datasets
  - Risk of old versions rolling off by accident
    - Especially if using incremental image copies
- Use catalogued datasets instead, with 'meaningful' naming convention
  - Adds informational value (e.g. date and time of the backup)

- Db2 11 – Partition-level inline image copy
  - Faster partition-level recovery from inline image copy
  - Create partition-level inline image copies if using TEMPLATE with &PA or &PART
    - No new option or keyword on REORG
    - Support substring notation with &PA as long as substring ensures uniqueness
    - Support writing to tape as long as STACK YES not specified
  - This is already the behavior for inline FlashCopy image copies in Db2 10

Image copy backups ...

- Schedule a daily production job to check for unrecoverable objects
  - Ensure a valid backup exists and there is enough recovery log data to recover from it
  - If IIC are used, ensure a FIC is also available
  - Check should take into account LOG NO events and materializing REORGs
  - Db2 11 lifted many restrictions on PIT recovery prior to materializing REORG
    - PIT recovery restrictions lifted for
      - ✓ LOB table spaces
      - ✓ XML table spaces
      - ✓ PBR table spaces
        - » Including when immediate ALTERs have occurred since materializing REORG
    - PIT recovery restrictions still in place
      - ✓ Table space conversion
      - ✓ PBG table spaces
      - ✓ PBG partition pruning
      - ✓ Online DROP COLUMN

Design intelligently

- 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

Design intelligently ...

- 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

Design intelligently …

- Db2 12 recover utility - SCOPE UPDATED
  - Default … only recover objects that have changed
- Db2 12 Rebuild Index SCOPE PENDING
  - Default … SCOPE ALL
- Recommendation
  - RECOVER Tablespace SCOPE UPDATED + REBUILD Index SCOPE PENDING
  - RECOVER Tablespace SCOPE ALL + REBUILD Index SCOPE ALL

Design intelligently …

- Objects with longest end-to-end recovery time need to be recovered first
  - Size of the object
  - Update rate since last image copy
  - Number and size of indexes

- Optimize job scheduling to avoid ‘dead times’

Design intelligently …

- Create automated procedures to create efficient recovery jobs
  - Considerations
    - Table prioritization
    - Virtual Tape/Tape optimization
  - Frequency
    - Execute after nightly backup jobs
      - ✓ Recovery jobs updated daily and ready to execute
      - ✓ Execute when CPU is available (middle of night)
    - Procedures in place to execute efficiently at time of recover
      - ✓ Automated process(REXX) to execute and create recovery jobs at time of recovery
      - ✓ Needs to be efficient
      - ✓ Procedures need to be tested and proved out periodically

