IDUG
Leading the DB2 User Community since 1988

IDUG Db2 Tech Conference NA
Philadelphia, PA | April 29 - May 3, 2018

#IDUGDb2

DB2 Recovery best practices: learn the latest to address key scenarios

Dale McInnis

IBM Canada Ltd.

Session code: C17

May 3: 1035-11:35

Platform: LUW

Agenda

- Introduction
- Recovery Granularity
  - Object level, e.g. table recovery
  - Tablespace level
  - Partition level
  - Database level
- Making backups run faster
- Additional Documentation

Why am I backing up?

- Hardware / Software failure protection
  - Much better methods available
    - H/W Clusters, HADR, Q Repl, CDC, …
- Moving to a new platform (H/W or OS)
  - If the same OS then consider HADR
  - If different OS then consider logical replication
- Populate a QA/Dev system
  - If the entire DB is not required consider Transportable Schemas or the REBUILD option on restore
- Logical Protection
  - Someone messed up the data and you have to roll it back
  - If error is limited in scope them perhaps you can use DB2 Tooling
    - Recovery Export
    - High Performance Unload

What types of backup should I use?

Types of backups

- Traditional DB2 Backup
  - Full, incremental or delta
- Advanced Copy Services (ACS) backups
- Offloaded ACS backups
- Non-traditional
  - Q Repl
  - HADR
  - …
- My rule of thumb
  - If the elapsed time for the backup exceeds 6-8 hours (after tuning) then strongly consider an ACS backup

Failure anatomy

- Average cost of database downtime per minute:
  - $7900.00 – up by 41% over the last 2 years
- 80% of outages impacting mission-critical services are due to
  - Human Error
  - More than 50% of these outages are caused by change, configuration, release integration and hand-off issues
- Average outages last
  - 90 Minutes
- The average database to DBA ratio tends to be around
  - 5TB per DBA

*Source: Manageforce.com

Recovery granularity

- Multiple levels of granularity for recovery is available:
  - Object level, e.g. table recovery
  - Tablespace level
  - Partition level
  - Database level
- Always recover at the lowest level of granularity
  - If a disk goes bad only restore the tablespaces affected, not the entire database

Agenda

- Introduction
- Recovery Granularity
  - Object level, e.g. table recovery
  - Tablespace level
  - Partition level
  - Database level
- Making backups run faster
- Additional Documentation

DB2 Dropped Table Recovery

- Suppose table tab1 was dropped at time T1
- Restore DB from the backup image
  - Db2 restore db mydb from /path
- Identity the dropped table ID and corresponding DDLs from the list history command
  - Db2 list history dropped table all for mydb
- Rollforward to a PIT specifying the time with one second ahead of the timestamp when the table was dropped
  - Db2 rollforward db mydb to <timestamp> recover dropped table 0000000014c8a00020004 to $home/tab1

DB2 Dropped Table Recovery

- After recovery the table specify the complete option to complete the rollforward
  - Db2 rollforward db mydb to end of logs and complete

- Recreate the dropped table
  - Db2 create table tab1 (c1 integer) in userspace1
  - Db2 import from $home/tab1 of del insert into tab1

Dropped table recovery example

Step 1: (Get the DDL for the table which was dropped)

```
db2 =>list history dropped table all for dbnoauto
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log  Backup ID
-- --- ------------------ ---- --- ------------ ------------ --------------
 D  T  20170913233705                              000000000001a8500030002
----------------------------------------------------------------------------
 "DMCINNIS"."TEST" resides in 1 tablespace(s):
 00001 DATA_SMS1
----------------------------------------------------------------------------
   Comment: DROP TABLE
Start Time: 20170913233705
  End Time: 20170913233705
    Status: A
----------------------------------------------------------------------------
 EID: 109
DDL: CREATE TABLE "DMCINNIS"."TEST" ( "COL1" INTEGER , "COL2" INTEGER )  IN "DATA_SMS1" ;
----------------------------------------------------------------------------
```

Dropped table recovery example

Step 2: (Restore the table space from the backup image – table space level restore)

```
db2 => RESTORE DATABASE dbnoauto TABLESPACE (DATA_SMS1) FROM
C:\dmcinnis\dmcinnisdocs\offlinebackups TAKEN AT 20170913230239 without prompting
DB20000I  The RESTORE DATABASE command completed successfully.
```

Step 3: (Roll forward the db to end of logs and complete with recover dropped table <table-id>) command

```
db2 => ROLLFORWARD DATABASE dbnoauto TO END OF LOGS AND COMPLETE RECOVER DROPPED TABLE
0000000000001a8500030002 to "C:\dmcinnis\dmcinnisdocs\dropped table"
                      Rollforward Status
 Input database alias                   = dbnoauto
 Number of nodes have returned status   = 1

 Node number                            = 0
 Rollforward status                     = not pending
 Next log file to be read               =
 Log files processed                    =  -
 Last committed transaction             = 2017-07-13-11.14.07.000000 UTC
DB20000I  The ROLLFORWARD command completed successfully.
```

Dropped table recovery example

Step 4: (Make sure the data file is created in the mentioned directory with the data)

```
C:\dmcinnis\dmcinnisdocs\dropped table\NODE0000>type data
5,2
10,5
15,5
```

Step 8: (Run the DDL which creates the Table definition)

```
C:\Program Files\IBM\SQLLIB_01\BIN>db2 -tvf "C:\dmcinnis\dmcinnisdocs\R and
D\Scripts\droppedtableddl\droppedtabledef.sql" -z "C:\dmcinnis\dmcinnisdocs\R and
D\Scripts\droppedtableddl\droppedtabledef.log"
CONNECT TO DBNOAUTO
   Database Connection Information
 Database server        = DB2/NT 9.1.3
 SQL authorization ID   = DMCINNIS
 Local database alias   = DBNOAUTO

CREATE TABLE "DMCINNIS"."TEST" ( "COL1" INTEGER , "COL2" INTEGER )  IN "DATA_SMS1"
DB20000I  The SQL command completed successfully.
CONNECT RESET
DB20000I  The SQL command completed successfully.
```

Dropped table recovery example

Step 5: (Import/Load the data into the table)

```
C:\Program Files\IBM\SQLLIB_01\BIN>db2 -tvf "C:\dmcinnis\dmcinnisdocs\R and
D\Scripts\droppedtableddl\droppedtableimport.sql" -z "C:\dmcinnis\dmcinnisdocs\R and
D\Scripts\droppedtableddl\droppedtableimport.log"

CONNECT TO DBNOAUTO

   Database Connection Information

 Database server        = DB2/NT 9.1.3
 SQL authorization ID   = DMCINNIS
 Local database alias   = DBNOAUTO

IMPORT FROM "C:\dmcinnis\dmcinnisdocs\dropped table\NODE0000\data" OF DEL METHOD P(1, 2) MESSAGES
"C:\dmcinnis\dmcinnisdocs\R and D\Scripts\droppedtableddl\importlog.msg" INSERT INTO TEST(COL1, COL2)

Number of rows read         = 3
Number of rows skipped      = 0
Number of rows inserted     = 3
Number of rows updated      = 0
Number of rows rejected     = 0
Number of rows committed    = 3

CONNECT RESET
DB20000I  The SQL command completed successfully.
```

Dropped table recovery example

Step 6: (Make sure the table is accessible and able to retrieve data)

```
db2 => connect to dbnoauto

   Database Connection Information

 Database server        = DB2/NT 9.1.3
 SQL authorization ID   = DMCINNIS
 Local database alias   = DBNOAUTO

db2 => select col1, col2 from test

COL1        COL2
----------- -----------
          5           2
         10           5
         15           5

  3 record(s) selected.
```

IBM DB2 Advanced Recovery Feature
Save time, reduce errors, and meet SLAs

DB2 Merge Backup

- Improve speed of your backup & recovery processes
- Minimize application impact

DB2 Recovery Expert

- Recover faster with greater granularity while protecting your critical business data
- Eliminate data errors before they compound into costly business mistakes
- Track and report data changes in response to auditing requests

Optim High Performance Unload

- Extract large amounts of data quickly and with minimal impact on system productivity
- Perform full data and system migrations from one DB2 instance to another

Backup
Recover
Unload

All products support DB2 11.1 10.5, 10.1, 9.7, **for LUW**

Recovery

DB2 Recovery Expert
Granular and Advanced Data Recovery

Solution	▪ DB2 Recovery Expert performs rich log analysis and recovery operations which help organizations protect their valuable data
Installation	▪ Client, server, web browser
Audience	▪ DB2 LUW customers needing to meet availability SLAs
What's New	▪ Support DB2 11.1 features and functions ▪ Improved scalability ▪ Web UI interface, no thick user interface ▪ Support for pureScale environments

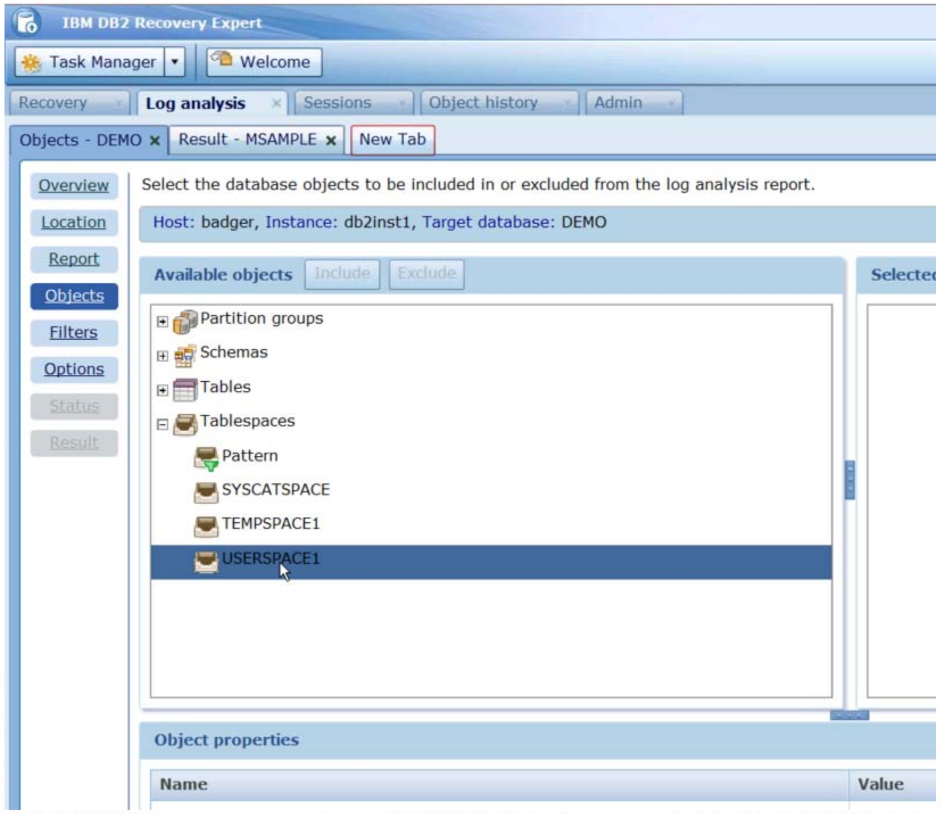
DB2 Recovery Expert
Example Scenarios

Recovery

Scenarios

An accident occurs

- **I ran the weekly batch job instead of the monthly job**
- **I dropped a production table instead of a test table**
- **I ran a delete with the wrong date parameters and ended up deleting too many rows**

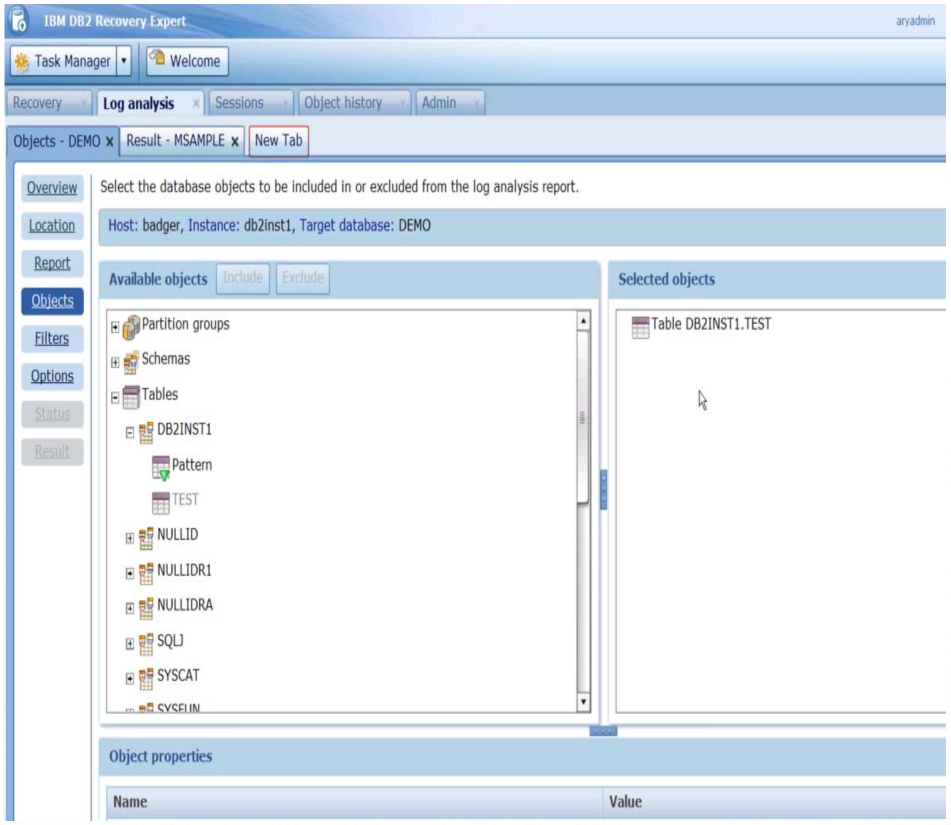


Select the database objects to be included in or excluded from the log analysis report.

Host: badger, Instance: db2inst1, Target database: DEMO

DB2 Recovery Expert LUW
Recovery Features

- Dropped object restore
- Recover tables, table spaces, stored procedures, etc.
- Point-in-time object recovery, with REDO or UNDO SQL generation
- Determines best recovery path and resources; user can override

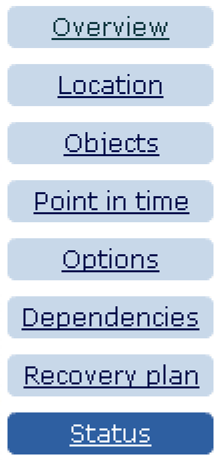


Select the database objects to be included in or excluded from the log analysis report.

