Db2 for z/OS Mass Data Recovery

Hints and Tips

John Campbell
Email: john.Campbell@triton.co.uk

Agenda

- Introduction
- Common issues
- Factors that affect the recovery elapsed time
- Image copy backups
- Design intelligently
- Stress test
- Index COPY/RECOVER vs. REBUILD INDEX
- FlashCopy – CHECK INDEX/DATA/LOB SHRLEVEL CHANGE
- FlashCopy – System level and object level backup/recovery
- Application design
- IBM Safeguarded Copy function
- Solutions
- Summary

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 and inconsistent
    - Bad Disaster Restart e.g., using stale coupling facility structures
  - Data corrupted at local site caused by
    - Bad application program
    - Operational error
    - Db2, IRLM, z/OS, 3rd party product code failure
    - CF microcode failure
    - DASD 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 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
- Taking regular backups is necessary but far from sufficient for anything other than minor recoveries
- If not prepared, practised and optimised, will lead to extended application service downtime
  - Possibly many hours to several days

Common issues

- Lack of planning, intelligent design, optimisation, practice and maintenance
- No prioritised list of application objects and interdependencies
  - Data dependencies and integrity management “buried” in applications
  - Heavily dependent on application knowledge and support
- Procedures for taking backups and executing recovery compromised by lack of investment in technical configuration
- Use of tape (VTS)
  - Cannot share tape volume across multiple jobs
  - Even though VTS is a DASD cache it is known to z/OS as tape device
    - Same serialisation characteristics as all tape devices
    - A single virtual volume cannot be shared by different jobs or systems at the same time

Common issues …

- Results in any or all of the following
  - No estimate of overall elapsed time to complete
  - Elongated overall elapsed time to complete recovery
  - Performance bottlenecks so that recovery performance does not scale
  - Surprises caused by changing technical configuration
  - Unrecoverable objects

Image copy backups

- Always take dual inline image copies as part of LOG NO events (REORG, LOAD REPLACE)
- Schedule daily a production job to check for unrecoverable objects
  - Ensure valid full image copy (FIC) backup dataset exists and enough recovery log available to recover forward from it
  - If incremental image copy (IIC) are used, ensure a full image copy (FIC) is also available
  - Check should take into account LOG NO events
- Use as much DASD as possible for optimal recovery speed
- Consider shortening the full image copy (FIC) cycle time (<= 24 hours) for Db2 Catalog/Directory and most critical application objects
  - Objective to reduce log apply time

Image copy backups ….

- Consider use of incremental image copy (IIC)
  - Keep IIC on DASD, otherwise perform regular MERGECOPY in background
- Do not use GDGs for image copy datasets
  - Risk of older versions rolling off by accident
  - Use catalogued datasets with “meaningful” naming convention
- Use partition-level image copy
  - Faster partition-level recovery from inline image copy
  - Create partition-level image copy using TEMPLATE with &PA or &PART

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 required during a recovery event
    - Objective to bring back critical application services as soon as possible
    - Without this list, either have to wait for “whole world” to be recovered, or take a risk of bringing back 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
  - Maximize parallelism in the RESTORE phase
    - For partitioned tablespace use parallelism by partition
      - LISTDEF utility statement with PARTLEVEL option will build a list of partitions for an object and automatically handle partitions that have been added or pruned
    - Use PARALLEL for parallel processing for image copies on DASD
    - Use PARALLEL(n) TAPEUNITS(n) for image copies stacked on tape
  - Optimize the use of Fast Log Apply (FLA)
    - Schedule up to 51 RECOVER jobs per Db2 member
    - Recover list of objects rather than individual objects
      - Optimal 20-30 objects per RECOVER
    - Spread the jobs around all Db2 data sharing members

Design intelligently …

- Objects with longest end-to-end recovery time should be recovered first
  - Size of object
  - Update rate since last image copy taken
  - Number and size of indexes
- Optimize job scheduling to avoid “dead times”

Stress test

- Practice regular full-scale “fire drills” for mass recovery of entire application or even the whole system
- Objectives
  - Validate that procedures are in good working order
    - Both for local and remote DR recovery
  - Maintain readiness for mass recovery execution
  - Find out what the actual service level achievement is
    - Break down the elapsed time for each job: RESTORE/LOG APPLY/REBUILD INDEX
    - If elapsed time needs to be improved, look for possible optimizations

