

Data and AI

Mass Db2 Application Recovery....
“The Nightmare of a Db2 Professional”

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

Agenda

- **Background**
- **Common Issues**
- **Db2 Logging Considerations**
- **Db2 Catalog/Directory Best Practices**
- **Backups**
- **Recovery**
- **Recovery Testing**
- **Application Design**

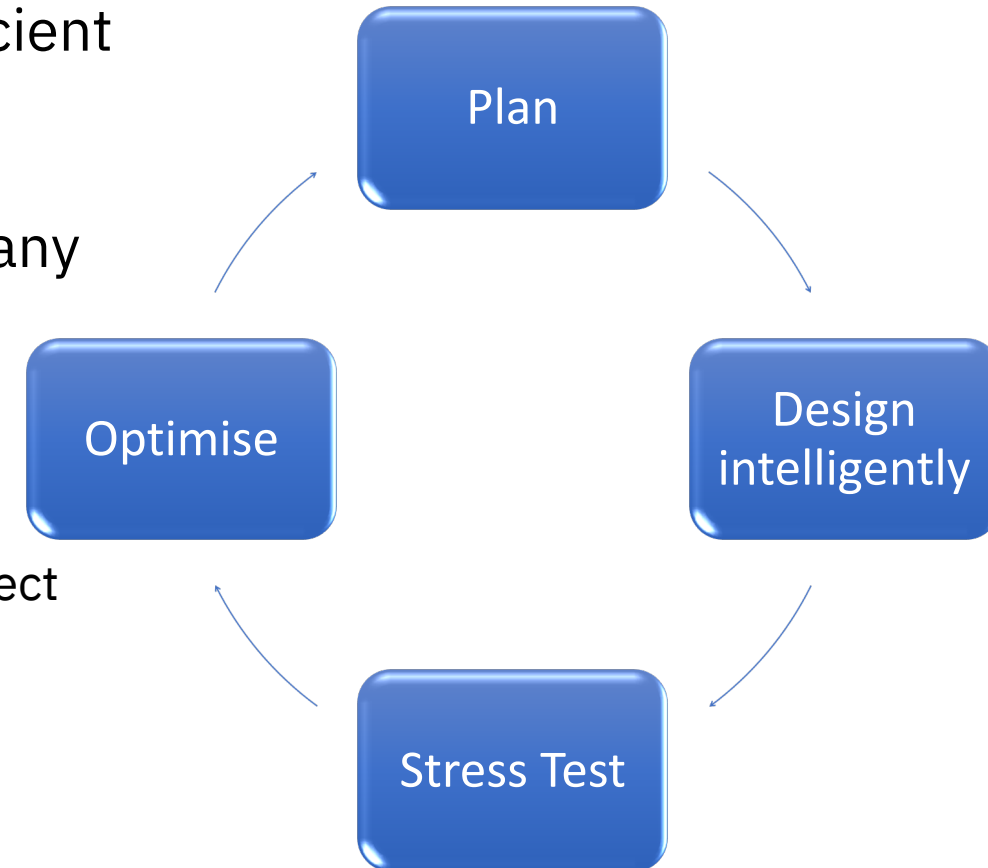
Db2 recovery background

- 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 ...

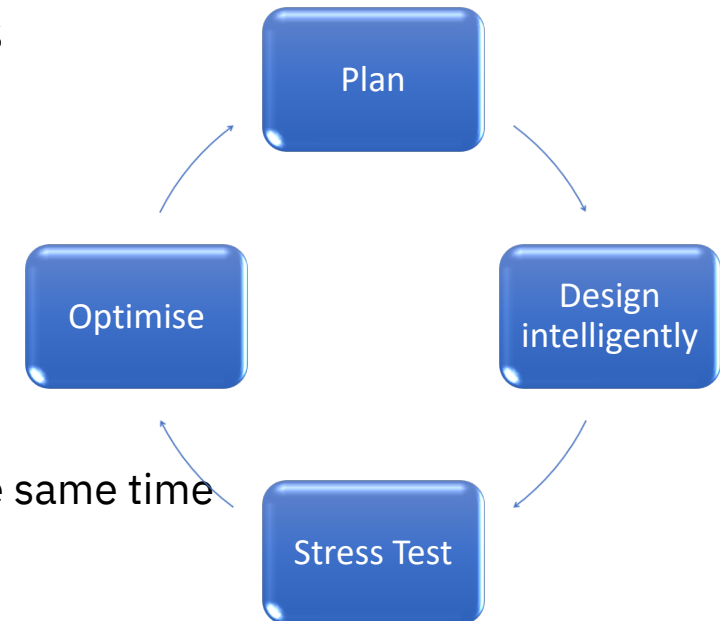
- **Db2 log-based recovery of multiple objects is a very rare event ...**
 - ... but statistically, it is more frequent than a true DR event (flood, fire, etc.)**
- Taking regular backups is necessary but far from sufficient for anything beyond minor application recovery
- If not prepared, practiced 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



