IDUG

2025

Atlanta, GA | June 8-12

NA Db2 TECH CONFERENCE

Unlocking Recovery - The Key to Solving Everyday Problems

Michael Roecken, *IBM*

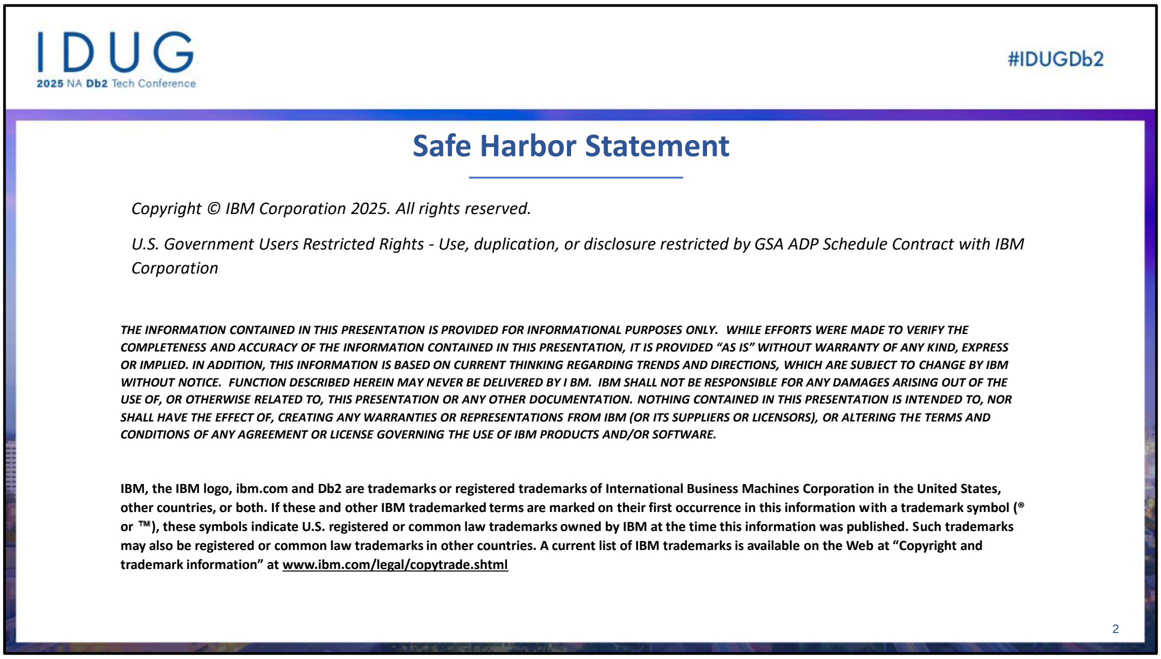
Session Code: A15

Platform:

LUW

Michael Roecken is a senior software developer with Db2 for Linux, UNIX, and Windows platforms at the IBM Toronto Lab. Michael has worked since 2000 designing, implementing and supporting various features and capabilities in the areas of: backup/restore, crash/rollforward recovery, high availability/disaster recovery, logging/transaction management and upgrade/migration.

Please connect with Michael on X at @roecken and LinkedIn.



Safe Harbor Statement

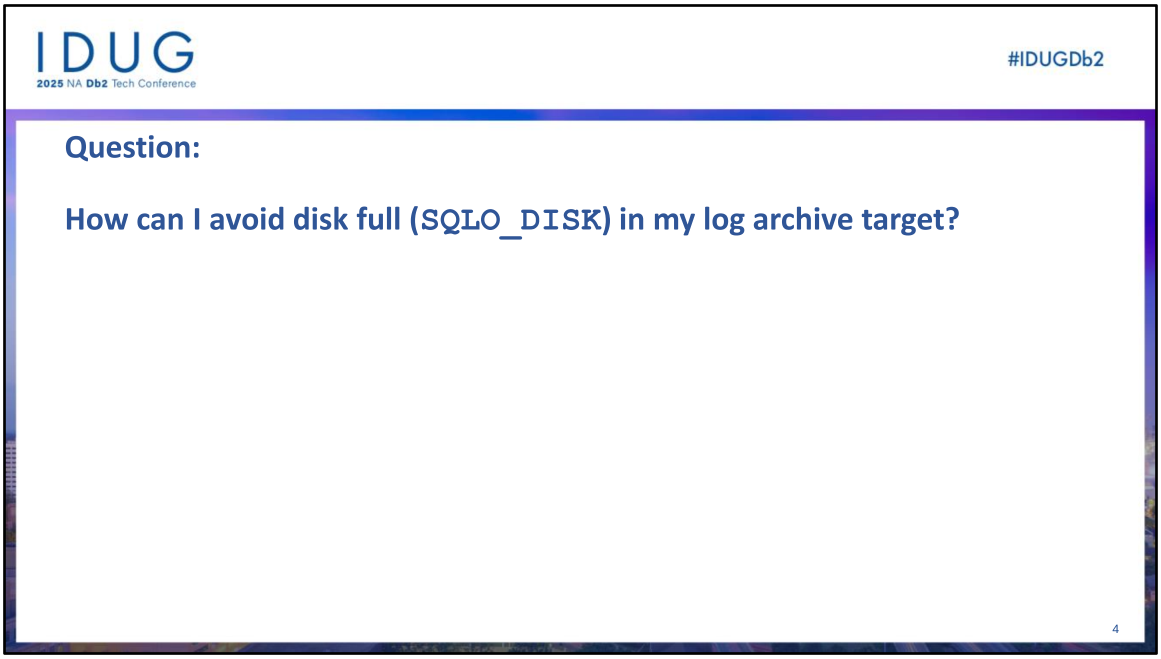
U.S. Government Users Restricted Rights - Use, duplication, or disclosure restricted by GSA ADP Schedule Contract with IBM Corporation

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 CURRENT THINKING REGARDING TRENDS AND DIRECTIONS, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. FUNCTION DESCRIBED HEREIN MAY NEVER BE DELIVERED BY I BM. 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

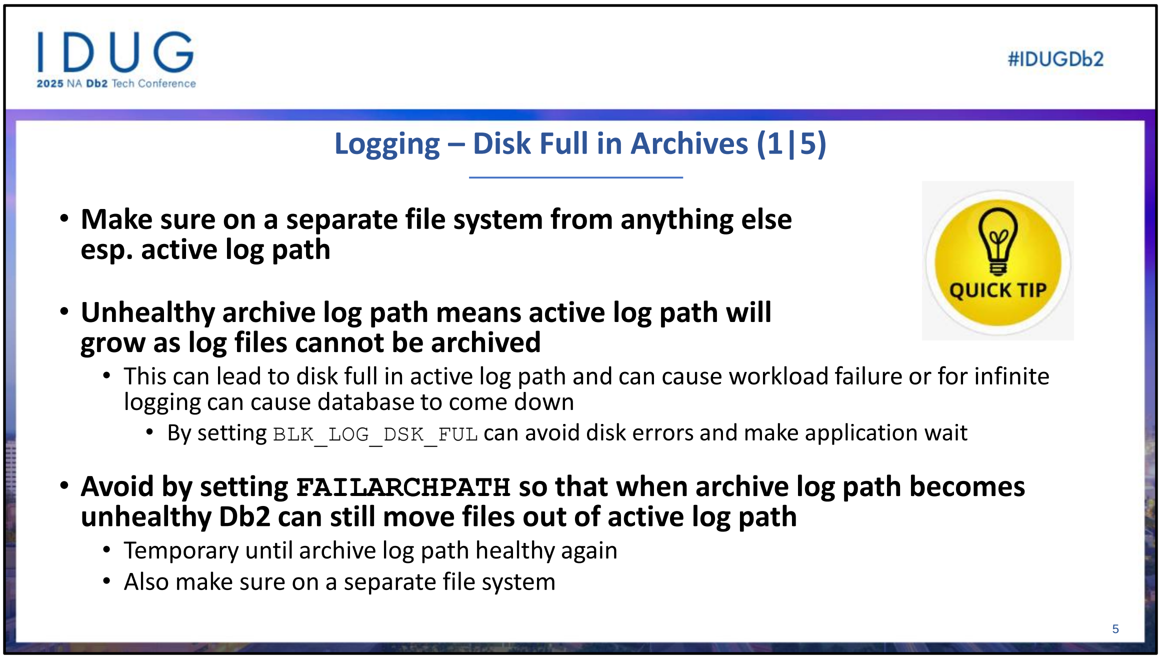
Agenda

- **Logging**
  - **Disk Full in Archives**
    - **Automatic Pruning**
- Backup & Restore
  - Performance, ITP & BAR Stats
  - History File Improvements
- Recovery
  - Host1 to Host2



Question:

How can I avoid disk full (`SQLO_DISK`) in my log archive target?



Logging – Disk Full in Archives (1|5)

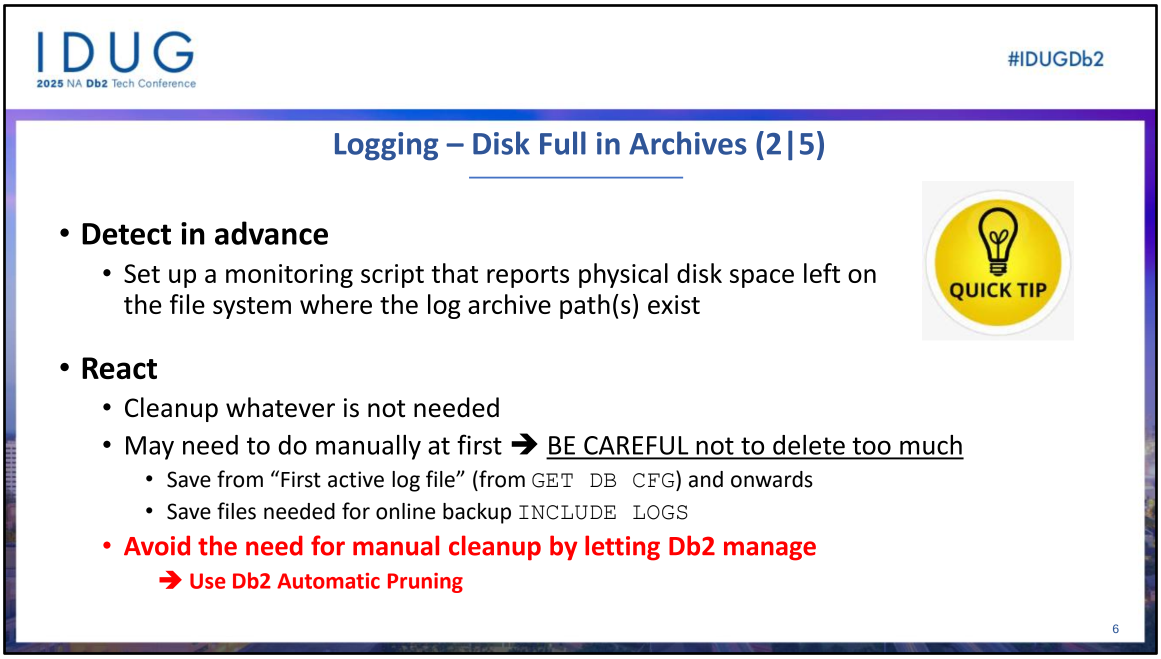
- **Make sure on a separate file system from anything else esp. active log path**
- **Unhealthy archive log path means active log path will grow as log files cannot be archived**
  - This can lead to disk full in active log path and can cause workload failure or for infinite logging can cause database to come down
    - By setting `BLK_LOG_DSK_FUL` can avoid disk errors and make application wait
- **Avoid by setting `FAILARCHPATH` so that when archive log path becomes unhealthy Db2 can still move files out of active log path**
  - Temporary until archive log path healthy again
  - Also make sure on a separate file system

db2diag.log would show something like this:

```
EDUID   : 397                  EDUNAME: db2logmgr (SAMPLE)
FUNCTION: DB2 UDB, data protection services, sqlpgArchiveLogDisk,
probe:2530
MESSAGE : ZRC=0x850F000C=-2062614516=SQLO_DISK "Disk full."
          DIA8312C Disk was full.
DATA #1 : String, 66 bytes
/db2archives/db2inst/NODE0000/SQL00001/LOGSTREAM0000/S0000031.LOG

EDUID   : 397                  EDUNAME: db2logmgr (SAMPLE)
FUNCTION: DB2 UDB, data protection services, sqlpgArchiveLogFile,
probe:3150
MESSAGE : ADM1848W  Failed to archive log file "S0000031.LOG" to
          "/db2archives/db2inst/SAMPLE/NODE0000/LOGSTREAM0000/C00
          00000/" from
          "/home/db2inst/db2inst/NODE0000/SQL00001/LOGSTREAM0000/".
```

Configuration parameters for database logging

- https://www.ibm.com/docs/en/db2/12.1?topic=logging-configuration-parameters



IDUG
2025 NA Db2 Tech Conference

#IDUGDb2

Logging – Disk Full in Archives (2|5)

QUICK TIP

- **Detect in advance**
  - Set up a monitoring script that reports physical disk space left on the file system where the log archive path(s) exist
- **React**
  - Cleanup whatever is not needed
  - May need to do manually at first ➔ BE CAREFUL not to delete too much
    - Save from “First active log file” (from `GET DB CFG`) and onwards
    - Save files needed for online backup `INCLUDE LOGS`
  - **Avoid the need for manual cleanup by letting Db2 manage**

    ➔ Use Db2 Automatic Pruning

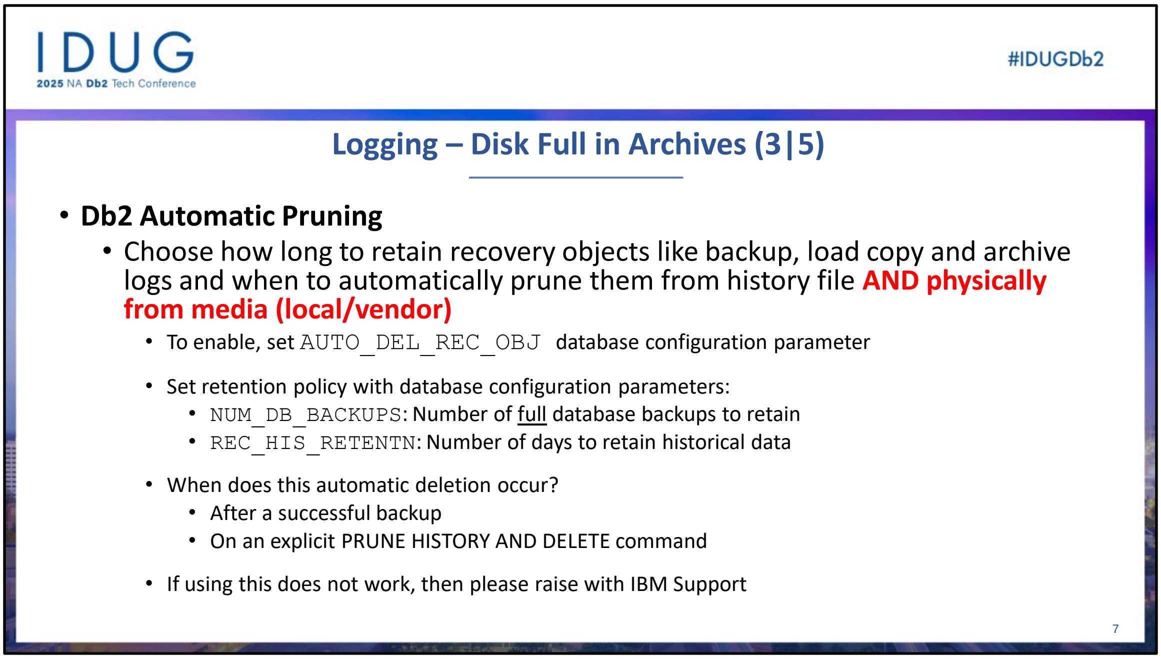
6

Configuration parameters for database logging

- https://www.ibm.com/docs/en/db2/12.1?topic=logging-configuration-parameters

Automating recovery history file pruning

- https://www.ibm.com/docs/en/db2/12.1?topic=file-automating-recovery-history-pruning



IDUG
2025 NA Db2 Tech Conference

#IDUGDb2

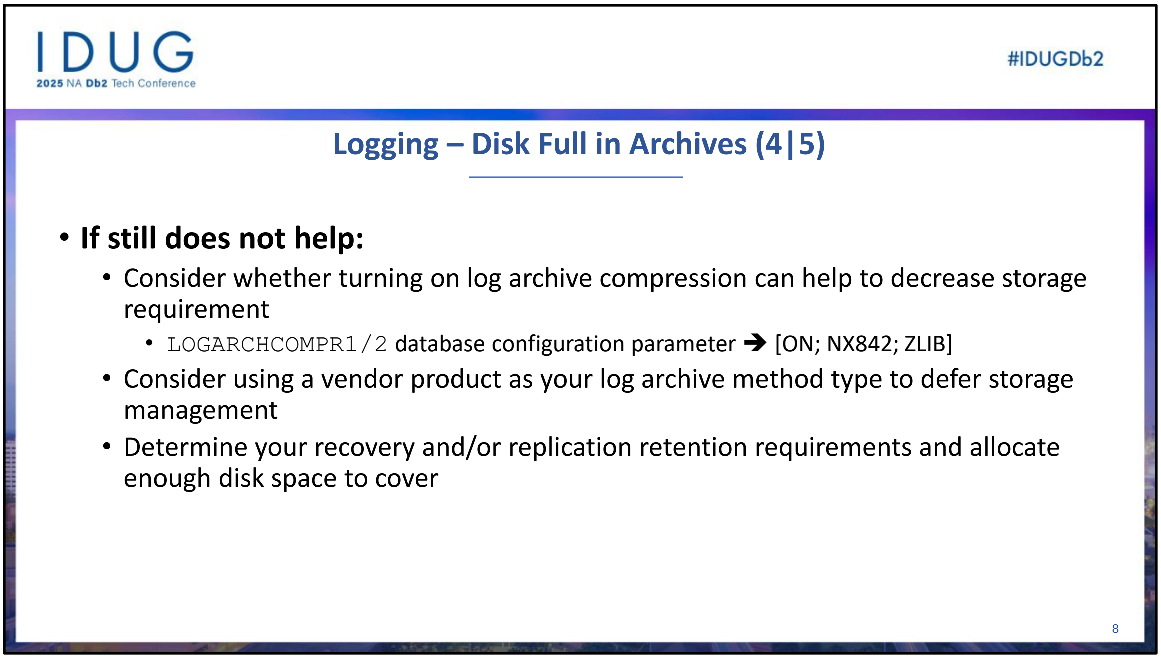
Logging – Disk Full in Archives (3|5)

- **Db2 Automatic Pruning**
  - Choose how long to retain recovery objects like backup, load copy and archive logs and when to automatically prune them from history file **AND physically from media (local/vendor)**
    - To enable, set `AUTO_DEL_REC_OBJ` database configuration parameter
    - Set retention policy with database configuration parameters:
      - `NUM_DB_BACKUPS`: Number of full database backups to retain
      - `REC_HIS_RETENTN`: Number of days to retain historical data
    - When does this automatic deletion occur?
      - After a successful backup
      - On an explicit PRUNE HISTORY AND DELETE command
    - If using this does not work, then please raise with IBM Support

7

Automating recovery history file pruning

- https://www.ibm.com/docs/en/db2/12.1?topic=file-automating-recovery-history-pruning

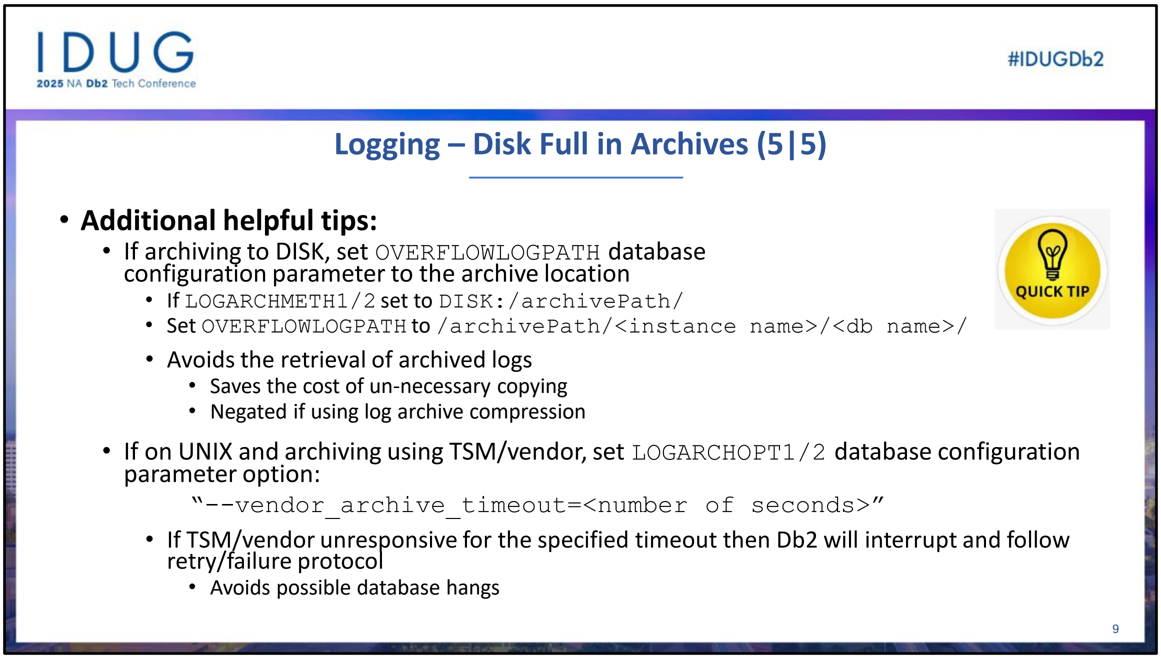


Logging – Disk Full in Archives (4|5)

- **If still does not help:**
  - Consider whether turning on log archive compression can help to decrease storage requirement
    - `LOGARCHCOMPR1/2` database configuration parameter ➔ [ON; NX842; ZLIB]
  - Consider using a vendor product as your log archive method type to defer storage management
  - Determine your recovery and/or replication retention requirements and allocate enough disk space to cover

