

Are backups really needed anymore?

Dale McInnis
IBM Canada Ltd.

Session Code: D04
Tuesday Nov 9 08:30 | Platform: Linux, UNIX and Windows



In this presentation we will discuss the need, or lack there of, for database backups. There are many different techniques available today that can be used in lieu of a traditional backup. These techniques will be discussed and positioned with respect to the Recovery Point Object and Recovery Time Objective they provide.

Agenda

- Review of current backup and recovery functionality
- Are backups really needed anymore?
- Protecting yourself from physical corruption
- Protecting yourself from logical corruption
- What do I do for peta-byte size DBs?

Objective 1: Review of current backup and recovery functionality

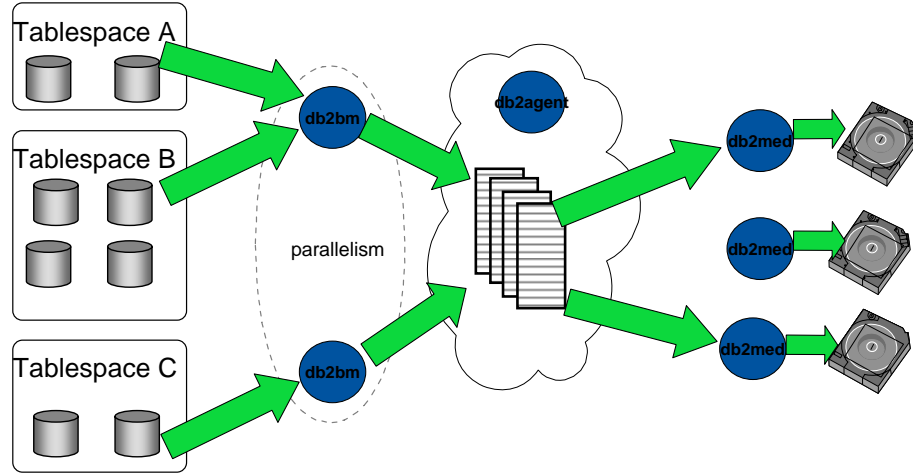
Objective 2: Are backups really needed anymore?

Objective 3: Protecting yourself from physical corruption

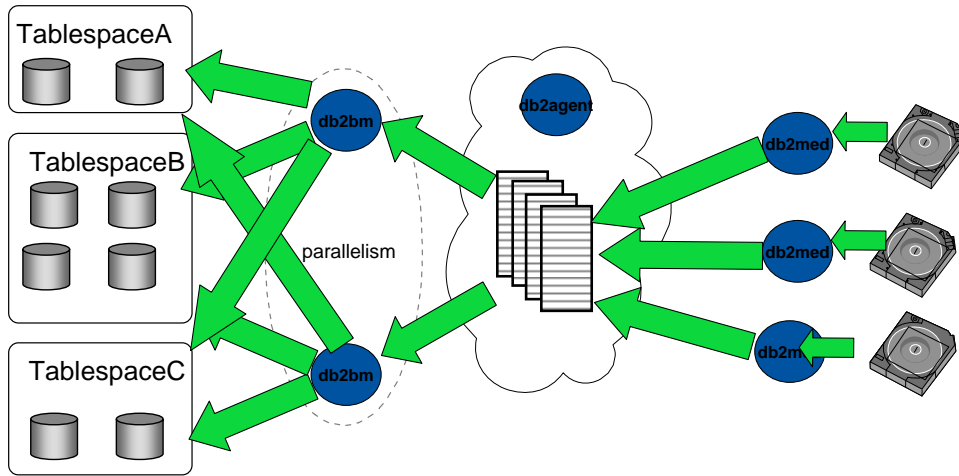
Objective 4: Protecting yourself from logical corruption

Objective 5: What do I do for the peta-byte size DBs?

Backup Process Model



Restore Process Model



Self-Tuning Backup/Restore

- Automatic, optimized settings for:
 - # buffers
 - Parallelism
 - Buffersize
- Based on:
 - # table spaces
 - Extentsize of each table space
 - # CPUs
 - Available memory
- Automatic settings chosen by default if any of the above parameters is not provided,
 - Not specified on command line
 - Set to 0 in API
 - Overrides BACKBUFSZ, RESTBUFSZ
- Built in compression capabilities
- Backup will always have precedence over all other automatic maintenance utilities

Florida State moved to V8.2 and found a 10x improvement in backup from their hand tuning efforts. Backup went from 4 hours and 15 minutes on V8.1 to 25 minutes in V8.2 *without* any work on the DBA's part.



Online Backup with 20 tape drives
2.81 TB/Hr
Restore from tape
2.1 TB/HR
Offline Backup to /dev/null 3.02TB/Hr



How can I make backup run faster?

- **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

DB2 UDB Backup and Restore Details

- Physical copy of table space pages from disk.
- Pages from possibly multiple table spaces are multiplexed and written to the target device(s) in randomized order.
 - Optimized for backup performance.
- Not a file based backup.
 - Integrated support to seamlessly handle raw containers.
 - I/O buffers of backup image data written directly to:
 - Destination devices.
 - 3rd party storage management software (ex. TSM).
- Not copied to the backup images:
 - Temporary table spaces.
 - Free DMS extents.

Logs Stored in Backup Images

- **Ensures backup will be restorable if archived logs misplaced**
- Supports all types of online backups, eg:
 - Database, Table space, Incremental, Compressed
- **Mechanism**
 - After the table space data is backed up, the tail log file will be truncated
 - All logs needed to restore the backup and roll forward to the time corresponding to the end of the backup, are placed in the backup image
 - Logs are backed up in parallel, similar to the mechanism used for table space data
 - Logs are placed at end of backup image, and compressed, if backup compression is specified

```

BACKUP DB <dbname> ONLINE to <backup_path> INCLUDE LOGS

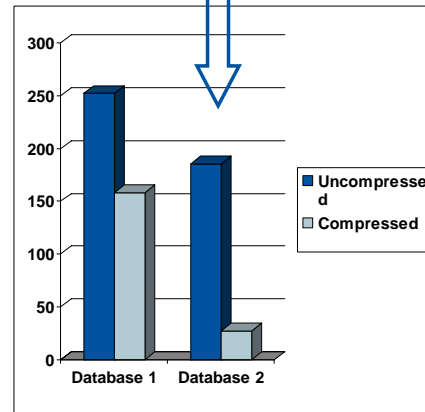
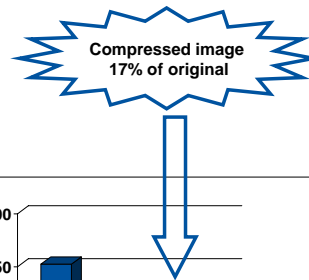
RESTORE DB <dbname> ONLINE FROM <backup_path> LOGTARGET <log_path>
ROLLFORWARD DB <dbname> TO END OF LOGS AND STOP
                                OVERFLOW LOG PATH <log_path>

// Or, to restore just the LOGS
RESTORE DB <dbname> LOGS FROM <backup_path> LOGTARGET <log_target>
  