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, VTS or DASD? Standard ICs, are ICs stacked? FCICs, SLBs?
 - Degree of parallelism?
- Log scan
 - Backup frequency
 - Archive logs needed to recover?
 - Archive logs on tape, VTS or DASD?
 - Degree of parallelism?
- Log apply
 - Update frequency and update patterns
 - Maximal fast log apply?
- Recover/Rebuild indexes

Db2 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
 - 4GB – 1 Byte (Db2 11), 768GB (Db2 12)
 - 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

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

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 DSNTESTSQ
 - 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 ...

- Exploitation of FlashCopy technology
 - Data set FlashCopy 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
 - Db2 Backup 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
- 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)

Image copy backup considerations

- 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
 - Including checking for 2 valid backups within a retention period
 - 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

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

Identifying the scope of data corruption

- Before developing a recovery plan it is vital to identify the scope of data the data corruption
- 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
 - CHECK utilities exploiting FlashCopy enables the ability to non-disruptively identify scope of data corruption
- Important FlashCopy parameters:
 - 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

Design for intelligent Db2 recover

- 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

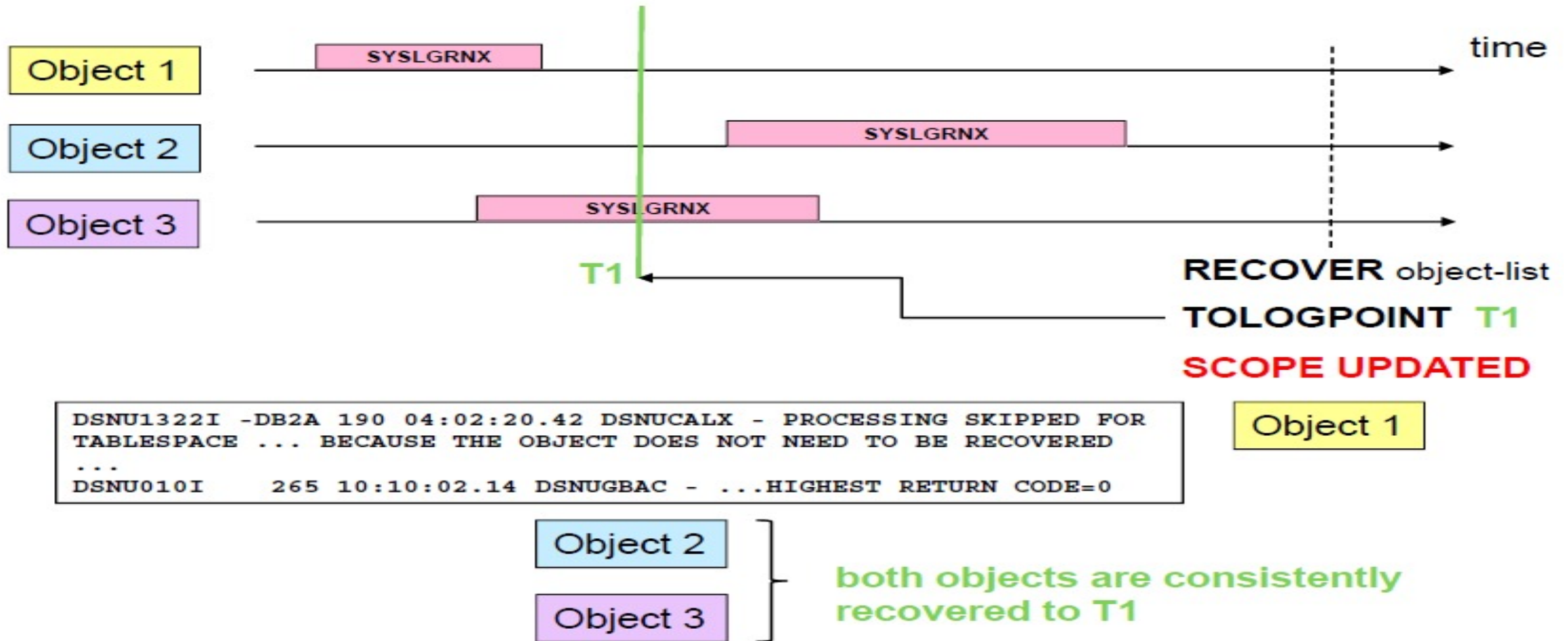
Design for intelligent Db2 recover ...

