DB2 10 Availability Enhancements

Haakon Roberts
IBM
haakon@us.ibm.com

Session Code: A08

March 12 2010 2:45pm
Platform: z/OS

Disclaimer/Trademarks

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

The information on the new product is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information on the new product is for informational purposes only and may not be incorporated into any contract. The information on the new product is not a commitment, promise, or legal obligation to deliver any material, code or functionality. The development, release, and timing of any features or functionality described for our products remains at our sole discretion. *

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the pages of the presentation:

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both: AIX, AS/400, DataJoiner, DataPropagator, DB2, DB2 Connect, DB2 Extenders, DB2 OLAP Server, DB2 Universal Database, Distributed Relational Database Architecture, DRDA, eServer, IBM, IMS, iSeries, MVS, Net.Data, OS/390, OS/400, PowerPC, pSeries, RS/6000, SQL/400, SQL/DS, Tivoli, VisualAge, VM/ESA, VSE/ESA, WebSphere, z/OS, zSeries

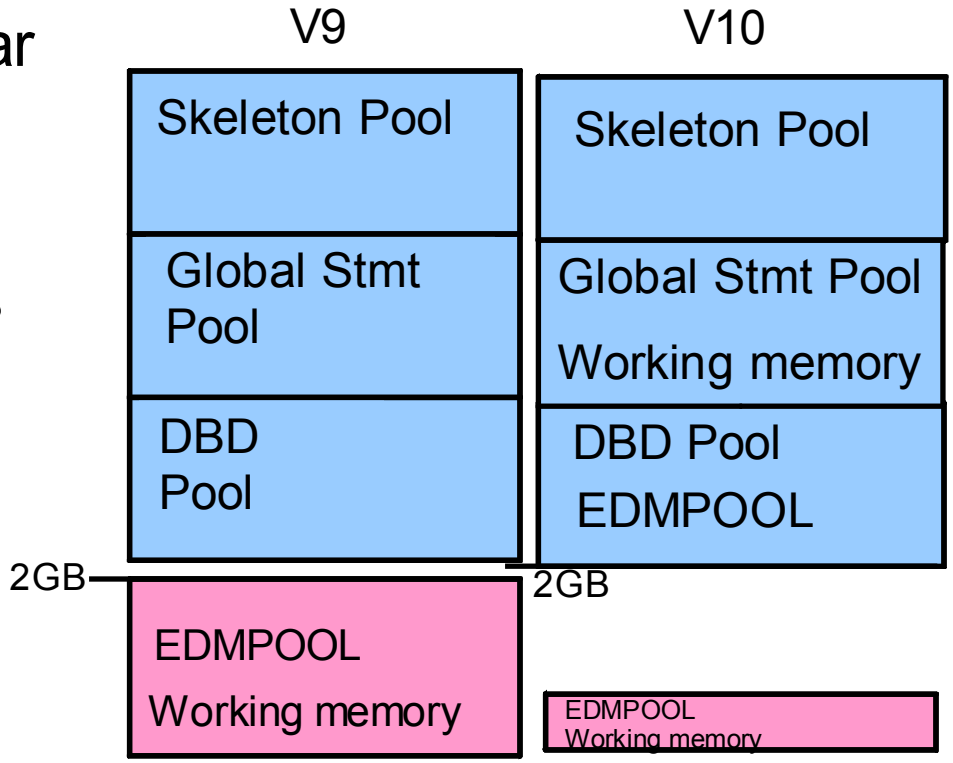
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Other company, product, or service names may be trademarks or service marks of others.

Agenda

- Virtual Storage & Scalability
- Online Schema Evolution
- Access Currently Committed Data
- REORG INDEX avoidance
- REORG Enhancements
- Backup/Recovery Enhancements
- Logging Enhancements
- Summary