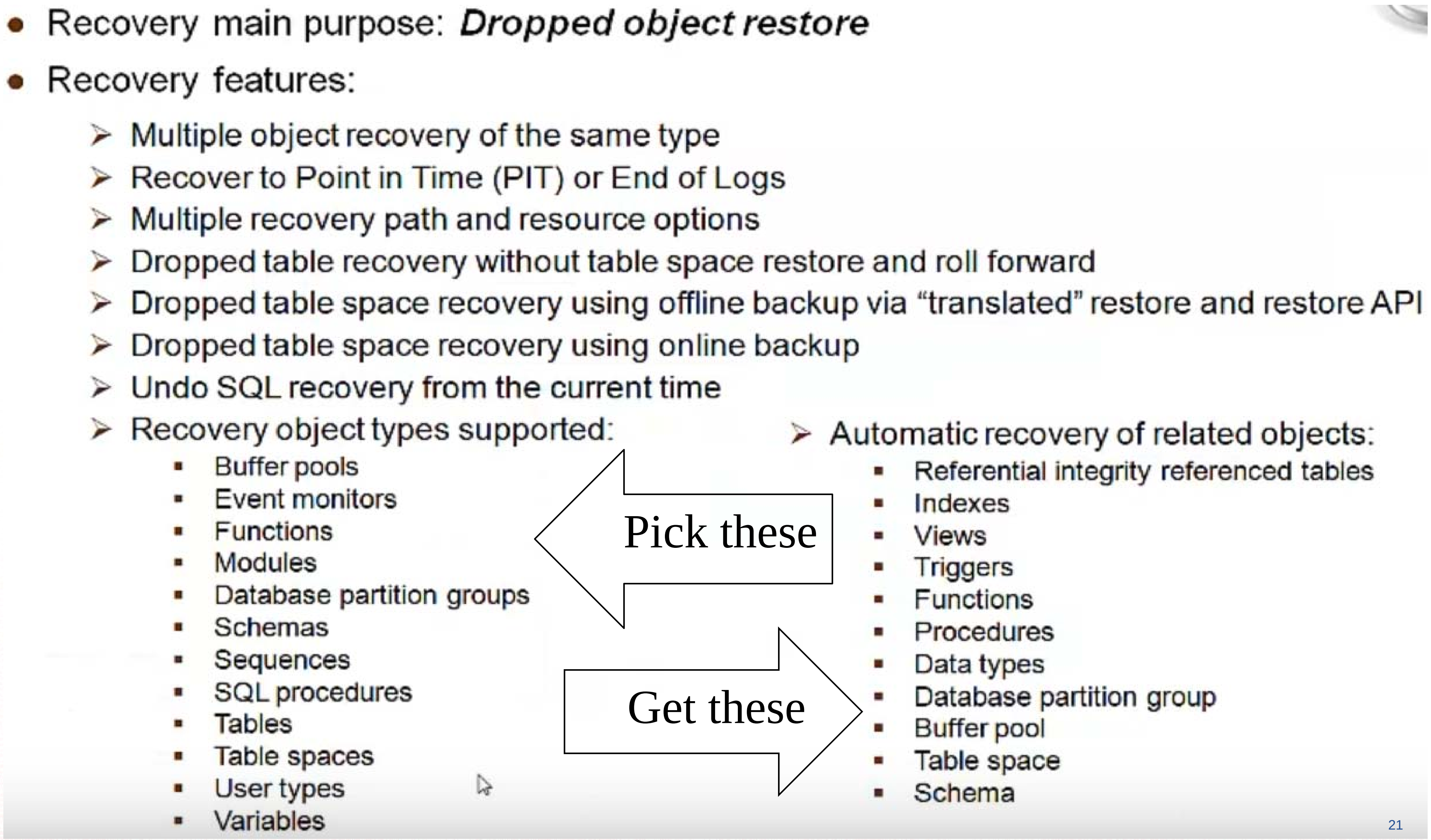
Host: badger, Instance: db2inst1, Target database: DEMO

Recovery Process Overview

- To recover database objects, identify
  - The objects to be recovered
  - Target recovery time
  - Available recovery resources (logs, backups, …)
- Recovery Expert identifies dependent or related objects and potential recovery plans
- Review applicable recovery plans and choose the one that aligns best with your preferred strategy



Recovery Feature and Capabilities

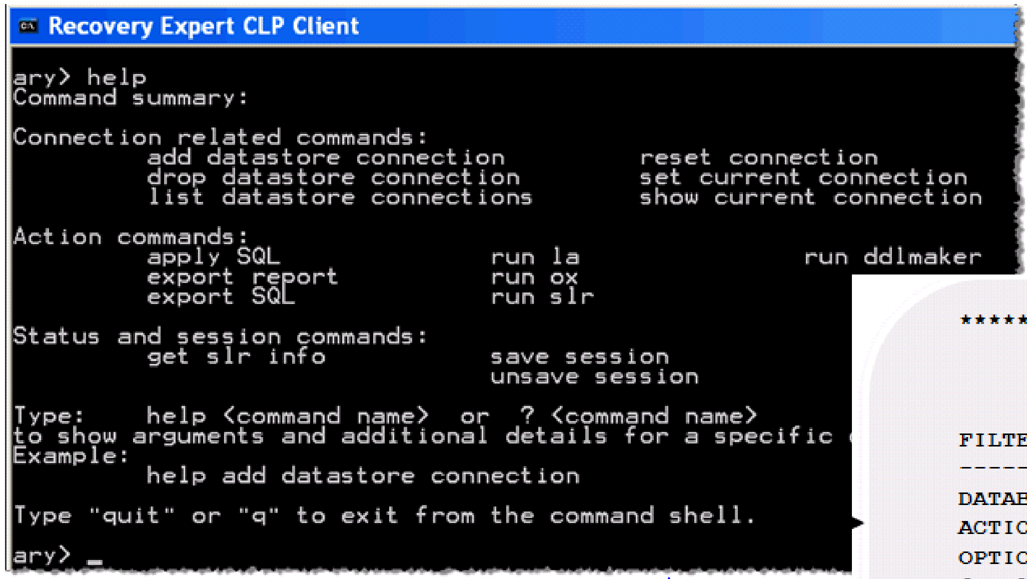


- Recovery main purpose: ***Dropped object restore***
- Recovery features:
  - Multiple object recovery of the same type
  - Recover to Point in Time (PIT) or End of Logs
  - Multiple recovery path and resource options
  - Dropped table recovery without table space restore and roll forward
  - Dropped table space recovery using offline backup via “translated” restore and restore API
  - Dropped table space recovery using online backup
  - Undo SQL recovery from the current time
  - Recovery object types supported:
    - Buffer pools
    - Event monitors
    - Functions
    - Modules
    - Database partition groups
    - Schemas
    - Sequences
    - SQL procedures
    - Tables
    - Table spaces
    - User types
    - Variables
  - Automatic recovery of related objects:
    - Referential integrity referenced tables
    - Indexes
    - Views
    - Triggers
    - Functions
    - Procedures
    - Data types
    - Database partition group
    - Buffer pool
    - Table space
    - Schema

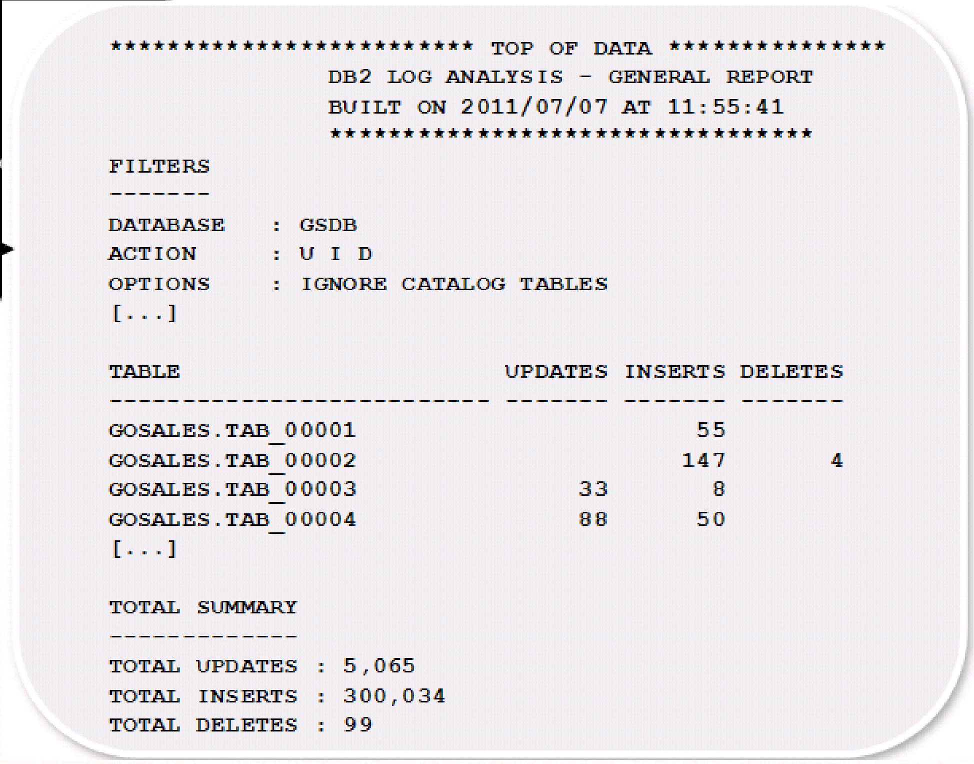
Pick these

Get these

DB2 Recovery Expert Command Line Tools


Recovery Expert CLP Client

```
ary> help
Command summary:

Connection related commands:
        add datastore connection          reset connection
        drop datastore connection         set current connection
        list datastore connections        show current connection

Action commands:
        apply SQL               run la                  run ddlmaker
        export report           run ox
        export SQL              run slr

Status and session commands:
        get slr info            save session
                                unsave session

Type:   help <command name>  or  ? <command name>
to show arguments and additional details for a specific
Example:
        help add datastore connection

Type "quit" or "q" to exit from the command shell.

ary> _
```



```
************************ TOP OF DATA ***************
              DB2 LOG ANALYSIS - GENERAL REPORT
              BUILT ON 2011/07/07 AT 11:55:41
              ********************************
FILTERS
-------
DATABASE    : GSDB
ACTION      : U I D
OPTIONS     : IGNORE CATALOG TABLES
[...]

TABLE                      UPDATES INSERTS DELETES
-------------------------- ------- ------- -------
GOSALES.TAB_00001                       55
GOSALES.TAB_00002                      147       4
GOSALES.TAB_00003               33       8
GOSALES.TAB_00004               88      50
[...]

TOTAL SUMMARY
-------------
TOTAL UPDATES : 5,065
TOTAL INSERTS : 300,034
TOTAL DELETES : 99
```

What's New - DB2 Recovery Expert
DB2 11.1 Features

- Enhancements for DB2 pureScale
- Support for databases using native encryption
- CLP enhancements

  - Use Log Analysis to monitor changes to a database and give the DBA the ability to quickly restore or correct erroneous data even in pureScale environments

  - If you use native encryption for any DB2 10.5 or DB2 11.1 databases DB2 Recovery Expert will be able to recognize this and handle

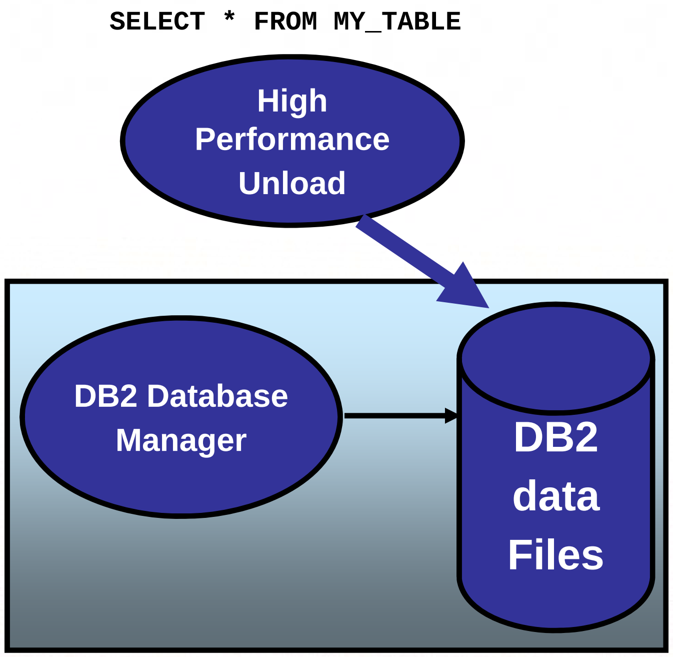
  - Command line processor (CLP) commands and command syntax were enhanced to support the product centralized repository design.

Unload

Optim High Performance Unload

Meet service level agreements and ease application upgrades by extracting data quickly and efficiently

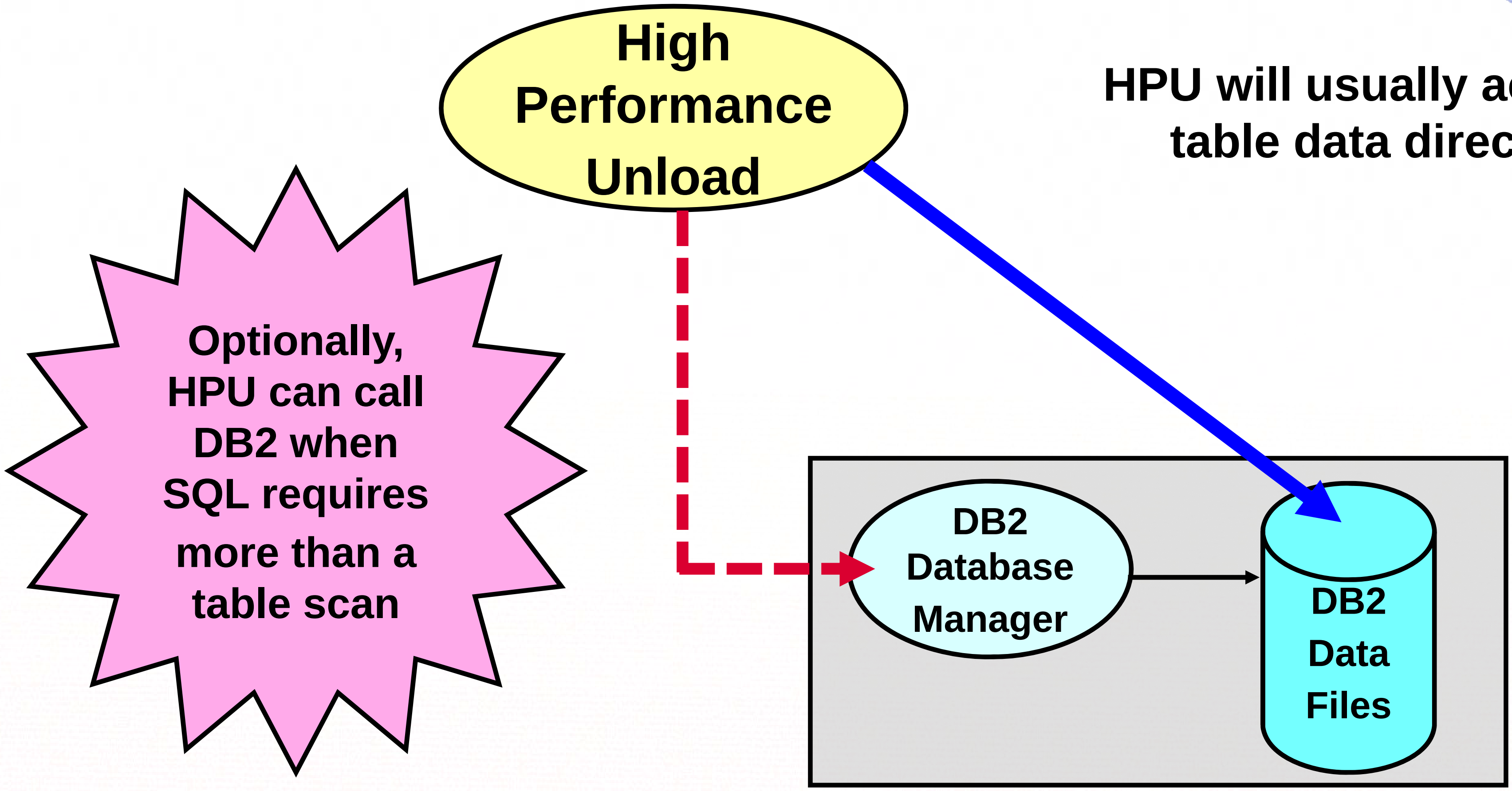
- Fast unload capability
  - Often 10 to 12 times faster than export
  - Parallel processing for higher speeds
- Flexibility to meet all use cases
  - Full Partitioned DB2 support
  - Subset of SELECT syntax used to filter columns and/or rows
  - Many automatic data type conversions and formats available
  - Offline or Online Backups
- Repartitioning feature
  - Built-in “DB2 Splitter”
    - Unload and split in a single operation



Conceptual Overview: What Makes HPU So Fast?