- 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

Point-In-Time (PIT) Recovery in Db2 12 with SCOPE UPDATED

- New in Db2 12, SCOPE UPDATED option for RECOVER utility
 - Goal is to speed up the elapsed time for PIT recovery (TORBA/TOLOGPOINT)
 - SCOPE UPDATED is the default
 - Objects are excluded from recovery that have not changed since the given recovery point
 - Avoids wasting time restoring the image copy for a given object
 - Potentially a great performance optimization
 - Might be known to a few customers by running RECOVER under DIAGNOSE(607) for Db2 11
 - As before, SCOPE ALL forces the recovery of any object even if it has not been changed
 - How does Db2 know if the objects have been changed or not after the recovery point?
 - By reading the entries in SYSLGRNX

Point-In-Time (PIT) Recovery in Db2 12 with SCOPE UPDATED ...



Point-In-Time (PIT) Recovery in Db2 12 with SCOPE UPDATED ...

- Recovery of unchanged tablespaces
 - Determined at execution time
 - Only applies to point in time recovery
 - Db2 11 execute with **DIAGNOSE(607)**
 - Db2 12 RECOVERY **SCOPE UPDATED**
 - Default
- REBUILD INDEX
 - Db2 11 & 12 **SCOPE PENDING**
 - **Default SCOPE ALL**
- Point in Time Recovery Recommendation
 - Db2 11
 - RECOVER ... DIAGNOSE(607) + REBUILD Index ... SCOPE PENDING
 - Db2 12
 - RECOVER ... SCOPE UPDATED + REBUILD INDEX ... SCOPE PENDING

```
DSNU1322I -DB2A 190 04:02:20.42 DSNUCALX - PROCESSING SKIPPED FOR  
TABLESPACE ... BECAUSE THE OBJECT DOES NOT NEED TO BE RECOVERED  
...  
DSNU010I 265 10:10:02.14 DSNUGBAC - ...HIGHEST RETURN CODE=0
```

Point-In-Time (PIT) Recovery in Db2 12 with SCOPE UPDATED ...

- Problem statement
 - If a PIT recovery of SYSLGRNX is performed then can run into a big issue when performing a PIT recovery for application objects after the PIT recovery of SYSLGRNX
 - Scenario
 - No data changes have been done to the application object between the time of COPY and the recovery point of SYSLGRNX
 - But data changes are done later after the the recovery point of SYSLGRNX
 - Db2 will not detect that situation because the entries of SYSLGRNX have been eliminated by the PIT recovery of SYSLGRNX
 - THE OBJECT DOES NOT NEED TO BE RECOVERED
 - Db2 will incorrectly exclude the recovery of the application object which will lead to data inconsistencies
 - REPORT RECOVERY utility also reads SYSLGRNX and will not report the changes
 - The RECOVER job ends with RC04 and DSNU1322I message for each excluded object
 - DSNU1322I =D2LC 318 12:10:14.52 DSNUCALX - PROCESSING SKIPPED FOR TABLESPACE DSNDB06.SYSTSLVH BECAUSE THE OBJECT DOES NOT NEED TO BE RECOVERED

Point-In-Time (PIT) Recovery in Db2 12 with SCOPE UPDATED ...