```

You never have to worry about where are the logs required to make the DB transactionally consistent ever again.
 This is now the default for all online backups for DB2 9 for non-DPF DBs

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



9

Test System

126 GB database
4-way p660 with 4GB memory
Backup target
SSA drive

Database 1 : 251 GB Random Data

Uncompressed

Time : 2:25
Size : 252 GB

Compressed

Time : 4:26
Size : 185 GB

Database 2 : 158 GB ERP Data

Uncompressed

Time : 0:56
Size : 158 GB

Compressed

Backup Compression : Usage

```
DB2 BACKUP DATABASE <dbname> ... COMPRESS
DB2 RESTORE DATABASE <dbname> ...
```

```
DB2 BACKUP DATABASE <dbname> ... [COMPRESS
[COMPLIB lib-name [EXCLUDE]]
[COMPROPTS options] ]
DB2 RESTORE DATABASE <dbname> ...
[COMPLIB lib-name]
[COMPROPTS options]
```

```
DB2 BACKUP DB <dbname> COMPRESS COMPLIB
encrypt.a COMPROPTS "publickey"
DB2 RESTORE DB <dbname>
COMPROPTS "privatekey"
```

Typical usage

- Uses built-in LZ compression
- Automatically selected on restore
- Heterogeneous restore supported

Advanced usage

- Allows user-provided compression library
 - Library stored in backup image and used on restore, by default
 - For heterogeneous restore, need to provide platform-specific library on restore
- COMPROPTS allows options to be passed through to the library

Example : Encrypted Backups

- This example shows how a user-specified library can be used to support encryption

10

Need to document that the autonomic behavior is enabled by specifying number of Buffers = 0 and/or buffersize = 0 and/or parallelism = 0 which will be the default value if none is specified. CLP will be modified to supply the new default values of 0.

The parallelism calculation is common to both backup and restore and is located in sqluxGetDegreeParallelism (sqluagents.C)

Parallelism = min (# tablespaces, #CPUs * fudgeFactor); where fudgeFactor = 1 compressed backups, 3 for backup, 3 for restore)
 Never set Parallelism greater than 2 * # sessions as this would be a waste of resources

Backup Calculations:

Assume # buffers = 2 * # sessions

Buffersize is calculated to the lowest common multiple of the minimum buffer size and the extent size using Euler's algorithm for greatest common denominator, and

converting that into the LCM

Once this calculation is completed the # of buffers is calculated as minimum of (# sessions * 2), (available memory / backup buffer size). If there is not enough memory then

DB2 9 BAR Enhancements

- **Redirected Restore Script builder**
 - New feature added to RESTORE utility
 - Builds CLP script with a complete list of container paths
- **Rebuild Partial DB**
 - 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
 - Lots more on this later!

New redirected restore script builder integrated into RESTORE utility. No longer struggle to remember what containers were used, just run the restore with “GENERATE” option to provide a CLP script which you can edit using your favorite editor.

DB2 9.5 Backup and Recovery Enhancements

- **Single System View(SSV) in DPF**
 - Configuration
 - Hierarchical in nature, can be overridden on a partition level
 - Backup
 - Single command resulting in a single point of consistency across all partitions (DPF Aware Backup)
 - 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)
- **Logs in Backup now supported in DPF**
- **Rollforward to end of backup simplified:**
 - `db2 rollforward db foo to end of backup and stop`

12

Provide policy based definitions to maintain the required backup images and log files for you.

May move Recovery History File into a set of SQL tables to increase our flexibility wrt advanced policy maintenance

SSV –

1) Provide a single configuration file for ALL DPF partitions, but be able to override at a partition level

2) Provide a single backup command to backup all partitions, using a single backup timestamps and a consistency point in the logs across all partitions for all online backups. Still have the ability to backup partitions separately. If any node fails during the backup then the backup is considered useless and will be deleted on all partitions

Review : 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

Now before I tell you about what we're doing here,... let me review a bit of information about the history file

rec_his_retentn - Recovery history retention period configuration parameter

Configuration Type

Database

Parameter Type

Configurable

Default [Range]

366 [-1; 0 -- 30 000]

Unit of Measure

Days

This parameter is used to specify the number of days that historical information on backups should be retained.

If the recovery history file is not needed to keep track of backups, restores, and loads, this parameter can be set to a small number.

If value of this parameter is -1, the number of entries indicating full database backups

(and any table space backups that are associated with the database backup) will correspond with the value specified by the num_db_backups parameter.

Other entries in the recovery history file can only be pruned by 8 explicitly using the available commands or APIs.

No matter how small the retention period, the most recent full database backup plus its restore set will always be kept, unless you use the PRUNE utility with the FORCE option.

AUTO_DEL_REC_OBJ : The solution

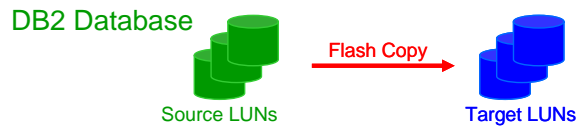
- A new database configuration parameter that controls whether the underlying logs, backups and other associated objects are deleted when the history file is pruned
- AUTO_DEL_REC_OBJ
 - OFF (default, existing behaviour)
 - ON
- When does this automatic deletion occur ?
 - After a successful backup
 - 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

DB2 9.5 support for Advanced Copy Services (ACS)

- New DB2 Advanced Copy Services (ACS) API enables integration with storage hardware
- A backup operation that uses DB2 ACS is called a snapshot backup
- Integrated into IBM Data Server is a DB2 ACS API driver for the following storage hardware:
 - IBM TotalStorage SAN Volume Controller
 - IBM Enterprise Storage Server Model 800
 - IBM System Storage DS6000
 - IBM System Storage DS8000
 - IBM N Series
 - NetApp V-series
 - XIV