Unload

SELECT * FROM MY_TABLE

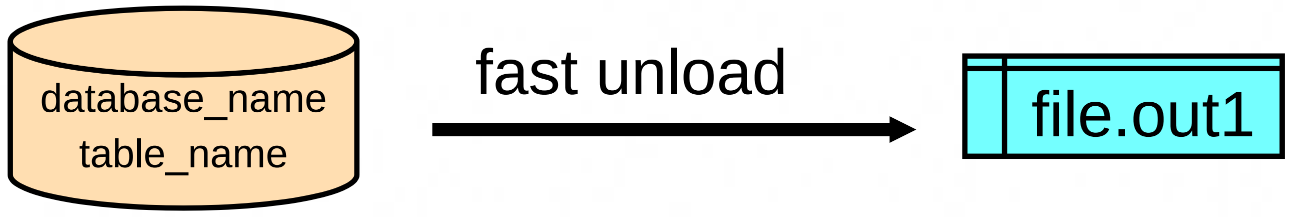

High Performance Unload
Optionally, HPU can call DB2 when SQL requires more than a table scan
DB2 Database Manager
DB2 Data Files

HPU will usually access table data directly

Unload Data Using the Command Line

```
db2hpu -d database_name -t table_name > file.out1
```

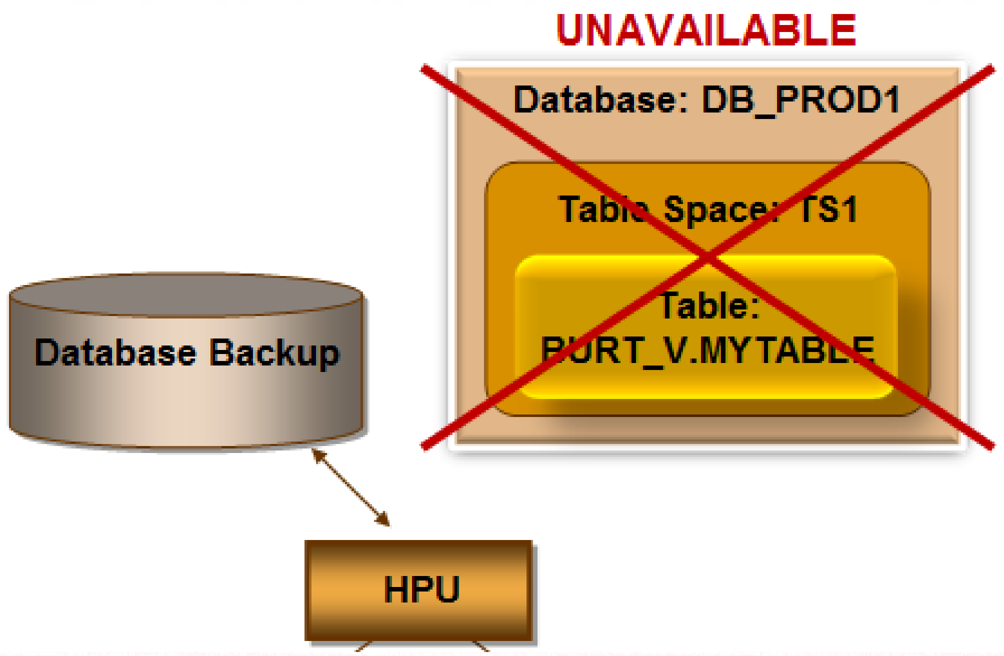
(use a control file (-f option) for more complex operations)



- Source can also be
  - Databases or table spaces
    - All user (non-system) tables
  - Tables
    - Individual
    - Multiple tables
    - Summary and alias tables
  - Backups
    - Full, delta, incremental backups
  - All or listed partitions
- Output can also be
  - Files
    - Tape devices
    - Named pipes (UNIX only)
  - Formats
    - PC/IXF
    - Delimited
    - Undelimited
    - DSNTIAUL
    - XML

Extracting from DB2 Backups

- HPU can extract tables from full , delta, incremental, merge backups
  - Use catalog information in backups if provided
    - Allows extraction even if DB2 is down
    - Allows extraction on a different system



What's New – Optim High Performance Unload New Features

- Optim High Performance Unload will now be able to handle DB2 native encryption in DB2 11.1 and DB2 10.5 databases

- Other new features:
  - New JSON output formats
  - Interface with Big Data environments
  - Native support of DB2 variables

Agenda

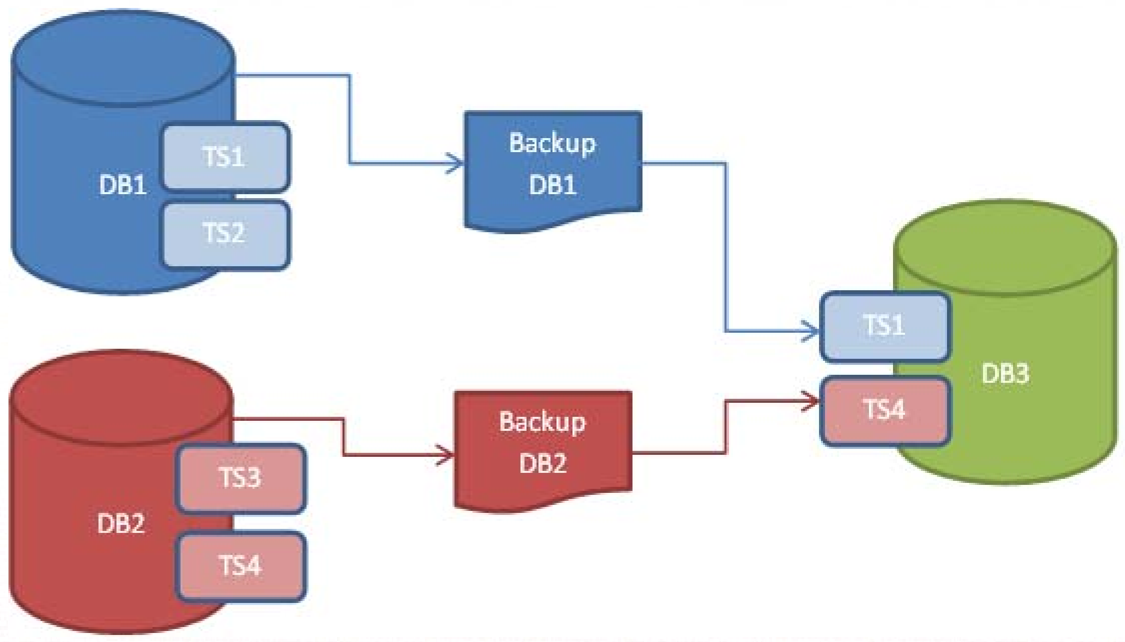
- Introduction
- Recovery Granularity
  - Object level, e.g. table recovery
  - Tablespace level
  - Partition level
  - Database level
- Making backups run faster
- Best Practices

Table level Recovery

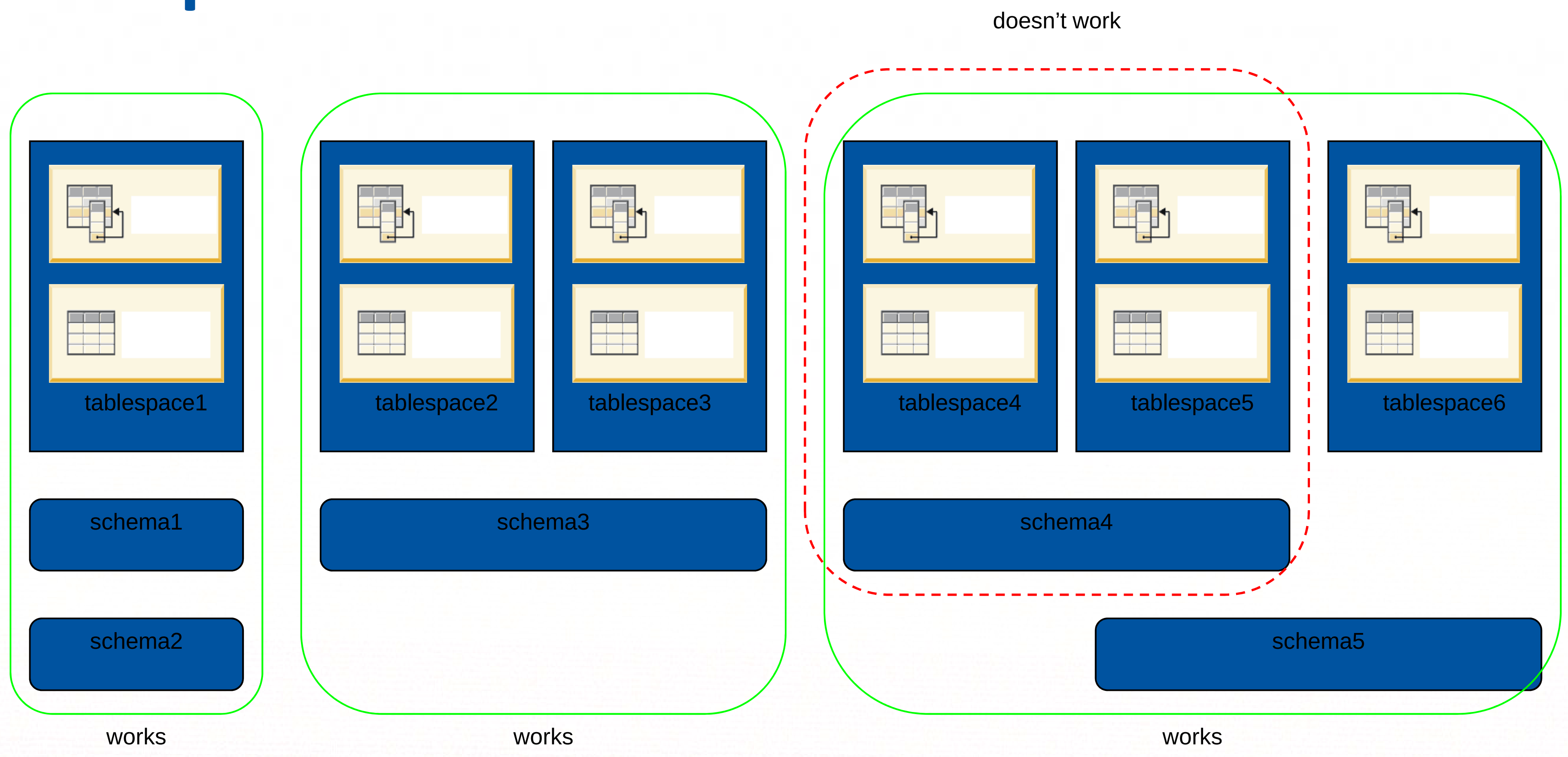
- Finest granularity of backup is the tablespace
- Question: Do I need to restore all of the tablespaces in a backup?
- Restore can be performed at either the same level of the backup OR at the tablespace level
- If a full backup image is taken it is possible to restore only a subset of the tablespaces(s) in that image.

Transportable Schema

Ability to restore a tablespace from a database into a completely separate database



Transport Sets



What will be transported

- Tables / CGTTs / MQTs / ‘normal + stats’ Views
- Generated columns
  - Expression
  - Identity
  - Row change timestamp
  - Row change token
- UDFs / UDTs + generated functions
- Constraints
  - Check
  - Foreign key
  - Unique / primary
  - Functional dependency
- Indexes
- Triggers
- Sequences
- Procedure – not external routine executable
- Object authorizations / privileges / Security / Access control / Audit
- Table Statistics + profiles / hints (plan lockdown)
- Packages

What will NOT be transported

- Created Global variables
- Aliases
- Jobs
- Index extensions
- Hierarchical tables, typed tables, typed views
- Nicknames
- Structured types
- methods
- Servers
- Wrappers
- Functional mappings & templates
- OLE DB External functions
- sourced Procedures
- XML - sysxmlstrings and sysxmlpaths collisions an issue

DPF and pureScale environments are not be supported

Agenda

- Introduction
- Recovery Granularity
  - Object level, e.g. table recovery
  - Tablespace level
  - Partition level
  - Database level
- Making backups run faster
- Best Practices

DPF Recovery

- Each partition contains
  - 1/nth of the data
  - Independent log stream
  - Can be recovered independently
- Minimal Recovery Time (MRT) is more complex to determine
  - When the "LIST TABLESPACES SHOW DETAIL" command or "GET SNAPSHOT FOR TABLESPACES" command is executed, one of the attributes that may be displayed is the "Minimum recovery time".
- Only need to recover on the partitions that have been affected by the problem, which may be a subset of the partitions

Agenda

- Introduction

- Recovery Granularity
  - Object level, e.g. table recovery
  - Tablespace level
  - Partition level
  - Database level

- Making backups run faster

- Best Practices

DPF Backup : Difficulties Faced by DBAs

- Current procedure for taking offline backup of DPF database is not fully parallel:

```
db2_all “<<+0<   db2 backup db foo”
db2_all “<<-0<|| db2 backup db foo”
```

- The catalog must be backed up separately
  - Simultaneous exclusive connections to catalog and non-catalog nodes not allowed, currently

DPF Backup : Difficulties Faced by DBAs

- Each partition gets a different timestamp:

```
$ db2_all “db2 backup db foo”
Backup successful. The timestamp for this
   backup image is : 20170201143307

Backup successful. The timestamp for this
   backup image is : 20170201143310

Backup successful. The timestamp for this
   backup image is : 20170201143310
```

DPF Backup : Difficulties Faced by DBAs

- What is the minimum set of logs needed to roll forward a DPF database?
  - Based on the time that the backup finishes on the last partition to finish
  - Roll forward all partitions to at least this time

- How do you determine this time?

```
 $ db2 rollforward db foo to 2016-01-01 and stop
 SQL1275N  The stoptime passed to roll-forward must be
greater than or equal to "2017-10-01-19.27.43.00 UTC",
because database "BAZ" on node(s) "0,1" contains
information later than the specified time.

 $ db2 rollforward db foo to 2017-10-01-19.27.43.00 and
stop
 DB20000I  The ROLLFORWARD command completed successfully.
```

The solution : Single System View Backup

- The BACKUP command is enhanced to take a list of database partitions
  - All partitions backed up in parallel (including catalog)
  - All other BACKUP options are supported as usual

```
$ db2 backup db foo on all dbpartitionnums
$ db2 backup db foo on dbpartitionnums (0, 30)
$ db2 backup db foo on all dbpartitionnums except
     dbpartitionnums (10, 20)
```