- Recommendations
 - Apply PTF for APAR PH20056
 - RECOVER will internally change SCOPE UPDATED to ALL for PIT recovery of Catalog/Directory objects
 - RECOVER will internally change SCOPE UPDATED to ALL for any object after SYSLGRNX has been recovered to a prior point in time
 - DSNU124I message will be issued to indicate the override
 - RC will still be 0
 - No plan to change the default from SCOPE UPDATED to SCOPE ALL
 - Until PTF applied for APAR PH20056, explicitly specify SCOPE ALL on PIT recoveries
 - Any pre-existing “canned” job to recover the Catalog/Directory should be modified

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

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

Optimizing Db2 recovery

- 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'

Optimizing Db2 recovery ...

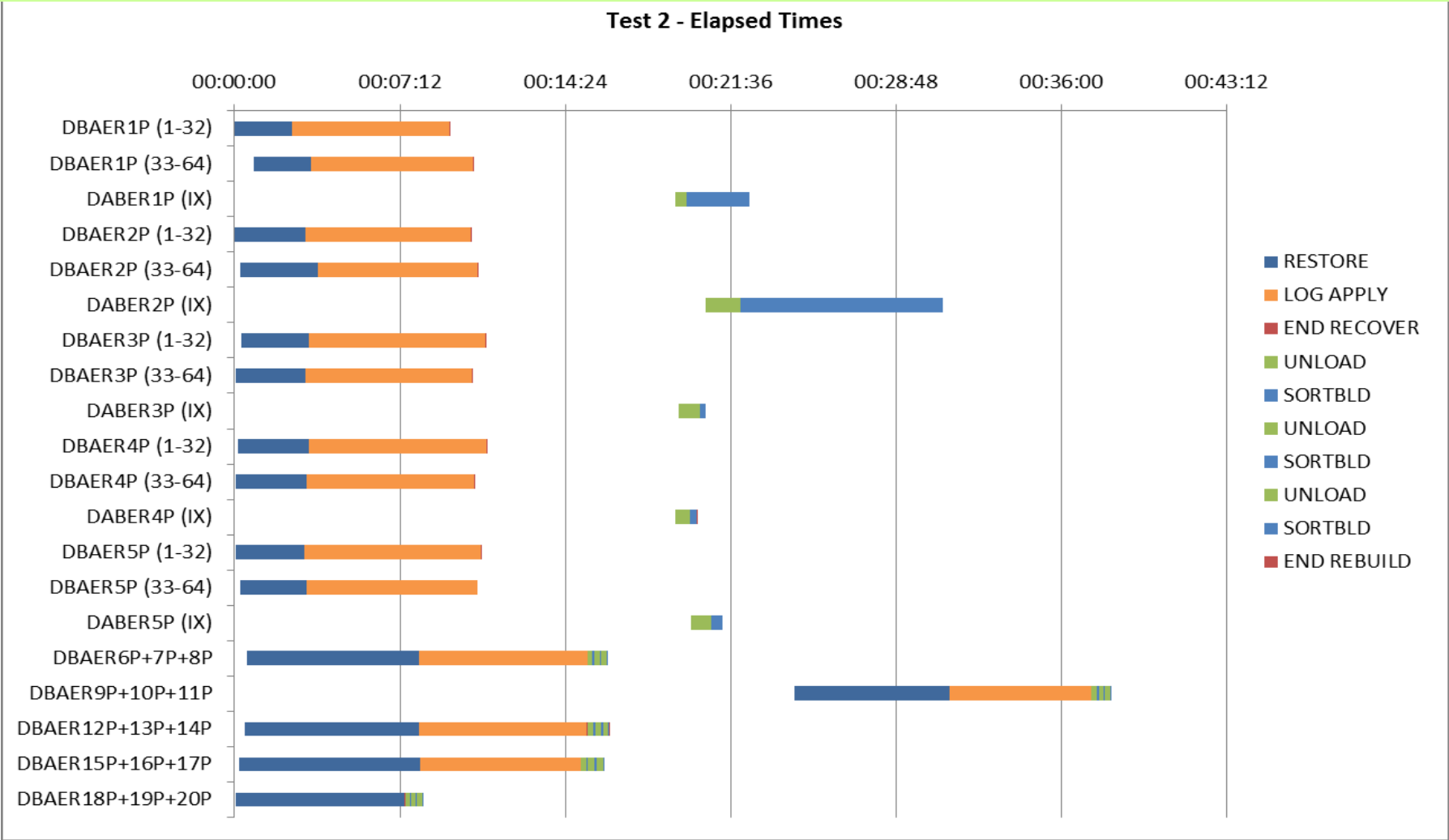
- Create automated procedures to create efficient recovery jobs
 - Considerations
 - Table prioritization
 - Virtual Tape/Tape optimization
 - Frequency
 - Execute after nightly backup jobs
 - Recovery jobs updated daily and ready to execute
 - Execute when CPU is available (middle of night)
 - Procedures 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