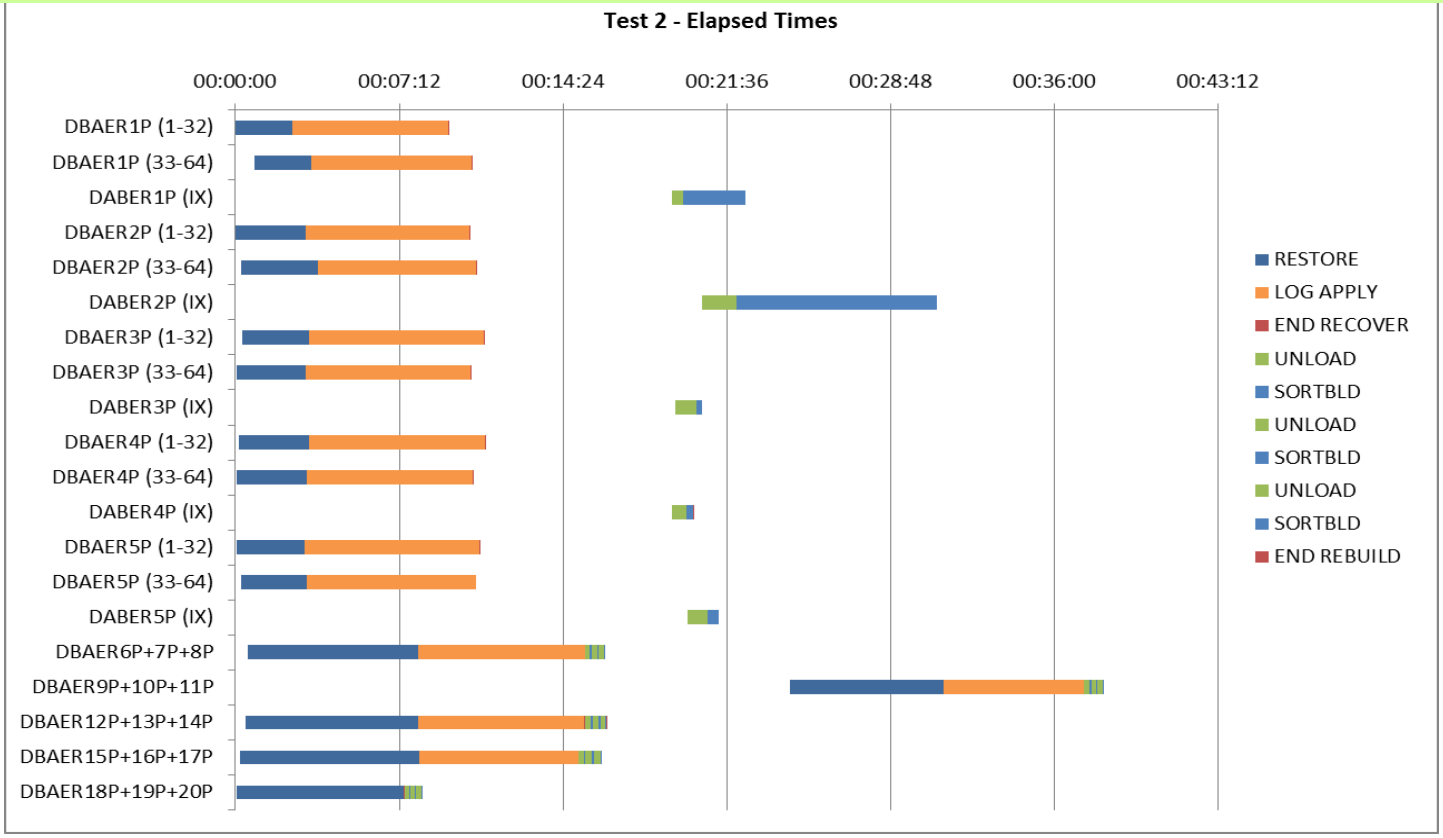
Stress test

- Practice regular full-scale ‘fire drills’ for mass recovery of an entire application or even the entire system

- Objectives:
  - Validate that procedures are in working order
    - Both for local and remote DR recovery
  - Maintain readiness on mass recovery execution
  - Find out what the actual service level is
    - Break down the elapsed time of each job: RESTORE/LOG APPLY/REBUILD INDEX
    - If elapsed time needs to be improved further, look for possible optimizations