Integrated Flash Copy Backup & Restore

- Flashcopy backup/restore just like any other DB2 backup
- Backup
 1. Identify LUN(s) associated with the database
 2. Identify free target LUN(s) for the copy
 3. Execute **DB2 BACKUP DB sample USE SNAPSHOT**
 4. Issue **DB2 STOP I/O** command to suspend database write I/Os
 5. Issue storage commands necessary to do the actual flash copy
 6. Issue DB2 RESUME I/O command to return DB2 to normal
- Restore
 1. Execute **DB2 RESTORE DB sample USE SNAPSHOT**
 2. Issue **DB2 ROLLFORWARD ...** command to rollforward
 3. Issue DB2 ROLL FORWARD command



16

Flash copy - DS6xxx, DS8xxx, ESS, N-Series and NetApp - no support for DS4xxx

We are working on getting the CIMS interface support which would provide support for EMC Symmetrix - meanwhile NetWorker is attempting to code to our interface for Symmetrix

In history file, etc

Comes with limited license of TDP

Enabling DB2 Advanced Copy Services (ACS)

- To use DB2 Advanced Copy Services (ACS) to perform snapshot backup operations, you must install, activate, and configure DB2 ACS.
 - Installation of DB2 ACS:
 - The files and libraries required for DB2 Advanced Copy Services (ACS) are installed by the IBM Data Server installer.
 - DB2 ACS supports a subset of hardware and operating systems:
 - Add a port for the DB2 ACS agent in the TCP/IP services file. For example: db2acs 5400/tcp # DB2 ACS service port.
 - Configure DB2 ACS:
 - Files for DB2 ACS are copied to sqllib/acs directory by db2icrt
 - Run the setup.sh script from the sqllib/acs directory without any parameters.
 - This will lead you through an interactive, text-based wizard that will configure DB2 ACS.
 - The wizard creates a configuration profile file and modifies the /etc/initab on the machine to trigger the launch of the DB2 ACS daemons.
 - Additional manual configuration may be required for some of the supported disk storage systems.

Performing a snapshot backup or restore

- DB2 ACS must be activated and configured for a DB2 database
- You can perform a snapshot backup using the **BACKUP DATABASE** command with the **USE SNAPSHOT** parameter
`db2 backup db sample use snapshot`
- A SNAPSHOT Backup can be ONLINE or OFFLINE
 - For Online SNAPSHOT backups DB2 will perform SET WRITE SUSPEND and SET WRITE RESUME automatically
 - DB2 Recovery History file is updated to record backup
- You can restore a snapshot backup using the **RESTORE DATABASE** command with the **USE SNAPSHOT** parameter
`db2 restore db sample use snapshot taken at 20080125080000`
- DB2 LIST UTILITIES command will show the stages of processing for SNAPSHOT Backup and Restore utilities but will not show progress as a percentage completed

Mapping of DB2 databases to storage subsystem volumes for using snapshot backups

- To use SNAPSHOT backup and restore functions DB2 databases need to be carefully configured on storage system volumes:
 - **Location** of database path and table spaces
The database path as well as all system, user and temporary table spaces need to be assigned to a set of storage system volumes that do not overlap any other DB2 database or even another database partition of the same database.
 - **Location** of active database logs
The active and mirror (mirrorlogpath) database logs need to be defined on a separate volume group. This ensures that the recovery logs are not overwritten during a snapshot restore.
 - **Location** of archive logs
Log files archived to local disk need to be defined on a separate volume group to avoid being overlaid by a snapshot restore

DB2 active log file backup and recovery considerations

- For the BACKUP utility with the USE SNAPSHOT option
 - INCLUDE LOGS is the default for both ONLINE and OFFLINE backups
 - Logs need to be placed on a separate volume group
- New options for handling logs by the RESTORE utility when the USE SNAPSHOT option is specified:
 - LOGTARGET EXCLUDE (default):
 - If this option is specified, then no log directories will be restored from the backup image.
 - If a path belonging to the database is restored and a log directory will implicitly be overwritten because of this, an error will be returned.
 - LOGTARGET INCLUDE:
 - If this option is specified and the backup image contains log directories, then they will be restored.
 - If existing log directories on disk conflict with the log directories in the backup image, then an error will be returned.
 - LOGTARGET INCLUDE FORCE:
 - Allow existing log directories in the current database to be overwritten and replaced when restoring the snapshot image.
 - Use this option with caution, and always ensure that you have backed up and archived all logs that might be required for recovery.

Limited options for Backup/Restore using SNAPSHOT

- The BACKUP command using SNAPSHOT can **NOT** include:
 - TABLESPACE (only Database level backups are allowed)
 - INCREMENTAL
 - COMPRESS
 - Performance options like BUFFER BUFFERSIZE, PARALLELISM or SESSIONS
- The RESTORE command using SNAPSHOT can **NOT** include:
 - REDIRECT
 - TABLESPACE
 - INCREMENTAL
 - REBUILD
 - Database or Automatic Storage path changes using TO, ON and DBPATH
 - Changes to database name using INTO
 - Changes to log path using NEWLOGPATH
 - LOGS ONLY
 - COMPRLIB
 - Performance options like BUFFER, BUFFERSIZE, PARALLELISM or SESSIONS

db2acsutil - Managing snapshot backup objects

- To list available snapshot backup objects, use **QUERY** parameter.
For example, to list available snapshot backup objects for the database manager instance named dbminst1, use the following syntax:

```
db2acsutil query instance dbminst1
```
- To check the progress of a given snapshot backup operation, use the **STATUS** parameter.
For example, to see the progress of snapshot backup operations that might be currently running on a database called database1, use the following syntax:

```
db2acsutil query status db database1
```
- To delete a particular snapshot backup object, use the **DELETE** parameter.
For example, to delete all snapshot backup objects for the database called database1 that are older than 10 days, use the following syntax:

```
db2acsutil delete older than 10 days ago db database1
```



Examples of db2acsutil command output

- Sample output for a snapshot backup with an active background copy.

```
db2acsutil query status db f01 instance db2inst1 dbpartitionnum 0
```

Instance	Database	Part	Image Time	Status
keon14	F01	0	20070719120848	Remotely mountable + Background_monitor pending (16 / 1024 MB)

- Sample output for a snapshot backup with a completed background copy.

```
db2acsutil query status db f01 instance db2inst1 dbpartitionnum 0 show details
```

```
Instance : keon14
Database : F01
Partition : 0
Image timestamp : 20070719120848
Host : machine1
Owner :
DB2 Version : 9.5.0
Creation time : Thu Jul 19 12:08:50 2007
First active log (chain:file) : 0:0
Metadata bytes : 6196
Progress state : Successful
Usability state : Remotely mountable + Repetitively restorable + Swap restorable
                  + Physical protection + Full copy
Bytes completed : 0
Bytes total : 0
```