DB2 10: 64 bit Evolution Virtual Storage Relief

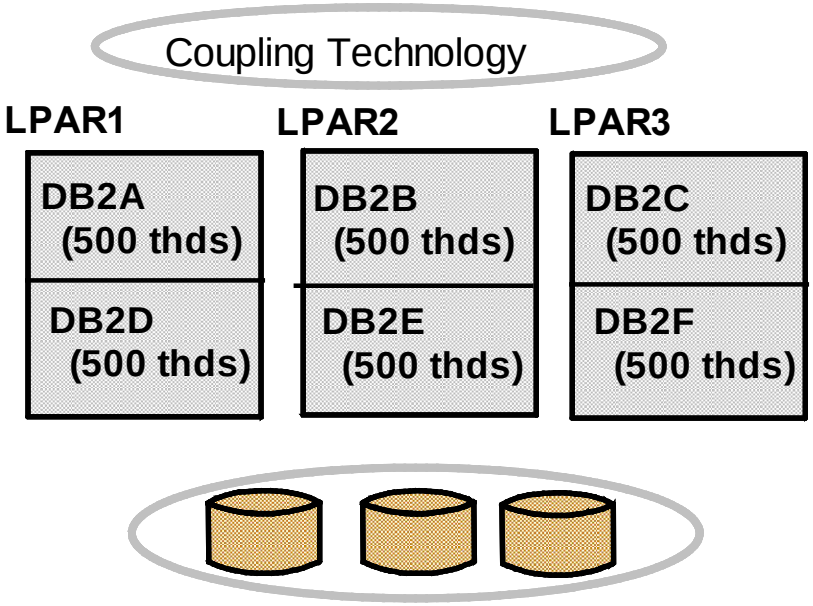
- DB2 9 helped (~ 10% – 15%)
- DB2 10: 5 to 10 times more threads, up to 20,000
  - Move 80% - 90% above bar
  - More concurrent work
  - Reduce need to monitor
  - Able to consolidate LPARs
  - Reduced cost
  - Easier to manage
  - Easier to grow



Scalability: Virtual storage constraint is still an important issue for many DB2 customers in 9.

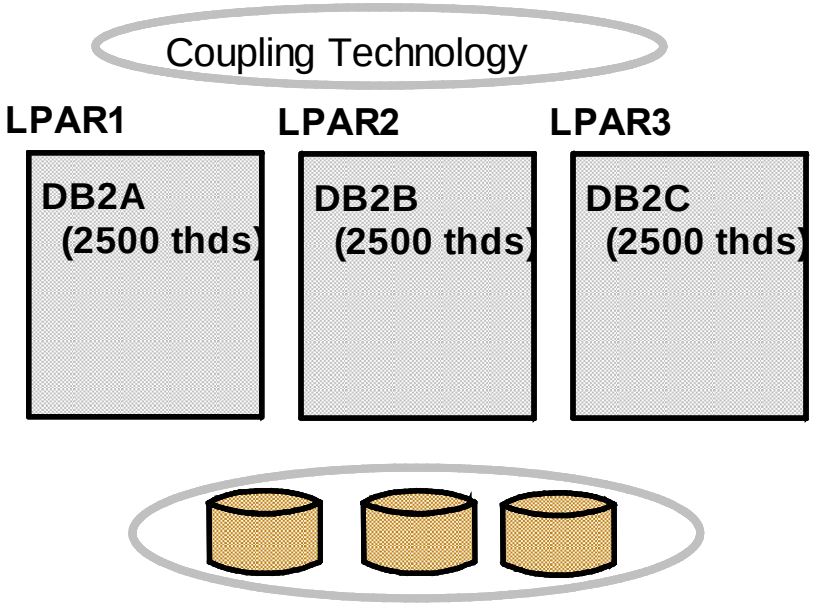
Running a Large Number of Active Threads

Today



- Data sharing and sysplex allows for efficient scale-out of DB2 images
- Sometimes multiple DB2s per LPAR

DB2 10



- More threads per DB2 image
- More efficient use of large n-ways
- Easier growth, lower costs, easier management
- Data sharing and Parallel Sysplex still required for very high availability and scale
- Rule of thumb: save ½% CPU for each member reduced, more on memory

Other System Scaling Improvements

- Other bottlenecks can emerge in extremely heavy workloads
  - Several improvements to reduce latching and other system serialization contention
  - New option to for readers to avoid waiting for inserters
  - Eliminate UTSERIAL lock contention for utilities
  - Use 64-bit common storage to avoid ECSA constraints
- Concurrent DDL/BIND/Prepare processes can contend with one another
  - Restructure parts of DB2 catalog to avoid the contention
- SPT01 64GB limit can be a constraint, especially if package stability is enabled
  - Allow many more packages by using LOBs