- Returns single timestamp

```
0000  DB20000I  BACKUP DATABASE completed successfully
0010  DB20000I  BACKUP DATABASE completed successfully

Backup successful. The timestamp for this backup image
   is : 20170102030405
```

The RECOVER Command

- DPF Aware Restore command, combining RESTORE and ROLLFORWARD
- Works with FULL DB backups by default, utilizing the recovery history files to locate the backup images
  - Db2set DB2_RECOVER_WITH_REBUILD=yes to allow for recover to use tablespace level backup images
- Able to specify which partitions to include OR exclude
- Examples (each applies to single- and multi-partitioned databases)
  - Recover a database to the end of logs from best available backup

```
RECOVER DB <dbname>
```

  - Recover a database to a particular point in time
    - Note that default time specification is in local time (not GMT)

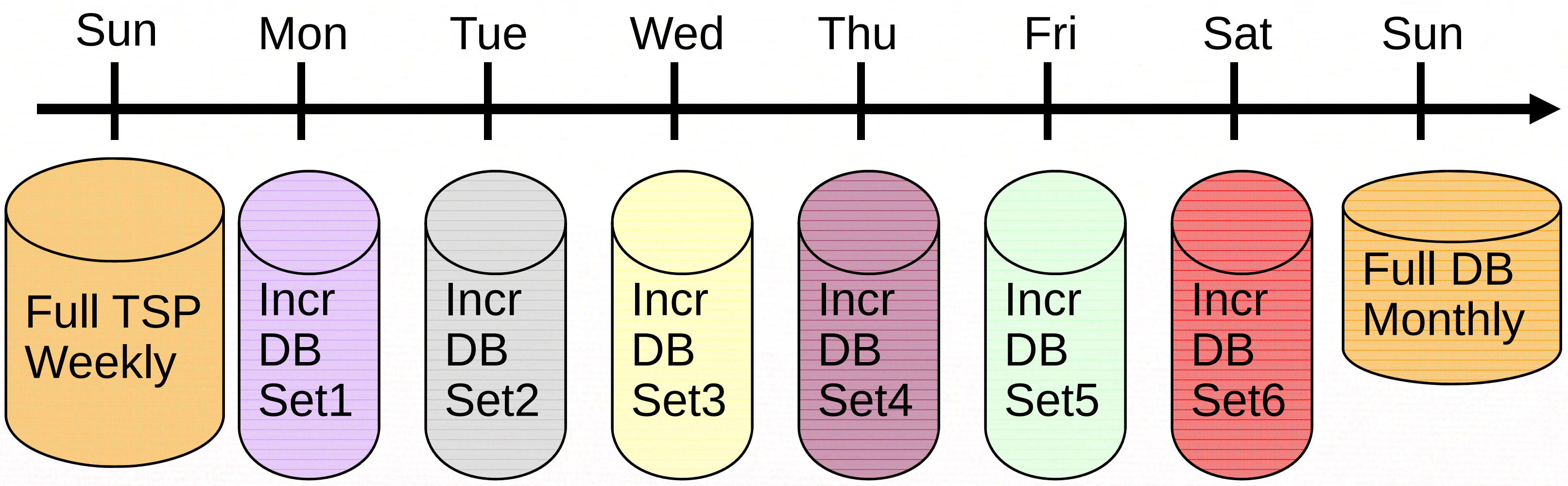
```
RECOVER DB <dbname>  TO 2016-12-31-04.00.00
```

  - Recover a point in time not represented in current history file

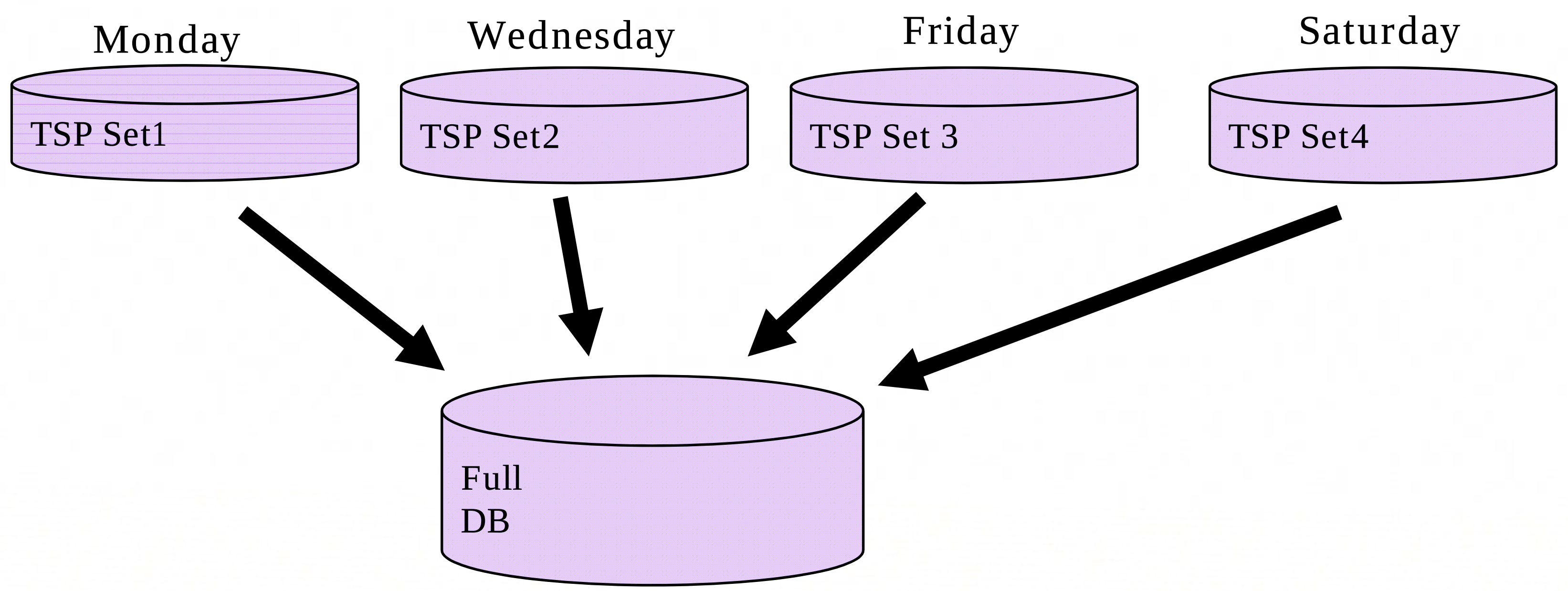
```
RECOVER DB <dbname>  TO 2016-12-31-04.00.00
  USING HISTORY FILE (/home/user/fromimage/db2rhist.asc
```

Rebuild Partial Database Scenarios

- Eliminate the need for FULL db backup
- Ability to rebuild entire DB, including DPF, from set of table space backup images
- Ability to bring only a portion of the DB online
- Scans the history file to build the list of images to restore
- **Design for recovery speed NOT backup performance**



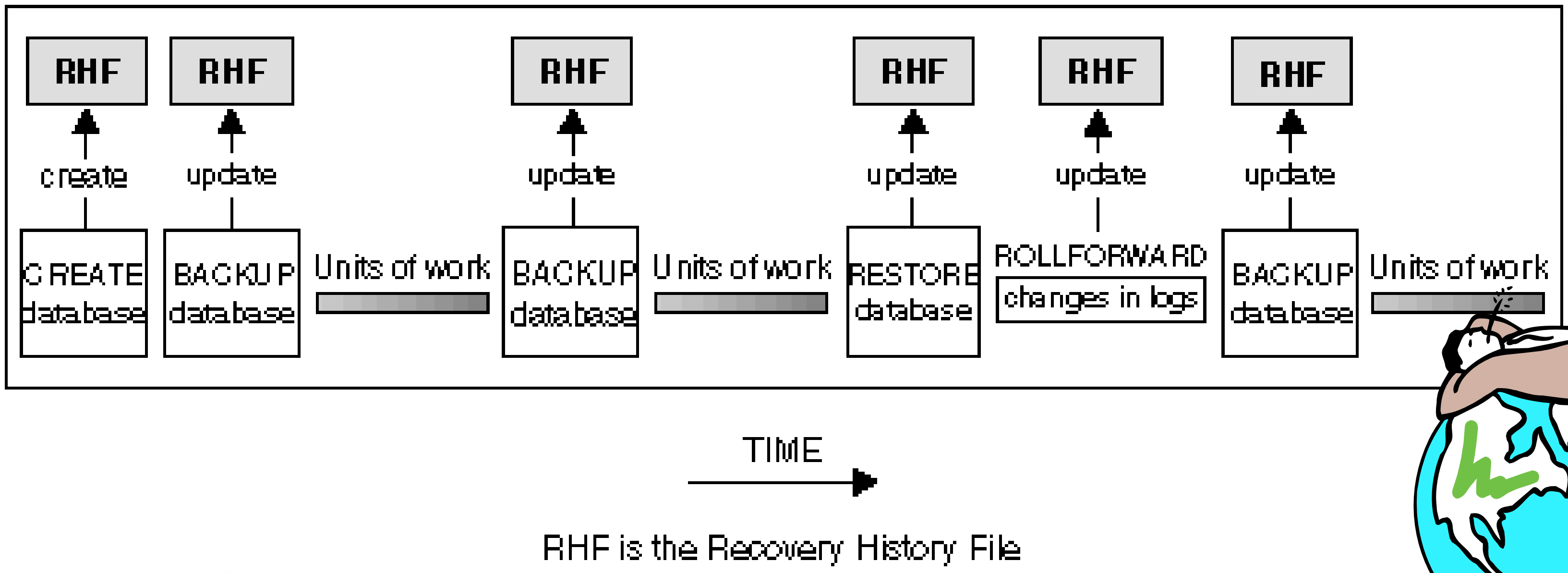
Rebuild DB from TSP Image



- Never require a FULL DB backup ever again
- Rebuild DB from a set of TSP images
- Records recovery point for each tablespace
- Multiple restores and a single rollforward command

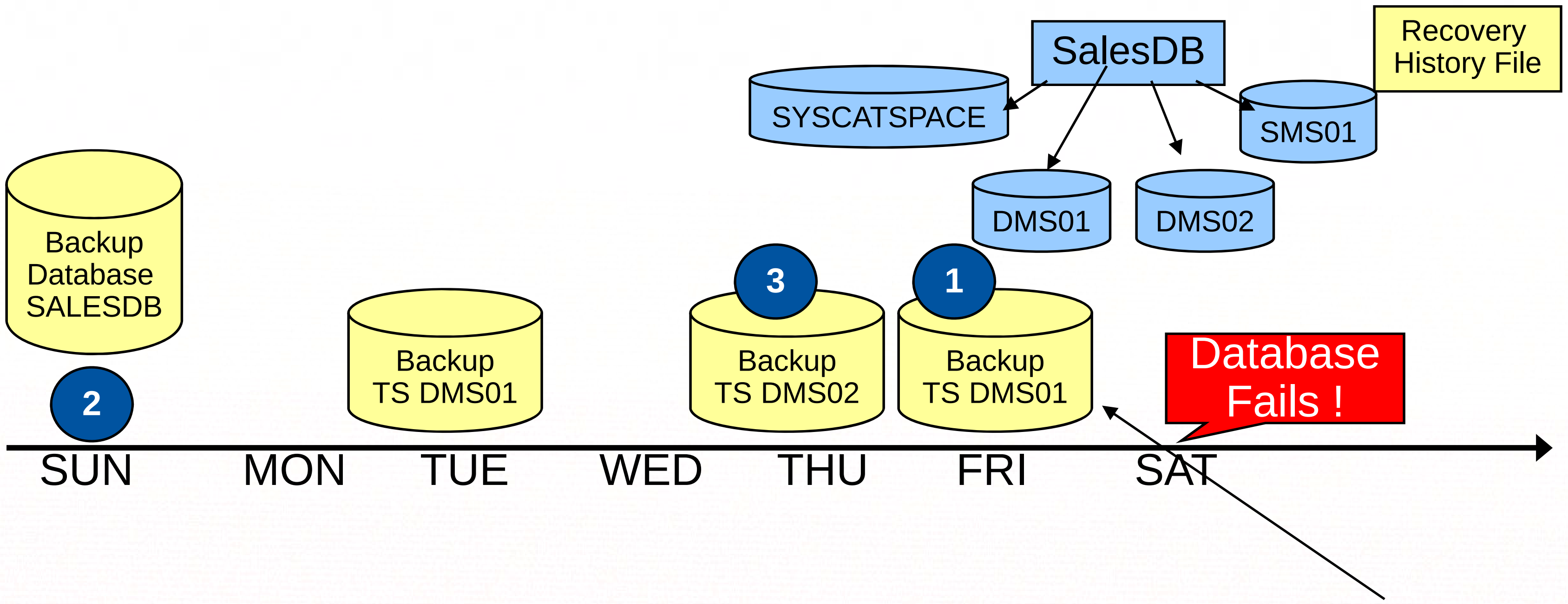
The importance of the Database and Table Space Recovery History Files

- **Database recovery file: db2rhist.asc**
- **Table space recovery file: db2tschg.his**
- **Do not delete! Maintain using ‘db2 prune’ command**



Recovery Scenario 1: Catastrophic Failure – you lost your IO Subsystem – the entire database is lost!