Index COPY/RECOVER vs. REBUILD INDEX

- Fast Log Apply (FLA) now used for RECOVER INDEX
  - Db2 will use list prefetch to read all index pages that are needed to apply log records for, before applying any log records
    - Potential for significant savings in elapsed time
  - Should now reconsider COPY YES and taking image copy backups for indexes
    - Use RECOVER INDEX in parallel with RECOVER TABLESPACE [PART] instead of wait for RECOVER TABLESPACE [PART} for all partitions to complete followed by REBUILD INDEX
  - Note: REBUILD INDEX is still preferred option as first step when recovering from 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 can be very disruptive
  - Even with SHRLEVL CHANGE, R/O access only allowed during creation of shadow datasets
- Db2 ZPARM CHECK_FASTREPLICATION
  - Strongly recommend to set REQUIRED whether FlashCopy is used or not
    - CHECK will fail if FlashCopy is not available
- Db2 ZPARM UTIL_TEMP_STORCLASS
  - Optional, can be used to specify a storage class to be used for shadow datasets
  - If blank, the shadow datasets are defined in the same storage class as the production datasets
  - If using DASD-based replication, specify a pool of volumes outside the DASD mirror

FlashCopy – System level and object level backup/recovery

- FlashCopy provides some interesting new options to help mass data recovery
- Point-In-Time backup of the entire system
  - If still on DASD, can be restored quickly
  - Can be used to create a “forensic” system
    - Quick cloning of the environment away in an isolated 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 analyze and reconcile the data contents of the forensic system vs. current damaged system
- FlashCopy (FC) image copy
  - If still on DASD, potential for significant elapsed time reduction for the RESTORE phase of RECOVER
  - Can be used to create a transaction consistent image copy with COPY SHRLEVEL CHANGE

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 DB)

Application design …

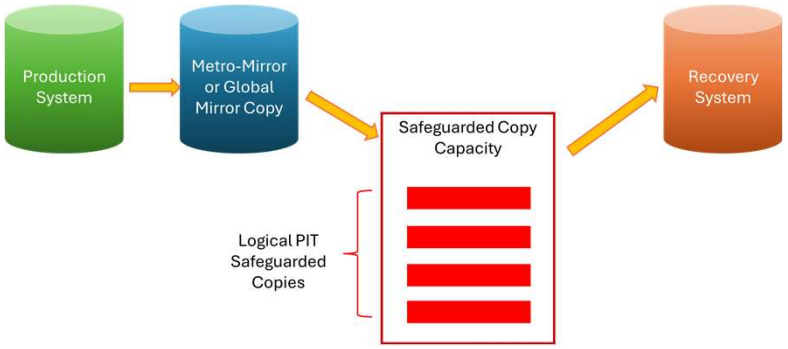
- Recommendations
  - Take frequent commits in long running batch jobs
    - Dynamic, table driven
    - Application must be capable of restarting automatically from the last intermediate commit point
  - Separate out active vs. inactive (historical) data
    - Use separate tables
    - Regular, aggressive pruning of older rows from active tables
    - Consider use of Db2 Archive Transparency feature
  - Data isolation to de-couple applications
    - Build “fire walls” to isolate data used by critical applications from non-critical applications

Application design …

- Recommendations …
  - Needs to be considered very carefully
    - Single integrated data source vs. high availability (and performance)
    - Evaluate cost vs. benefit
  - Potential solutions
    - Logical partitioning
    - Asynchronous processing
    - Data replication
    - Duplicate updates
  - Data dependencies across multiple data sources
    - Use volume level FlashCopy to create a cross-database point of consistency
    - Migrate all inter-dependent data to Db2
      - Greatly simplified operations, increased robustness, reduced overhead