Online Schema – V9

- Change of table - or index space attributes can require an outage
- Change of table space attributes
  - Unload data
  - Drop table space
  - Recreate table space, tables, indexes, views
  - Re-establish authorization & RI
  - Reload data
- Change of index space attributes
  - Alter index
    - Index placed in RBDP
  - Rebuild index
- Undo of DDL changes
  - Same as above

Online Schema – V10

- Execute ALTER statement
- Changes are cached & materialized by next REORG
  - SHRLEVEL REFERENCE|CHANGE
- Undo of DDL changes if not materialized
  - ALTER TABLESPACE… DROP PENDING CHANGES
  - All pending changes are removed
- Undo of DDL changes if materialized
  - Perform compensating ALTER & schedule REORG
    - Assumes no dependencies on prior ALTER have evolved

Online Schema - What Attributes are ALTERable?

- ALTER TABLESPACE
  - Page size (not XML) (BUFFERPOOL)
  - DSSIZE
  - SEGSIZE
  - Table space type
    - Single table simple -> PBG (inherit MC)
    - Single table segmented -> PBG
    - Classic partitioned -> PBR (inherit MC)
  - MEMBER CLUSTER
- ALTER INDEX
  - Page size (BUFFERPOOL)
    - In V9 this was immediate with RBDP set

ALTER TABLESPACE … MAXPARTITIONS m

ALTER TABLESPACE … SEGSIZE s

Online Schema – Details on Execute ALTER Statement

- Statement is validated
  - Semantic checking against effective catalog definition
- Assuming all checks out ok:
  - Statement is put on pending list
  - Table space is placed in AREOR (non-restrictive)
  - Statement completes with SQLCODE +610 to advertise the advisory state

SYSIBM.SYSPENDINGDDL:

DBNAME	TSNAME	DBID	PSID	OBJSCHEMA	OBJNAME	...	OPTION_ KEYWORD	OPTION_ VALUE	...	STATEM ENT_TE XT

Online Schema – Details on REORG

- Pending DDL is materialized
  - Catalog is updated with the new attributes
  - OBD is updated with the new attributes
  - Datasets are updated with the new attributes
  - Materialized SYSPENDINGDDL entries are removed
- Stats are collected
  - Default is TABLE ALL INDEX ALL UPDATE ALL HISTORY ALL unless overridden
  - Warning message is issued to indicate that some partition statistics may no longer be accurate
    (COLGROUP, KEYCARD, HISTOGRAM …)
- SYSCOPY entries are created to record pending-DDL materialization
- AREOR state is reset
- Dependent plans and packages are invalidated if table space type conversion occurs

Online Schema – SQL Restrictions

- Not permitted to mix immediate and deferred options in an ALTER statement (SQLCODE -20385)
- Many immediate DDL statements are not allowed while there is pending DDL awaiting materialization (-20385)
  - CREATE/DROP/ALTER
    - E.g. alter of FREEPAGE for a partition
- ALTERing table space type supports only single-table table spaces
- Most deferred ALTERs other than changing table space type are supported only for UTS
  - Alter of page size supported for LOBs

Online Schema – Utility Restrictions

- Pending DDL only materialized by REORG at table space level
- Pending DDL only materialized by REORG SHRLEVEL REFERENCE or CHANGE
  - REORG SHRLEVEL(NONE) and part-level REORGs are not blocked, but do not materialize pending DDL
- Restrict RECOVER across or prior to materializing REORGs
  - REPORT RECOVERY will identify image copies taken before REORG materialization as ineligible for recovery use (# brackets).
  - UNLOAD from these image copies is permitted

Online Schema - Optimizations

- Undefined table or index spaces
  - ALTERs take immediate effect
- ALTER BUFFERPOOL (no pagesize change)
  - ALTERs take immediate effect
  - Unless other pending operations exist

Currently Committed

- Requirement
  - Read applications acquire locks on data
  - Overhead, contention, concurrency issues
  - ISO(UR) avoids contention, but does not return committed data
  - Ported applications particularly prone to timeouts
- Currently committed
  - Return currently committed data without waiting for locks
  - Supported for uncommitted inserts or deletes
  - No support in 10 for uncommitted updates