- Yes, you can rebuild your entire database from all the tablespace backups!
- Necessary Table spaces will be restored automatically based on Recovery History data

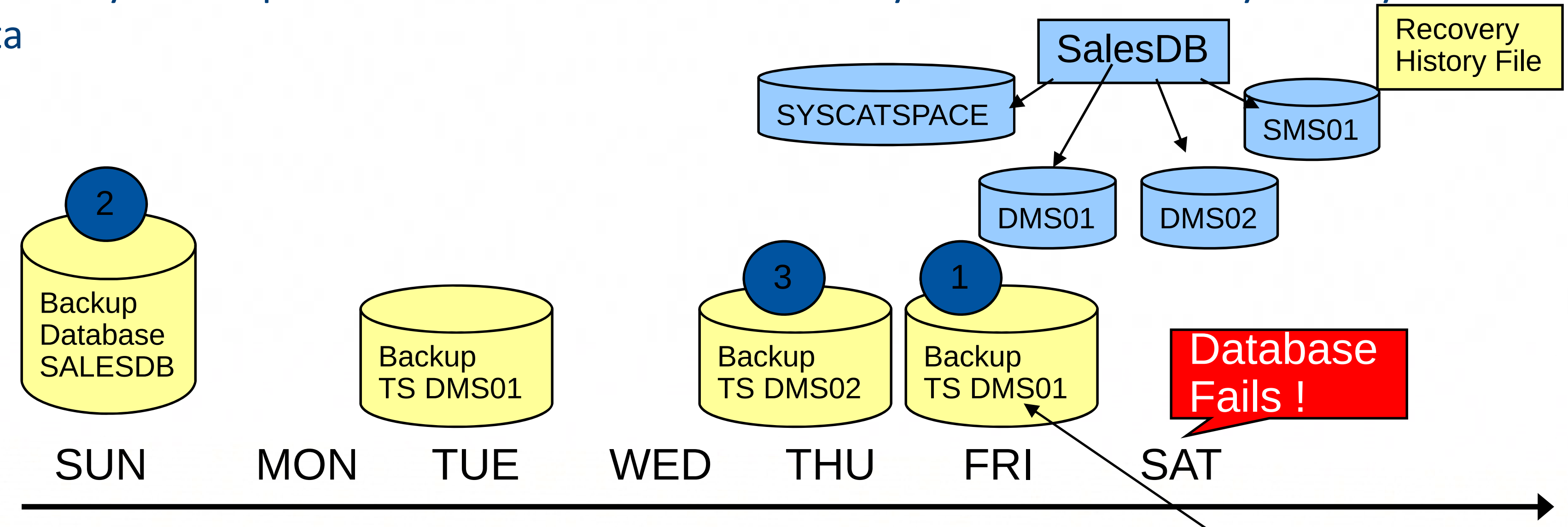


Recovery Scenario 1:
Database recovery to end of logs

- Use restore with the rebuild option:
  db2_all "db2 \"restore db salesdb rebuild with all tablespaces use tsm replace existing”
  - Reads backup history file, and figures out which tablespace backup images are needed and the corresponding logs
- Issue rollforward:
  db2 "rollforward db salesdb to end of logs and stop”

Database Recovery using Rebuild

- Necessary Table spaces will be restored automatically based on Recovery History data

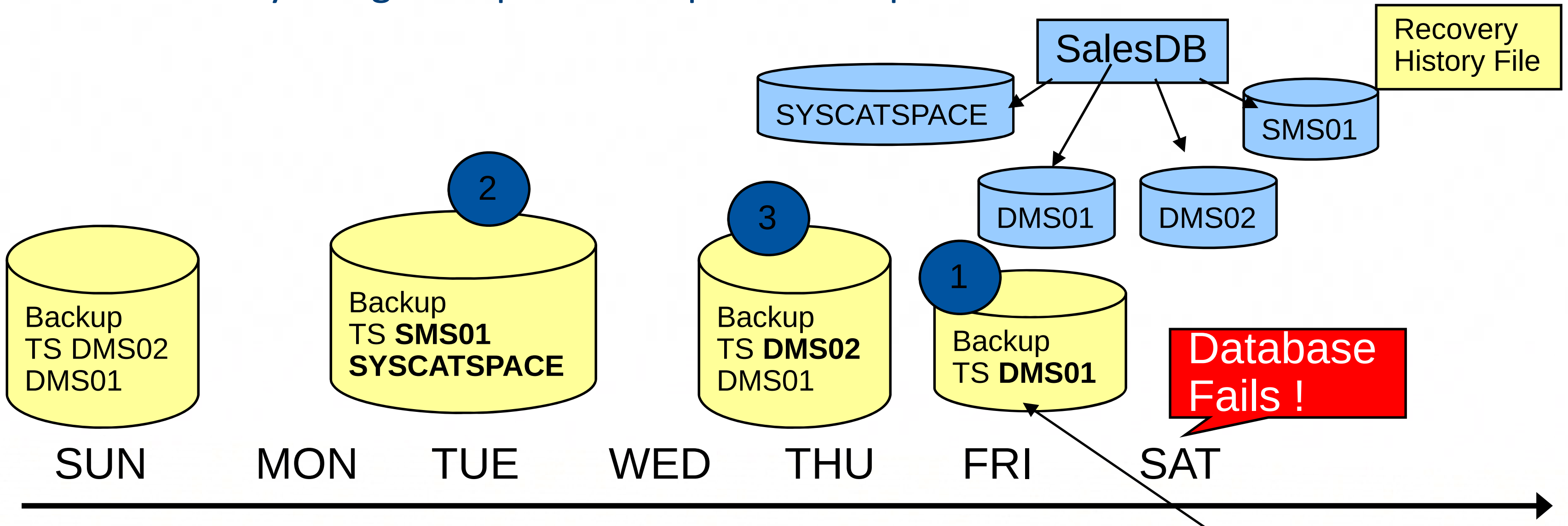


1. RESTORE DB SALESDB
   REBUILD WITH ALL TABLESPACES IN DATABASE TAKEN AT *Friday*
   (DB2 Restores Friday, Sunday and Thursday)

2. ROLLFORWARD DB SALESDB TO END OF LOGS AND STOP

Database Recovery - Rebuild from TS Backups

- Database Recovery using multiple Table space backups

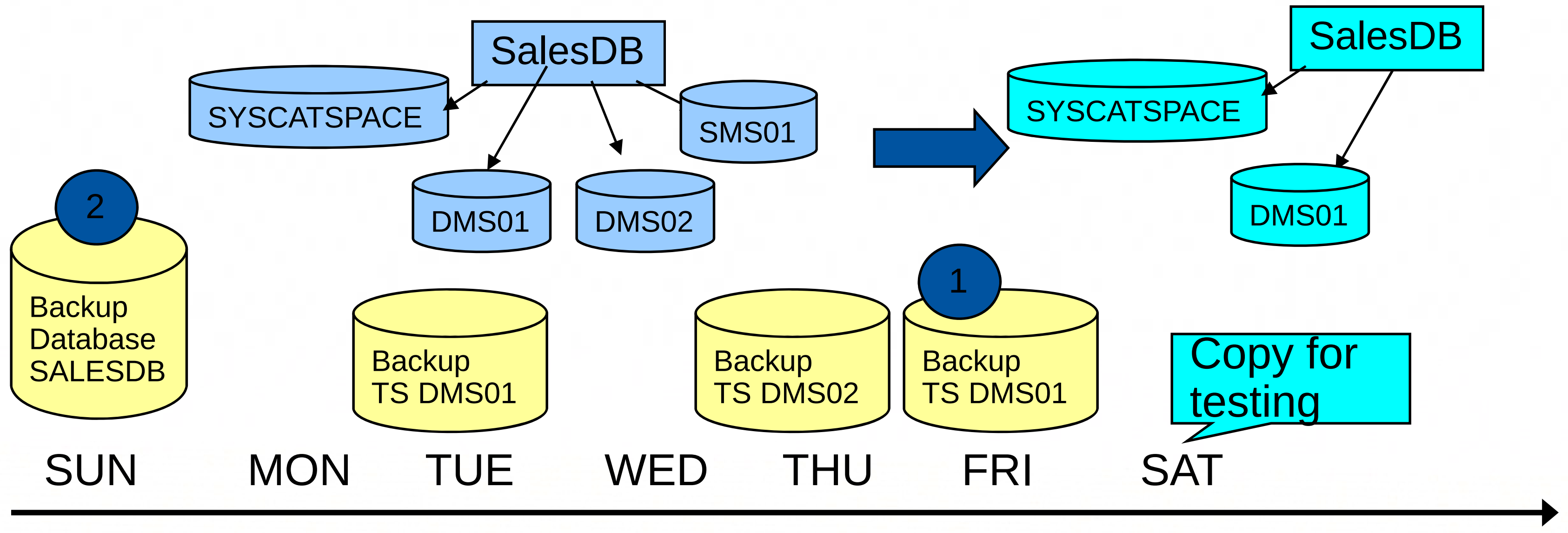


1. RESTORE DB SALESDB
   REBUILD WITH ALL TABLESPACES IN DATABASE TAKEN AT *Friday*
   (DB2 Restores Friday, Tuesday and Thursday)

2. ROLLFORWARD DB SALESDB TO END OF LOGS AND STOP

Database Partial Copy using REBUILD

- Can Create a Database copy with a subset of table spaces



1. RESTORE DB SALESDB
   REBUILD WITH TABLESPACE(SYSCATSPACE,DMS01) TAKEN AT *Friday*
   (DB2 Restores Friday, and Sunday (just SYSCATSPACE TS)

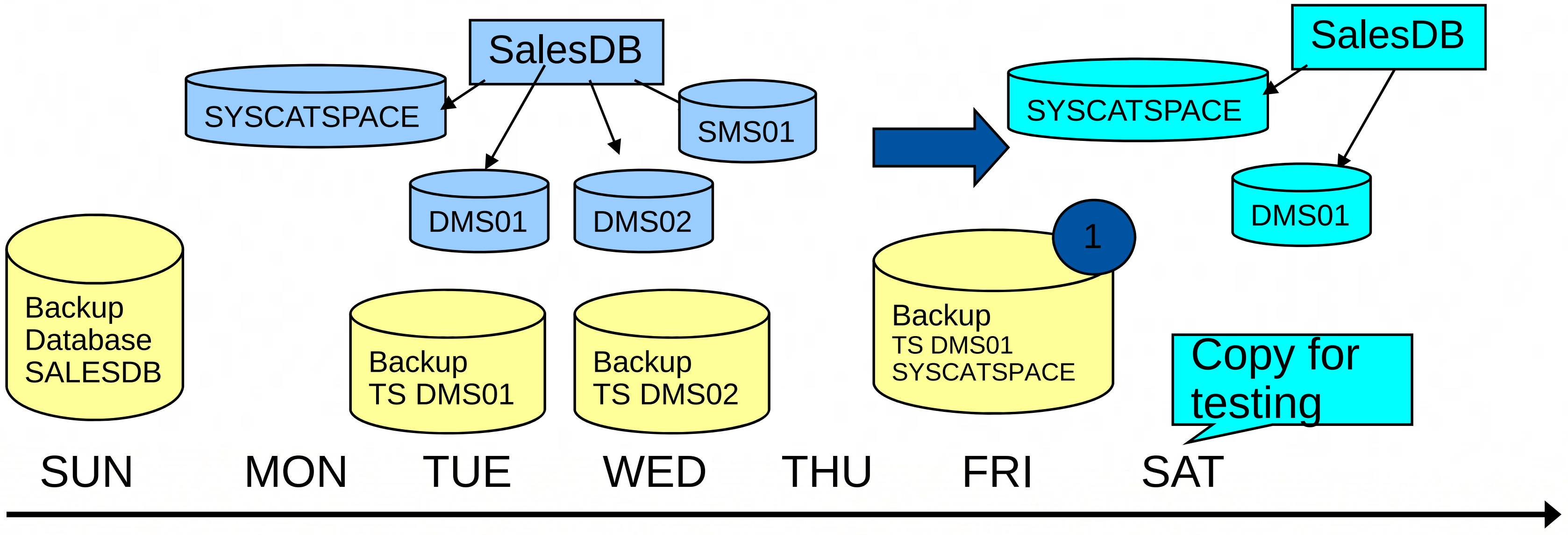
OR

RESTORE DB SALESDB REBUILD WITH ALL TABLESPACES IN DATABASE EXCEPT TABLESPACE(SMS01,DMS02) TAKEN AT *Friday*

2. ROLLFORWARD DB SALESDB TO END OF LOGS AND STOP

Database Rebuild using one TS Backup image

- Can Create a partial Database copy from one Table space Backup

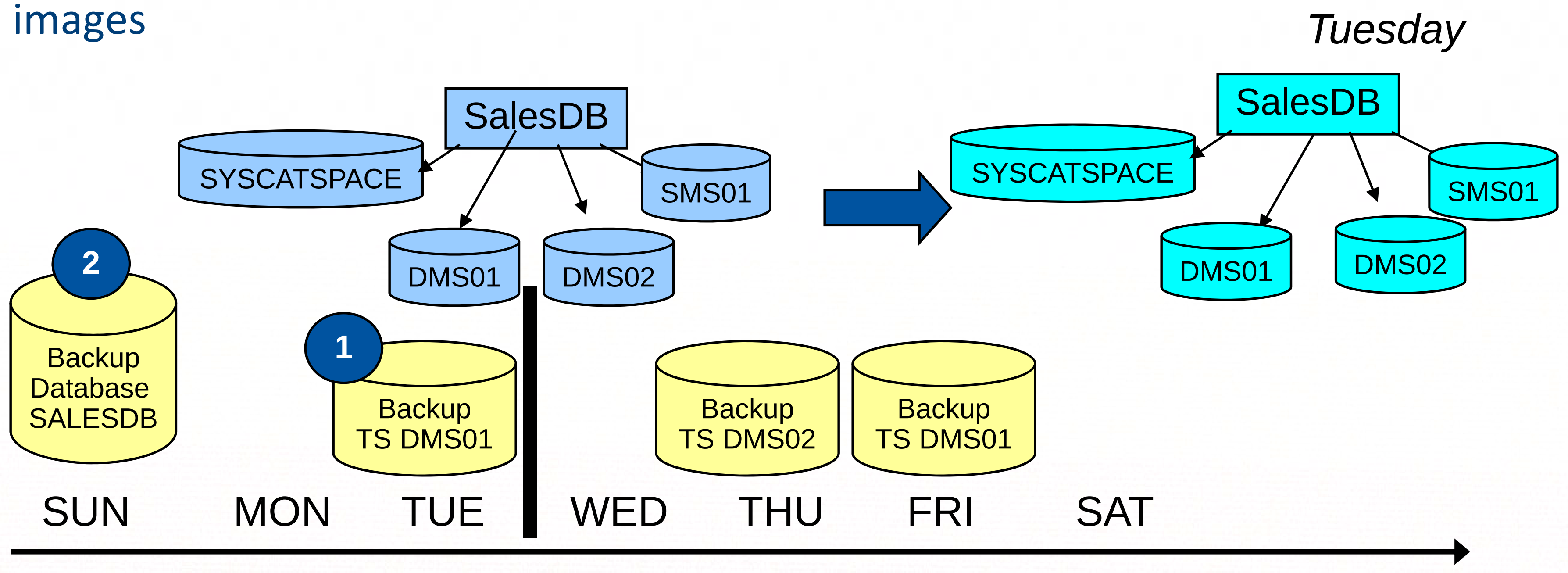


1. RESTORE DB SALESDB REBUILD WITH ALL TABLESPACES IN IMAGE TAKEN AT *Friday*

2. ROLLFORWARD DB SALESDB TO END OF LOGS AND STOP

Recovery Scenario 2: Logical Corruption – something happened on Wednesday, you need to go back to Tuesday night.