DB2 Advanced Copy Services supported OS and HW

Hardware	AIX operating system with storage area network (SAN) storage	AIX operating system with network file system (NFS) storage	Linux operating system with network file system (NFS) storage
IBM TotalStorage SAN Volume Controller	Fully supported.	Not Supported	Not Supported
IBM Enterprise Storage Server Model 800	Full support except: incremental copying is not supported	Not Supported	Not Supported
IBM System Storage DS6000	Full support except: incremental copying is not supported	Not Supported	Not Supported
IBM System Storage DS8000	Full support except: incremental copying is not supported	Not Supported	Not Supported
IBM System Storage N Series	Fully supported.	Fully supported.	Fully supported.
NetApp V-series	Fully supported.	Fully supported.	Fully supported.
XIV	Fully Supported	Not Supported	Not Supported

DB2 9.7 Backup and Recovery Enhancements

- **Deduplication device support**
 - Native support for all deduplication devices
 - Starting with V 9.7 FP3

Transportable Schema – V 9.7 FP2

- The objective of this solution is to simplify the migration of a Sybase database solution to a DB2 database solution.
- The Sybase database architecture includes database objects called dbspaces. Each dbspace is independent and self contained; and Sybase database administrators (DBAs) can easily copy all the data in a dbspace from one database to another. When users migrate a Sybase database solution to a DB2 database, the data in the Sybase dbspaces is mapped to DB2 schemas. However, the DB2 database architecture does not facilitate copying data between DB2 databases by schema.
- Performance objective – 100 GB in under 20 minutes
 - 80GB out of 200 GB backup image consisting of 400 tables
- Customers want to ensure we minimize the amount of I/O that is done

Transportable Schema

Provide the illusion that we can restore any table space(s) into any database and then recreate all of the logical objects based off of the physical objects in the table space(s).

Essentially Oracle's Transportable Tablespaces feature on steroids (they only transport tables, indexes, views, RI constraints and triggers)

Interface will require the user to specify both the list of schemas to be transported as well as the list of table spaces (for now).

Restore will now do multiple operations

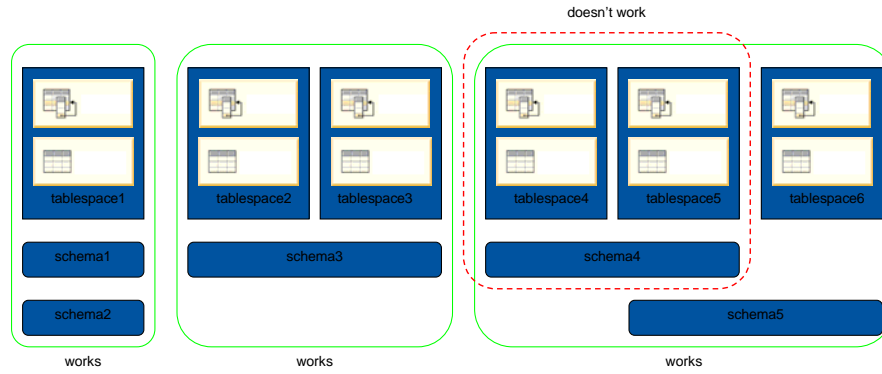
- Restore the syscatspace and specified table spaces from the backup image
- Roll them forward to a consistency point
- Validate the schemas specified
- Transfer ownership of the specified table spaces to the target DB
- Recreate the schema in the targetDB

Transportable Schema con't

Restore command changes

- Add "TRANSPORT INTO" option to identify this is a transport option
- Add "Stage into" option to identify which staging DB to use and to not delete it once the transport complete
- Reuse the "TABLESPACE" option to identify which table spaces to transport
- Add Schema list option to identify which schemas will be transported
 - With this release ALL schemas in the supplied table space list must be specified
 - This is added for future extensibility, we do require the call to specify the set of table spaces to transport. In the future we could drop the table space list parameter and determine it ourselves.

Transport Sets



Example

Transport Schema 1 and Schema 2 to the target DB successfully

- Restore the mydata1 and myindex1 table spaces into TARGETDB

```
db2 restore db originaldb tablespace ("mydata1","myindex1")
      schema("schema1","schema2") transport into targetdb redirect
db2 set tablespace containers for 2 using (path '/targetdb/mydata1')
db2 set tablespace containers for 3 using (path '/targetdb/myindex1')
db2 restore db originaldb continue
```



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

Agenda

- Review of current backup and recovery functionality
- **Are backups really needed anymore?**
- Protecting yourself from physical corruption
- Protecting yourself from logical corruption
- What do I do for peta-byte size DBs?

32

Objective 1: Review of current backup and recovery functionality

Objective 2: Are backups really needed anymore?

Objective 3: Protecting yourself from physical corruption

Objective 4: Protecting yourself from logical corruption

Objective 5: What do I do for the peta-byte size DBs?

Use the correct tool for the job

- **Disk Failure**
 - Best protected by a RAID file system, e.g. RAID 5
- **CPU Failure**
 - Best protected by a cluster manager e.g. HACMP, TSA
- **Hardware Failure**
 - Have redundant component, e.g. dual power supplies, dual NICs, dual paths to the storage devices, ...
- **Application / Migration errors**
 - Use DB2 backup/restore utilities



What purpose does a DB2 backup provide

- Protects you from physical corruption
- Allows you to restore only what is needed for recovery, e.g. recovery only a table space not entire DB
- Protects from Application errors.
- Can be used to copying a database.
 - Populating development or test systems.
 - Moving database to new hardware.
- Before and after software updates.
 - Database migration.

Setting your goals

- RPO
- RTO
- Two goals:
 - Ensure you can recover the data you need
 - Minimal impact to running queries - balance your backup process across partitions



September 2010

35

This is the time to be practical. You can achieve any RPO/RTO combination with enough money. This is the balance of money vs. what you really need. Figure this out early – so you have the right budget approval.

RPO = Recovery Point Objective

RPO refers to the maximum amount of data loss that is acceptable in a recovery scenario. This will affect how you configure and manage your transaction logs, and how you approach your backup strategy. This value may also be different for a disaster recovery scenario.

RTO = Recovery Time Objective

RTO refers to the time taken to complete the recovery process. An RTO should state how the objective is calculated, based on:

- When the problem is diagnosed and logged as an issue
- When the logged issue has been researched and accepted as a problem
- When the recovery procedure begins

Determining the RTO and RPO will depend on the demands of the data warehouse as determined in the planning stages. For example, in a single line-of-business data warehouse, a table space outage may render the data warehouse unusable. However, in an enterprise data warehouse supporting multiple line-of-business feeds, a single table space outage may only impact a small number of business users. Similarly, in some environments, maintaining up-time (RTO) may be more critical than suffering a certain amount of data loss (RPO).