Configuration parameters for database logging

- https://www.ibm.com/docs/en/db2/12.1?topic=logging-configuration-parameters



IDUG
2025 NA Db2 Tech Conference

#IDUGDb2

Logging – Disk Full in Archives (5|5)

- **Additional helpful tips:**
  - If archiving to DISK, set `OVERFLOWLOGPATH` database configuration parameter to the archive location
    - If `LOGARCHMETH1/2` set to `DISK:/archivePath/`
    - Set `OVERFLOWLOGPATH` to `/archivePath/<instance name>/<db name>/`
    - Avoids the retrieval of archived logs
      - Saves the cost of un-necessary copying
      - Negated if using log archive compression
  - If on UNIX and archiving using TSM/vendor, set `LOGARCHOPT1/2` database configuration parameter option:

    ```
    "--vendor_archive_timeout=<number of seconds>"
    ```

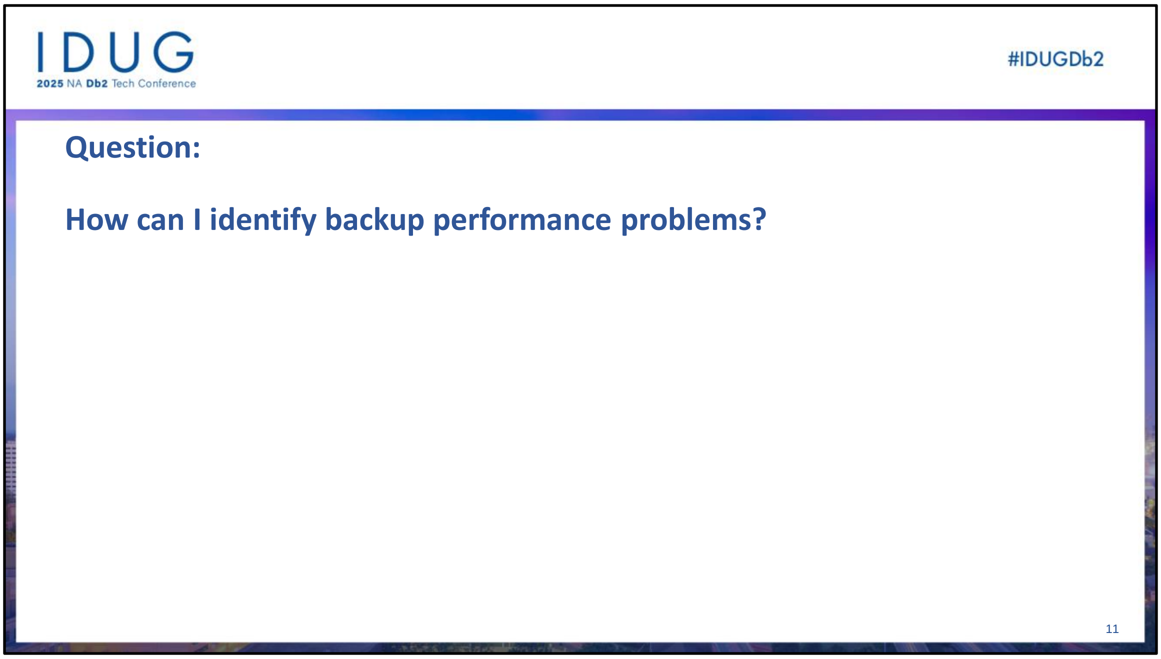
    - If TSM/vendor unresponsive for the specified timeout then Db2 will interrupt and follow retry/failure protocol
      - Avoids possible database hangs

Using an overflow log path

- https://www.ibm.com/docs/en/db2/12.1?topic=parameters-using-overflow-log-path

Agenda

- Logging
  - Disk Full in Archives
    - Automatic Pruning
- **Backup & Restore**
  - **Performance, ITP & BAR Stats**
  - **History File Improvements**
- Recovery
  - Host1 to Host2



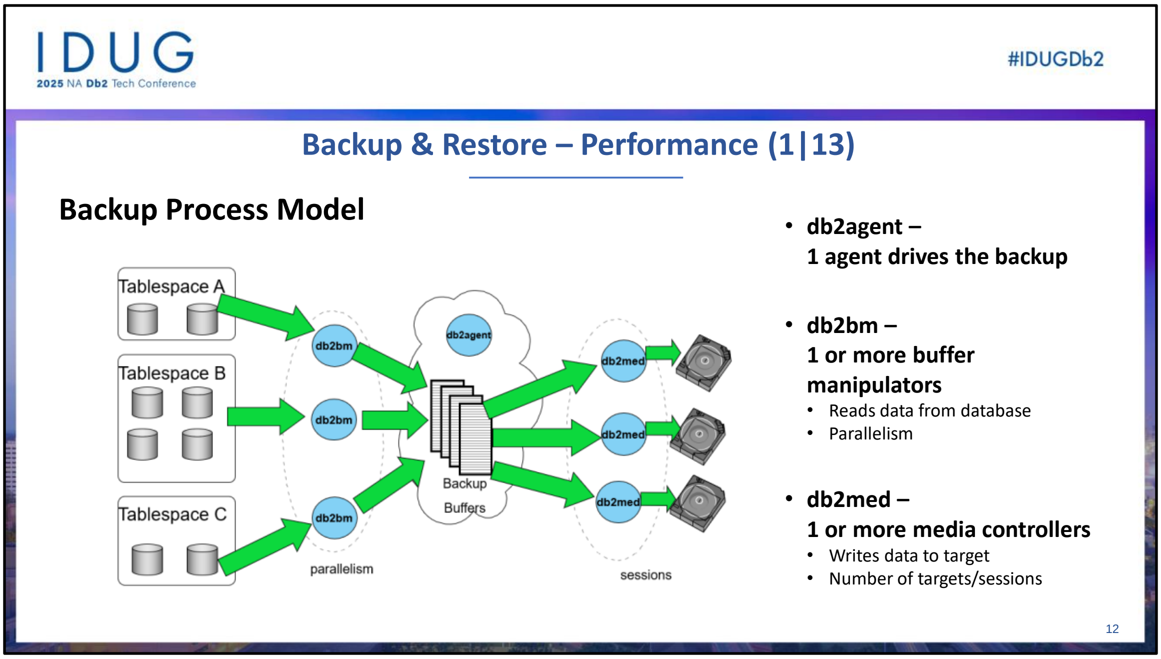
IDUG
2025 NA Db2 Tech Conference

#IDUGDb2

Question:

How can I identify backup performance problems?

11



IDUG
2025 NA Db2 Tech Conference

#IDUGDb2

Backup & Restore – Performance (1|13)

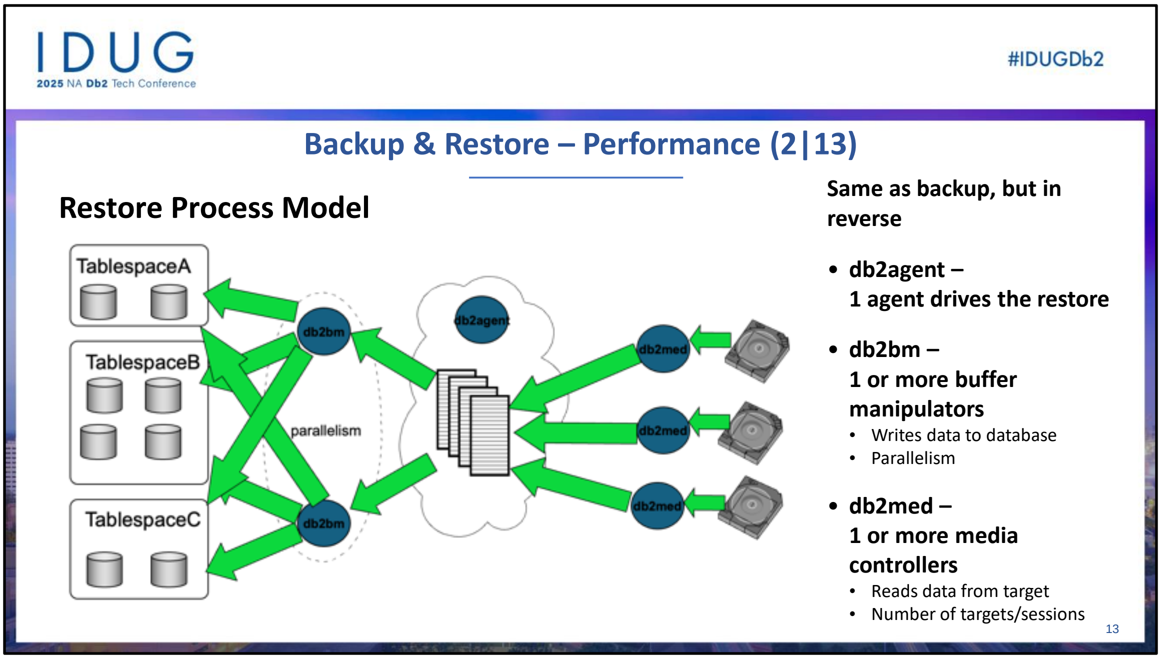
Backup Process Model

- **db2agent –**
  1 agent drives the backup
- **db2bm –**
  1 or more buffer manipulators
  - Reads data from database
  - Parallelism
- **db2med –**
  1 or more media controllers
  - Writes data to target
  - Number of targets/sessions

12

BACKUP DATABASE comma

- https://www.ibm.com/docs/en/db2/12.1?topic=commands-backup-database



IDUG
2025 NA Db2 Tech Conference

#IDUGDb2

Backup & Restore – Performance (2|13)

Restore Process Model

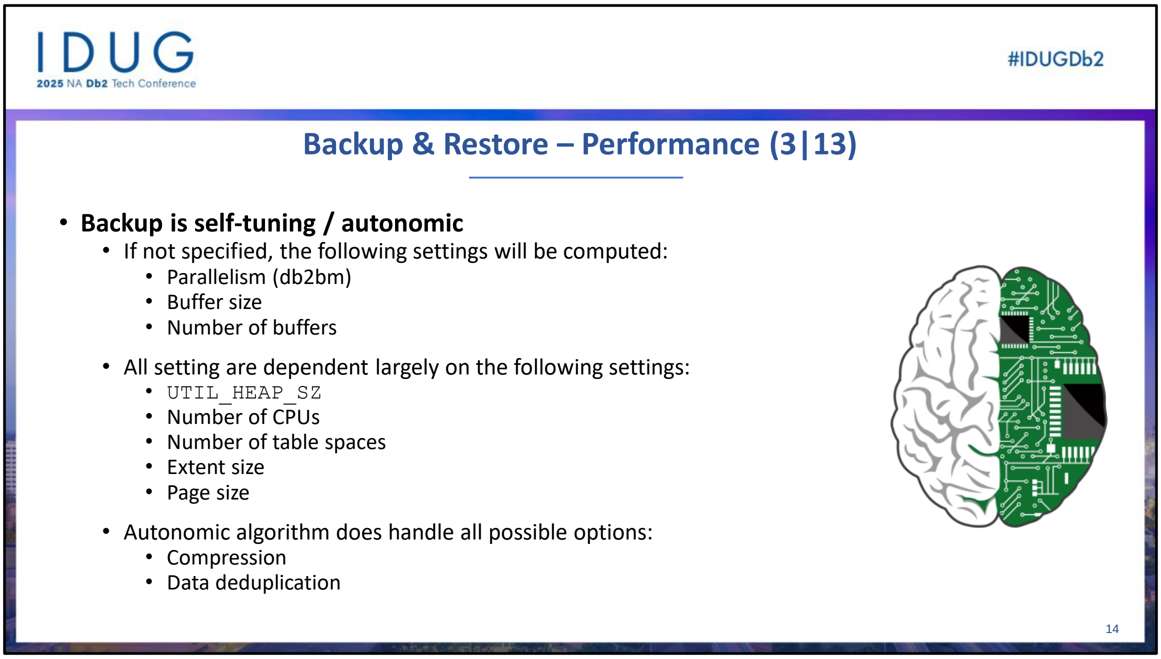
Same as backup, but in reverse

- **db2agent –**
  1 agent drives the restore
- **db2bm –**
  1 or more buffer manipulators
  - Writes data to database
  - Parallelism
- **db2med –**
  1 or more media controllers
  - Reads data from target
  - Number of targets/sessions

13

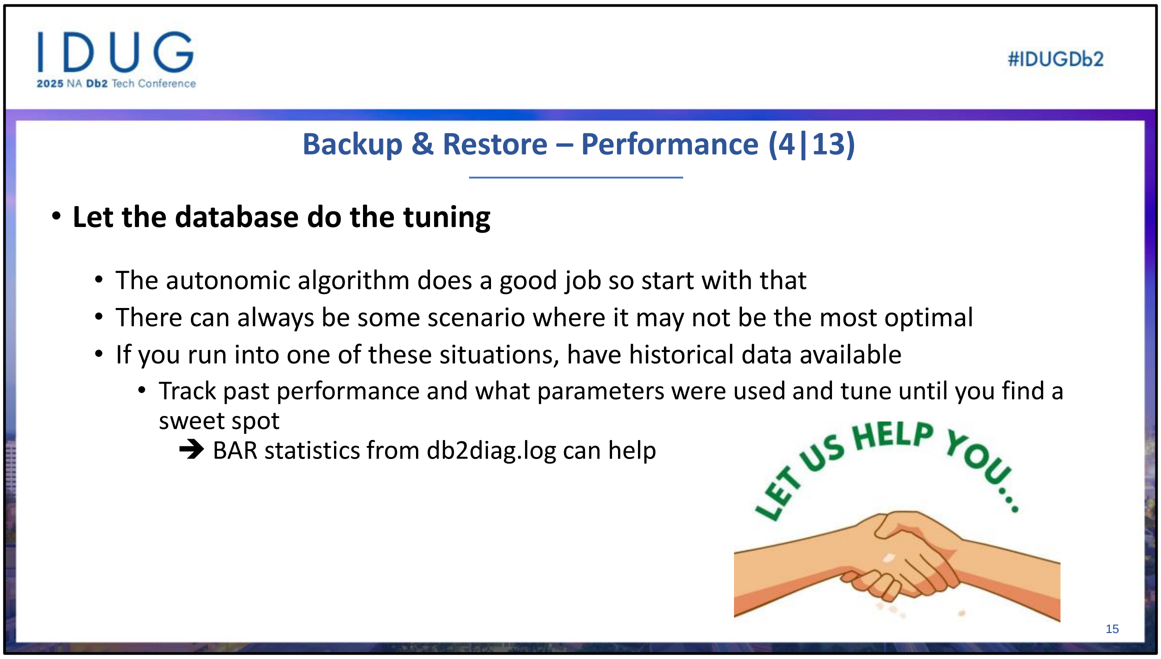
BACKUP DATABASE comma

- https://www.ibm.com/docs/en/db2/12.1?topic=commands-backup-database



Backup & Restore – Performance (3|13)

- **Backup is self-tuning / autonomic**
  - If not specified, the following settings will be computed:
    - Parallelism (db2bm)
    - Buffer size
    - Number of buffers
  - All setting are dependent largely on the following settings:
    - `UTIL_HEAP_SZ`
    - Number of CPUs
    - Number of table spaces
    - Extent size
    - Page size
  - Autonomic algorithm does handle all possible options:
    - Compression
    - Data deduplication



IDUG 2025 NA Db2 Tech Conference

#IDUGDb2

Backup & Restore – Performance (4|13)

- **Let the database do the tuning**
  - The autonomic algorithm does a good job so start with that
  - There can always be some scenario where it may not be the most optimal
  - If you run into one of these situations, have historical data available
    - Track past performance and what parameters were used and tune until you find a sweet spot

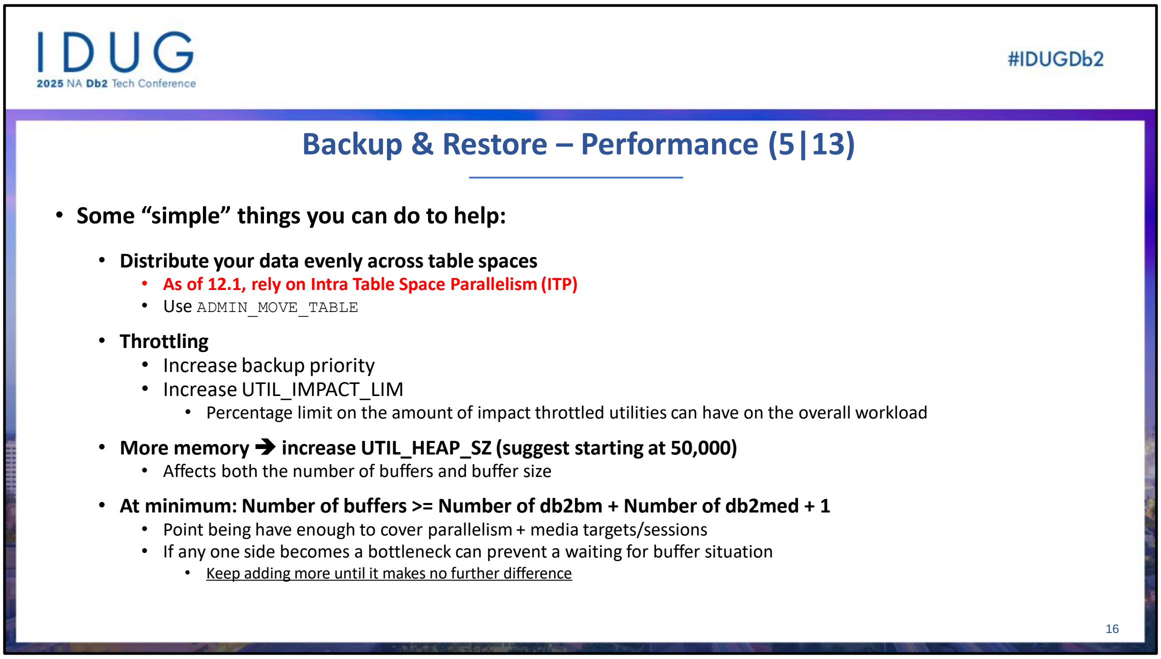
      ➔ BAR statistics from db2diag.log can help

15

** In the future this may be less of a need due to Intra Tablespace Parallelism feature

Optimizing backup performance

- https://www.ibm.com/docs/en/db2/12.1?topic=backup-optimizing-performance



IDUG
2025 NA Db2 Tech Conference

#IDUGDb2

Backup & Restore – Performance (5|13)

- **Some “simple” things you can do to help:**
  - **Distribute your data evenly across table spaces**
    - **As of 12.1, rely on Intra Table Space Parallelism (ITP)**
    - Use `ADMIN_MOVE_TABLE`
  - **Throttling**
    - Increase backup priority
    - Increase UTIL_IMPACT_LIM
      - Percentage limit on the amount of impact throttled utilities can have on the overall workload
  - **More memory ➔ increase UTIL_HEAP_SZ (suggest starting at 50,000)**
    - Affects both the number of buffers and buffer size
  - **At minimum: Number of buffers >= Number of db2bm + Number of db2med + 1**
    - Point being have enough to cover parallelism + media targets/sessions
    - If any one side becomes a bottleneck can prevent a waiting for buffer situation
      - Keep adding more until it makes no further difference

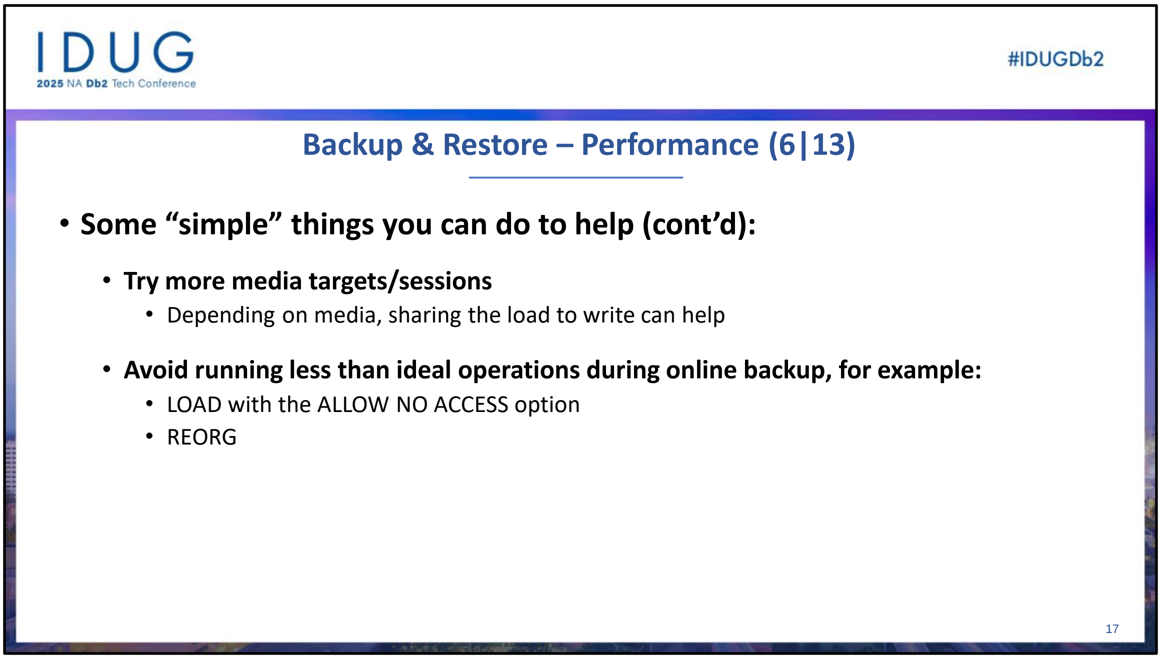
16

Optimizing backup performance

- https://www.ibm.com/docs/en/db2/12.1?topic=backup-optimizing-performance

Compatibility of online backup and other utilities

- https://www.ibm.com/docs/en/db2/12.1?topic=backup-compatibility-online-other-utilities



Backup & Restore – Performance (6|13)