- Database point in time recovery, using REBUILD
- You can restore the database to a point in time, using your tablespace images



Recovery Scenario 2:
Logical Corruption - Database Recovery to Point in Time

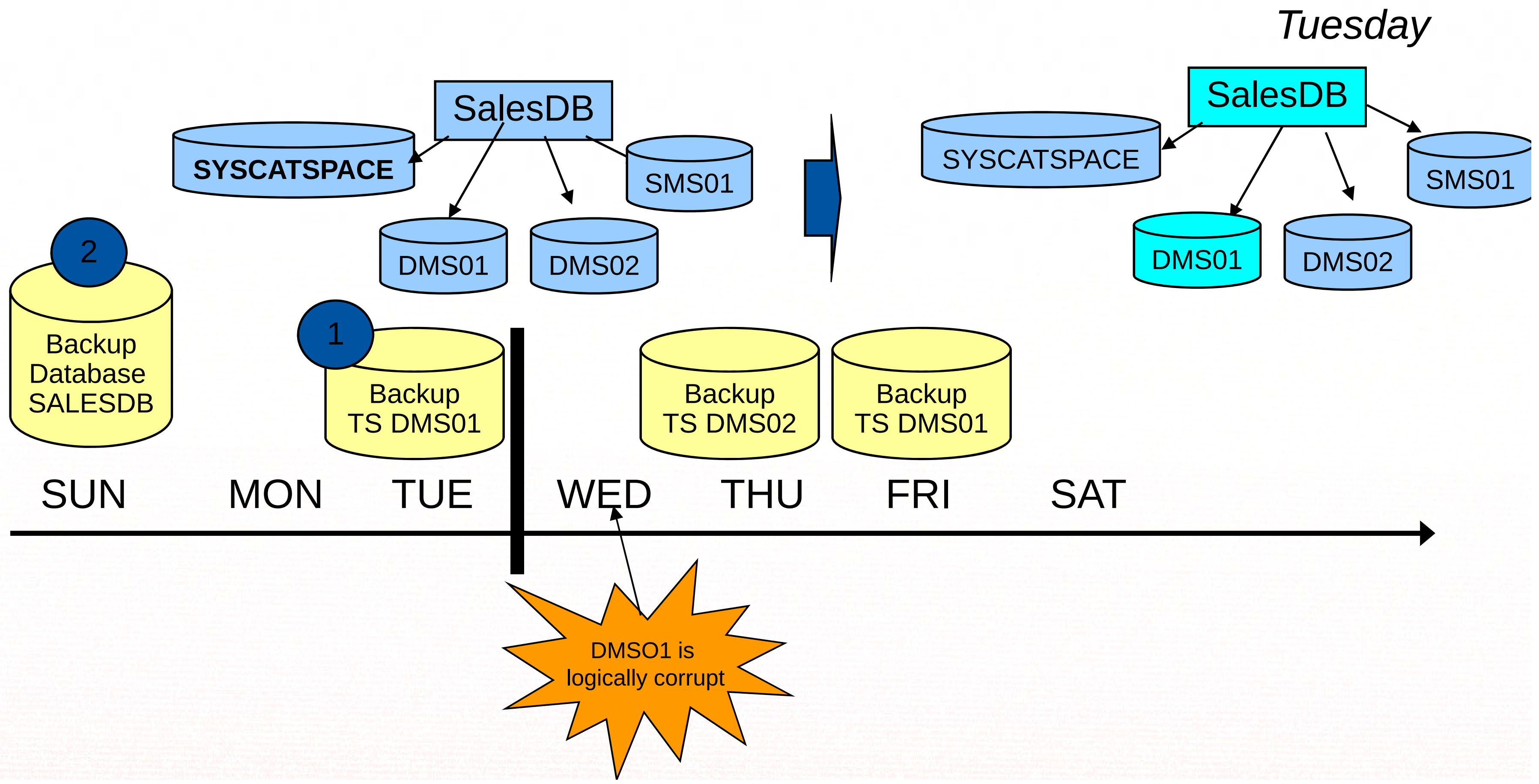
- Use restore with the rebuild option:
  db2_all "db2 \"restore db salesdb rebuild with all tablespaces use tsm replace existing"
  - Reads backup history file, and figures out which tablespace backup images are needed and the corresponding logs

- Issue rollforward:
  db2 "rollforward db salesdb to <timestamp>"

- Must recover all database partitions to the same point in time

Recovery Scenario #3: Logical Corruption: User Error on Wednesday, only affects a single table space.

- Table space point in time recovery using REBUILD
- Can Restore Catalogs and Selected Table space to point in time.
- Need to understand MRT – Minimum Recovery Time

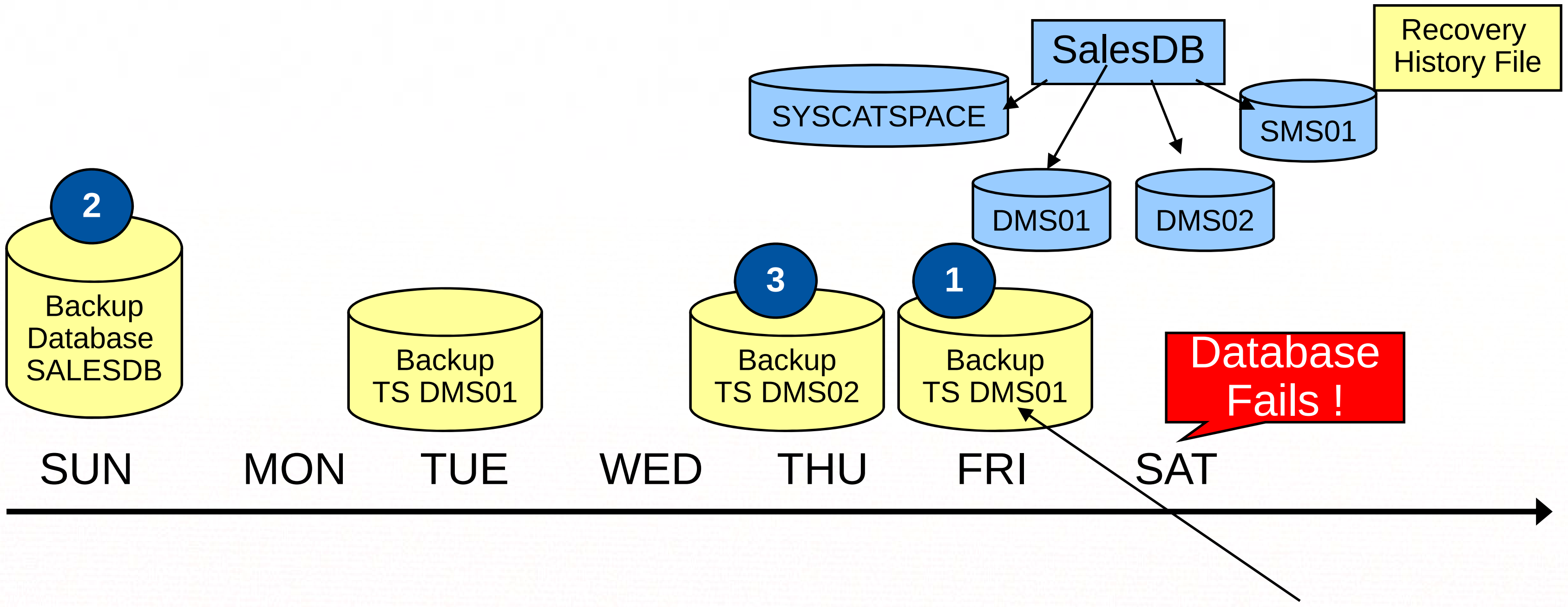


Recovery Scenario #3: Logical Corruption: User Error on Wednesday, only affects a single table space.

- A single tablespace needs to restored
- From the catalog node:
  - Start restore:
    db2_all “db2 \"restore db salesdb into tempDB rebuild with tablespace (syscatspace, dms01) use tsm taken at <tuesday date> replace existing
  - Rollforward:
    db2 "rollforward db tempdb to <point in time> and stop tablespace (syscatspace, dms01)"

Recovery Scenario 4: A single disk fails, so data is corrupted only on a single logical partition.

- Partition Level Recovery To End of Logs
- Necessary table partition will be restored automatically based on Recovery History data



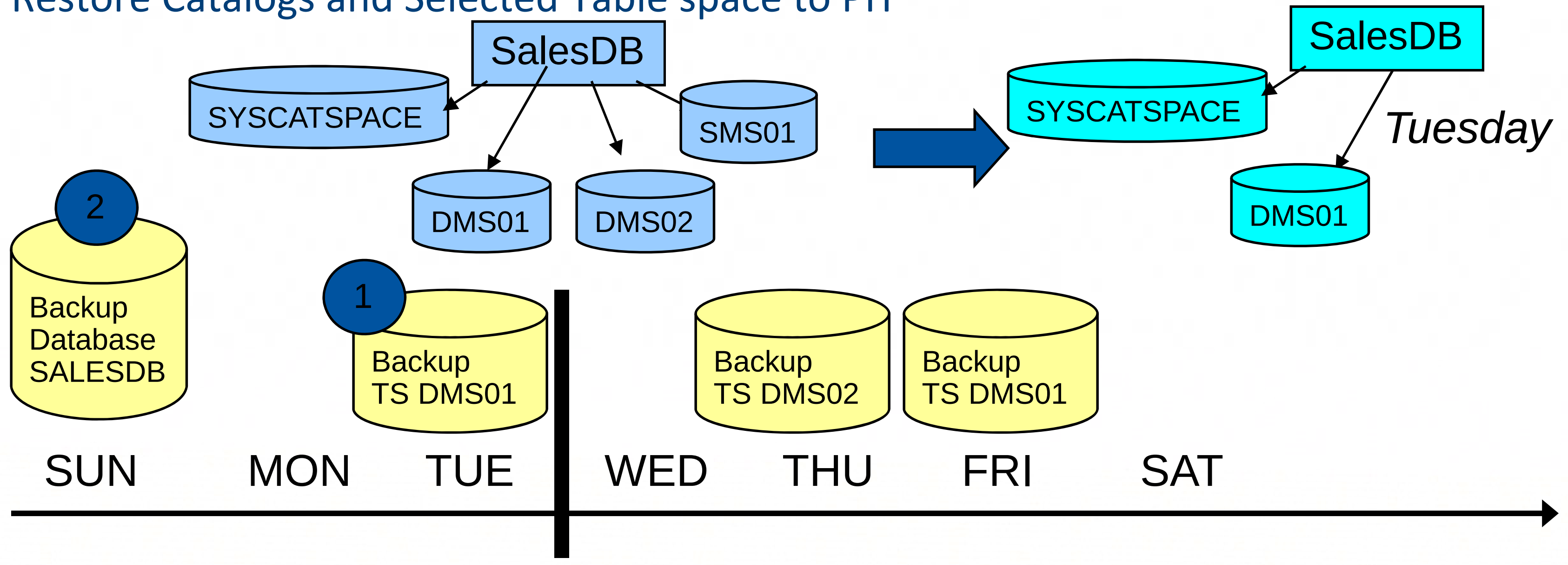
Recovery Scenario 4: A single disk fails, so data is corrupted only on a single logical partition.

- A single logical partition is corrupted (disk failure!)
- The example, db partition 3 is corrupted
- From the catalog node:
  - Start restore:
    db2_all "<<+3<db2 \"restore db salesdb tablespace (dms01, dms02) online use tsm taken at 20170401143616 replace existing
  - Check status:
    db2 rollforward db salesdb query status
    Result:
    3 **TBS pending** S0000000.LOG-S0000003.LOG 2017-03-29-07.42.44.000000 Local
    note: only partition 3 was restored, so only partition 3 needs to be rolled forward
  - Rollforward:
    db2 "rollforward db salesdb to end of logs on dbpartitionnum (3) tablespace (dms01, dms02) online"

- Can I use the RECOVER command?
- RECOVER DB SAMPLE TO END OF LOGS ON DBPARTITIONNUMS(3)
  - Uses last full DB backup

Table space point in time recovery using REBUILD

- Can Restore Catalogs and Selected Table space to PIT



1. RESTORE DB SALESDB
   REBUILD WITH TABLESPACE(SYSCATSPACE,DMS01) TAKEN AT *Tuesday*
   (DB2 Restores Tuesday, and Sunday (just SYSCATSPACE TS)

2. ROLLFORWARD DB SALESDB TO *Tuesday* AND STOP

Agenda

- Introduction
- Recovery Granularity
  - Object level, e.g. table recovery
  - Tablespace level
  - Partition level
  - Database level
- Making backups run faster
- Best Practices

Autonomics

Both backup and restore are self tuning

- If not specified the following settings will be computed
  - Parallelism
  - Buffer size – capped to 4097 x 4K pages
  - # of buffers

- All setting are dependent largely on the following settings:
  - UTIL_HEAP_SZ
  - # of CPUs
  - # of table spaces
  - Extent size
  - Page size

- Autonomics do handle all possible options
  - Compression
  - Data deduplication
  - …

How can I make backup run faster?

- **Distribute your data evenly across the tablespaces**
- **Let the DB tune the backup parameters**
  - Ensure you heap size is large enough (util_heap_sz 50000)
- **If backing up to a file system create multiple objects**
  - If the target device is a raid 5 file system creating multiple backup objects will speed up the process
  - Backup db sample to /mydb/backup, /mydb/backup, /mydb/backup, …
  - For TSM – open multiple sessions

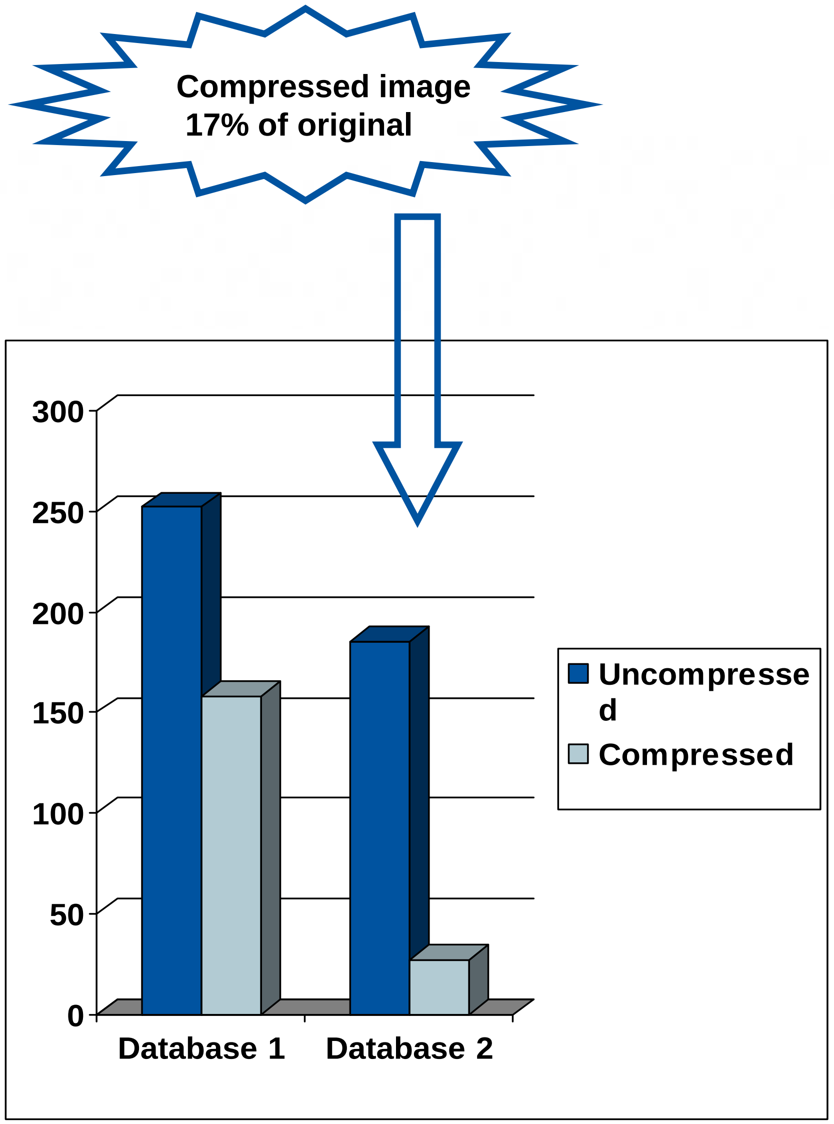
What about backup compression?

- **Compression can be used in 4 different areas**
  1. Table compression
     - Row level compression in the database
     - 10.1 introduced adaptive compression
  2. DB2 Backup compression
     - compressed while backing up
     - Requires additional CPU resources
     - Avoid is backup target is a data deduplication device
  3. TSM Software compression (if you have TSM)
  4. Storage (ie. hardware) level compression
     - Depends on your storage devices
- **Recommendation:**
  - Start with the following:
    - If you have table compression, you may not see a huge benefit with using db2 backup compression as well.
    - If you have storage level compression, then you do may not need to enable to TSM software compression.
- Test the combinations of compression types, to see what is the best fit, it will depend on the resource usage in your environment.