Reconcile your recovery objectives with the design and configuration of your data warehouse to ensure they are achievable

Plan your strategy so that you can be comfortable with the stated objectives, ensuring that the hardware available can service the recovery times expected. Again, the earlier in the project cycle that the objectives are known the more likely you will be able to influence sizing, hardware and configuration to meet these objectives.

The 1st goal: Ensure you can recovery the data you need, as quickly as you need to. Granularity of backups allows you to recover only what you need, this increases performance. Using 'rebuild' option allows you to bring online the tbsps you need most, first.

The 2nd goal... balance across your system refers to one of the key philosophies of an IBM Smart Analytics System (ie. Balanced Warehouse). You want to think about spreading your workloads evenly across your partitioned environment, and this applies to backup and other utilities as much as it does to the query workload.

Another expression I like ... "Query is King" in a warehouse – so we want to design a backup strategy that minimizes impact on running queries, as well maintaining system 'up time'

Goal for backup speed -> To complete the backup cycle before the next cycle is scheduled.

Why - So money can be spent on hardware that focuses on query speed and not on hardware that is focused on backup speed for 5% of the week.

Agenda

- Review of current backup and recovery functionality
- Are backups really needed anymore?
- **Protecting yourself from physical corruption**
- Protecting yourself from logical corruption
- What do I do for peta-byte size DBs?

36

Objective 1: Review of current backup and recovery functionality

Objective 2: Are backups really needed anymore?

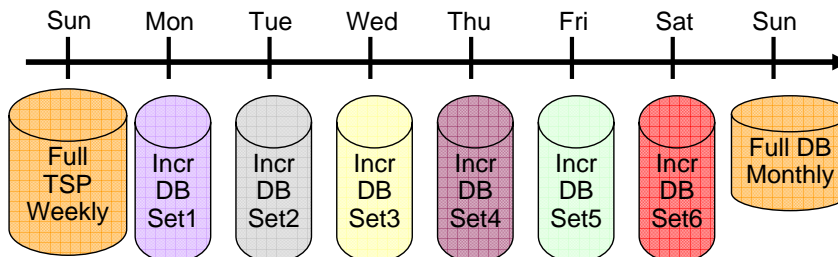
Objective 3: Protecting yourself from physical corruption

Objective 4: Protecting yourself from logical corruption

Objective 5: What do I do for the peta-byte size DBs?

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



37

Removing the need to perform full DB backups is providing the ability for DBAs to shorten their backup maintenance windows.

MUST have one full tablespace backup of EACH tablespace accessible (can be combined with other tablespaces), these are all recorded in the recovery history file.

Under the covers we use the recovery history file to locate the most recent full tablespace backup of each tablespace, restore each, then rollforward each tablespace from the time it was backed up.

In above diagram, no need for full DB monthly. However if taken they will shorten the recovery windows

Scenario for Backup:

- Log archiving enabled, to TSM:
db2 update db cfg for salesdb using LOGARCHMETH1 TSM
- Incremental backup
db2 db2 update dbm cfg using TRACKMOD YES
- Online daily backup of the catalog partition:
db2 -v "backup db salesdb on dbpartitionnums (0) online use tsm open 2 sessions util_impact_priority 33 include logs"
- Online partitioned tablespace backup – 2x per week
db2 "backup db salesdb on dbpartitionnums (1,5) tablespace(pdts_in, pdts_ini) online use tsm open 2 sessions util_impact_priority 33 include logs"
note: Rolling schedule – start with partitions (1,5), then (2,6), (3,7), (4,8)

September 2010

38

Sample backup scenarios

- daily backup of the catalog partition,
- full backup of all table spaces on Sunday and Wednesday,
- a full database backup every quarter.
- All backups were throttled at 33%. Examples are using TSM
- The TSM client was installed on each database partition. Archive logging was enabled and configured to use TSM using the following command:

```
db2 update db cfg for bcudb1 using LOGARCHMETH1 TSM
```

- Incremental backup was enabled by setting the TRACKMOD configuration parameter and performing a full database backup.

```
db2 db2 update dbm cfg using TRACKMOD YES
```

- To verify that the transaction logs are being archived to TSM:

```
bcuinst2@beluga002:~> db2pd -log -db bcudb1
```

```
Database Partition 0 -- Database BCUDB1 -- Active -- Up 0 days 22:26:00
```

Logs:

```
Current Log Number 48
Pages Written 599
Cur Commit Disk Log Reads 0
Cur Commit Total Log Reads 0
Method 1 Archive Status n/a
Method 1 Next Log to Archive 48
Method 1 First Failure n/a
Method 2 Archive Status n/a
Method 2 Next Log to Archive n/a
Method 2 First Failure n/a
Log Chain ID 1
Current LSN 0x00000000A77AFE45
```

```
Address StartLSN State Size Pages Filename
0x00002B98F8B98FB0 00000000A7558010 0x00000000 12800 12800 S0000048.LOG
0x00002B98F8B99810 00000000AA758010 0x00000000 12800 12800 S0000049.LOG
0x00002B98F8B9A070 00000000AD958010 0x00000000 12800 12800 S0000050.LOG
0x00002B98F8B9A8D0 00000000B0B58010 0x00000000 12800 12800 S0000051.LOG
0x00002B98F8B9B130 00000000B3D58010 0x00000000 12800 12800 S0000052.LOG
0x00002B98F8B9B990 00000000B6F58010 0x00000000 12800 12800 S0000053.LOG
0x00002B98F8B9C1F0 00000000BA158010 0x00000000 12800 12800 S0000054.LOG
0x00002B98F8B9CA50 00000000BD358010 0x00000000 12800 12800 S0000055.LOG
0x00002B98F8B9D2B0 00000000C0558010 0x00000000 12800 12800 S0000056.LOG
```

Online daily database backup of catalog partition

The catalog database partition contains only one database partition. An example of the command to back up to disk is shown below:

Monday ~ Sunday

```
db2 -v "backup db bcudb1 on dbpartitionnums (0) online use tsm open 2 sessions util_impact_priority 33 include logs"
```

Part Result

Recovery Scenario 1: Partition level recovery to end of logs

- 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 20100401143616 replace existing`
 - Check status:
`db2 rollforward db salesdb query status`
Result:
3 TBS pending S0000000.LOG-S0000003.LOG 2010-03-29-07.42.44.000000 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\"`