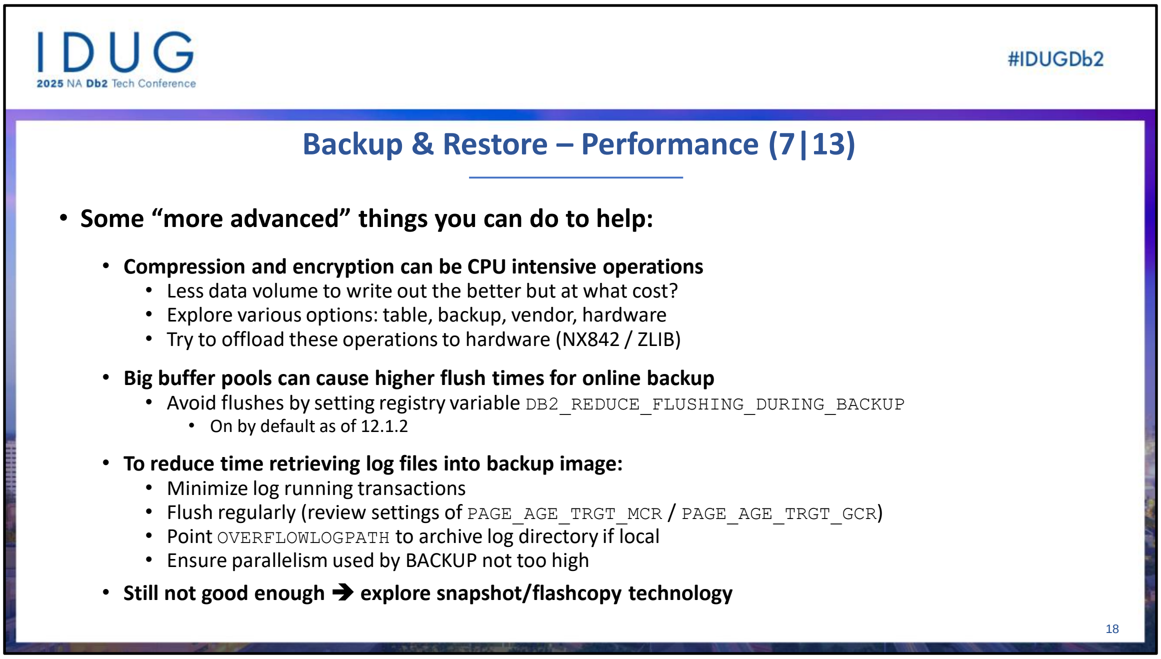
- **Some “simple” things you can do to help (cont’d):**
  - **Try more media targets/sessions**
    - Depending on media, sharing the load to write can help
  - **Avoid running less than ideal operations during online backup, for example:**
    - LOAD with the ALLOW NO ACCESS option
    - REORG

Optimizing backup performance

- https://www.ibm.com/docs/en/db2/12.1?topic=backup-optimizing-performance

Compatibility of online backup and other utilities

- https://www.ibm.com/docs/en/db2/12.1?topic=backup-compatibility-online-other-utilities



IDUG 2025 NA Db2 Tech Conference

#IDUGDb2

Backup & Restore – Performance (7|13)

- **Some “more advanced” things you can do to help:**
  - **Compression and encryption can be CPU intensive operations**
    - Less data volume to write out the better but at what cost?
    - Explore various options: table, backup, vendor, hardware
    - Try to offload these operations to hardware (NX842 / ZLIB)
  - **Big buffer pools can cause higher flush times for online backup**
    - Avoid flushes by setting registry variable `DB2_REDUCE_FLUSHING_DURING_BACKUP`
      - On by default as of 12.1.2
  - **To reduce time retrieving log files into backup image:**
    - Minimize log running transactions
    - Flush regularly (review settings of `PAGE_AGE_TRGT_MCR` / `PAGE_AGE_TRGT_GCR`)
    - Point `OVERFLOWLOGPATH` to archive log directory if local
    - Ensure parallelism used by BACKUP not too high
  - **Still not good enough ➔ explore snapshot/flashcopy technology**

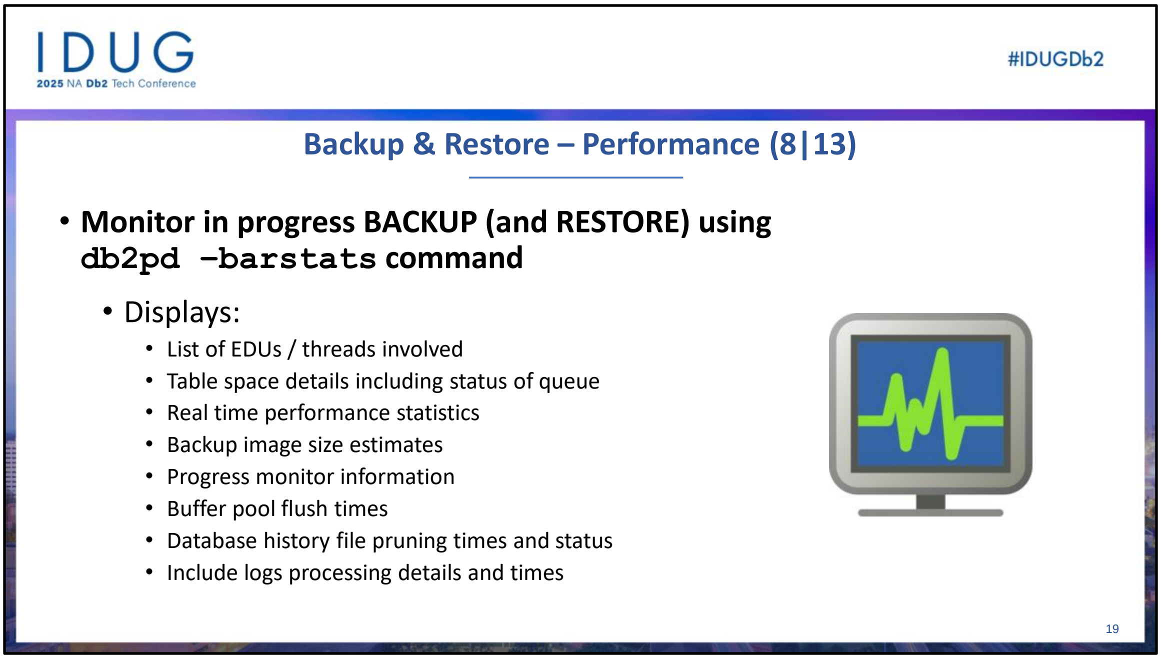
18

Optimizing backup performance

- https://www.ibm.com/docs/en/db2/12.1?topic=backup-optimizing-performance

Hardware accelerated backup and log file compression

- https://www.ibm.com/docs/en/db2/12.1?topic=considerations-hardware-accelerated-backup-log-file-compression



IDUG
2025 NA Db2 Tech Conference

#IDUGDb2

Backup & Restore – Performance (8|13)

- **Monitor in progress BACKUP (and RESTORE) using `db2pd -barstats` command**
  - Displays:
    - List of EDUs / threads involved
    - Table space details including status of queue
    - Real time performance statistics
    - Backup image size estimates
    - Progress monitor information
    - Buffer pool flush times
    - Database history file pruning times and status
    - Include logs processing details and times

19

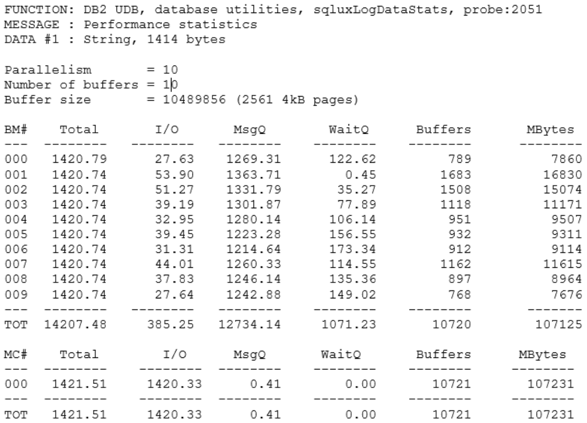
Monitoring backup and restore performance with db2pd -barstats

- https://www.ibm.com/docs/en/db2/12.1?topic=strategy-backup-restore-monitoring-db2pd-barstats
- https://www.ibm.com/docs/en/db2/12.1?topic=barstats-example-output-backup-performance-monitoring#c0070441

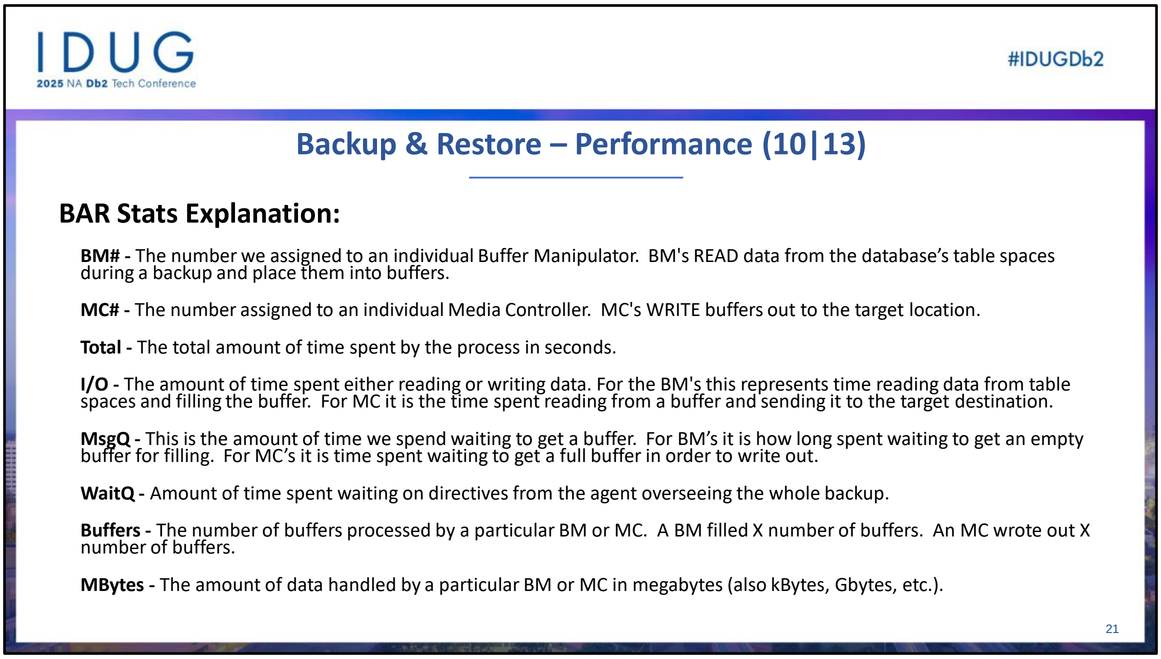
Backup & Restore – Performance (9|13)

BAR Stats Example:

- Backup to vendor
- Parallelism of 10 (BMs)
- 1 session (MC)
  - Media target
- util_heap_sz 30000



```
FUNCTION: DB2 UDB, database utilities, sqluxLogDataStats, probe:2051
MESSAGE : Performance statistics
DATA #1 : String, 1414 bytes

Parallelism       = 10
Number of buffers = 10
Buffer size       = 10489856 (2561 4kB pages)

BM#    Total      I/O      MsgQ      WaitQ      Buffers    MBytes
---  --------  --------  --------  --------  --------  --------
000   1420.79     27.63   1269.31    122.62       789      7860
001   1420.74     53.90   1363.71      0.45      1683     16830
002   1420.74     51.27   1331.79     35.27      1508     15074
003   1420.74     39.19   1301.87     77.89      1118     11171
004   1420.74     32.95   1280.14    106.14       951      9507
005   1420.74     39.45   1223.28    156.55       932      9311
006   1420.74     31.31   1214.64    173.34       912      9114
007   1420.74     44.01   1260.33    114.55      1162     11615
008   1420.74     37.83   1246.14    135.36       897      8964
009   1420.74     27.64   1242.88    149.02       768      7676
---  --------  --------  --------  --------  --------  -------
TOT  14207.48    385.25  12734.14   1071.23     10720    107125

MC#    Total      I/O      MsgQ      WaitQ     Buffers    MBytes
---  --------  --------  --------  --------  --------  --------
000   1421.51   1420.33      0.41      0.00     10721    107231
---  --------  --------  --------  --------  --------  --------
TOT   1421.51   1420.33      0.41      0.00     10721    107231
```



Backup & Restore – Performance (10|13)

BAR Stats Explanation:

BM# - The number we assigned to an individual Buffer Manipulator. BM's READ data from the database's table spaces 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 table spaces and filling the buffer. For MC it is the time spent reading from a buffer and sending it to the target destination.

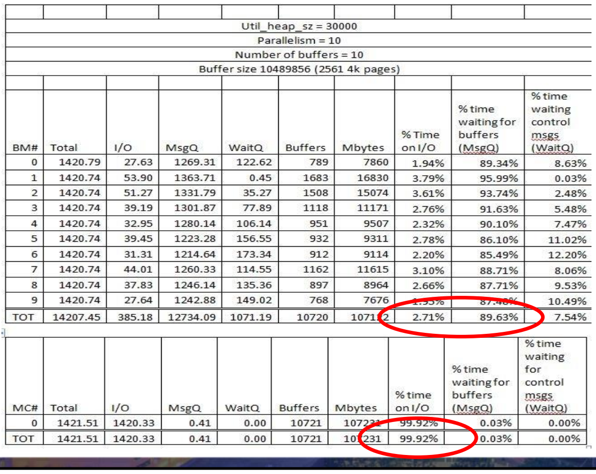
MsgQ - This is the amount of time we spend waiting to get a buffer. For BM's it is how long spent waiting to get an empty buffer for filling. For MC's it is time spent waiting to get a full buffer in order to write out.

WaitQ - 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.

MBytes - The amount of data handled by a particular BM or MC in megabytes (also kBytes, Gbytes, etc.).

Backup & Restore – Performance (11|13)

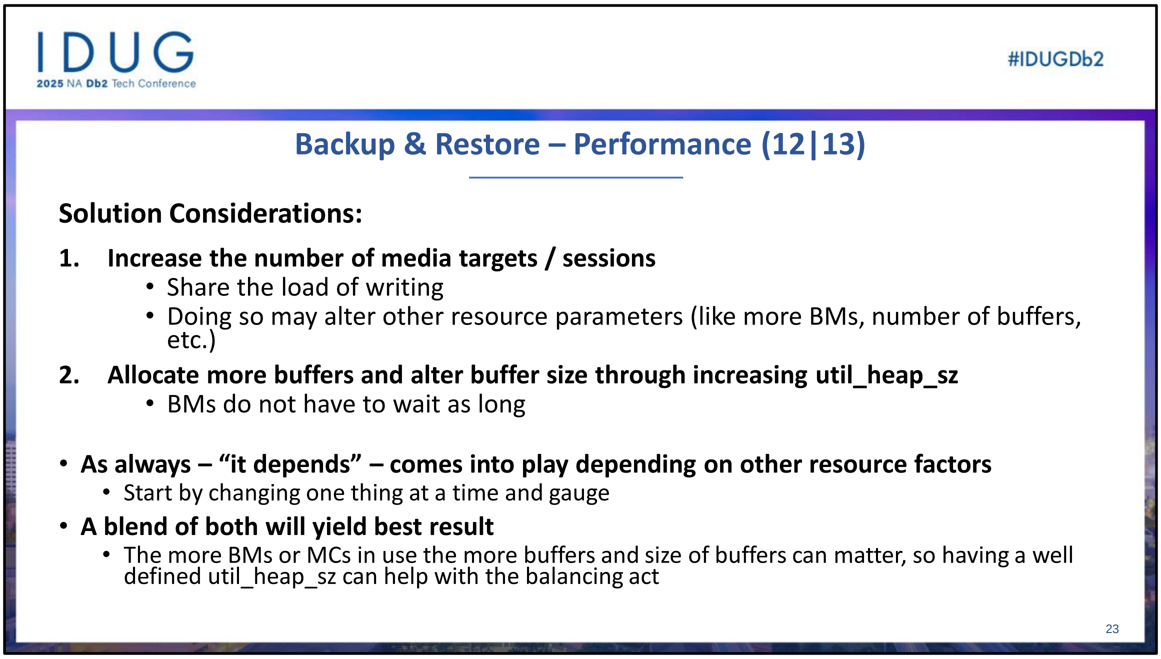


Util_heap_sz = 30000
Parallelism = 10
Number of buffers = 10
Buffer size 10489856 (2561 4k pages)

BM#	Total	I/O	MsgQ	WaitQ	Buffers	Mbytes	% Time on I/O	% time waiting for buffers (MsgQ)	% time waiting control msgs (WaitQ)
0	1420.79	27.63	1269.31	122.62	789	7860	1.94%	89.34%	8.63%
1	1420.74	53.90	1363.71	0.45	1683	16830	3.79%	95.99%	0.03%
2	1420.74	51.27	1331.79	35.27	1508	15074	3.61%	93.74%	2.48%
3	1420.74	39.19	1301.87	77.89	1118	11171	2.76%	91.63%	5.48%
4	1420.74	32.95	1280.14	106.14	951	9507	2.32%	90.10%	7.47%
5	1420.74	39.45	1223.28	156.55	932	9311	2.78%	86.10%	11.02%
6	1420.74	31.31	1214.64	173.34	912	9114	2.20%	85.49%	12.20%
7	1420.74	44.01	1260.33	114.55	1162	11615	3.10%	88.71%	8.06%
8	1420.74	37.83	1246.14	135.36	897	8964	2.66%	87.71%	9.53%
9	1420.74	27.64	1242.88	149.02	768	7676	1.95%	87.48%	10.49%
TOT	14207.45	385.18	12734.09	1071.19	10720	107122	2.71%	89.63%	7.54%

MC#	Total	I/O	MsgQ	WaitQ	Buffers	Mbytes	% time on I/O	% time waiting for buffers (MsgQ)	% time waiting for control msgs (WaitQ)
0	1421.51	1420.33	0.41	0.00	10721	107231	99.92%	0.03%	0.00%
TOT	1421.51	1420.33	0.41	0.00	10721	107231	99.92%	0.03%	0.00%

- BMs:
  - On average spent 2.71% of time waiting to read data from the database
  - Spent 89.63% of the time waiting for buffers
- MCs:
  - On average spent 99.92% of the time waiting for I/O (in this case vendor)
- Conclusion:
  - MC cannot free up a buffer until the vendor confirms it has been written
  - BMs must wait for the MCs to free the buffers
  - Bottleneck is in writing to the target device



IDUG 2025 NA Db2 Tech Conference

#IDUGDb2

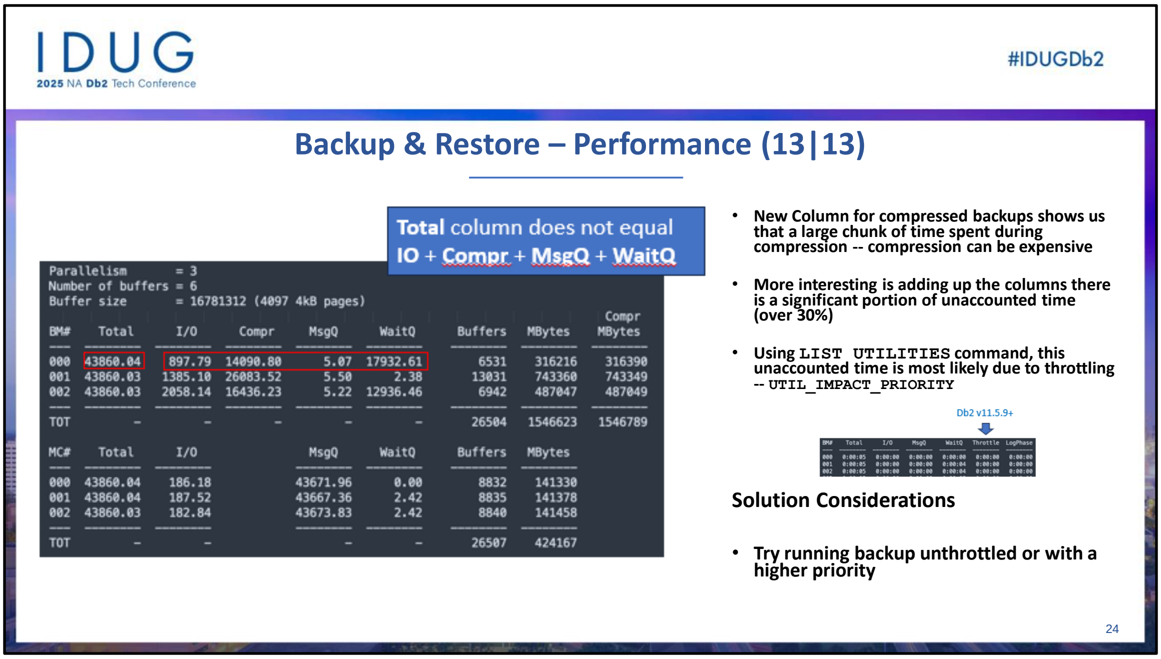
Backup & Restore – Performance (12|13)

Solution Considerations:

1. **Increase the number of media targets / sessions**
   - Share the load of writing
   - Doing so may alter other resource parameters (like more BMs, number of buffers, etc.)
2. **Allocate more buffers and alter buffer size through increasing util_heap_sz**
   - BMs do not have to wait as long

- **As always – "it depends" – comes into play depending on other resource factors**
  - Start by changing one thing at a time and gauge
- **A blend of both will yield best result**
  - The more BMs or MCs in use the more buffers and size of buffers can matter, so having a well defined util_heap_sz can help with the balancing act

23



IDUG
2025 NA **Db2** Tech Conference

#IDUGDb2

Backup & Restore – Performance (13|13)

Total column does not equal
IO + Compr + MsgQ + WaitQ

```
Parallelism       = 3
Number of buffers = 6
Buffer size       = 16781312 (4097 4kB pages)

                                                                         Compr
BM#    Total      I/O      Compr     MsgQ      WaitQ      Buffers   MBytes    MBytes
---  --------  --------  --------  --------  --------    --------  --------  --------
000  43860.04    897.79  14090.80      5.07  17932.61        6531    316216    316390
001  43860.03   1385.10  26083.52      5.50      2.38       13031    743360    743349
002  43860.03   2058.14  16436.23      5.22  12936.46        6942    487047    487049
---  --------  --------  --------  --------  --------    --------  --------  --------
TOT         -         -         -         -         -       26504   1546623   1546789

MC#    Total      I/O                MsgQ      WaitQ      Buffers   MBytes
---  --------  --------            --------  --------    --------  --------
000  43860.04    186.18            43671.96      0.00        8832    141330
001  43860.04    187.52            43667.36      2.42        8835    141378
002  43860.03    182.84            43673.83      2.42        8840    141458
---  --------  --------            --------  --------    --------  --------
TOT         -         -                   -         -       26507    424167
```

- **New Column for compressed backups shows us that a large chunk of time spent during compression -- compression can be expensive**
- **More interesting is adding up the columns there is a significant portion of unaccounted time (over 30%)**
- **Using `LIST UTILITIES` command, this unaccounted time is most likely due to throttling -- `UTIL_IMPACT_PRIORITY`**

Db2 v11.5.9+

```
BM#    Total       I/O       MsgQ      WaitQ     Throttle  LogPhase
---  --------  --------  --------  --------  --------  --------
000   0:00:05   0:00:00   0:00:00   0:00:00   0:00:00   0:00:00
001   0:00:05   0:00:00   0:00:00   0:00:04   0:00:00   0:00:00
002   0:00:05   0:00:00   0:00:00   0:00:04   0:00:00   0:00:00
```

Solution Considerations

- **Try running backup unthrottled or with a higher priority**

24

First we see a new Column for compression, which tells us that this is a compressed backup.