Optimizing Db2 recovery - 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

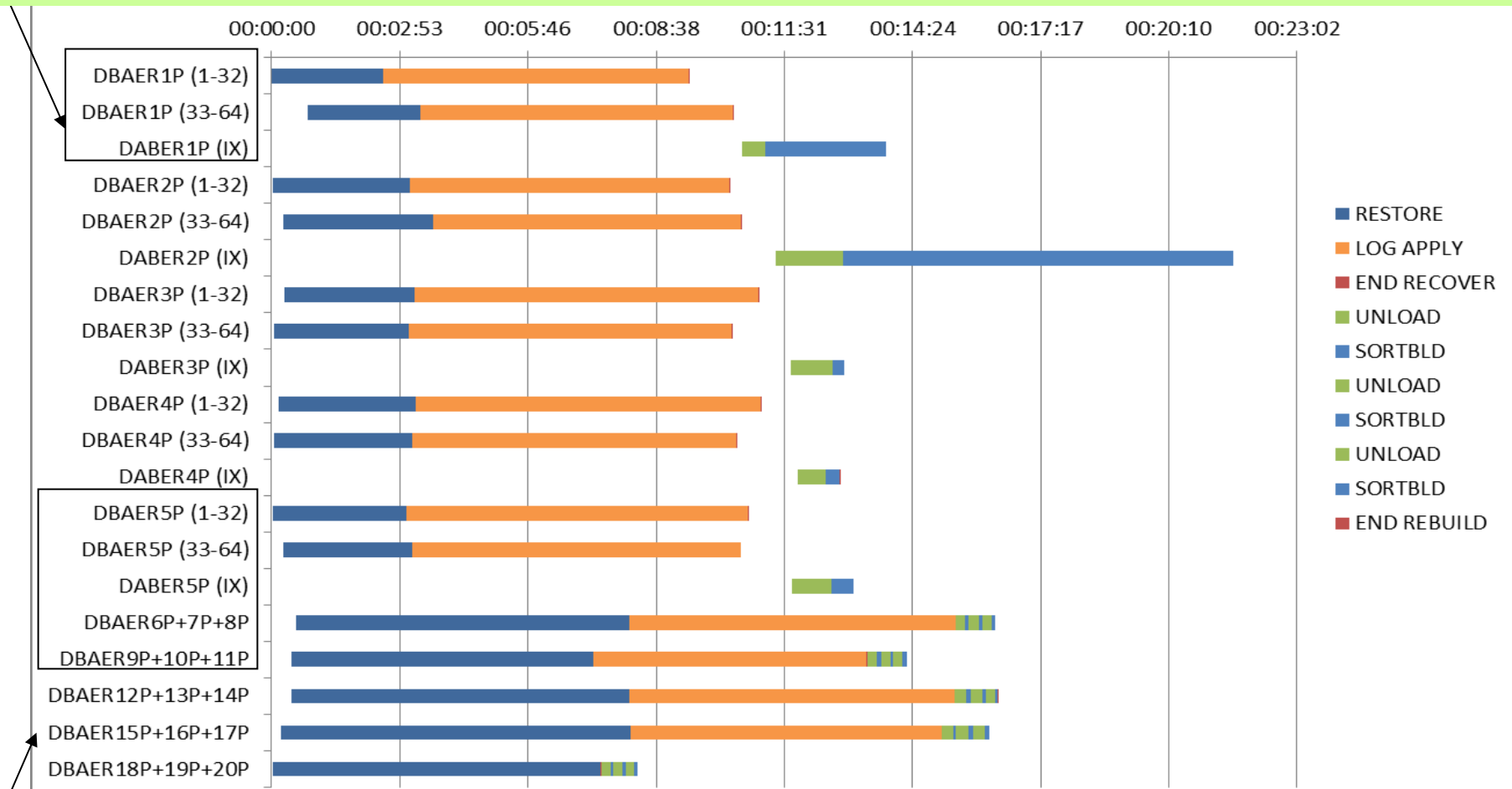
Optimizing Db2 recovery - stress test ...