September 2010

39

Recover table space on single database partition

The first step is to issue the restore command and specify the table space to be restored and the database partition that is the target of the restore. The command illustrated below targets database partition 3 and the table space DMS01 and associated index table space DMS02.

```
bcuinst2@beluga002:~/scr> db2_all "<<+3<db2 \"restore db salesdb tablespace (dms01, dms02) online use tsm taken at 20100401143616 replace existing \" "
```

DB20000I The RESTORE DATABASE command completed successfully.

beluga003: db2 "restore db bcudb1 ... completed ok

When the restore has completed, a QUERY STATUS prior to the rollforward shows that database partition 3 is in rollforward pending state. The table space in this partition must be consistent with the other partitions and therefore a rollforward to end of logs is expected.

```
bcuinst2@beluga002:~/scr> db2 rollforward db salesdb query status
```

Rollforward Status

Input database alias = salesdb

Number of nodes have returned status = 9

Node	Rollforward status	Next log to be read	Log files processed	Last committed transaction
0	not pending	S0000000.LOG-S0000002.LOG	2010-04-01-09.35.31.000000	Local
1	not pending	S0000000.LOG-S0000003.LOG	2010-03-29-07.42.37.000000	Local
2	not pending	S0000000.LOG-S0000003.LOG	2010-03-29-07.42.36.000000	Local
3	TBS pending	S0000000.LOG-S0000003.LOG	2010-03-29-07.42.44.000000	Local
4	not pending	S0000000.LOG-S0000003.LOG	2010-03-29-07.42.36.000000	Local
5	not pending	S0000000.LOG-S0000003.LOG	2010-03-29-07.42.36.000000	Local
6	not pending	S0000000.LOG-S0000003.LOG	2010-03-29-07.42.36.000000	Local
7	not pending	S0000000.LOG-S0000003.LOG	2010-03-29-07.42.36.000000	Local
8	not pending	S0000000.LOG-S0000003.LOG	2010-03-29-07.42.36.000000	Local

```
-----
0 not pending S0000000.LOG-S0000002.LOG 2010-04-01-09.35.31.000000 Local
1 not pending S0000000.LOG-S0000003.LOG 2010-03-29-07.42.37.000000 Local
2 not pending S0000000.LOG-S0000003.LOG 2010-03-29-07.42.36.000000 Local
3 TBS pending S0000000.LOG-S0000003.LOG 2010-03-29-07.42.44.000000 Local
4 not pending S0000000.LOG-S0000003.LOG 2010-03-29-07.42.36.000000 Local
5 not pending S0000000.LOG-S0000003.LOG 2010-03-29-07.42.36.000000 Local
6 not pending S0000000.LOG-S0000003.LOG 2010-03-29-07.42.36.000000 Local
7 not pending S0000000.LOG-S0000003.LOG 2010-03-29-07.42.36.000000 Local
8 not pending S0000000.LOG-S0000003.LOG 2010-03-29-07.42.36.000000 Local
```

The ROLLFORWARD command is then issued for the table spaces concerned and on the database partition being recovered:

```
db2 \"rollforward db salesdb to end of logs on dbpartitionnum (3) tablespace (dms01, dms02) online\"
```

Rollforward Status

Input database alias = bcudb1

Number of nodes have returned status = 1

Node number = 3

Rollforward status = not pending

Next log file to be read =

Log files processed = -

Last committed transaction = 2010-03-04-13.53.03.000000 UTC

DB20000I The ROLLFORWARD command completed successfully.

Recovery Scenario 2: Database recovery to end of logs

- Use restore with the rebuild option:
db2 "restore db salesdb with all tablespaces in database use tsm taken at 20100401143616"
 - 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"

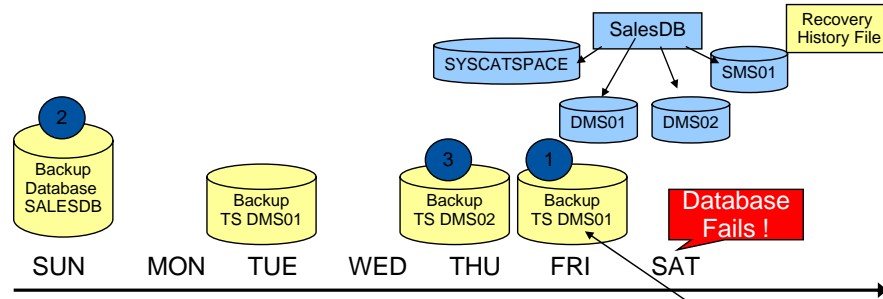
Note:

You can use the rebuild option to rebuild the table spaces you need most, first. Then bring those table spaces online, while the other table spaces are still being restored. This is a key advantage and allows you to get your applications up and running faster. (note: you need to think of this in your database design!!)

Also, remember that you will need to restore the syscat table space first and always. It holds the definition of all your database objects.

Database Recovery using Rebuild

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



- RESTORE DB SALESDB
REBUILD WITH ALL TABLESPACES IN DATABASE TAKEN AT *Friday*
(DB2 Restores Friday, Sunday and Thursday)
- ROLLFORWARD DB SALESDB TO END OF LOGS AND STOP
(Only One ROLLFORWARD needed (DB2 9))

Recovery Scenario 3: Database Recovery to Point in Time

- Use restore with the rebuild option:
`db2 "restore db salesdb with all tablespaces in database use tsm taken at 20100401143616"`
 - 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

September 2010

42

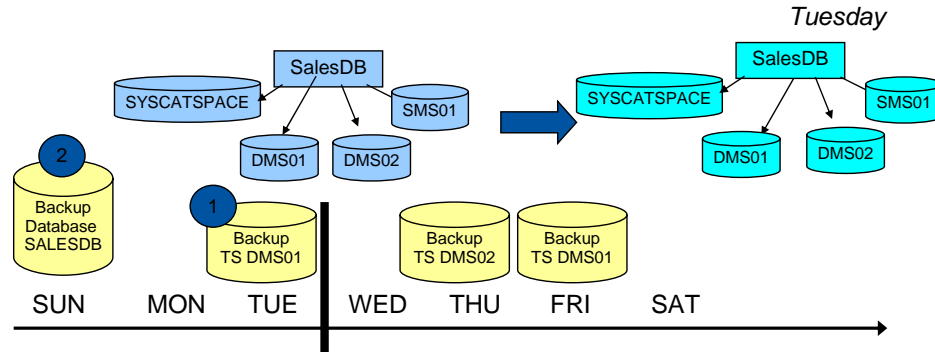
Point in Time (PIT) recovery refers to the earliest moment to which a database or table space can be recovered in a consistent state. PIT recovery of a single table space in an individual database partition is *not* possible in a DPF environment.. To be recovered to the same point in time, a single table space recovery requires rolling forward to end of logs for all database partitions. Should you need to perform a PIT recovery, this should be executed as a database recovery.