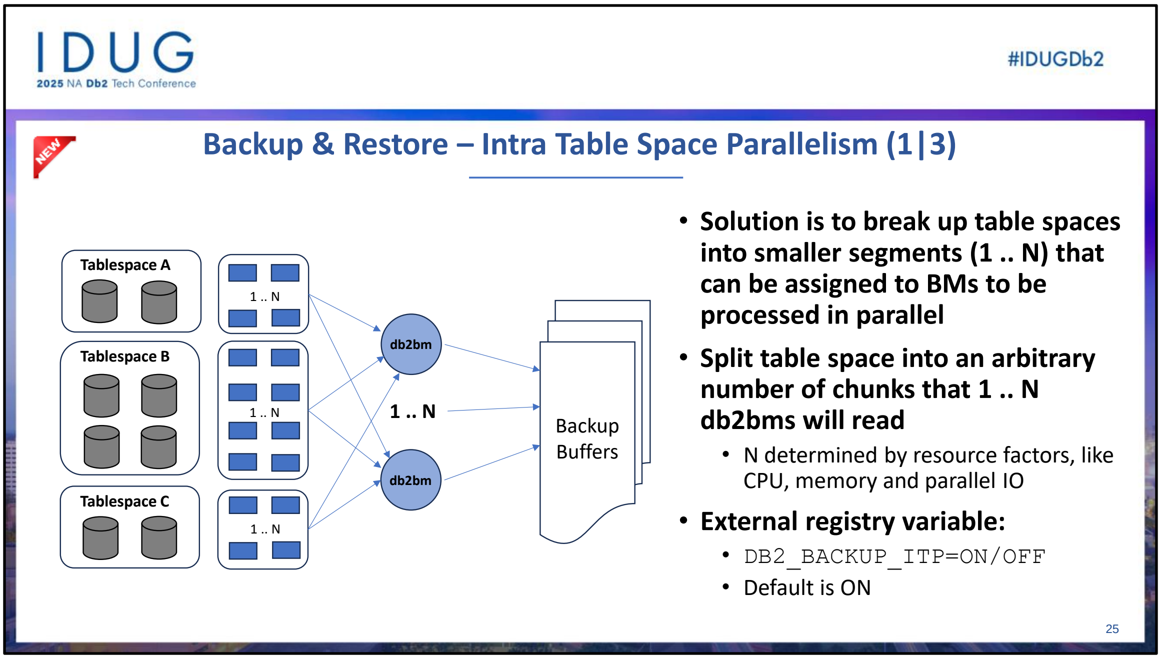
The BM stats shows us that a large chunk of time was spent during compression.

- This makes sense, considering that compression can be expensive.
- What's more interesting is that if you add up the individual columns, there is a significant portion of unaccounted time (over 30%).
- This unaccounted time is most likely due to throttling.

You can use the list utilities command to show what is currently throttled, and the current throttle priority settings.

A backup by default is not throttled. To throttle a backup, the UTIL_IMPACT_PRIORITY option needs to be specified.
So, you can try to run the backup unthrottled, or with a higher throttle priority.

Worth noting that in newer versions of Db2, we have a new throttle column that indicates how much time was spent on throttling.



IDUG
2025 NA Db2 Tech Conference

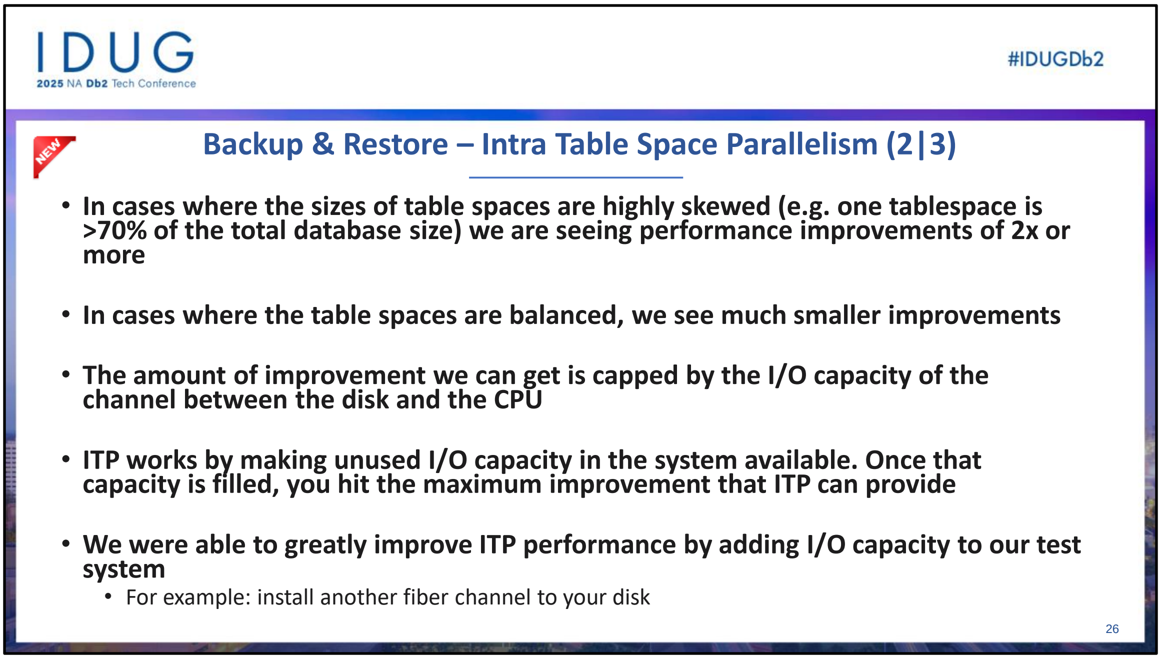
#IDUGDb2

NEW

Backup & Restore – Intra Table Space Parallelism (1|3)

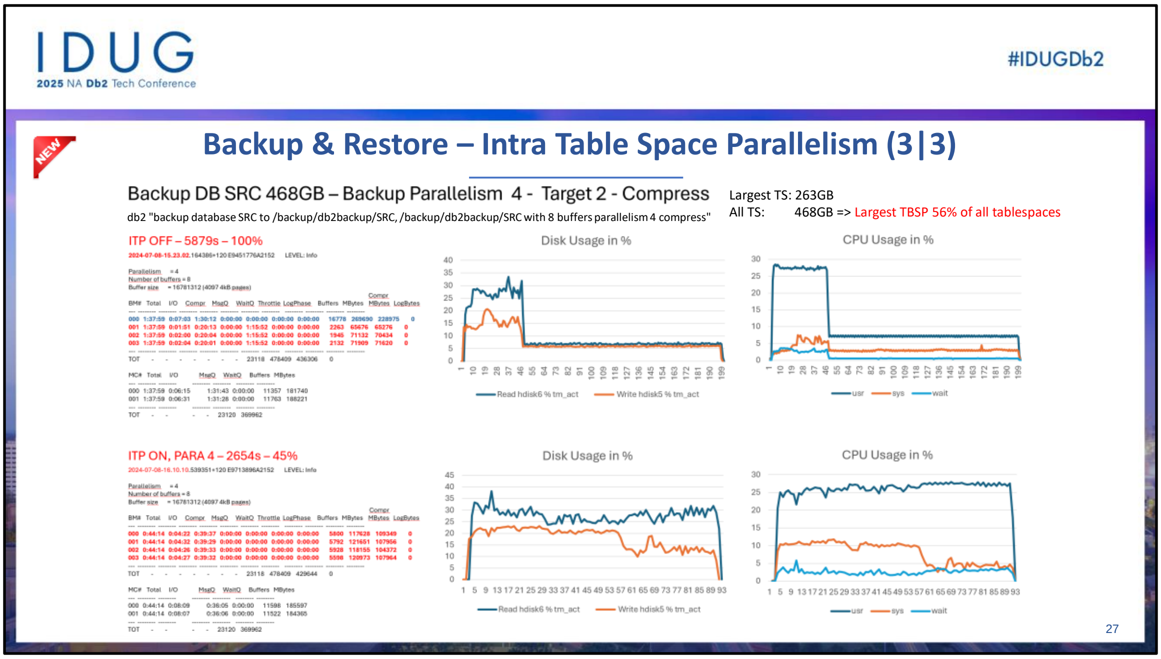
- **Solution is to break up table spaces into smaller segments (1 .. N) that can be assigned to BMs to be processed in parallel**
- **Split table space into an arbitrary number of chunks that 1 .. N db2bms will read**
  - N determined by resource factors, like CPU, memory and parallel IO
- **External registry variable:**
  - `DB2_BACKUP_ITP=ON/OFF`
  - Default is ON

25



Backup & Restore – Intra Table Space Parallelism (2|3)

- **In cases where the sizes of table spaces are highly skewed (e.g. one tablespace is >70% of the total database size) we are seeing performance improvements of 2x or more**
- **In cases where the table spaces are balanced, we see much smaller improvements**
- **The amount of improvement we can get is capped by the I/O capacity of the channel between the disk and the CPU**
- **ITP works by making unused I/O capacity in the system available. Once that capacity is filled, you hit the maximum improvement that ITP can provide**
- **We were able to greatly improve ITP performance by adding I/O capacity to our test system**
  - For example: install another fiber channel to your disk



Backup & Restore – Intra Table Space Parallelism (3|3)

Backup DB SRC 468GB – Backup Parallelism 4 - Target 2 - Compress

db2 "backup database SRC to /backup/db2backup/SRC, /backup/db2backup/SRC with 8 buffers parallelism 4 compress"

Largest TS: 263GB
All TS: 468GB => Largest TBSP 56% of all tablespaces

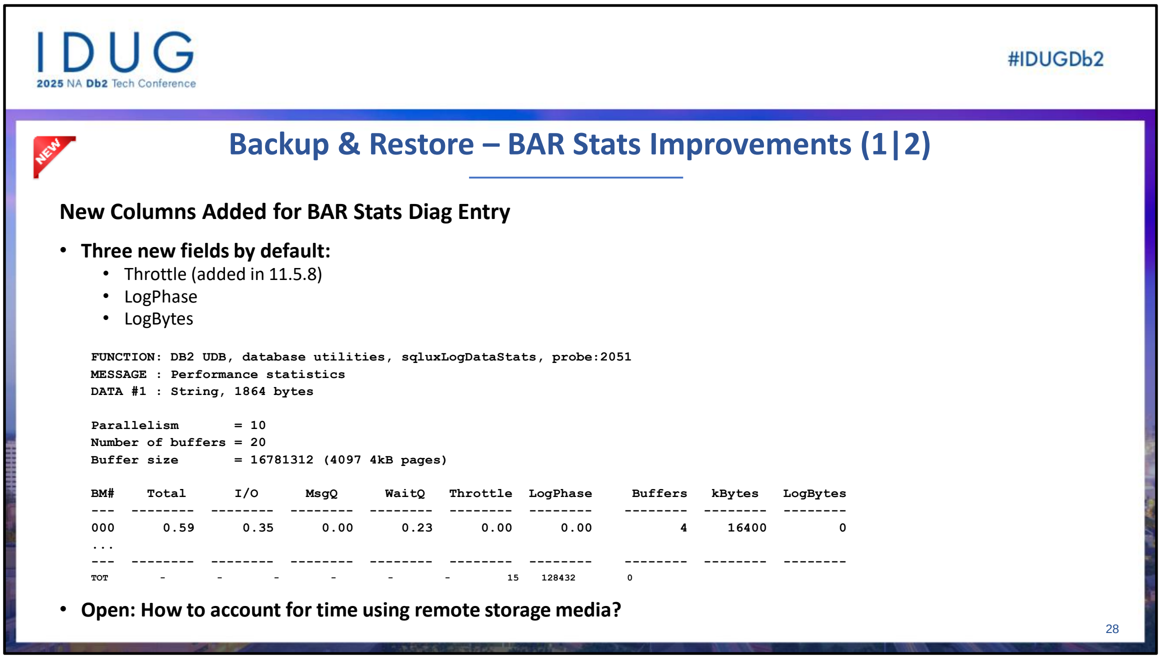
ITP OFF – 5879s – 100%

```
2024-07-08-15.23.02.164386+120 E9451776A2152      LEVEL: Info

Parallelism     = 4
Number of buffers = 8
Buffer size     = 16781312 (4097 4kB pages)

                                                                          Compr
BM#  Total    I/O      Compr    MsgQ     WaitQ    Throttle LogPhase   Buffers  MBytes   MBytes   LogBytes
---  -------- -------- -------- -------- -------- -------- --------   -------- -------- -------- --------
000  1:37:59  0:07:03  1:30:12  0:00:00  0:00:00  0:00:00  0:00:00    16778    269690   228975   0
001  1:37:59  0:01:51  0:20:13  0:00:00  1:15:52  0:00:00  0:00:00    2263     65676    65276    0
002  1:37:59  0:02:00  0:20:04  0:00:00  1:15:52  0:00:00  0:00:00    1945     71132    70434    0
003  1:37:59  0:02:04  0:20:01  0:00:00  1:15:52  0:00:00  0:00:00    2132     71909    71620    0
---  -------- -------- -------- -------- -------- -------- --------   -------- -------- -------- --------
TOT  -        -        -        -        -        -        -          23118    478409   436306   0

MC#  Total    I/O               MsgQ     WaitQ     Buffers  MBytes
---  -------- --------          -------- --------  -------- --------
000  1:37:59  0:06:15           1:31:43  0:00:00   11357    181740
001  1:37:59  0:06:31           1:31:28  0:00:00   11763    188221
---  -------- --------          -------- --------  -------- --------
TOT  -        -                 -        -         23120    369962
```

ITP ON, PARA 4 – 2654s – 45%

```
2024-07-08-16.10.10.539351+120 E9713896A2152      LEVEL: Info

Parallelism     = 4
Number of buffers = 8
Buffer size     = 16781312 (4097 4kB pages)

                                                                          Compr
BM#  Total    I/O      Compr    MsgQ     WaitQ    Throttle LogPhase   Buffers  MBytes   MBytes   LogBytes
---  -------- -------- -------- -------- -------- -------- --------   -------- -------- -------- --------
000  0:44:14  0:04:22  0:39:37  0:00:00  0:00:00  0:00:00  0:00:00    5800     117628   109349   0
001  0:44:14  0:04:32  0:39:29  0:00:00  0:00:00  0:00:00  0:00:00    5792     121651   107956   0
002  0:44:14  0:04:26  0:39:33  0:00:00  0:00:00  0:00:00  0:00:00    5928     118155   104372   0
003  0:44:14  0:04:27  0:39:32  0:00:00  0:00:00  0:00:00  0:00:00    5598     120973   107964   0
---  -------- -------- -------- -------- -------- -------- --------   -------- -------- -------- --------
TOT  -        -        -        -        -        -        -          23118    478409   429644   0

MC#  Total    I/O               MsgQ     WaitQ     Buffers  MBytes
---  -------- --------          -------- --------  -------- --------
000  0:44:14  0:08:09           0:36:05  0:00:00   11598    185597
001  0:44:14  0:08:07           0:36:06  0:00:00   11522    184365
---  -------- --------          -------- --------  -------- --------
TOT  -        -                 -        -         23120    369962
```

- POWER 8
- FC: 2x 8GBit ports – SDD and HDD on different ports
- AIX 7200-05-07-2346
- Backup target: HDD
- Backup source: SDD
- Db2 software compression used to slowdown I/O rate



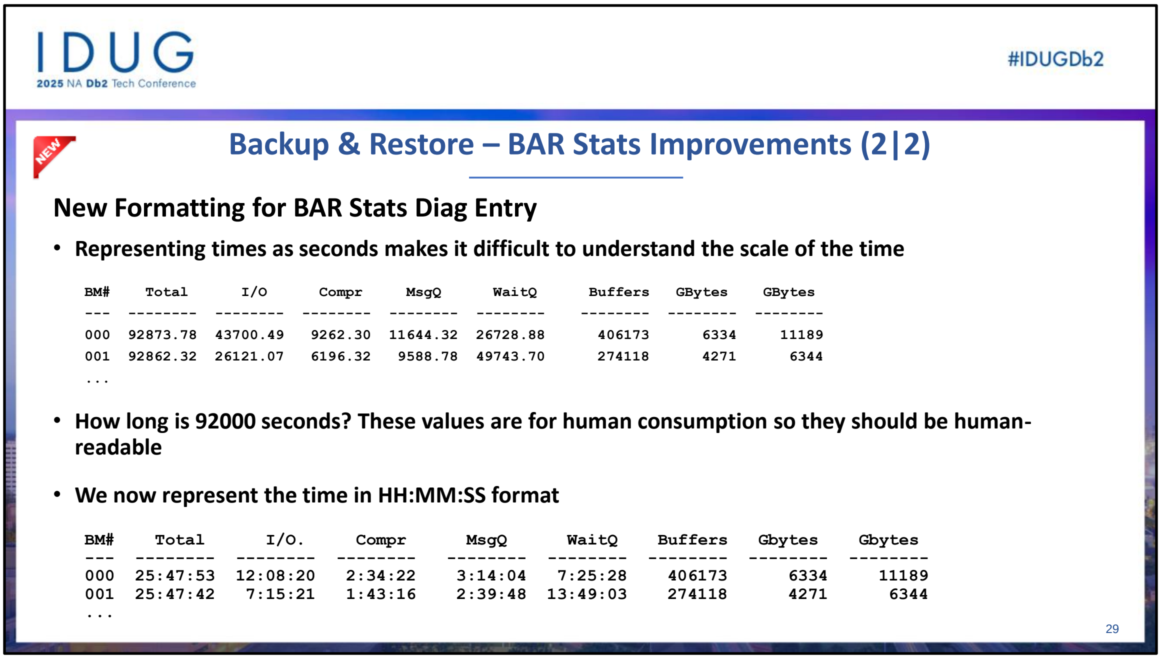
Backup & Restore – BAR Stats Improvements (1|2)

New Columns Added for BAR Stats Diag Entry

- **Three new fields by default:**
  - Throttle (added in 11.5.8)
  - LogPhase
  - LogBytes

```
FUNCTION: DB2 UDB, database utilities, sqluxLogDataStats, probe:2051
MESSAGE : Performance statistics
DATA #1 : String, 1864 bytes

Parallelism       = 10
Number of buffers = 20
Buffer size       = 16781312 (4097 4kB pages)

BM#    Total      I/O      MsgQ      WaitQ    Throttle  LogPhase     Buffers   kBytes   LogBytes
---  --------  --------  --------  --------  --------  --------    --------  --------  --------
000      0.59      0.35      0.00      0.23      0.00      0.00           4     16400         0
...
---  --------  --------  --------  --------  --------  --------    --------  --------  --------
TOT        -        -        -        -        -        -        15    128432        0
```

- **Open: How to account for time using remote storage media?**



NEW

Backup & Restore – BAR Stats Improvements (2|2)

New Formatting for BAR Stats Diag Entry

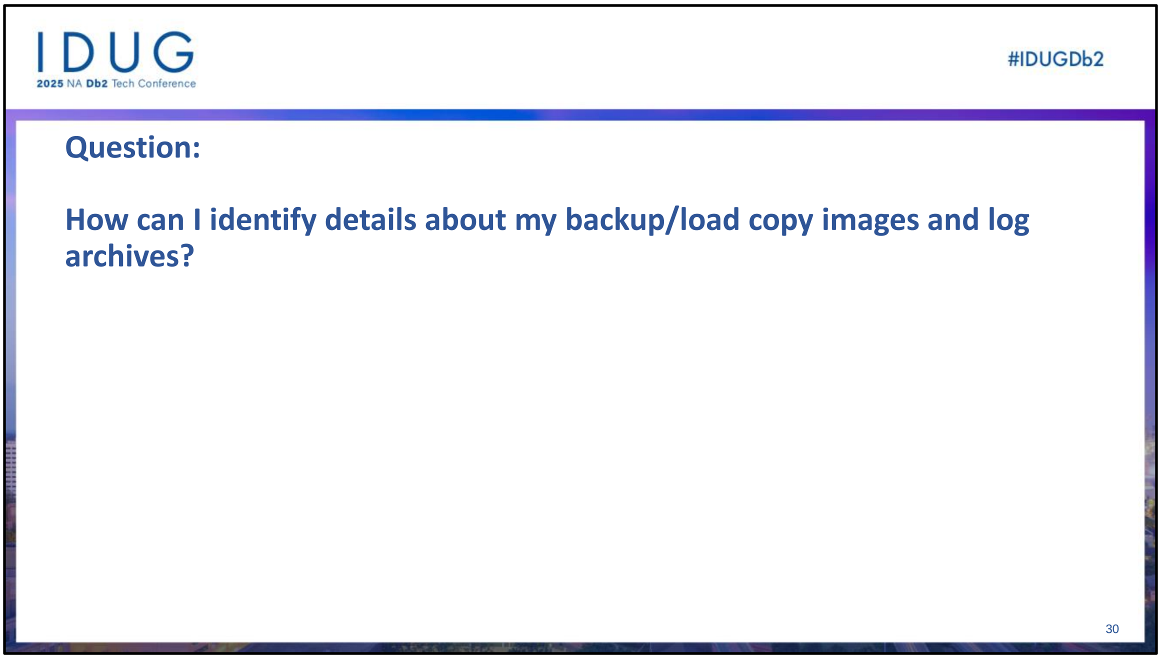
- **Representing times as seconds makes it difficult to understand the scale of the time**

```
BM#    Total      I/O      Compr     MsgQ      WaitQ       Buffers    GBytes     GBytes
---  --------  --------  --------  --------  --------    --------  --------   --------
000  92873.78  43700.49   9262.30  11644.32  26728.88      406173      6334      11189
001  92862.32  26121.07   6196.32   9588.78  49743.70      274118      4271       6344
...
```