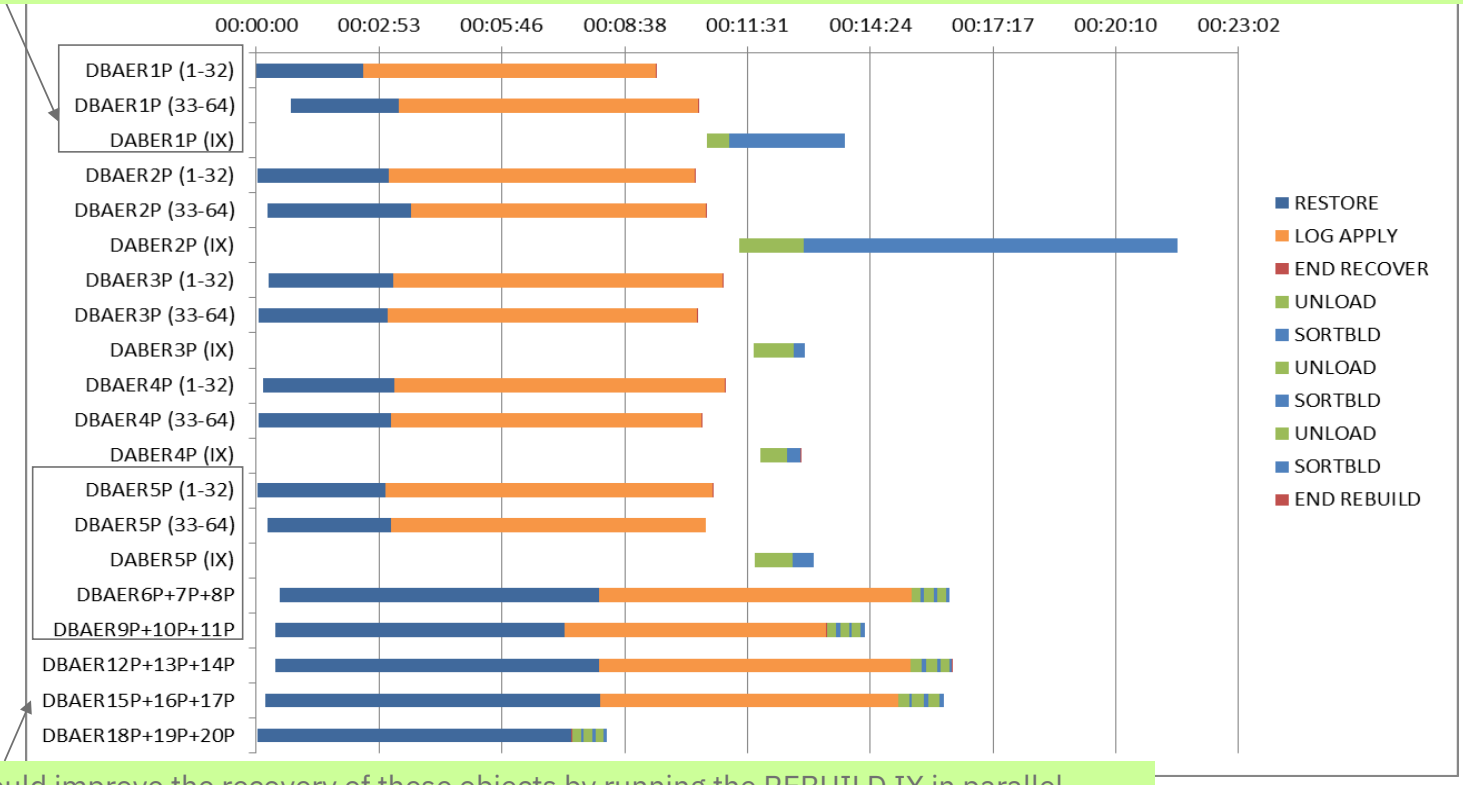
Stress test …

Lot of 'dead times' introduced by the job scheduling >> next slide will show how this test would have looked if it took only 1 minute to start REBUILD INDEX after the RECOVERY of all parts + if the job #17 had not been started late



Stress test …

Options for optimization: IC on DASD or FCIC to improve RESTORE, more frequent IC to reduce LOG APPLY, use COPY/RECOVER instead of REBUILD for the index



Could improve the recovery of these objects by running the REBUILD IX in parallel

COPY/RECOVER vs. REBUILD INDEX

- Fast Log Apply (FLA) now implemented for RECOVER INDEX
  - Previously Db2 would wait until a log record was to be applied before reading the associated index page into the local bufferpool where it would then be cached
  - Now Db2 will use list prefetch to read all the index pages that are needed to apply log records for, before applying any log record
  - Potential for significant savings in elapsed time
  - Enhancement taken back to V9 and V10 via APAR PI07694

- Should now reconsider COPY YES and image copies for indexes
  - Run RECOVER INDEX in parallel with RECOVER TABLESPACE [PART] vs. wait for RECOVER TABLESPACE [PART] to complete for all parts and then REBUILD INDEX
  - Especially for very large NPSIs

- Note: REBUILD INDEX still preferred option after index vs. table mismatches

FlashCopy – CHECK INDEX/DATA/LOB SHRLEVEL(CHANGE)

- CHECK is a critical tool in case of data corruption
- Without FlashCopy support, CHECK utilities can be very disruptive
  - Even with SHRLEVEL(CHANGE) – R/O access during creation of the shadow objects

- Db2 zparm CHECK_FASTREPLICATION
  - PREFERRED (default V9) >> Standard I/O will be used if FC cannot be used
  - REQUIRED (default V10) >> CHECK will fail if FC cannot be used ← **strongly recommended whether FlashCopy is available or not**