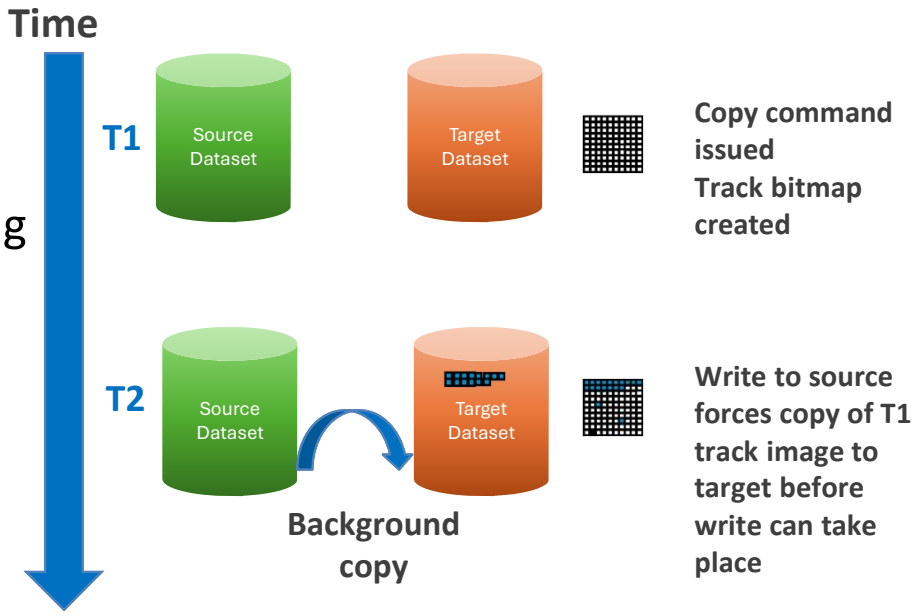
IBM Safeguarded Copy function

- Works in conjunction with IBM Copy Services
- Cyber resistant point-in-time copies of DASD volumes
  - Relationship with Metro Mirror or Global Mirror
  - Data is Encrypted
  - Data is immutable and cannot be changed
  - Uses thin provisioning
  - Policy driven (frequency, retention period)
- Used for forensic analysis, surgical recovery or catastrophic recovery

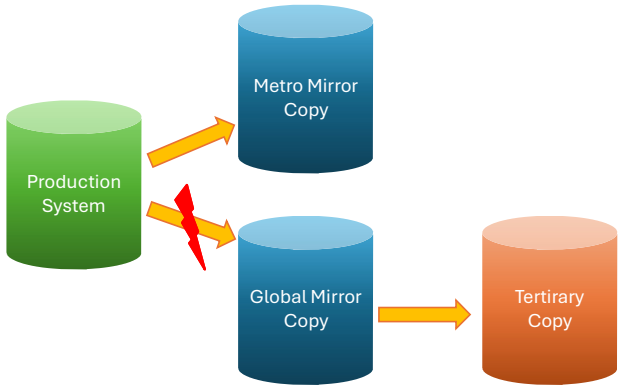


Solution #1

- Limited scope to logical data corruption e.g., specific application issue
- Use IBM Db2 Cloning Tool (or similar) for cloning and renaming a relatively small number of tablespaces by using dataset level Flash Copy and avoid disruption to production
- Could be cloned to the same or different z/OS system
- Allow application teams to query objects and compare against production



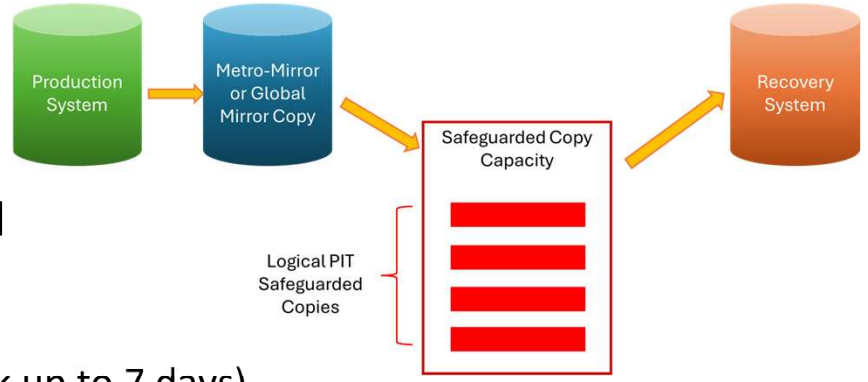
Solution #2



- Use for extensive data integrity checking without disrupting production and/or suspected widespread data corruption
- Suspend Global Mirror session
- Use volume level Flash Copy to create tertiary copy from Global Mirror secondary
- Resume Global Mirror
- IPL off tertiary copy in a network isolated LPAR to create forensic environment
- Perform extensive Db2 CHECK INDEX processing
- Allow application teams to query data and compare against production

Solution #3

- Use for malicious attack (ransomware) where cannot recover in either production or DR
- Predicated on prior implementation of Safeguarded Copy function with Global Mirror (DR)
  - Regular frequent point-in-time volume copies (going back up to 7 days)



- Create tertiary copy from last Safeguarded copy taken prior to problem being injected
- IPL off the tertiary copy and switch to DR
- If Db2 logs are good at DR, then potentially use log only forward data recovery to bring databases forward to just before problem injected and limit amount of data loss
- Need prepared business plan to compensate for lost transactions and batch processing

Summary

- Need to design for high performance and reduced elapsed time
  - Plan, design intelligently, stress test and optimize
    - Prioritize most critical applications
    - Understand application and data dependencies
    - Design for parallel recovery jobs
    - Optimize utilization of technical configuration
    - Intelligent creation and scheduling of recovery jobs
  - Design for DASD-based recovery for optimal performance
  - Practice regularly

Summary …

- Applications and data life cycle also have a role to play
  - Separate active/operational data from inactive/historical data
  - Perform regular aggressive archiving to history
  - Allow application toleration of unavailable historical data
  - Look at creating “fire walls” between applications

Questions?