- **How long is 92000 seconds? These values are for human consumption so they should be human-readable**

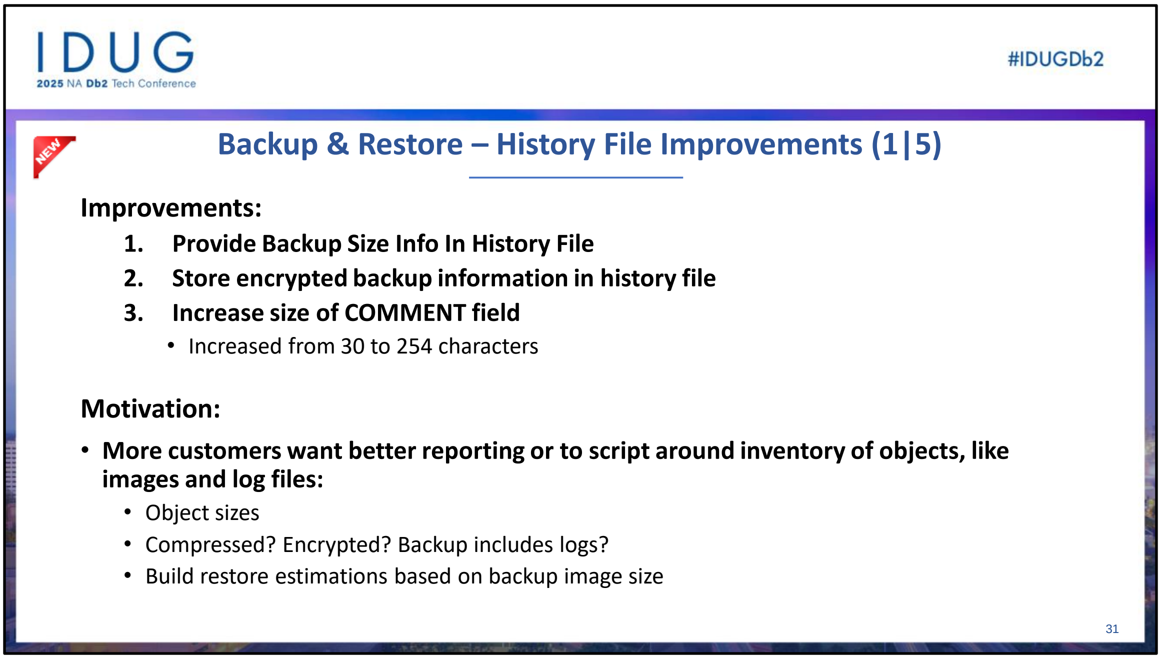
- **We now represent the time in HH:MM:SS format**

```
BM#     Total      I/O.      Compr       MsgQ      WaitQ    Buffers    Gbytes    Gbytes
---  --------- --------- ---------  --------- ---------  --------- --------- ---------
000   25:47:53  12:08:20   2:34:22    3:14:04   7:25:28     406173      6334     11189
001   25:47:42   7:15:21   1:43:16    2:39:48  13:49:03     274118      4271      6344
...
```



Question:

How can I identify details about my backup/load copy images and log archives?



NEW

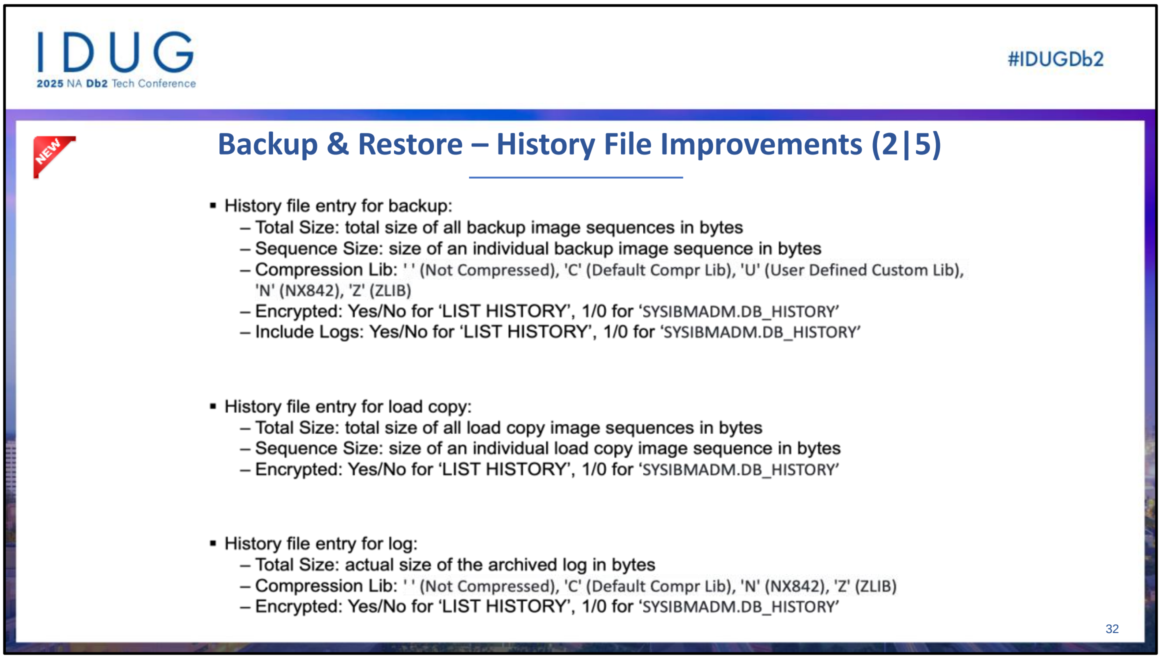
Backup & Restore – History File Improvements (1|5)

Improvements:

1. **Provide Backup Size Info In History File**
2. **Store encrypted backup information in history file**
3. **Increase size of COMMENT field**
   - Increased from 30 to 254 characters

Motivation:

- **More customers want better reporting or to script around inventory of objects, like images and log files:**
  - Object sizes
  - Compressed? Encrypted? Backup includes logs?
  - Build restore estimations based on backup image size



NEW

Backup & Restore – History File Improvements (2|5)

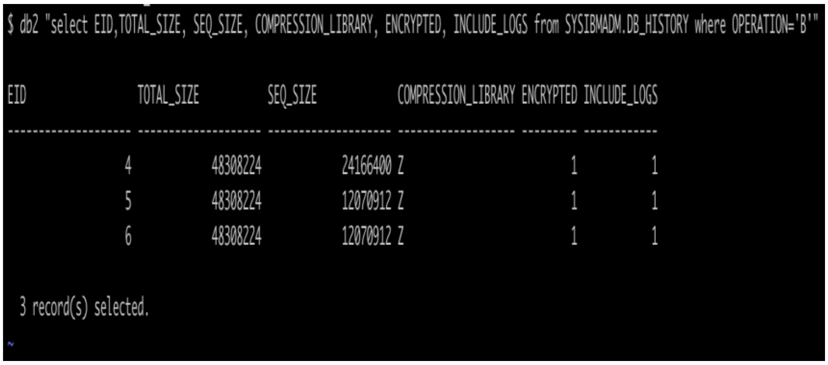
- History file entry for backup:
  - Total Size: total size of all backup image sequences in bytes
  - Sequence Size: size of an individual backup image sequence in bytes
  - Compression Lib: ' ' (Not Compressed), 'C' (Default Compr Lib), 'U' (User Defined Custom Lib), 'N' (NX842), 'Z' (ZLIB)
  - Encrypted: Yes/No for 'LIST HISTORY', 1/0 for 'SYSIBMADM.DB_HISTORY'
  - Include Logs: Yes/No for 'LIST HISTORY', 1/0 for 'SYSIBMADM.DB_HISTORY'

- History file entry for load copy:
  - Total Size: total size of all load copy image sequences in bytes
  - Sequence Size: size of an individual load copy image sequence in bytes
  - Encrypted: Yes/No for 'LIST HISTORY', 1/0 for 'SYSIBMADM.DB_HISTORY'

- History file entry for log:
  - Total Size: actual size of the archived log in bytes
  - Compression Lib: ' ' (Not Compressed), 'C' (Default Compr Lib), 'N' (NX842), 'Z' (ZLIB)
  - Encrypted: Yes/No for 'LIST HISTORY', 1/0 for 'SYSIBMADM.DB_HISTORY'

Backup & Restore – History File Improvements (3|5)

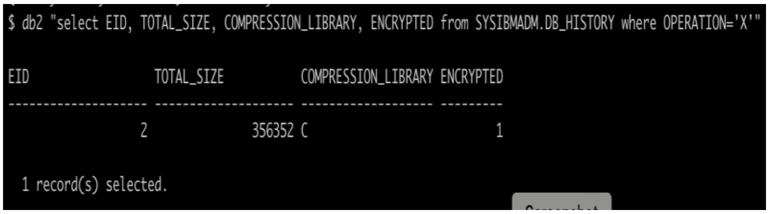
- SYSIBMADM.DB_HISTORY displays Total_Size, Seq_Size, Compression_Library, Encrypted, and IncludeLogs for an online backup image made up of three sequences with compression(ZLIB) and encryption enabled:



```
$ db2 "select EID,TOTAL_SIZE, SEQ_SIZE, COMPRESSION_LIBRARY, ENCRYPTED, INCLUDE_LOGS from SYSIBMADM.DB_HISTORY where OPERATION='B'"

EID                  TOTAL_SIZE           SEQ_SIZE             COMPRESSION_LIBRARY ENCRYPTED INCLUDE_LOGS
-------------------- -------------------- -------------------- ------------------- --------- ------------
                   4             48308224             24166400 Z                           1            1
                   5             48308224             12070912 Z                           1            1
                   6             48308224             12070912 Z                           1            1

  3 record(s) selected.

~
```

Backup & Restore – History File Improvements (4|5)

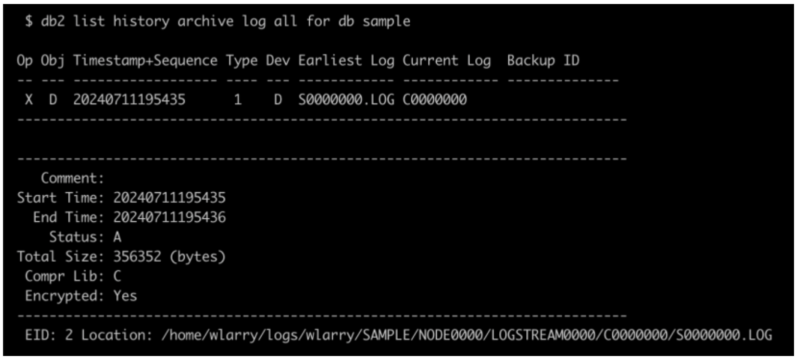
- **SYSIBMADM.DB_HISTORY displays Total Size, Compression Library and Encrypted for a log archive with both compression and encryption enabled:**



```
$ db2 "select EID, TOTAL_SIZE, COMPRESSION_LIBRARY, ENCRYPTED from SYSIBMADM.DB_HISTORY where OPERATION='X'"

EID                  TOTAL_SIZE           COMPRESSION_LIBRARY ENCRYPTED
-------------------- -------------------- ------------------- ---------
                   2               356352 C                           1

  1 record(s) selected.
```

Backup & Restore – History File Improvements (5|5)

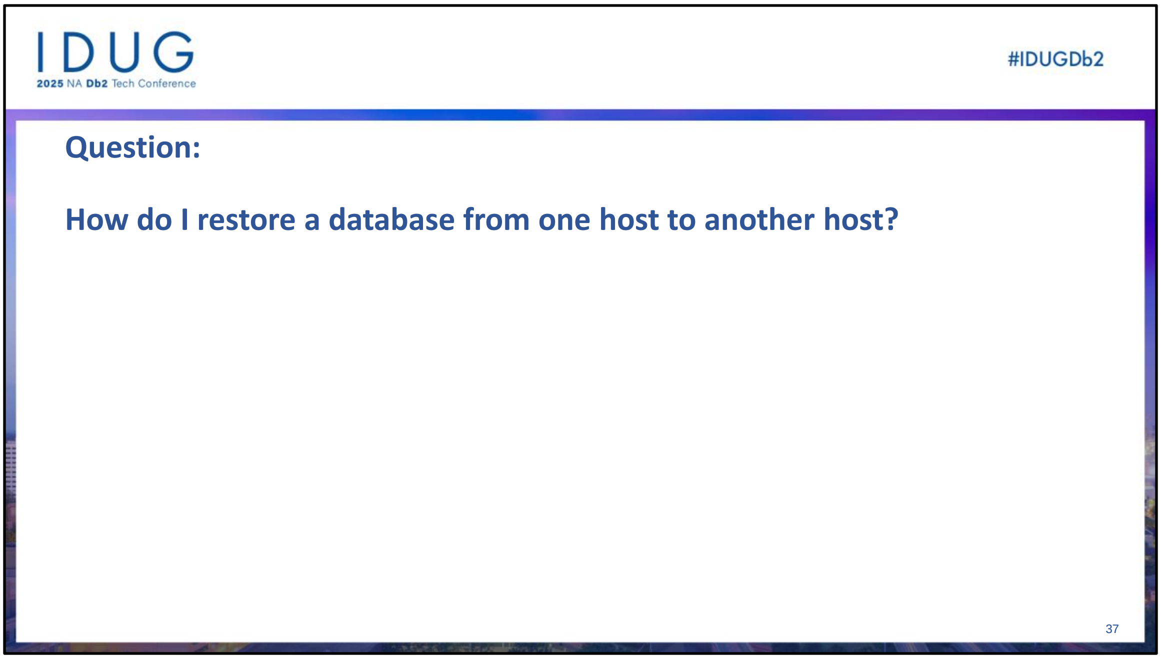
- **LIST HISTORY displays Total Size, Compression Library and Encrypted for a log archive with both compression and encryption enabled:**



```
 $ db2 list history archive log all for db sample

Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log  Backup ID
-- --- ------------------ ---- --- ------------ ------------ --------------
 X  D  20240711195435      1    D  S0000000.LOG C0000000
----------------------------------------------------------------------------


----------------------------------------------------------------------------
   Comment:
Start Time: 20240711195435
  End Time: 20240711195436
    Status: A
Total Size: 356352 (bytes)
 Compr Lib: C
 Encrypted: Yes
----------------------------------------------------------------------------
 EID: 2 Location: /home/wlarry/logs/wlarry/SAMPLE/NODE0000/LOGSTREAM0000/C0000000/S0000000.LOG
```

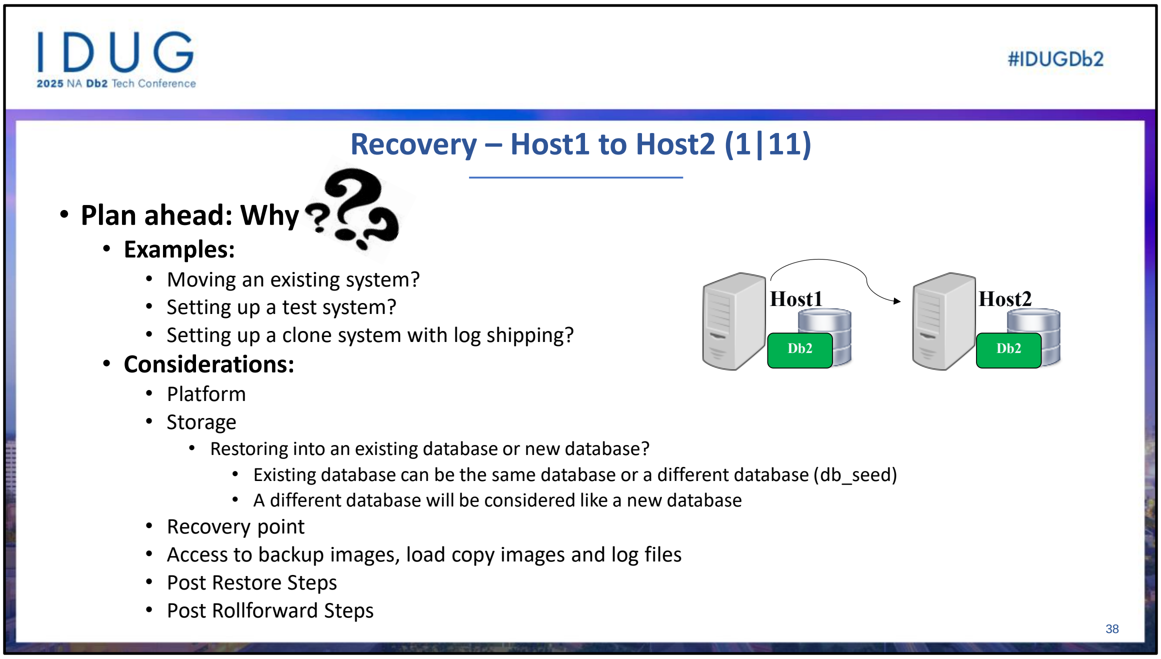
Agenda

- Logging
  - Disk Full in Archives
    - Automatic Pruning
- Backup & Restore
  - Performance, ITP & BAR Stats
  - History File Improvements
- **Recovery**
  - **Host1 to Host2**



Question:

How do I restore a database from one host to another host?



IDUG
2025 NA Db2 Tech Conference

#IDUGDb2

Recovery – Host1 to Host2 (1|11)

- **Plan ahead: Why**
  - **Examples:**
    - Moving an existing system?
    - Setting up a test system?
    - Setting up a clone system with log shipping?
  - **Considerations:**
    - Platform
    - Storage
      - Restoring into an existing database or new database?
        - Existing database can be the same database or a different database (db_seed)
        - A different database will be considered like a new database
    - Recovery point
    - Access to backup images, load copy images and log files
    - Post Restore Steps
    - Post Rollforward Steps

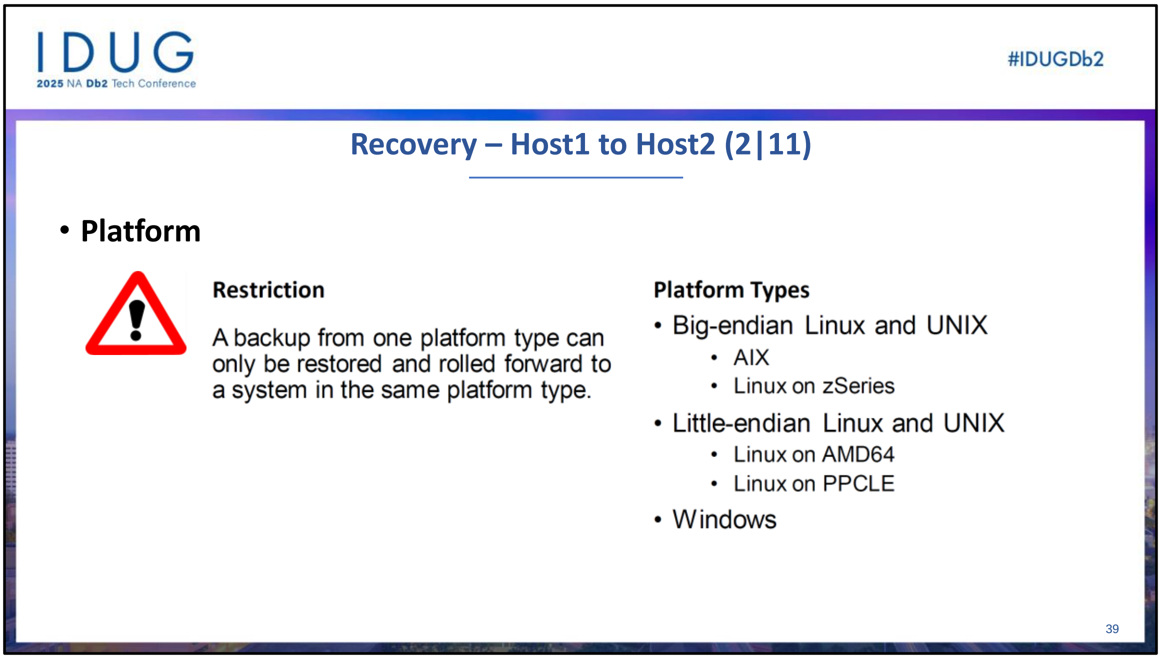
38

Restore overview

- https://www.ibm.com/docs/en/db2/12.1.0?topic=recovery-restore

Rollforward overview

- https://www.ibm.com/docs/en/db2/12.1.0?topic=recovery-rollforward



IDUG
2025 NA Db2 Tech Conference

#IDUGDb2

Recovery – Host1 to Host2 (2|11)

- **Platform**

Restriction

A backup from one platform type can only be restored and rolled forward to a system in the same platform type.