- Db2 zparm UTIL_TEMP_STORCLAS
  - Optional: can be used to specify a storage class for the shadow data sets
  - If blank, the shadow data sets are defined in the same storage class as the production page set
  - If using DASD-based replication, specify a pool of volumes that are not mirrored
    - Applies to Metro Mirror (PPRC) without Remote Pair FlashCopy (ZPARM FLASHCOPY_PPRC = REQUIRED), z/OS Global Mirror (XRC) and Global Mirror

FlashCopy – System-level and object-level backup/recovery

- FlashCopy provides interesting new options to help with mass data recovery
  - PIT backup of the entire system
    - Can be restored quickly – if still on DASD
    - Can also be used to create a 'forensic' system
      - ✓ Quick cloning if the environment away from main production system
      - ✓ Level restored will be to a point in time where the data is known to be good
      - ✓ Application teams can then analyse and reconcile the data contents of the forensic system vs. current damaged system
  - FC image copies
    - Potential for significant elapsed time reduction for the RESTORE phase
    - Can also be used to create a transaction-consistent image copy with COPY SHRLEVEL CHANGE

FlashCopy – System-level recovery outside of Db2's control

- Scope
  - Pool of volumes containing the entire Db2 environment: Db2 catalog/directory, all application data, active logs and BSDS for all members + associated ICF catalogs
- Backup
  - Need to guarantee the I/O consistency of the FlashCopy backup
  - Option 1: Use -SET LOG SUSPEND to temporarily 'freeze' all Db2 update activity
    - For Data Sharing, command must be issued on all the data sharing members
  - Option 2: Use Consistency Group FC to temporarily 'freeze' all I/O write activity
    - Advantage: Can include non-Db2 data in the backup
- Recovery
  - Need to use Db2 warm restart recovery to re-establish application consistency
    - FlashCopy backup is a 'fuzzy' copy - cannot be used as-is
  - For data sharing systems, must delete all CF structures before the Db2 restart
  - With Option 1, can also use RESTORE SYSTEM LOGONLY to bring the system forward (requires -SET LOG SUSPEND to set RBLP – starting point for the recovery)

FlashCopy – Application-level recovery outside of Db2's control

- Scope
  - Pool of volumes containing all objects for one application + associated ICF catalog

- No Db2 warm restart recovery to re-establish application consistency
  - Option 1: Application consistency must be established at the time of the backup
    - Start all table spaces to be copied in read only (RO) status
    - Run a QUIESCE with WRITE(YES) on all table spaces to be copied
    - FlashCopy
    - Start all table spaces in read-write (RW) status
  - Option 2: Must use RECOVER LOGONLY for each object + REBUILD indexes to re-establish the application consistency

RECOVER ... BACKOUT YES

- Db2 10 – BACKOUT YES option for point-in-time recovery
  - Backs out both data (except NOT LOGGED) and indexes (if defined as COPY YES)
    - COPY NO indexes must be rebuilt when backout complete
    - You can ALTER indexes to COPY YES and not produce image copies
      - ✓ SYSLGRNX entries build up
      - ✓ Use MODIFY to delete them (AGE or DATE) even without copies
  - True rollback, not run of generated SQL undo statements
  - Changes are backed out from the current state of the object → not for media recovery
  - Intent: Short backout, not hours/days
    - Fast Log Apply is not used
    - The recovery point must be contained within the Db2 system checkpoints that are recorded in the BSDS for each member
      - ✓ Message DSNU1545I-RECOVER does not process any of the objects and ends with RC8

Application design

- Common problems
  - Applications not committing frequently
  - No clear separation between active and inactive data
  - Critical applications tightly coupled to non-critical applications by shared data
  - Data inter-dependencies across multiple data sources (e.g. Db2/VSAM, Db2/IMS)
- Recommendations
  - Frequent commits in long-running batch applications
    - Dynamic, table driven
    - Application must be restartable from intermediate commit points
  - Separate active from inactive (historical) data
    - Use separate tables
    - Regular, aggressive pruning back of active tables
    - Application toleration of unavailable inactive data
    - Db2 11 – Transparent Archiving

Application design ...