Must recover all database partitions to the same point in time – otherwise db will be logically corrupt

Avoid falling into the trap of using backup and recovery to address problems in the ETL application where errors in loading data are allowed and the backup/recovery process is used to recover from such errors - this is only useful while the database is small.

Tablespace point in time recovery using REBUILD

- Can restore database to PIT



1. RESTORE DB SALESDB
REBUILD WITH ALL TABLESPACES IN DATABASE TAKEN AT *Tuesday*
(DB2 Restores Tuesday, and Sunday)
2. ROLLFORWARD DB SALESDB TO *Tuesday* AND STOP

September 2010

43

Use restore with the rebuild option:

db2 "restore db salesdb with all tablespaces in database use tsm taken at 20100401143616"

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.

Agenda

- Review of current backup and recovery functionality
- Are backups really needed anymore?
- Protecting yourself from physical corruption
- **Protecting yourself from logical corruption**
- What do I do for peta-byte size DBs?

Objective 1: Review of current backup and recovery functionality

Objective 2: Are backups really needed anymore?

Objective 3: Protecting yourself from physical corruption

Objective 4: Protecting yourself from logical corruption

Objective 5: What do I do for the peta-byte size DBs?



Protecting yourself from logical corruption

- Examples of logical corruption
 - Application erroneous updates, inserts, deletions
 - Dropped table(s)
 - Deleted row(s)
 - Dropped table space(s)



Methods to recover from logical corruption

- Utilize DB2 Backup images

- Recovery affected table space(s) and roll forward to a point in time (PIT)
- Must rollforward past last DDL statement or must also recovery the system catalog table space
- **Pros**
 - Utilizes existing backups, can also use snapshot backups
- **Cons**
 - Level of granularity is DB
 - Slow recovery if using traditional DB2 backup image
 - Complete DB outage while recovery is in progress
 - All transactions after the PIT must be manually reapplied



Methods to recover from logical corruption

Object level recovery via Staging DB

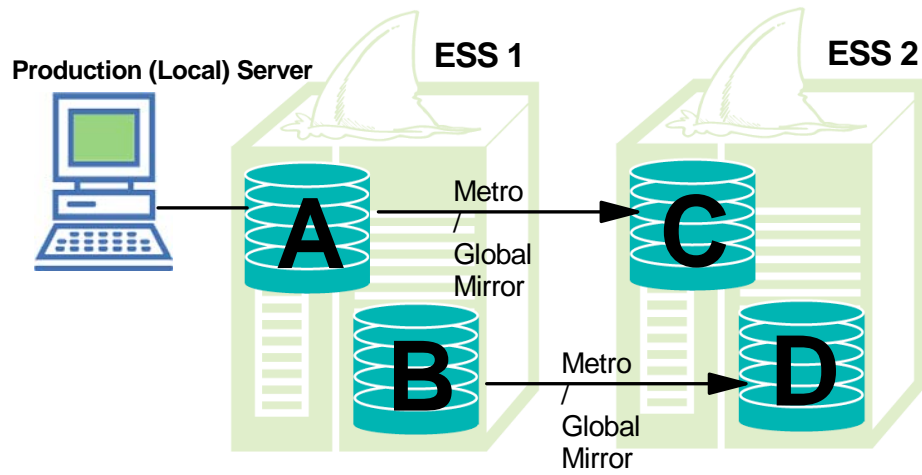
- Create a staging DB utilizing the “rebuild” option to restore all affected table spaces into a new DB
- Rollforward to a PIT before the erroneous operation
- Can use load from cursor between source and staging DB to recovery the affected objects
- **Pros**
 - Does not affect availability of source DB
- **Cons**
 - Requires space for the staging DB



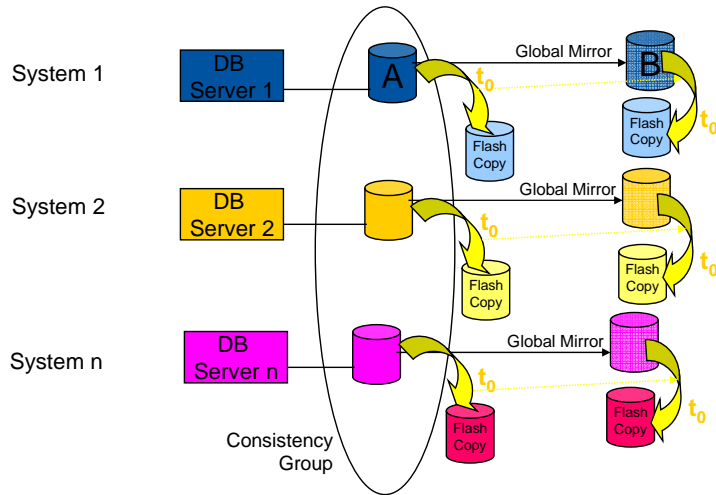
Methods to recover from logical corruption - Object level recovery via High Performance Unload

- **Pros**
 - Utilizes existing DB2 backups to extract objects
 - Only targeted objects are affected
- **Cons**
 - Unable to reapply log files generated after the backup
 - Must reapply transactions manually

DR using Storage Replication



Recommended Dual Data Center Solution
Global Mirror and Flash Copy – fully coordinated across DBs



FlashCopy at t_0 coordinated across all systems both on local and remote side



Storage Replication Best Practices

Ensure that

- **Consistency group contains ALL table space containers AND log files for ALL partitions.**
- JFS-2 block size (agblksize) is set to 4kBytes
- Direct I/O (DIO) or concurrent I/O (CIO) is enabled
- The logical volumes are not fragmented
- If LVM Striping is used, the stripe size is the same or a multiple of the page size
- Only table spaces of same page size reside on the same FS
 - mixing table spaces with larger page sizes may throw off alignment
 - no other files than the table space containers can be created on this FS at any time (i.e flat files, etc.) as this may throw off alignment
- Only database managed table spaces (DMS) or Automatic Storage (AS) are used

Agenda

- Review of current backup and recovery functionality
- Are backups really needed anymore?
- Protecting yourself from physical corruption
- Protecting yourself from logical corruption
- What do I do for peta-byte size DBs?