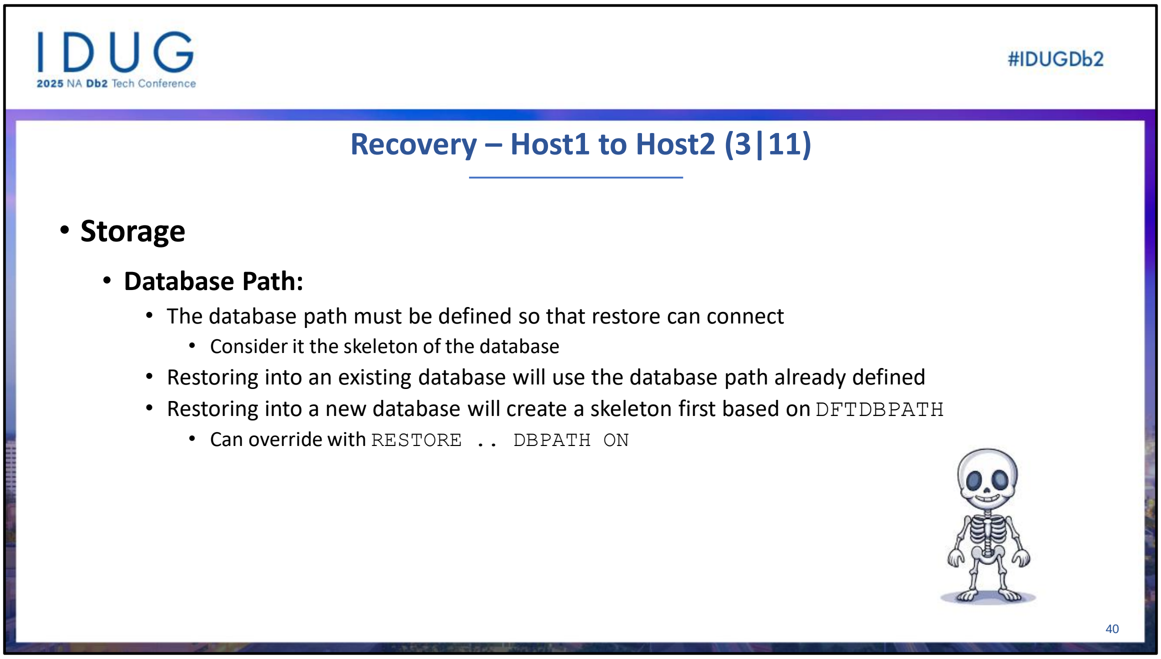
Platform Types

- Big-endian Linux and UNIX
  - AIX
  - Linux on zSeries
- Little-endian Linux and UNIX
  - Linux on AMD64
  - Linux on PPCLE
- Windows

39

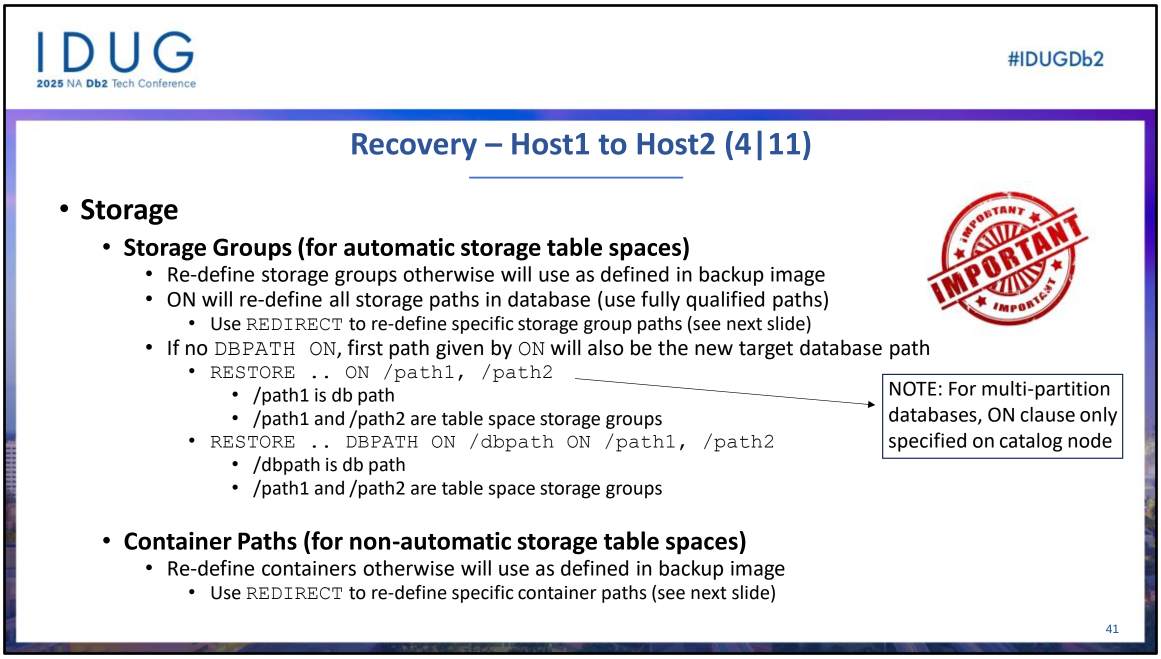
Backup and restore operations between different operating systems and hardware platforms

- https://www.ibm.com/docs/en/db2/12.1.0?topic=dbrs-backup-restore-operations-between-different-operating-systems-hardware-platforms



Recovery – Host1 to Host2 (3|11)

- **Storage**
  - **Database Path:**
    - The database path must be defined so that restore can connect
      - Consider it the skeleton of the database
    - Restoring into an existing database will use the database path already defined
    - Restoring into a new database will create a skeleton first based on `DFTDBPATH`
      - Can override with `RESTORE .. DBPATH ON`



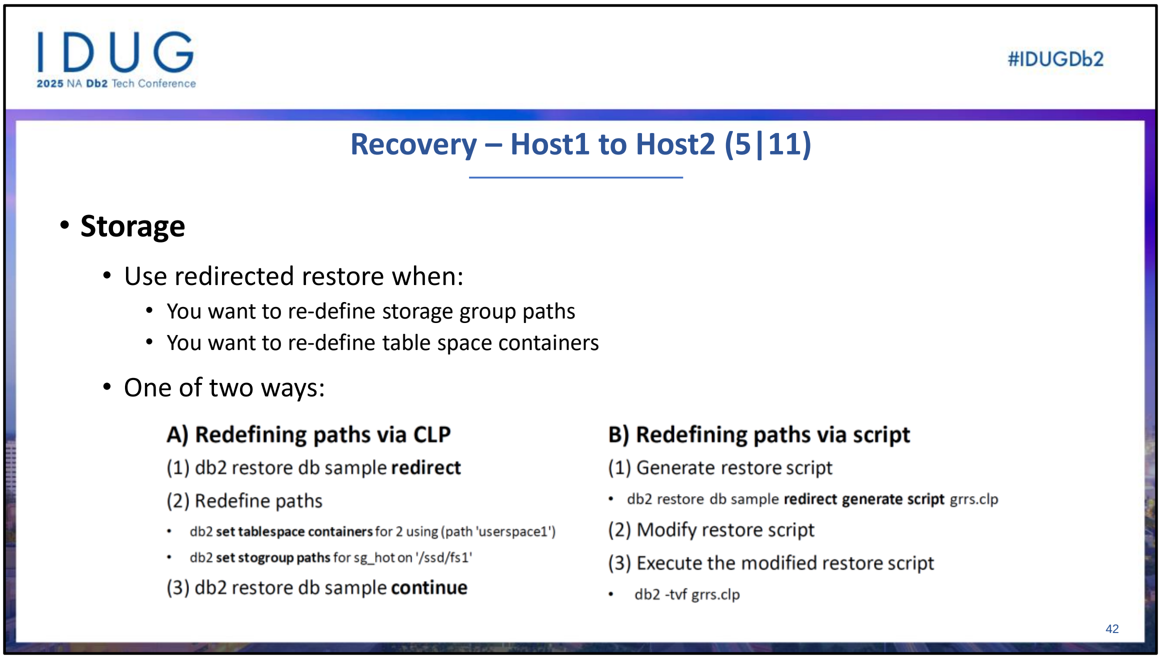
Recovery – Host1 to Host2 (4|11)

IMPORTANT

- **Storage**
  - **Storage Groups (for automatic storage table spaces)**
    - Re-define storage groups otherwise will use as defined in backup image
    - ON will re-define all storage paths in database (use fully qualified paths)
      - Use `REDIRECT` to re-define specific storage group paths (see next slide)
    - If no `DBPATH ON`, first path given by `ON` will also be the new target database path
      - `RESTORE .. ON /path1, /path2`
        - /path1 is db path
        - /path1 and /path2 are table space storage groups
      - `RESTORE .. DBPATH ON /dbpath ON /path1, /path2`
        - /dbpath is db path
        - /path1 and /path2 are table space storage groups

NOTE: For multi-partition databases, ON clause only specified on catalog node

  - **Container Paths (for non-automatic storage table spaces)**
    - Re-define containers otherwise will use as defined in backup image
      - Use `REDIRECT` to re-define specific container paths (see next slide)



IDUG
2025 NA **Db2** Tech Conference

#IDUGDb2

Recovery – Host1 to Host2 (5|11)

- **Storage**
  - Use redirected restore when:
    - You want to re-define storage group paths
    - You want to re-define table space containers
  - One of two ways:

A) Redefining paths via CLP

(1) db2 restore db sample **redirect**

(2) Redefine paths

- db2 **set tablespace containers** for 2 using (path 'userspace1')
- db2 **set stogroup paths** for sg_hot on '/ssd/fs1'

(3) db2 restore db sample **continue**

B) Redefining paths via script

(1) Generate restore script

- db2 restore db sample **redirect generate script** grrs.clp

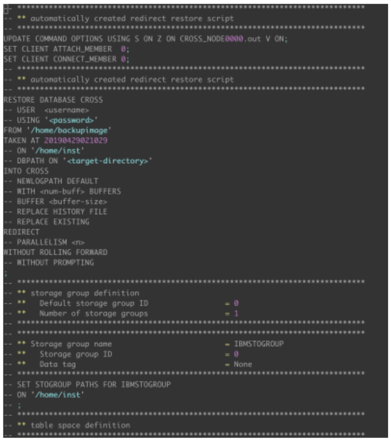
(2) Modify restore script

(3) Execute the modified restore script

- db2 -tvf grrs.clp

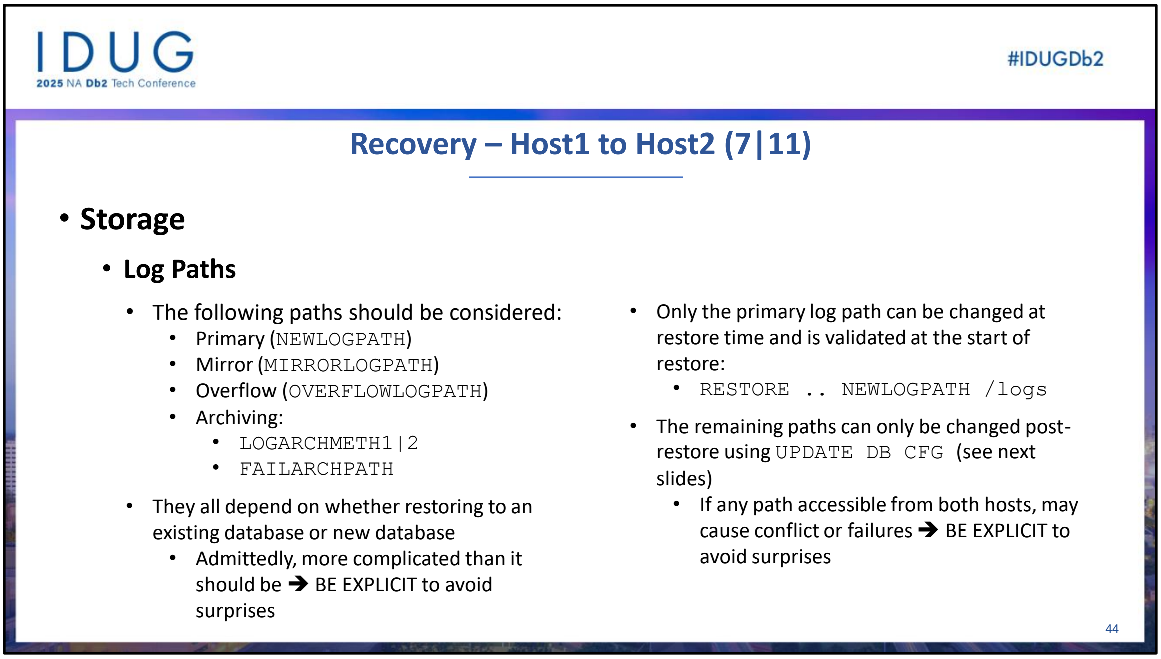
42

Recovery – Host1 to Host2 (6|11)



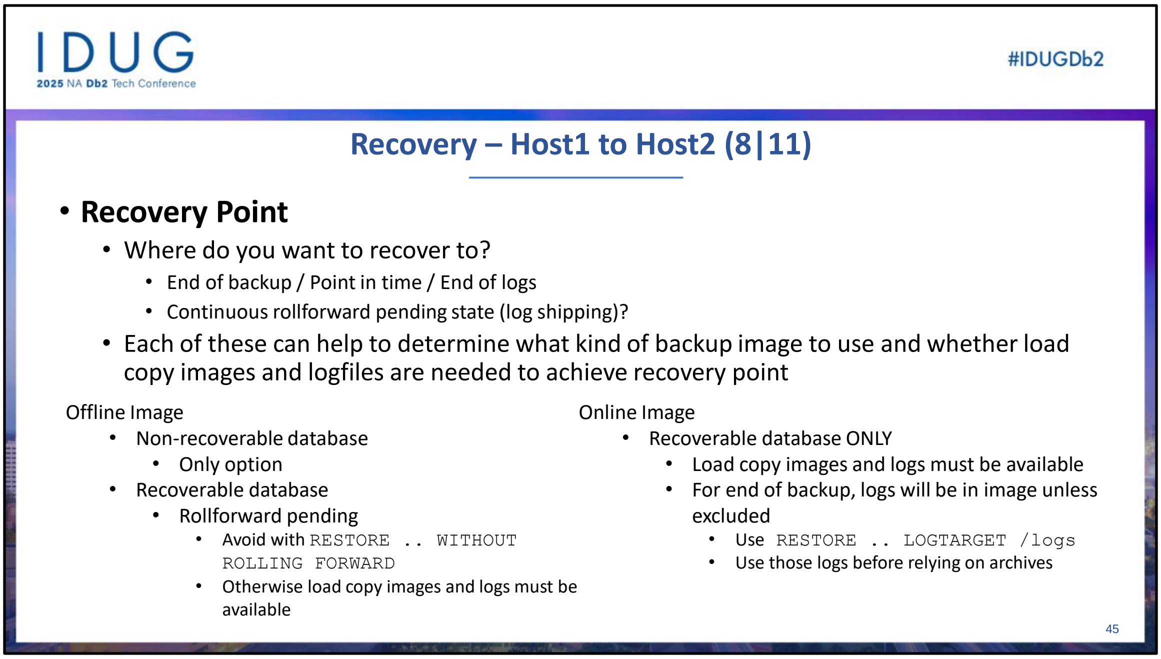
```
-- *****************************************************************************
-- ** automatically created redirect restore script
-- *****************************************************************************
UPDATE COMMAND OPTIONS USING S ON Z ON CROSS_NODE0000.out V ON;
SET CLIENT ATTACH_MEMBER  0;
SET CLIENT CONNECT_MEMBER 0;
-- *****************************************************************************
-- ** automatically created redirect restore script
-- *****************************************************************************
RESTORE DATABASE CROSS
-- USER  <username>
-- USING '<password>'
FROM '/home/backupimage'
TAKEN AT 20190429021029
-- ON '/home/inst'
-- DBPATH ON '<target-directory>'
INTO CROSS
-- NEWLOGPATH DEFAULT
-- WITH <num-buff> BUFFERS
-- BUFFER <buffer-size>
-- REPLACE HISTORY FILE
-- REPLACE EXISTING
REDIRECT
-- PARALLELISM <n>
WITHOUT ROLLING FORWARD
-- WITHOUT PROMPTING
;
-- *****************************************************************************
-- ** storage group definition
-- **   Default storage group ID                      = 0
-- **   Number of storage groups                      = 1
-- *****************************************************************************
-- *****************************************************************************
-- ** Storage group name                              = IBMSTOGROUP
-- **   Storage group ID                              = 0
-- **   Data tag                                      = None
-- *****************************************************************************
-- SET STOGROUP PATHS FOR IBMSTOGROUP
-- ON '/home/inst'
-- ;
-- *****************************************************************************
-- ** table space definition
-- *****************************************************************************
```

Example snippet of generated redirect restore script



Recovery – Host1 to Host2 (7|11)

- **Storage**
  - **Log Paths**
    - The following paths should be considered:
      - Primary (`NEWLOGPATH`)
      - Mirror (`MIRRORLOGPATH`)
      - Overflow (`OVERFLOWLOGPATH`)
      - Archiving:
        - `LOGARCHMETH1|2`
        - `FAILARCHPATH`
    - They all depend on whether restoring to an existing database or new database
      - Admittedly, more complicated than it should be ➔ BE EXPLICIT to avoid surprises
    - Only the primary log path can be changed at restore time and is validated at the start of restore:
      - `RESTORE .. NEWLOGPATH /logs`
    - The remaining paths can only be changed post-restore using `UPDATE DB CFG` (see next slides)
      - If any path accessible from both hosts, may cause conflict or failures ➔ BE EXPLICIT to avoid surprises



Recovery – Host1 to Host2 (8|11)

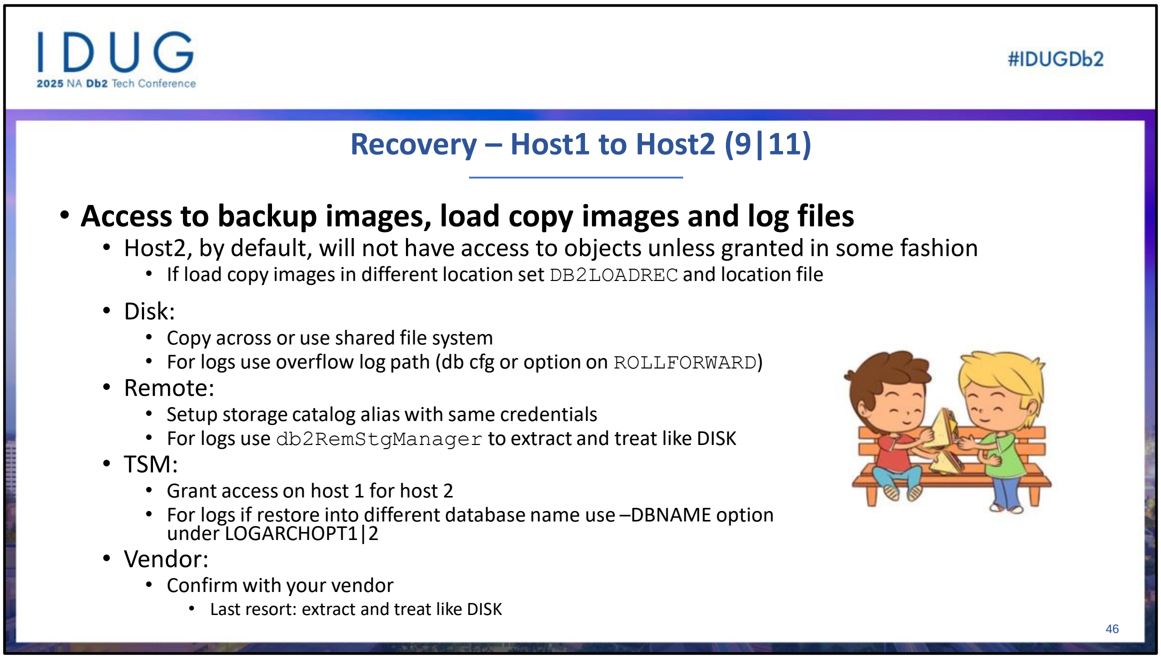
- **Recovery Point**
  - Where do you want to recover to?
    - End of backup / Point in time / End of logs
    - Continuous rollforward pending state (log shipping)?
  - Each of these can help to determine what kind of backup image to use and whether load copy images and logfiles are needed to achieve recovery point

Offline Image

- Non-recoverable database
  - Only option
- Recoverable database
  - Rollforward pending
    - Avoid with `RESTORE .. WITHOUT ROLLING FORWARD`
    - Otherwise load copy images and logs must be available

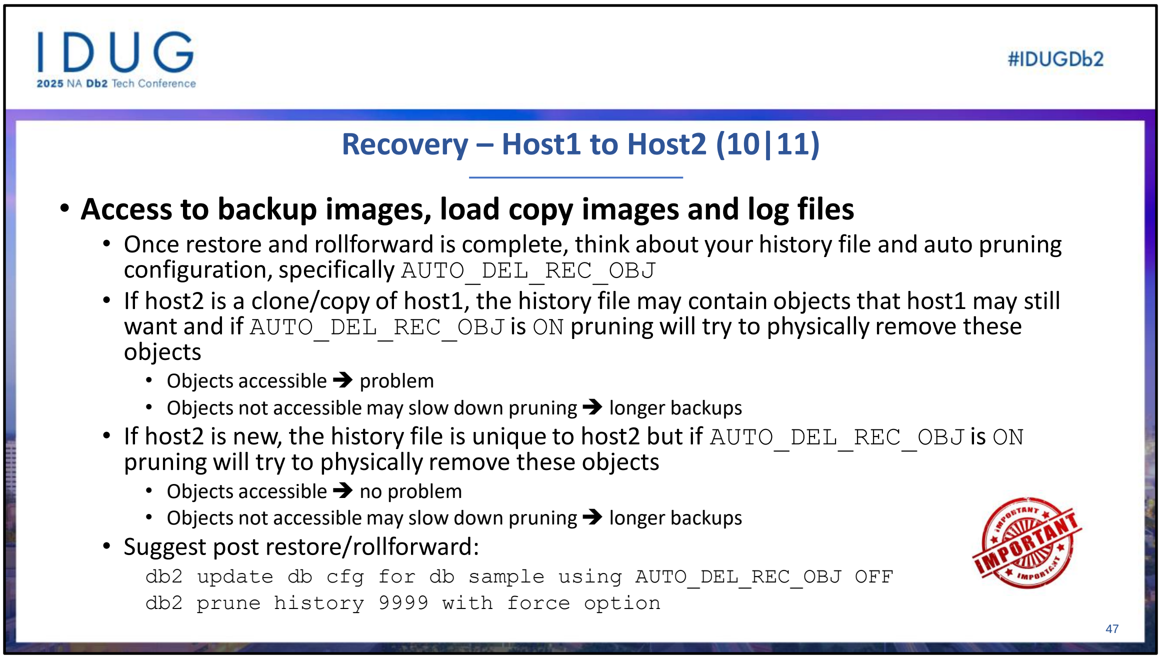
Online Image

- Recoverable database ONLY
  - Load copy images and logs must be available
  - For end of backup, logs will be in image unless excluded
    - Use `RESTORE .. LOGTARGET /logs`
    - Use those logs before relying on archives



Recovery – Host1 to Host2 (9|11)