Backup Compression

- DB2 backups can now be automatically compressed
  - •Using built-in LZ compression or an (optional) user-supplied compression library
  - •Can significantly reduce backup storage costs

- Performance characteristics
  - CPU costs typically increased (due to compression computation)
  - Media I/O time typically decreased (due to decreased image size)
  - Overall backup/restore performance can increase or decrease
    - Depending on whether CPU or media I/O is a bottleneck



11.1

DB2 Support for the NX842 Accelerator

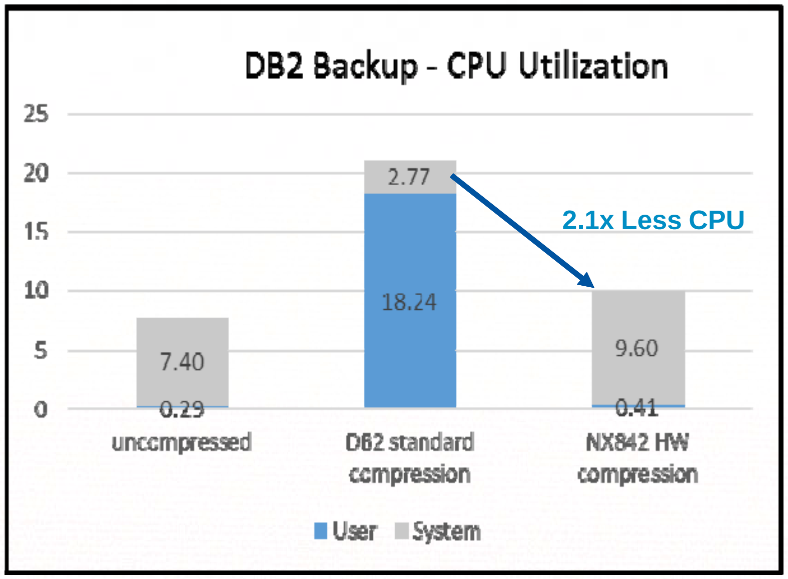
- DB2 backup and log archive compression now support the NX842 hardware accelerator on POWER 7+ and POWER 8 processors
- DB2 BACKUPs require the use of a specific NX842 library
  - `backup database <dbname> compress comprlib libdb2nx842.a`
- Backups can be compressed by default with NX842
  - Registry variable **DB2_BCKP_COMPRESSION** has to be set to **NX842**
  - Use the following backup command format:
  - `backup database <dbname> compress`
- Log archive compression is also supported
  - Update the database configuration parameter **LOGARCHCOMPR1** or **LOGARCHCOMPR2** to **NX842**

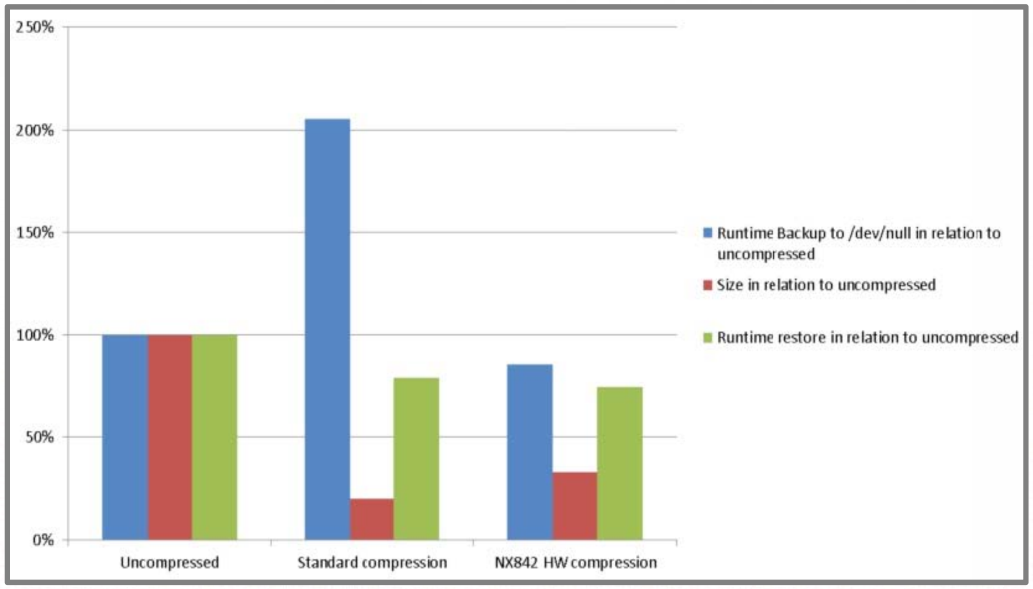
  -
    ```
    update database configuration for <dbname>
      using LOGARCHCOMPR1 NX842
    ```
  - Note: These two parameters can still take different values

11.1

DB2 Backup Compression Performance Results

- Preliminary results from early system testing
  - I/O bottleneck is usually the limiting factor in backup time
- About 50% DB2 backup size reduction compared to uncompressed
- Factor 2x less CPU consumption compared to DB2 standard compression





Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

The backup history file

- Each database has a backup history file that contains information about database recovery operations
  - BACKUP
  - RESTORE
  - ROLLFORWARD
  - Log file archive
- The history file is pruned periodically to keep its size manageable
  - After a full database backup
  - Using PRUNE HISTORY
- Automatic pruning is managed by two database config parameters
  - NUM_DB_BACKUPS
  - REC_HIS_RETENTN

How to manage the life cycle of backup assets

- The database configuration parameter that controls whether the underlying logs, backups and other associated objects are deleted when the history file is pruned
- AUTO_DEL_REC_OBJ
  - OFF (default, existing behaviour)
  - ON
- When does this automatic deletion occur ?
  - After a successful backup
  - On an explicit PRUNE HISTORY AND DELETE command
- What is deleted ?
  - Any full database backup images which exceed both NUM_DB_BACKUPS and REC_HIS_RETENTN db cfg parameters will be deleted
  - Any associated incremental backup images, load copy images, table space backup images or log files will be deleted

NO TABLESPACE Option for Backup Database

11.1

- DB2 History File
  - Contains information about log file archive location, log file chain etc.
  - If you use snapshot backups you want to have current information about log file location to perform point in time recovery (RECOVER command)
  - For RECOVER you need a current version of the history file
  - NO TABLESPACE backup allows you to create a current and consistent backup of the history file in a convenient way
  - If you use COMPRESS option of the BACKUP command, the created backup image is small
- BACKUP with NO TABLESPACE option
  - A NO TABLESPACE backup does not contain tablespaces
  - A no tablespace backup is used to restore the history file by using the HISTORY FILE option with the RESTORE DATABASE command
  - HISTORY FILE keyword is specified to restore only the history file from the backup image

Remote Storage Option for Utilities

11.1

- Remote storage is now accessible from:
  - INGEST, LOAD, BACKUP, and RESTORE
  - Accessed through the use of storage access aliases
- Supported Storage
  - IBM® SoftLayer® Object Storage
  - Amazon Simple Storage Service (S3)

Remote Storage Option for Utilities

11.1

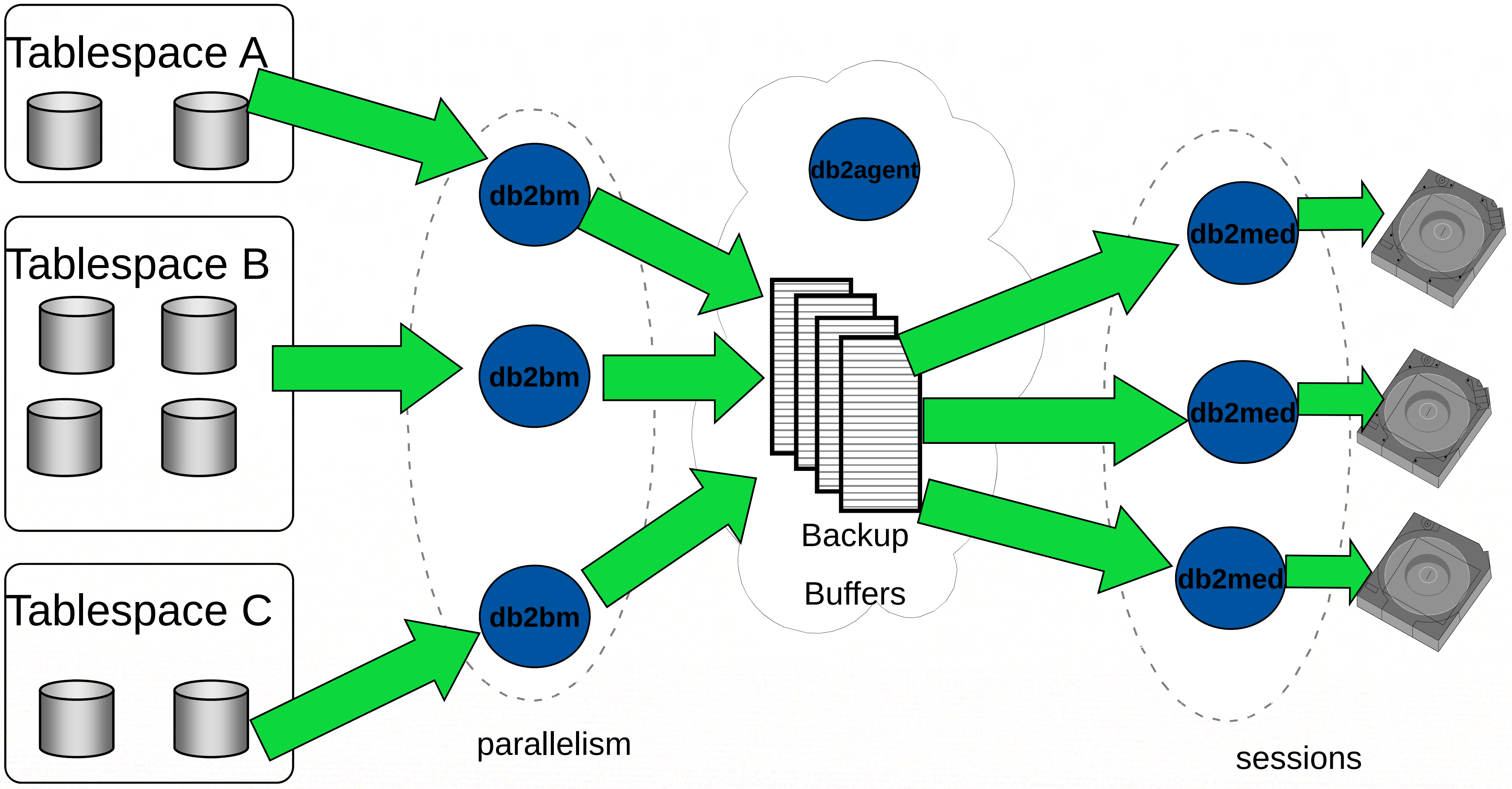
Catalog storage access first

- CATALOG STORAGE ACCESS ALIAS <alias> VENDOR (SOFTLAYER | S3) SERVER (DEFAULT | <endpoint>) USER <storage-user-ID> PASSWORD <storage-password> [CONTAINER <container-or-bucket>] [OBJECT <object>] [DBGROUP <group-ID> | DBUSER <user-ID>]

Backup to Swift

- db2 backup db testdb to db2remote://Alias//<storage-path>

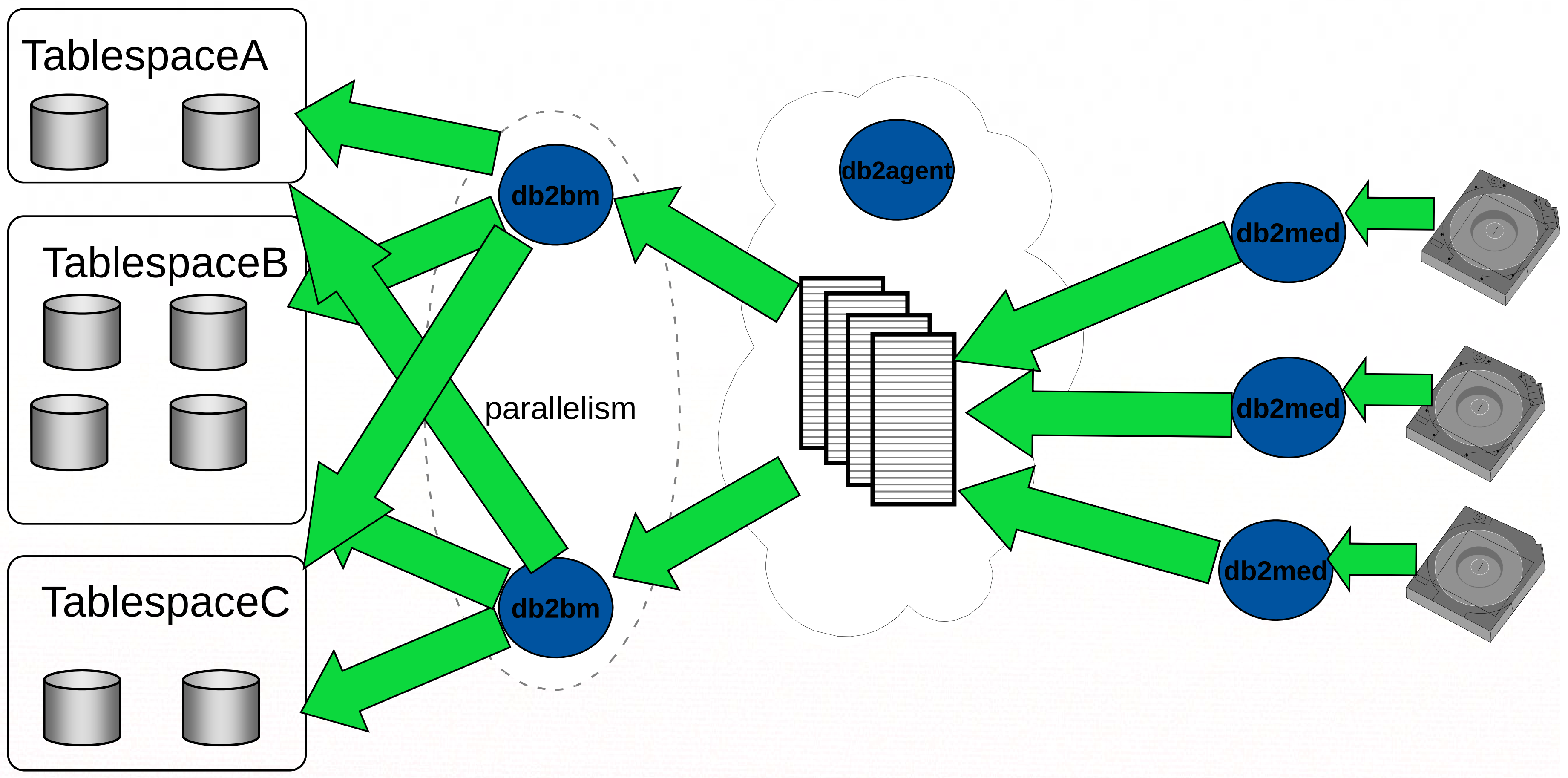
Backup Process Model



Backup Performance Tips

- Ensure data is distributed evenly across the tablespaces
- Ensure you are creating more than 1 backup file
  - Open x sessions to vendors
  - Create multiple output files on the filesystem by specifying the same filesystem multiple time
    - Db2 backup db sample to /db2backup, /db2backup, /db2backup
- Ensure there are sufficient buffers
  - Total number of buffers should be # BMs + # MCs + 2
  - Autonomics typically takes care of this
- Ensure parallelism is not set too high
  - Max value is the # of tablespaces
  - Recommended value is typically no more than 10

Restore Process Model



Restore Performance Tips

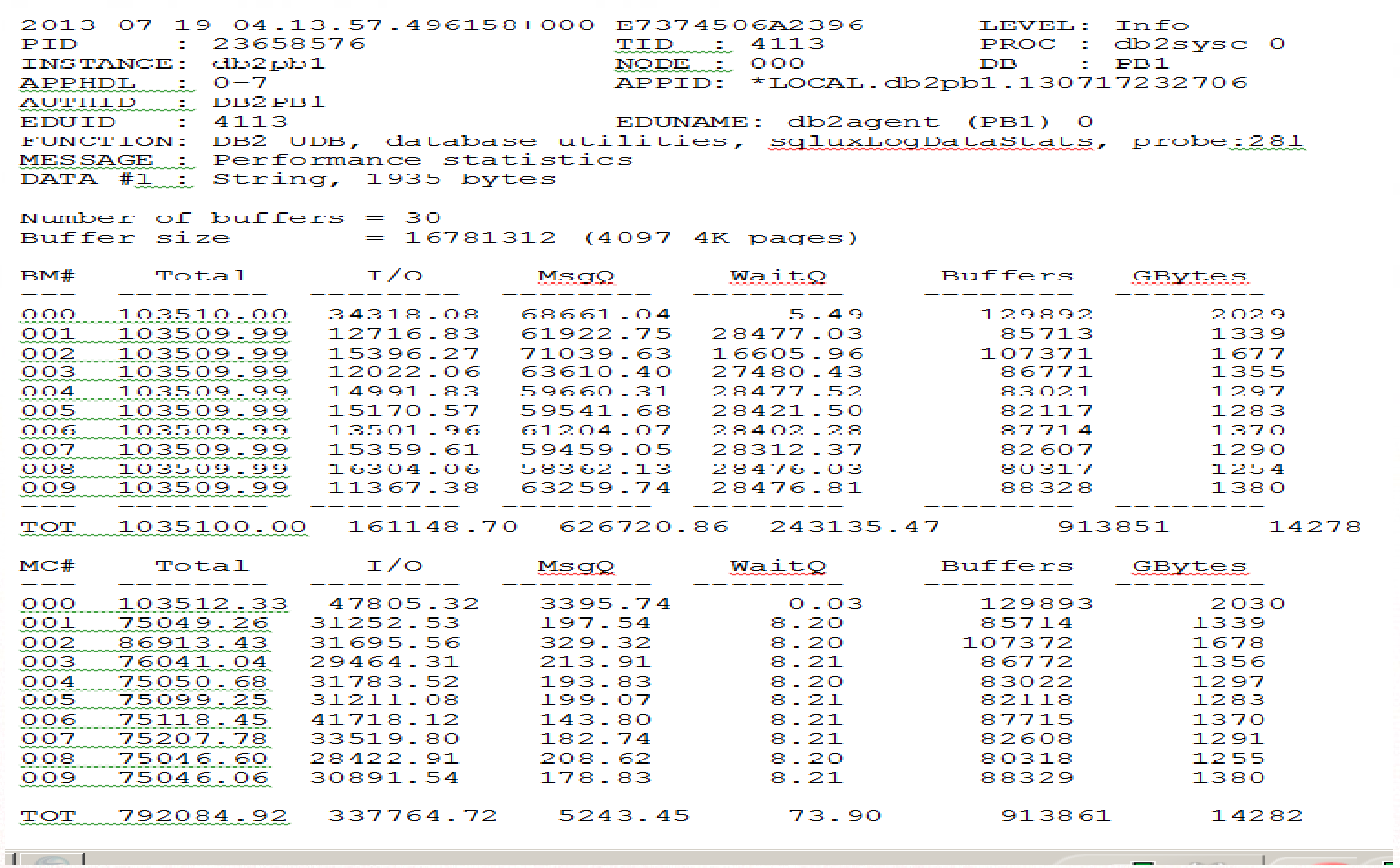
- Ensure you have more than 1 container per tablespace
  - Single container results in serial I/O
- Ensure your NUM_IO_SERVERS is sufficient
  - Number of tablespaces * number of containers per tablespace
  - Change back to automatic after recovery is completed
- Ensure you have enough restore buffers
  - # of BMs(Parallelism) + # of MCs (backup images) + 2

10.1

DB2_BAR_STATS

- New db2diag.log entry added at the end of each backup and restore operation
- Contains stats detailing where each BAR thread spent its time
- One entry per db2BM and per db2MED threads
- It was introduced in 9.7 under reg var control, and is enabled by default as of 10.1 FP2 (the reg var is no longer required).

Sample Output