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



Optimizing Db2 recovery - 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



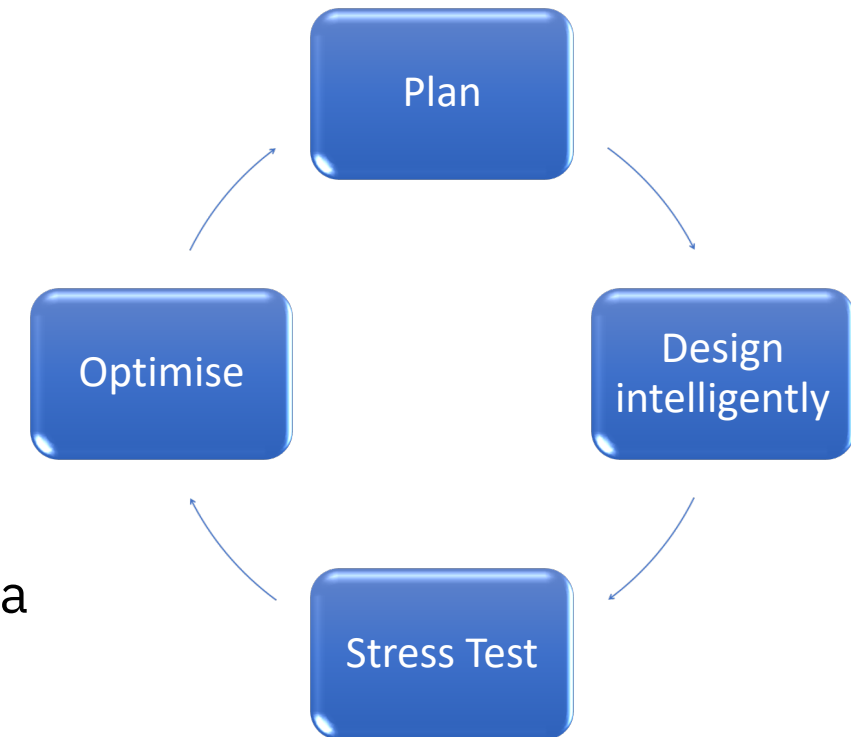
Could improve the recovery of these objects by running the REBUILD IX in parallel

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
 - Data isolation to de-couple applications
 - Objective – bring back critical applications first to resume availability
 - Incrementally bring back additional applications

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



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"

The World of Db2

Are you a member yet?

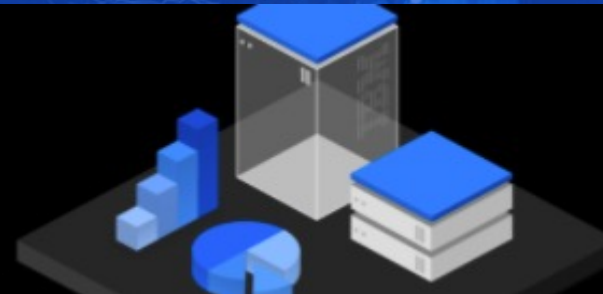
Join now

#Db2z #Db2World



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





TECHNICAL SESSION

On Demand
**Session 7. Primer on Insert
Algorithm 2 with Db2 12**





TECHNICAL SESSION

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





TECHNICAL SESSION

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





TECHNICAL SESSION

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





TECHNICAL SESSION

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





TECHNICAL SESSION

On Demand
**Session 2. Dos and Dont's about
Continuous Delivery**





TECHNICAL SESSION

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





TECHNICAL SESSION

On Demand
**Db2 for z/OS: Hot Topics and Best
Practices with John Campbell PART
2**



TECHNICAL SESSION

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



TECHNICAL SESSION

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



Thank You

IBM®

Data and AI

Mass Db2 Application Recovery....
“The Nightmare of a Db2 Professional”

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

IBM Data and AI

IBM