52

Objective 1: Review of current backup and recovery functionality

Objective 2: Are backups really needed anymore?

Objective 3: Protecting yourself from physical corruption

Objective 4: Protecting yourself from logical corruption

Objective 5: What do I do for the peta-byte size DBs?

Best Practices: Backup recommendations

- Backup Images:
 - Online database backups
 - Online table space backups
 - Incremental backups
 - Compression is good *
- Use the Single System View option (v9.5 +). 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

September 2010

53

To Summarize:

We are going to talking about a strategy that uses online table space backups. You will add in compression if your database is not already compressed, and you want to include logs in your backup image, as that will make recovery much easier.

Note: you need to think about processing the smallest amount of data possible. That will make your backups faster & your recovery faster – usually the most expensive operation is I/O. In a data warehouse that has 3 years of data, how much of that is historical data that is backed up over and over again and how much is active data that is changing and needs to be backed up

You can rebuild your database from:

- full tbsp backups (you can rebuild the db from these)
- full online tbsp backups
- Incremental backups
- Note:** always use the include logs feature

WRT to Load – you want to design your load processing so that in the event of a load failure or mistaken load, all you need to do is drop the staging table. If you keep your load files around, then you will need to backup the staging table tablespace. Once the staging table is attached to the main table (check explanation here) then it will be picked up in the next online tbsp backup.

In Version 9.5, when you perform a backup operation from the catalog node of a partitioned database, you can specify which partitions to include in the backup, or specify that all the database partitions should be included. The specified partitions will be backed up simultaneously, and the backup timestamp associated with all specified database partitions will be the same. Also, you can include database logs with a SSV backup; including logs in backup images is the default behavior for backup operations. Finally, when you restore from a SSV backup image, you can specify to roll forward to *end of logs*, which is the minimum recovery time calculated by the database manager.

Remember the two goals:

- Ensure you can recover the data you need
- Minimal impact to running queries - balance your backup process across partitions

Best Practices: Backup recommendations cont'd

- Example:
 - Full online table space backup of hot* and warm table spaces twice a week, include logs
 - Full online database backup of catalog partition daily, logs included
 - Full online database backup of database partitions on quarterly or monthly basis
 - Full table space backup after adding a new table space

September 2010

54

When you perform a backup operation from the catalog node of a partitioned database, you can specify which partitions to include in the backup, or specify that all the database partitions should be included. The specified partitions will be backed up simultaneously, and the backup timestamp associated with all specified database partitions will be the same. Also, you can include database logs with a SSV backup; including logs in backup images is the default behavior for backup operations. Finally, when you restore from a SSV backup image, you can specify to roll forward to *end of logs*, which is the minimum recovery time calculated by the database manager.

Remember the two goals:

Ensure you can recover the data you need

Minimal impact to running queries - balance your backup process across partitions

Table space backups

Table space backups are the recommended approach for a recovery strategy, executed as follows:

• Full online table space backup of hot and warm table spaces twice a week, logs included

Full online database backup of the catalog partition on a daily basis, logs included

Note: This is important so that when restoring the catalog partition and the tablespace being restored have the exact recollection (structure) of what the object being restored looked like. So - if you are restoring a TS that has a table where a column was altered... and you restore the catalog partition to a point before the column change, you have a mismatch and a problem.

• Operating system files backed up twice a week in tandem with table space backups

• Full online database backup of database partitions on a quarterly basis

• Full table space backup after adding a new table space

A table space backup strategy will execute a full backup of each table space in parallel across database partitions. When complete, the process will move to the second table space in the list and so on until all table spaces are backed up. Logs will be included in each backup image. Each table space will be backed up twice weekly. In a recovery scenario, no more than two to three days of transaction logs will be required. Staging tables should not need to be backed up and the temporary table space does not need to be backed up.

Incremental table space backups may be implemented for hot table spaces where a large volume of transaction logs is being generated on a daily basis. Incremental backups are enabled by setting the TRACKMOD configuration parameter to YES. Note that a full offline database backup is first required to activate this parameter.



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.

September 2010

55

Take advantage of table spaces, and use these to group your data into logical groupings. Having a good table space design will help you to only have to restore what you need. If you have multiple applications using your data warehouse, make sure you have each application's tables divided into separate table spaces. You can rebuild part of your database to get online quickly, while other table spaces remain offline.

Note: you need to think about processing the smallest amount of data possible. That will make your backups faster & your recovery faster – usually the most expensive operation is I/O.

WRT to Load – you want to design your load processing so that in the event of a load failure or mistaken load, all you need to do is drop the staging table. If you keep your load files around, then you do not need to backup the staging table table space, you can simply rebuild it in the event of a failure. Once the staging table is attached to the main table then it will be picked up in the next online tbspace backup.

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.

September 2010

56

Archive logging

Transaction logs record all changes made to the database and are then used to roll back or replay a transaction.

In a data warehouse environment it is recommended that archive logging be implemented to facilitate online backups in place of circular logging which does not support online backups. How you configure the transaction logs has an impact on performance in relation to workload, backup and recovery.

- Small transaction logs can reduce risk since logs contain less transactions thereby reducing risk of data loss in the event of failure.
- Small logs may increase the number of active logs because large transactions may span multiple log files.
- Recovery speed is directly affected by the number of archived logs that need to be retrieved
- Large transaction logs increase the number of transactions that may be lost in the event of failure
- Large transaction logs take up less system resources since less log switching takes place

It is recommended that each database partition have 50 primary logs of 50MB each (12,800 4K pages). This is a total of approximately 2.5GB of logs allocated per database partition. This may be modified depending on each environment but should always be with respect to volumes and objectives.

Run frequent table space backups to minimize the number of transaction logs required in a recovery scenario

Archiving transaction logs on a regular schedule can limit data loss. For example, if the maximum duration of lost transactions is sixty minutes, manually archive the logs every thirty minutes so as to minimize the risk of losing a log file in the event of failure. This requirement should also inform the data load application design process in terms of commit size.

Backup and Recovery

- If a DB2 online backup is not feasible due to conflicts with other utilities then consider SNAPSHOTS in DB2 9.5

OR

- **IBM Tivoli Storage Manager for Advanced Copy Services For DB2**
 - http://www.ibm.com/developerworks/tivoli/library/t-acbdb2_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>



Dale McInnis

IBM Canada Ltd.
dmcinnis@ca.ibm.com

Session: D04
Are backups really needed anymore?



Disclaimer

© Copyright IBM Corporation [current year]. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

IBM, the IBM logo, ibm.com, and DB2 are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

Other company, product, or service names may be trademarks or service marks of others.