```
2013-07-19-04.13.57.496158+000 E7374506A2396        LEVEL: Info
PID          : 23658576              TID    : 4113       PROC : db2sysc 0
INSTANCE: db2pb1                     NODE   : 000        DB   : PB1
APPHDL    : 0-7                      APPID: *LOCAL.db2pb1.130717232706
AUTHID    : DB2PB1
EDUID     : 4113                     EDUNAME: db2agent (PB1) 0
FUNCTION: DB2 UDB, database utilities, sqluxLogDataStats, probe:281
MESSAGE  : Performance statistics
DATA #1  : String, 1935 bytes

Number of buffers = 30
Buffer size       = 16781312 (4097 4K pages)

BM#     Total       I/O        MsgQ       WaitQ        Buffers    GBytes
---   ---------  ---------  ---------  ---------    ---------  ---------
000   103510.00  34318.08   68661.04        5.49       129892       2029
001   103509.99  12716.83   61922.75   28477.03        85713       1339
002   103509.99  15396.27   71039.63   16605.96       107371       1677
003   103509.99  12022.06   63610.40   27480.43        86771       1355
004   103509.99  14991.83   59660.31   28477.52        83021       1297
005   103509.99  15170.57   59541.68   28421.50        82117       1283
006   103509.99  13501.96   61204.07   28402.28        87714       1370
007   103509.99  15359.61   59459.05   28312.37        82607       1290
008   103509.99  16304.06   58362.13   28476.03        80317       1254
009   103509.99  11367.38   63259.74   28476.81        88328       1380
---   ---------  ---------  ---------  ---------    ---------  ---------
TOT   1035100.00  161148.70  626720.86  243135.47       913851      14278

MC#     Total       I/O        MsgQ       WaitQ        Buffers    GBytes
---   ---------  ---------  ---------  ---------    ---------  ---------
000   103512.33  47805.32    3395.74       0.03       129893       2030
001   75049.26  31252.53    197.54       8.20        85714      1339
002   86913.43  31695.56    329.32       8.20       107372      1678
003   76041.04  29464.31    213.91       8.21        86772      1356
004   75050.68  31783.52    193.83       8.20        83022      1297
005   75099.25  31211.08    199.07       8.21        82118      1283
006   75118.45  41718.12    143.80       8.21        87715      1370
007   75207.78  33519.80    182.74       8.21        82608      1291
008   75046.60  28422.91    208.62       8.20        80318      1255
009   75046.06  30891.54    178.83       8.21        88329      1380
---   ---------  ---------  ---------  ---------    ---------  ---------
TOT   792084.92  337764.72    5243.45      73.90       913861      14282
```

Explanation

BM# - the number we assigned to an individual Buffer Manipulator. BM's READ data from the databases tablespace during a backup and place them into buffers.

MC# - the number assigned to an individual Media Controller. MC's WRITE buffers out to the target location.

Total - The total amount of time spent by the process in seconds.

I/O - The amount of time spent either reading or writing data. For the BM's this represents time reading data from tablespace, and filling the buffer. For MC it's time spent reading from buffer and sending it to the target destination.

MsgQ - This is the amount of time we spend waiting to get a buffer. For BM's it's how long is spent waiting to get an empty buffer for filling. For MC's it's time spent waiting to get a full buffer in order to write out.

Wait Q - Amount of time spent waiting on directives from the agent overseeing the whole backup.

Buffers - The number of Buffers Processed by a particular BM or MC. A BM filled X number of buffers. An MC wrote out X number of buffers.

GBytes - The amount of data handled by a particular BM or MC in Gbytes.

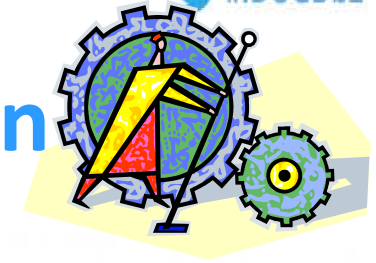
Analysis of DB2_BAR_STATS output

Num	ber of buff	ers = 30							
Buf	fer size	1678	1312 (4097	4K pages)					
BM#	Total	I/O	MsgQ	WaitQ	Buffers	GBytes	% Time on I/O	% time waiting for buffers	% time waiting for other threads
---	--------	--------	--------	--------	--------	--------			
0	103510	34318.08	68661.04	5.49	129892	2029	33.15%	66.33%	0.01%
1	103509.99	12716.83	61922.75	28477.03	85713	1339	12.29%	59.82%	27.51%
2	103509.99	15396.27	71039.63	16605.96	107371	1677	14.87%	68.63%	16.04%
3	103509.99	12022.06	63610.4	27480.43	86771	1355	11.61%	61.45%	26.55%
4	103509.99	14991.83	59660.31	28477.52	83021	1297	14.48%	57.64%	27.51%
5	103509.99	15170.57	59541.68	28421.5	82117	1283	14.66%	57.52%	27.46%
6	103509.99	13501.96	61204.07	28402.28	87714	1370	13.04%	59.13%	27.44%
7	103509.99	15359.61	59459.05	28312.37	82607	1290	14.84%	57.44%	27.35%
8	103509.99	16304.06	58362.13	28476.03	80317	1254	15.75%	56.38%	27.51%
9	103509.99	11367.38	63259.74	28476.81	88328	1380	10.98%	61.11%	27.51%
---	--------	--------	--------	--------	--------	--------			
TOT	1035099.9	161148.7	626720.8	243135.42	913851	14274	15.57%	60.55%	23.49%
MC#	Total	I/O	MsgQ	WaitQ	Buffers	GBytes	% time on I/O	% time waiting for buffers	% time waiting for agent
---	--------	--------	--------	--------	--------	--------			
0	103512.33	47805.32	3395.74	0.03	129893	2030	46.18%	3.28%	0.00%
1	75049.26	31252.53	197.54	8.2	85714	1339	41.64%	0.26%	0.01%
2	86913.43	31695.56	329.32	8.2	107372	1678	36.47%	0.38%	0.01%
3	76041.04	29464.31	213.91	8.21	86772	1356	38.75%	0.28%	0.01%
4	75050.68	31783.52	193.83	8.2	83022	1297	42.35%	0.26%	0.01%
5	75099.25	31211.08	199.07	8.21	82118	1283	41.56%	0.27%	0.01%
6	75118.45	41718.12	143.8	8.21	87715	1370	55.54%	0.19%	0.01%
7	75207.78	33519.8	182.74	8.21	82608	1291	44.57%	0.24%	0.01%
8	75046.6	28422.91	208.62	8.2	80318	1255	37.87%	0.28%	0.01%
9	75046.06	30891.54	178.83	8.21	88329	1380	41.16%	0.24%	0.01%
---	--------	--------	--------	--------	--------	--------			
TOT	792084.88	337764.7	5243.4	73.88	913861	14279	42.61%	0.57%	0.01%

Agenda

- Introduction
- Recovery Granularity
  - Object level, e.g. table recovery
  - Tablespace level
  - Partition level
  - Database level
- Making backups run faster
- Best Practices

Best Practices – Your database design



- Leverage intelligent database design practices!
  - All referential tables are in the same table space, or set of table spaces.
  - All staging tables (for load) are in a separate table space
  - Active vs. In-active data – separated by table space
    - Range partitioning – further identify active vs. inactive by designing ranges
  - Local vs. Global Indexes – if you have global indexes (ie. Prior to v9.7), then keep those in a table space that you back up with it's matching data table space. Rebuilding your indexes can be time consuming!
- Goal: you want to be set up to restore only what you need.

Best Practices: Backup recommendations

- Backup Images:
  - Online database backups
  - Online table space backups
  - Incremental backups
  - Compression is good *
- Use the Single System View option. Let DB2 manage the backups and timestamps for you.
- Always use lan-free option to write to tape drives directly via SAN
  - DO NOT archive directly to tape!
  - configure FAILARCHPATH in case primary-log-destination becomes unavailable
- Include the log records in each backup image – it's easier that way!
- Operating System files backed up twice per week

Best Practices - Logging

- Use archive logging – not circular logs for your production environment
- Goal: ease of maintenance
  - Include logs in your backup image
  - Use TSM to handle your archiving for you
- Small transaction logs:
  - Reduce risk since logs contain fewer transactions
  - Increases the number of logs required, since large transactions may span multiple log files, plus more overhead in system resources.
- Large transaction logs:
  - Increase risk since logs contain more transactions
  - Benefit: reduces the amount of system resources used, as less log switching takes place
- Recommendation: Start with 50 primary logs, of about 50MB each.
- Use Flash/SSD storage

Best Practices - Recovery recommendations

- For table space recovery, use the restore command with the REBUILD option
- Use the log manager, not user exits
- Use db2adutl to retrieve log files from tape to disk – keep ahead of the log files you need .

Backup and Recovery

- **If a DB2 online backup is not feasible due to conflicts with other utilities then consider SNAPSHOTS**

OR

- **IBM Tivoli Storage Manager for Advanced Copy Services For DB2**
  - http://www.ibm.com/developerworks/tivoli/library/t-acsdb2_1/index.html
- **When using TSM** always **use Lan-Free option**
- **Review SAP 20 TB BI Benchmark Results**
  - Full backup in < 7 hours
  - http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101012
- **Best Practices for backup and recovery in ISAS**
  - http://www.ibm.com/developerworks/data/bestpractices/isasrecovery/index.html

More Information

- DB2 LUW Best Practices: https://www.ibm.com/developerworks/data/bestpractices
- Best Practices: Building a Recovery Strategy for an IBM Smart Analytics System Database http://www.ibm.com/developerworks/data/bestpractices/isasrecovery/index.html
- Best Practices: Multi-Temperature Data Management http://www.ibm.com/developerworks/data/bestpractices/multitemperature/index.html
- Article: DB2 instance recovery for IBM Smart Analytics System http://www.ibm.com/developerworks/data/library/techarticle/dm-1010db2instancerecovery/index.html?ca=drs-

Thank You

IDUG
Leading the DB2 User Community since 1988

IDUG Db2 Tech Conference NA
Philadelphia, PA | April 29 - May 3, 2018

#IDUGDb2

Dale McInnis
IBM Canada Ltd.
dmcinnis@ca.ibm.com

Session code: C17

Please fill out your session evaluation before leaving!