- Recommendations ...
  - Data isolation to de-couple applications
    - Build 'fire walls'
      - ✓ Isolate data used by critical applications from non-critical applications
    - Needs to be considered very carefully
      - ✓ Single integrated data source vs. higher availability (and performance)
      - ✓ Evaluate cost vs benefit
    - Possible techniques
      - ✓ Logical partitioning
      - ✓ Asynchronous processing
      - ✓ Data replication
      - ✓ Duplicate updates
    - Objective – bring back critical applications first to resume availability
      - ✓ Incrementally bring back additional applications
  - Data inter-dependencies across multiple data sources (e.g. Db2/VSAM, Db2/IMS)
    - Use volume-based FlashCopy to establish a cross-database point of consistency
    - Migrate all inter-dependent data to Db2
      - ✓ Greatly simplified operations, increased robustness, reduced overhead

Modify Recovery

- Db2 10 & 11 Behavior
  - Cleaning up all image copies will result in tablespace being left in COPY-Pending (DSNU572I)
  - Deletes recovery information from the Db2 Catalog/Directory but all physical datasets remain available
    - Customer has to manually delete or implement automation outside of Db2 control to delete the datasets

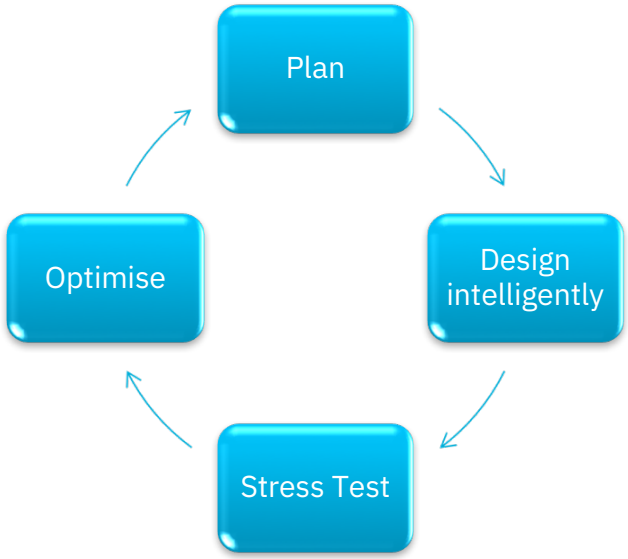
- Db2 12 New MODIFY RECOVERY options
  - NOCOPYPEND
    - When all of the recovery information is erased from SYSCOPY & SYSLGRNX tablespace will not be placed in COPY-Pending status
  - DELETEDS
    - In addition to erasing the recovery information from SYSCOPY & SYSLGRNX the backup datasets are also deleted

Retry of LPL and GRECP Recovery

- Db2 11 behavior
  - During Db2 restart
    - Automatically recovery GRECP and LPL tablespaces
    - Primary use case: Group Restart
  - In some cases not all GRECP or LPL conditions cleared
    - Additional action is required
- Db2 12 behavior
  - Retry of automatic LPL and GRECP recovery
    - Periodically up to 3 times, based on internal interval
    - Reduce instances where DBAs must issue –START DB commands
- APARs
  - Db2 V11 – PI75286 retrofits Db2 GRECP recovery retry functionality to V11
  - Db2 V11 & V12 – PI86567 – New messages
    - DSNB317I
      - ✓ Issued the first time that Db2 detects there are no objects in GRECP
    - DSNB318I
      - ✓ Issued to identify objects which Db2 will not longer attempt automatic GRECP recovery

Summary

- Need to design for high performance and reduced elapsed time
  - Plan, design intelligently, stress test and optimise
    - Prioritise most critical applications
    - Understand application and data interdependencies
    - Design for parallel recovery jobs
    - Optimised utilisation of technical configuration
    - Intelligent creation and scheduling of recovery jobs
  - Design for DASD-based recovery for optimal performance
  - Practice regularly
- Applications and data life cycle also have a role to play…
  - Separate active/operational data from inactive/historical data
  - Perform regular aggressive archiving to historical
  - Allow application toleration of unavailable historical data
  - Look at creating 'fire walls' between applications



Questions