- **Access to backup images, load copy images and log files**
  - Host2, by default, will not have access to objects unless granted in some fashion
    - If load copy images in different location set `DB2LOADREC` and location file
  - Disk:
    - Copy across or use shared file system
    - For logs use overflow log path (db cfg or option on `ROLLFORWARD`)
  - Remote:
    - Setup storage catalog alias with same credentials
    - For logs use `db2RemStgManager` to extract and treat like DISK
  - TSM:
    - Grant access on host 1 for host 2
    - For logs if restore into different database name use –DBNAME option under LOGARCHOPT1|2
  - Vendor:
    - Confirm with your vendor
      - Last resort: extract and treat like DISK

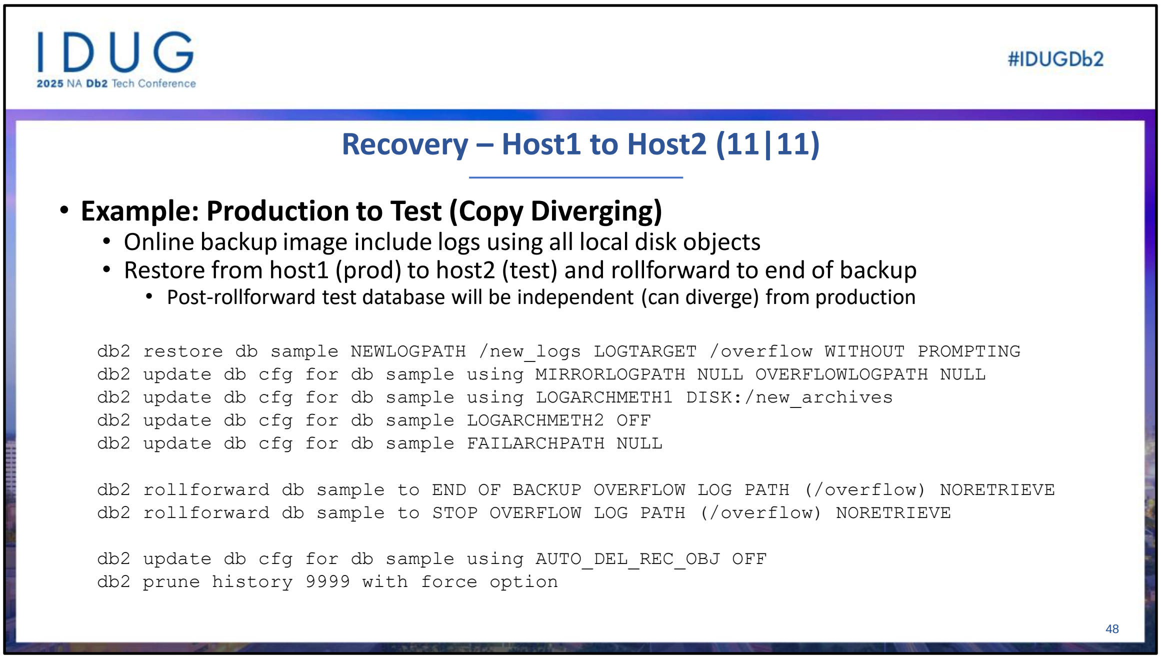


Recovery – Host1 to Host2 (10|11)

- **Access to backup images, load copy images and log files**
  - Once restore and rollforward is complete, think about your history file and auto pruning configuration, specifically `AUTO_DEL_REC_OBJ`
  - If host2 is a clone/copy of host1, the history file may contain objects that host1 may still want and if `AUTO_DEL_REC_OBJ` is `ON` pruning will try to physically remove these objects
    - Objects accessible ➔ problem
    - Objects not accessible may slow down pruning ➔ longer backups
  - If host2 is new, the history file is unique to host2 but if `AUTO_DEL_REC_OBJ` is `ON` pruning will try to physically remove these objects
    - Objects accessible ➔ no problem
    - Objects not accessible may slow down pruning ➔ longer backups
  - Suggest post restore/rollforward:

```
db2 update db cfg for db sample using AUTO_DEL_REC_OBJ OFF
db2 prune history 9999 with force option
```

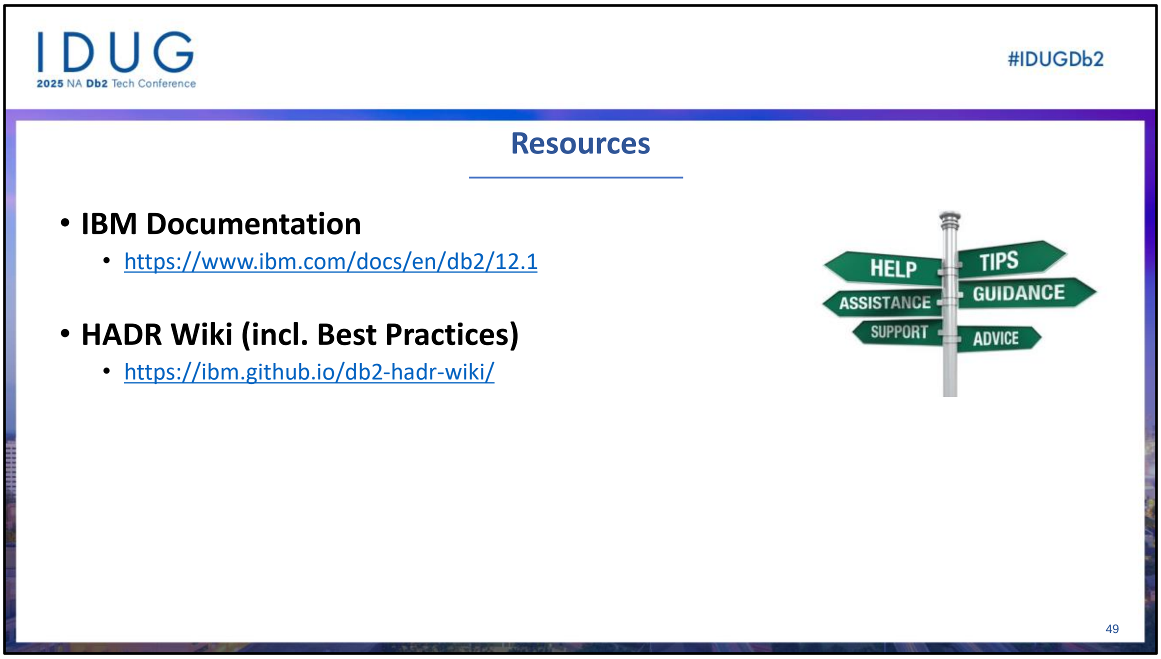
IMPORTANT



Recovery – Host1 to Host2 (11|11)

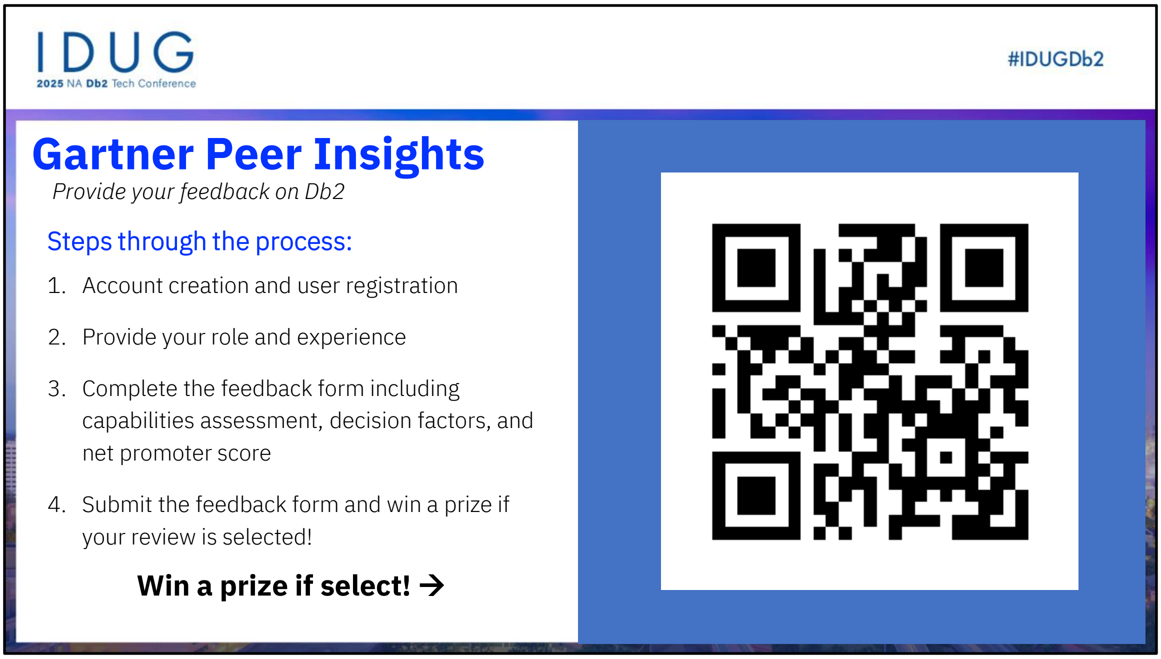
- **Example: Production to Test (Copy Diverging)**
  - Online backup image include logs using all local disk objects
  - Restore from host1 (prod) to host2 (test) and rollforward to end of backup
    - Post-rollforward test database will be independent (can diverge) from production

```
db2 restore db sample NEWLOGPATH /new_logs LOGTARGET /overflow WITHOUT PROMPTING
db2 update db cfg for db sample using MIRRORLOGPATH NULL OVERFLOWLOGPATH NULL
db2 update db cfg for db sample using LOGARCHMETH1 DISK:/new_archives
db2 update db cfg for db sample LOGARCHMETH2 OFF
db2 update db cfg for db sample FAILARCHPATH NULL

db2 rollforward db sample to END OF BACKUP OVERFLOW LOG PATH (/overflow) NORETRIEVE
db2 rollforward db sample to STOP OVERFLOW LOG PATH (/overflow) NORETRIEVE

db2 update db cfg for db sample using AUTO_DEL_REC_OBJ OFF
db2 prune history 9999 with force option
```



Resources

- **IBM Documentation**
  - https://www.ibm.com/docs/en/db2/12.1
- **HADR Wiki (incl. Best Practices)**
  - https://ibm.github.io/db2-hadr-wiki/



IDUG
2025 NA **Db2** Tech Conference

#IDUGDb2

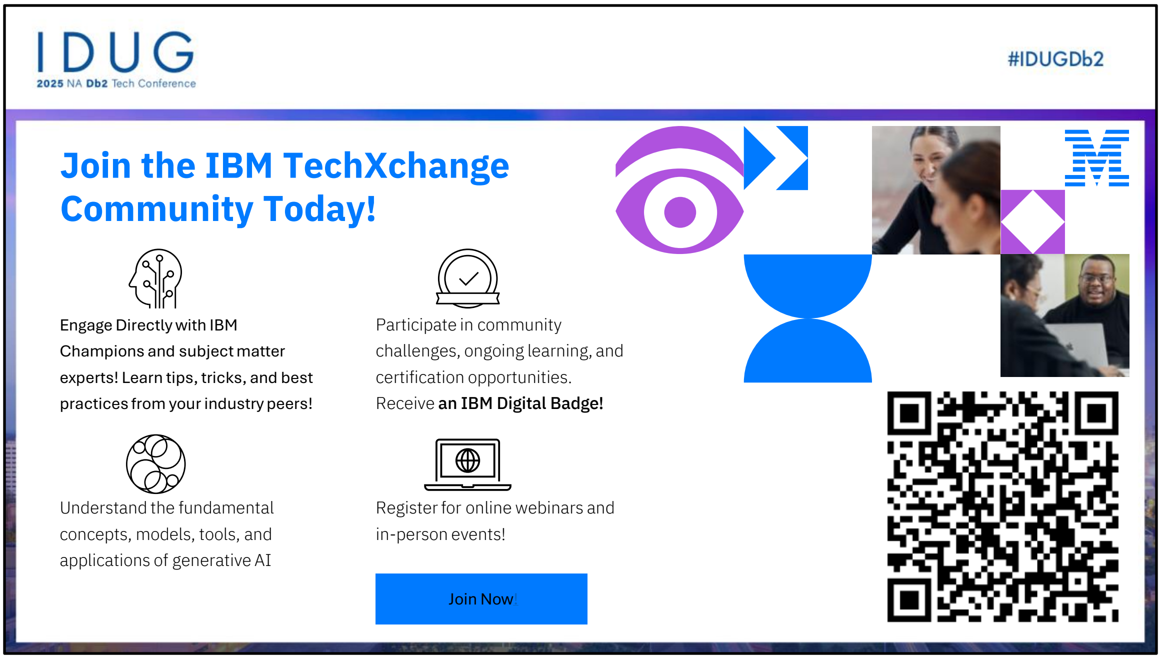
Gartner Peer Insights

Provide your feedback on Db2

Steps through the process:

1. Account creation and user registration
2. Provide your role and experience
3. Complete the feedback form including capabilities assessment, decision factors, and net promoter score
4. Submit the feedback form and win a prize if your review is selected!

Win a prize if select! →



IDUG 2025 NA Db2 Tech Conference

#IDUGDb2

Join the IBM TechXchange Community Today!

Engage Directly with IBM Champions and subject matter experts! Learn tips, tricks, and best practices from your industry peers!

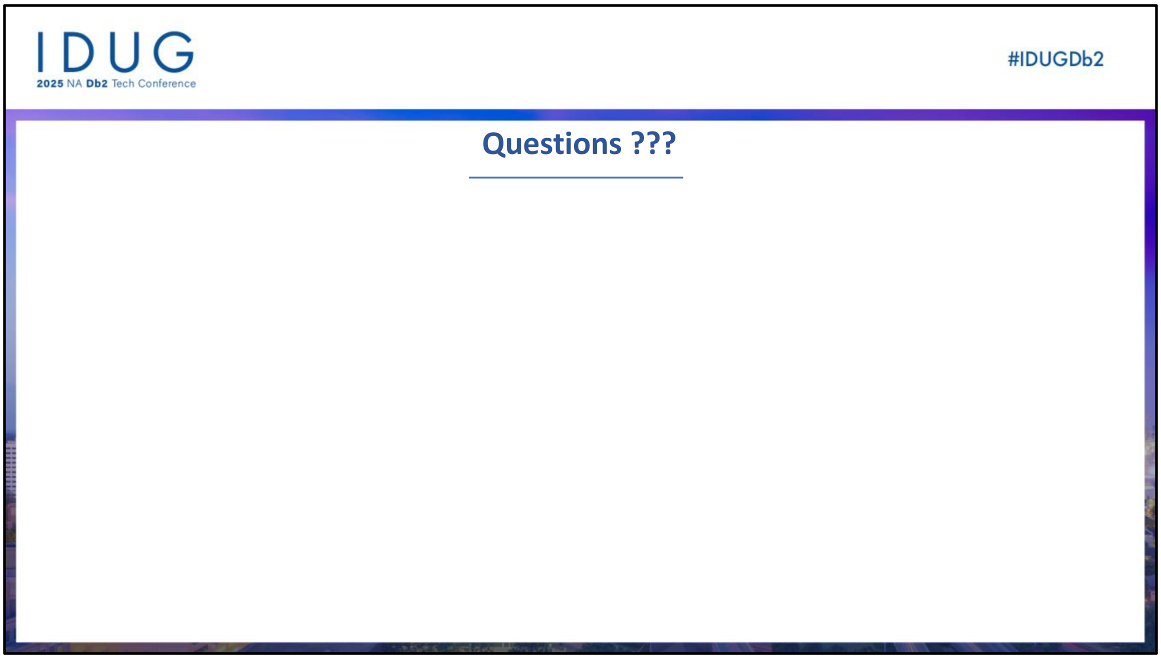
Participate in community challenges, ongoing learning, and certification opportunities. Receive **an IBM Digital Badge!**

Understand the fundamental concepts, models, tools, and applications of generative AI

Register for online webinars and in-person events!

Join Now

Webinars are bi-weekly



IDUG
2025 NA Db2 Tech Conference

#IDUGDb2

Questions ???

IDUG

2025

Atlanta, GA | June 8-12

NA Db2 TECH CONFERENCE

Unlocking Recovery - The Key to Solving Everyday Problems

Michael Roecken, *IBM*

Contact: 𝕏 @roecken

Session Code: A15

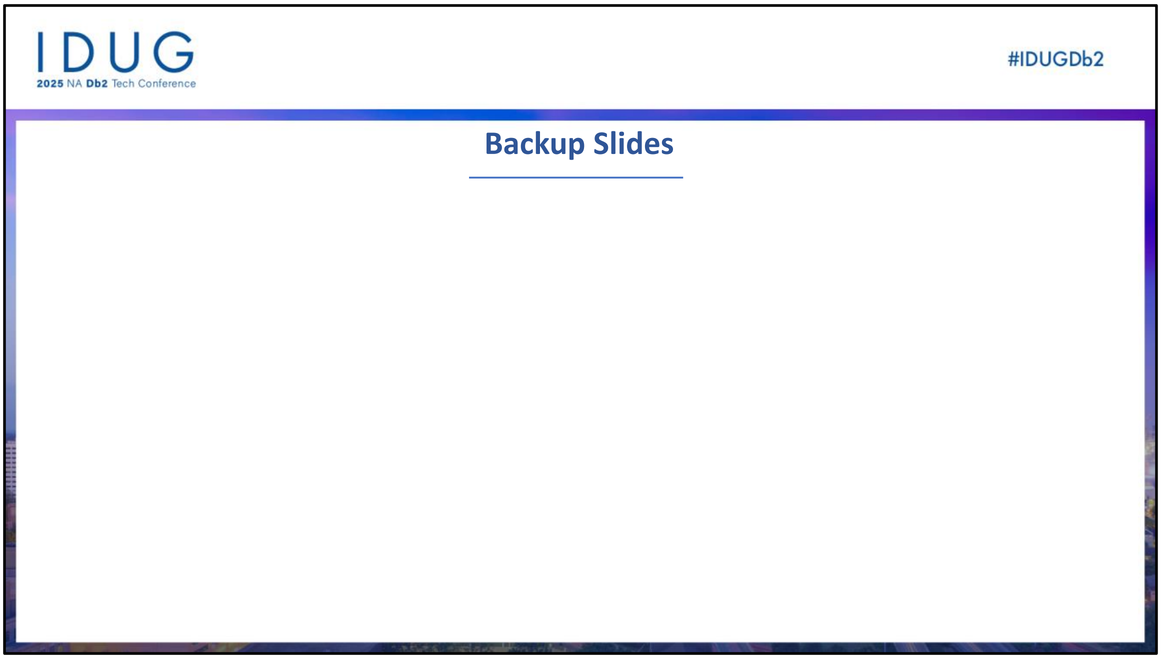
Please fill out your session evaluation

Platform:

LUW

IDUG

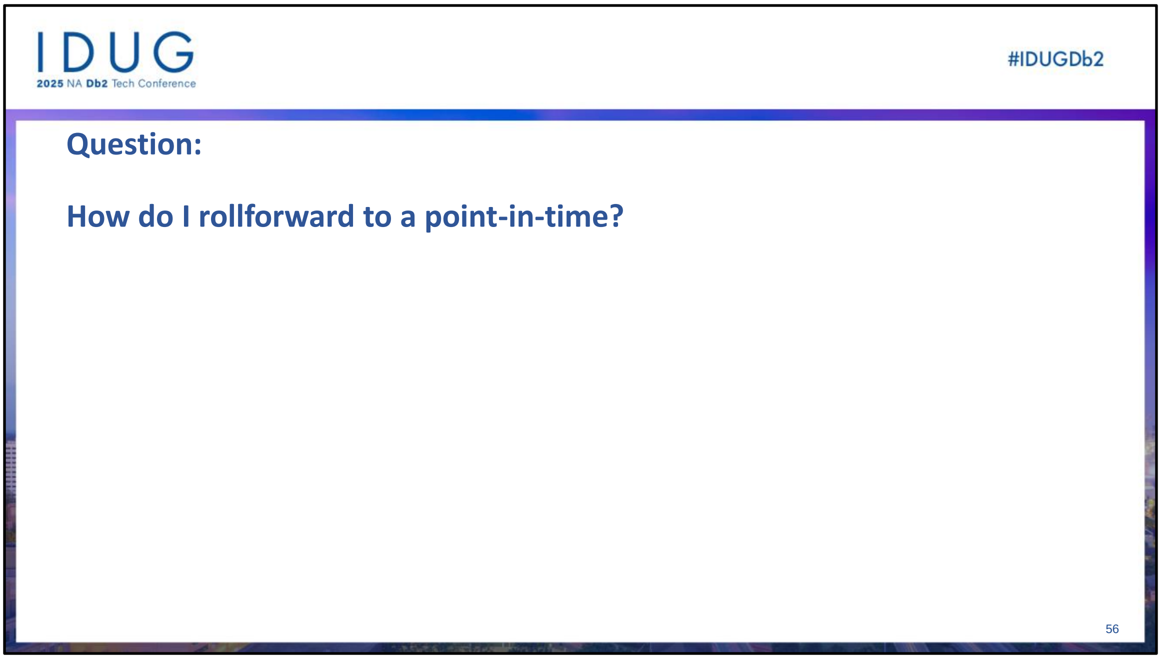
2025 NA **Db2** Tech Conference



IDUG
2025 NA Db2 Tech Conference

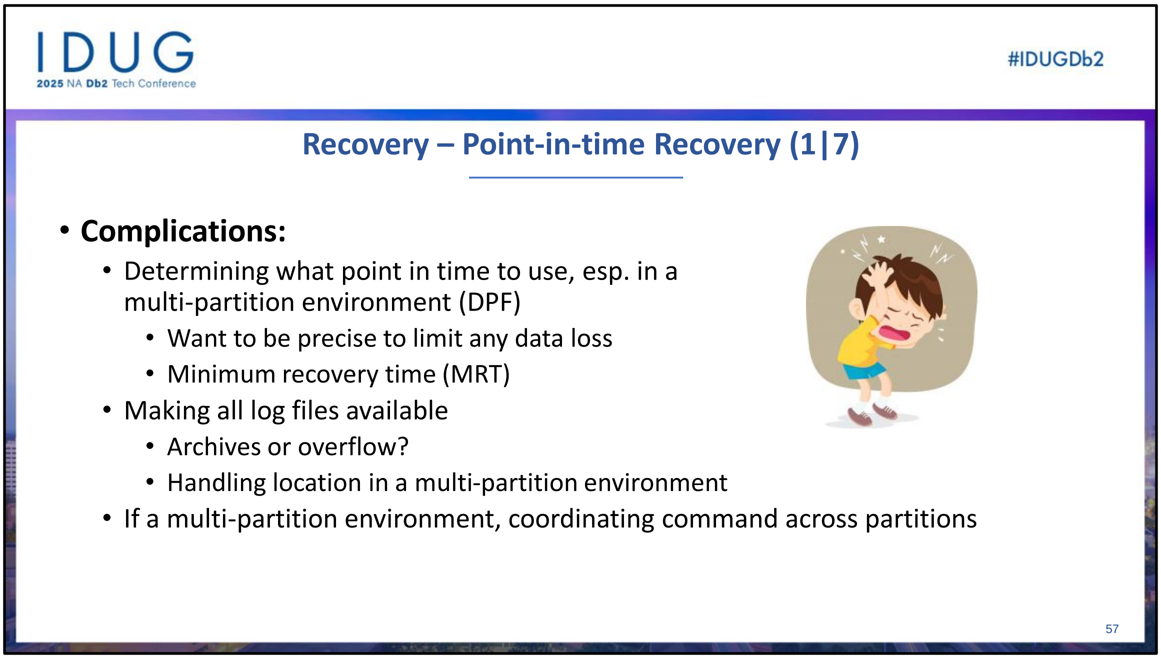
#IDUGDb2

Backup Slides



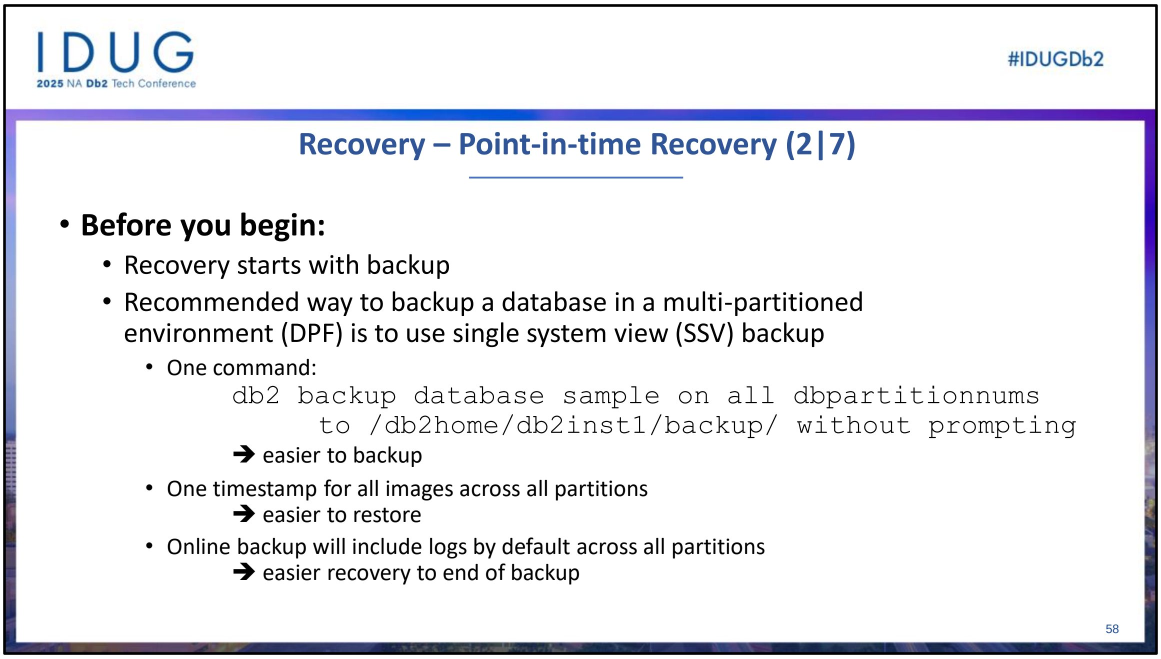
Question:

How do I rollforward to a point-in-time?



Recovery – Point-in-time Recovery (1|7)

- **Complications:**
  - Determining what point in time to use, esp. in a multi-partition environment (DPF)
    - Want to be precise to limit any data loss
    - Minimum recovery time (MRT)
  - Making all log files available
    - Archives or overflow?
    - Handling location in a multi-partition environment
  - If a multi-partition environment, coordinating command across partitions

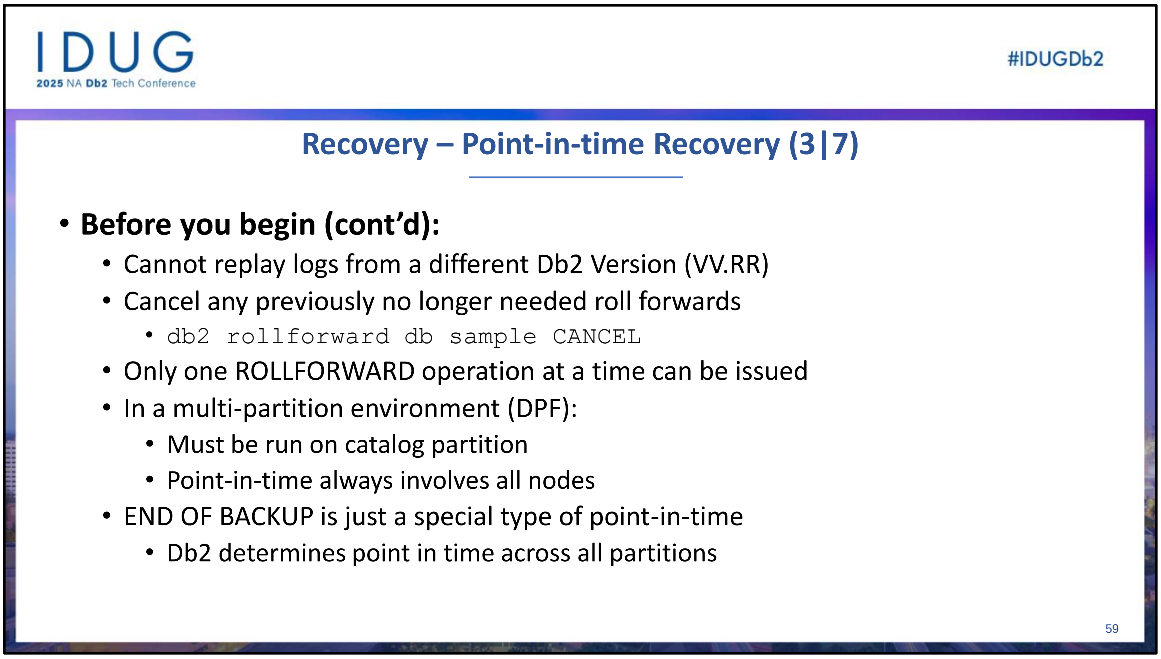


Recovery – Point-in-time Recovery (2|7)

- **Before you begin:**
  - Recovery starts with backup
  - Recommended way to backup a database in a multi-partitioned environment (DPF) is to use single system view (SSV) backup
    - One command:

```
db2 backup database sample on all dbpartitionnums
        to /db2home/db2inst1/backup/ without prompting
```

      ➔ easier to backup
    - One timestamp for all images across all partitions
      ➔ easier to restore
    - Online backup will include logs by default across all partitions
      ➔ easier recovery to end of backup

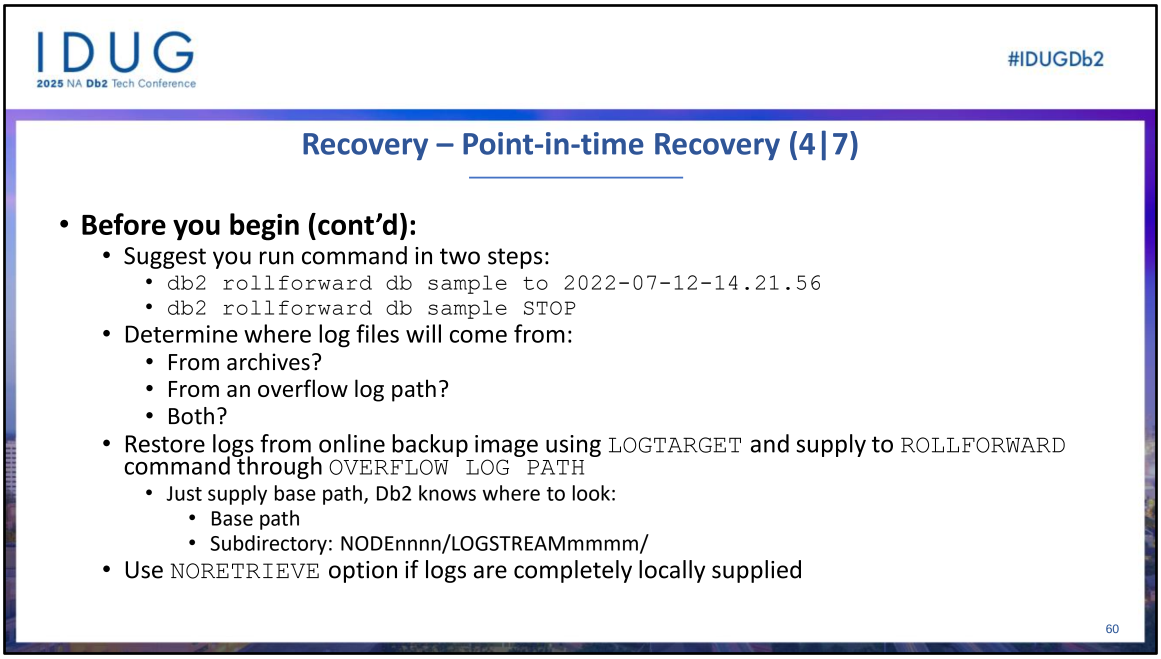


Recovery – Point-in-time Recovery (3|7)

- **Before you begin (cont'd):**
  - Cannot replay logs from a different Db2 Version (VV.RR)
  - Cancel any previously no longer needed roll forwards
    - `db2 rollforward db sample CANCEL`
  - Only one ROLLFORWARD operation at a time can be issued
  - In a multi-partition environment (DPF):
    - Must be run on catalog partition
    - Point-in-time always involves all nodes
  - END OF BACKUP is just a special type of point-in-time
    - Db2 determines point in time across all partitions

ROLLFORWARD DATABASE command

- https://www.ibm.com/docs/en/db2/12.1?topic=commands-rollforward-database



IDUG 2025 NA Db2 Tech Conference

#IDUGDb2

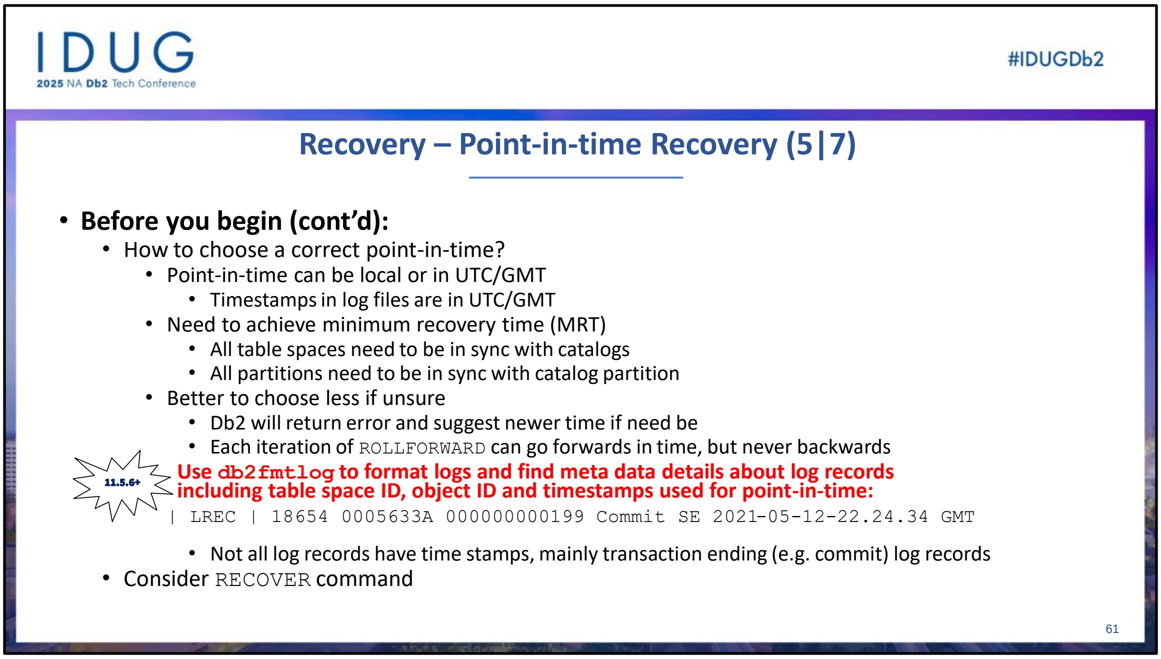
Recovery – Point-in-time Recovery (4|7)

- **Before you begin (cont'd):**
  - Suggest you run command in two steps:
    - `db2 rollforward db sample to 2022-07-12-14.21.56`
    - `db2 rollforward db sample STOP`
  - Determine where log files will come from:
    - From archives?
    - From an overflow log path?
    - Both?
  - Restore logs from online backup image using `LOGTARGET` and supply to `ROLLFORWARD` command through `OVERFLOW LOG PATH`
    - Just supply base path, Db2 knows where to look:
      - Base path
      - Subdirectory: NODEnnnn/LOGSTREAMmmmm/
  - Use `NORETRIEVE` option if logs are completely locally supplied

60

ROLLFORWARD DATABASE command

- https://www.ibm.com/docs/en/db2/12.1?topic=commands-rollforward-database



IDUG
2025 NA Db2 Tech Conference

#IDUGDb2

Recovery – Point-in-time Recovery (5|7)

- **Before you begin (cont'd):**
  - How to choose a correct point-in-time?
    - Point-in-time can be local or in UTC/GMT
      - Timestamps in log files are in UTC/GMT
    - Need to achieve minimum recovery time (MRT)
      - All table spaces need to be in sync with catalogs
      - All partitions need to be in sync with catalog partition
    - Better to choose less if unsure
      - Db2 will return error and suggest newer time if need be
      - Each iteration of `ROLLFORWARD` can go forwards in time, but never backwards

11.5.6+

Use `db2fmtlog` to format logs and find meta data details about log records including table space ID, object ID and timestamps used for point-in-time:

```
| LREC | 18654 0005633A 000000000199 Commit SE 2021-05-12-22.24.34 GMT
```

- Not all log records have time stamps, mainly transaction ending (e.g. commit) log records
- Consider `RECOVER` command

ROLLFORWARD DATABASE command

- https://www.ibm.com/docs/en/db2/12.1?topic=commands-rollforward-database

How to determine the minimum recovery time for a rollforward operation

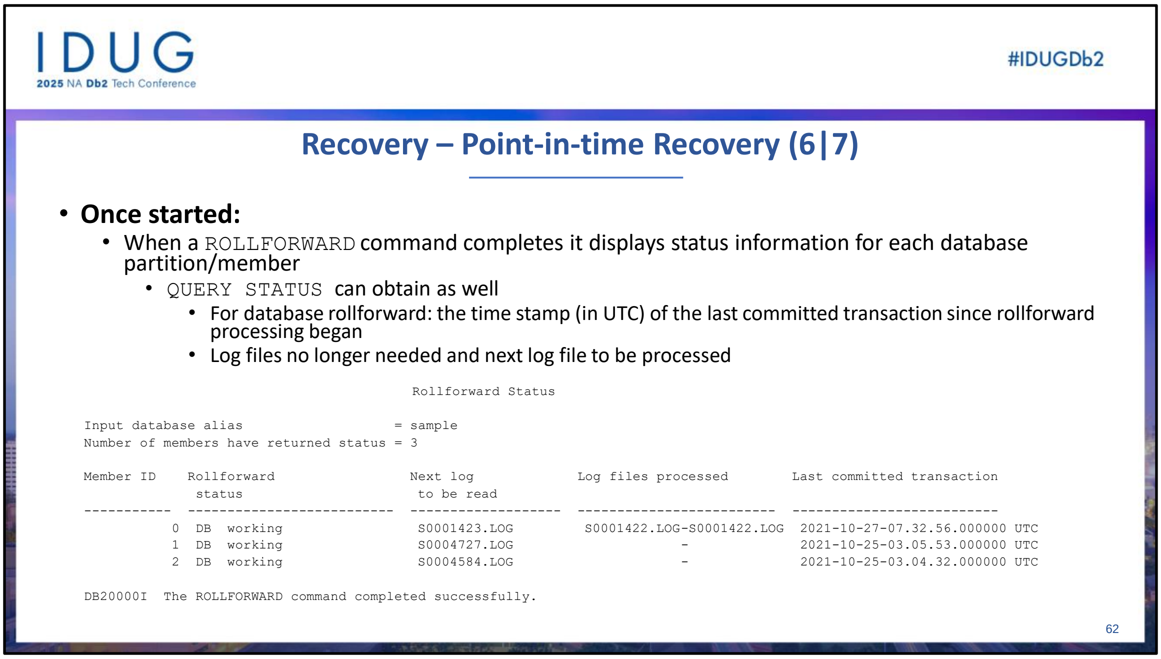
- https://www.ibm.com/support/pages/how-determine-minimum-recovery-time-rollforward-operation

db2fmtlog – Format and display log file information command

- https://www.ibm.com/docs/en/db2/12.1?topic=commands-db2fmtlog-format-display-log-file-information-tool

RECOVER DATABASE command

- https://www.ibm.com/docs/en/db2/12.1?topic=commands-recover-database



IDUG
2025 NA Db2 Tech Conference
#IDUGDb2

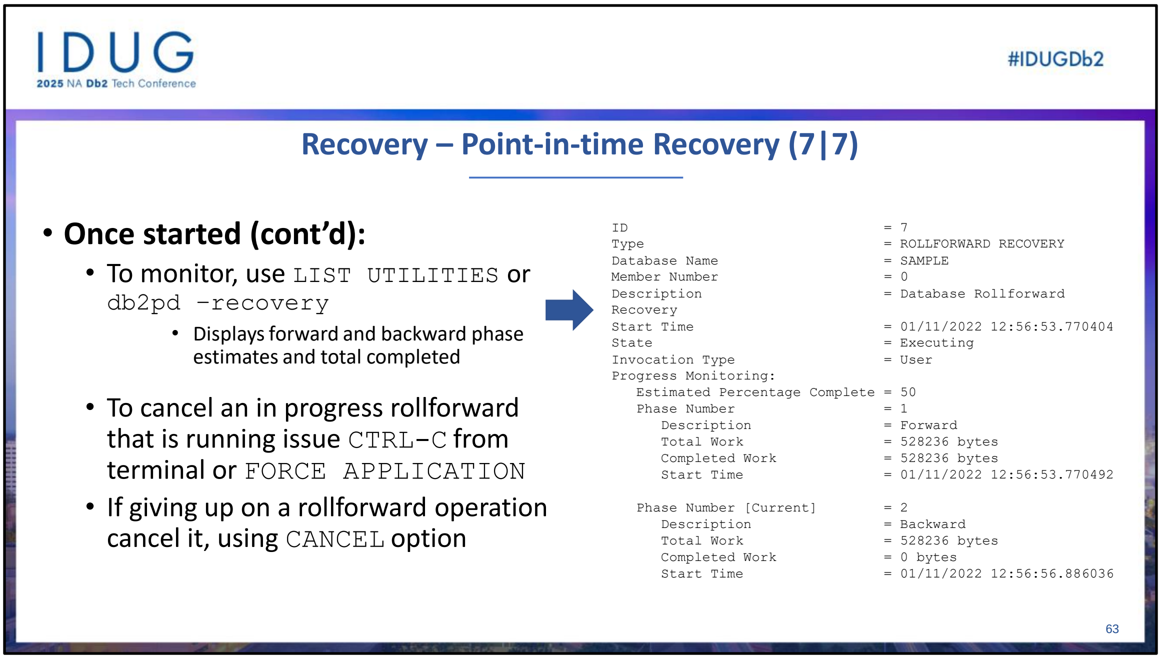
Recovery – Point-in-time Recovery (6|7)

- **Once started:**
  - When a `ROLLFORWARD` command completes it displays status information for each database partition/member
    - `QUERY STATUS` can obtain as well
      - For database rollforward: the time stamp (in UTC) of the last committed transaction since rollforward processing began
      - Log files no longer needed and next log file to be processed

```
                                         Rollforward Status

Input database alias                   = sample
Number of members have returned status = 3

Member ID    Rollforward                 Next log             Log files processed        Last committed transaction
              status                      to be read
-----------  --------------------------  -------------------  -------------------------  --------------------------
           0  DB  working                 S0001423.LOG         S0001422.LOG-S0001422.LOG  2021-10-27-07.32.56.000000 UTC
           1  DB  working                 S0004727.LOG                      -             2021-10-25-03.05.53.000000 UTC
           2  DB  working                 S0004584.LOG                      -             2021-10-25-03.04.32.000000 UTC

DB20000I  The ROLLFORWARD command completed successfully.
```

ROLLFORWARD DATABASE command

- https://www.ibm.com/docs/en/db2/12.1?topic=commands-rollforward-database



Recovery – Point-in-time Recovery (7|7)

- **Once started (cont'd):**
  - To monitor, use `LIST UTILITIES` or `db2pd -recovery`
    - Displays forward and backward phase estimates and total completed
  - To cancel an in progress rollforward that is running issue `CTRL-C` from terminal or `FORCE APPLICATION`
  - If giving up on a rollforward operation cancel it, using `CANCEL` option

```
ID                                  = 7
Type                                = ROLLFORWARD RECOVERY
Database Name                       = SAMPLE
Member Number                       = 0
Description                         = Database Rollforward
Recovery
Start Time                          = 01/11/2022 12:56:53.770404
State                               = Executing
Invocation Type                     = User
Progress Monitoring:
   Estimated Percentage Complete = 50
   Phase Number                     = 1
      Description                   = Forward
      Total Work                    = 528236 bytes
      Completed Work                = 528236 bytes
      Start Time                    = 01/11/2022 12:56:53.770492

   Phase Number [Current]           = 2
      Description                   = Backward
      Total Work                    = 528236 bytes
      Completed Work                = 0 bytes
      Start Time                    = 01/11/2022 12:56:56.886036
```

ROLLFORWARD DATABASE command

- https://www.ibm.com/docs/en/db2/12.1?topic=commands-rollforward-database