Currently Committed - Syntax

- New BIND Option
  - CONCURRENTACCESSRESOLUTION(USECURRENTLYCOMMITTED | WAITFOROUTCOME)
- New PREPARE Attribute
  - PREPARE … USE CURRENTLY COMMITTED | WAIT FOR OUTCOME
- New bind option in CREATE/ALTER of PROCEDURE, FUNCTION
  - CONCURRENT ACCESS RESOLUTION U[SE CURRENTLY COMMITTED] / W[AIT FOR OUTCOME]
- Defaults are today's "wait for outcome" behavior

Currently Committed

- Currently Committed semantic applicable to UTS on V10 NFM
- If contention is with uncommitted insert then CC applies to ISO(CS) or ISO(RS)
- If contention is with uncommitted delete then CC applies only to ISO (CS) with CURRENTDATA(NO)
- Statement level overrides package/plan level which overrides system level
- Row and page locking is supported
- Not applicable to table, partition or table space locks
  - Not applicable when LOCK TABLE IN EXCLUSIVE used
  - Not applicable when lock holder is performing mass delete
  - Not applicable if lock holder has escalated

Currently Committed

- Simple scenario
  - Reader encounters row
  - Standard lock avoidance fails – row is possibly uncommitted
  - Request conditional lock on row
  - If lock not available & held by inserter – skip row
  - If lock not available & held by deleter – return row
- Deleted rows are now pseudo-deleted
  - No loss in space reuse: space available after deleter commits
  - No need to log entire record on delete
- Update not supported in V10

Currently Committed

- Currently committed allows committed data to be returned without waiting
- BUT – does not guarantee that DB2 will do so
- In some cases DB2 may revert to unconditional locking
- Consider currently committed as 2nd generation lock avoidance
- Instrumentation
  - New counter **QISTRCCI** added to the Data Manager Statistics block DSNDQIST (IFCID 002), to show the number of rows skipped by read transactions using currently committed semantic encountering uncommitted inserts.
  - New counter **QISTRCCD** added to the Data Manager Statistics block DSNDQIST (IFCID 002), to show the number of rows accessed by read transactions using currently committed semantic encountering uncommitted deletes.

REORG INDEX Avoidance

- Ability to list prefetch index leaf pages based on index non-leaf information for range scans
  - May greatly reduce sync I/O waits for queries using disorganized indexes
  - REORG INDEX, CHECK INDEX, RUNSTATS expected to benefit
- Improved caching of non-leaf pages
  - Reduce getpages for root page
- Enable sequential detection & index look-aside for parent key lookup on RI insert
- New IFCID359 to track leaf page splits
- All available in V10 CM

REORG Limitations prior to V10

- REORG of base cannot move rows between partitions if LOB columns exist
  - No move between PBG parts
  - No alter of limitkey
  - No REORG REBALANCE
- REORG DISCARD orphans LOBs
- REORG does not support multiple part ranges
  - LISTDEF does not support part ranges
- No REORG SHRLEVEL CHANGE for LOBs
- Need to reduce outage duration for online REORG

REORG Enhancements

- Introduce new AUX keyword for REORG
  - UTS or classic partitioned
  - Allows movement of base rows by REORG even though LOB columns exist
    - Essential for PBG
  - Allows REBALANCE even though LOB columns exist
  - Would allow pruning of PBGs even though LOB columns exist…
  - Allows DISCARD to delete associated LOB values
  - Default is AUX NO unless:
    - Multi-part REORG of PBG with LOB columns
    - REBALANCE of PBR/classic partitioned with LOB columns
    - REORG of PBR/classic partitioned with multiple parts in REORP
  - No mapping table change
  - Restrictions
    - No XML column support

REORG Enhancements

- REORG & LISTDEF support for multiple part ranges
  - REORG TABLESPACE… PART 1,23:48,596,3042:3800
  - Retrofit REORG support to V9 in PK87762
- Allow REORG to cancel threads
  - Option to cancel all or just read claimers to ensure drain succeeds
    - FORCE(NO|READERS|ALL)
- Support REORG SHRLEVEL REFERENCE or CHANGE if REORP
  - Previously SHRLEVEL NONE was only option after alter of limitkey
  - Provides restartability
- Support REORG SHRLEVEL CHANGE for REBALANCE
- Reduce outage by updating inline stats after drain released in UTILTERM

REORG Enhancements

- REORG SHRLEVEL CHANGE for LOB page sets
  - No mapping table required
  - No access to base table, but not permitted if base is NOT LOGGED
- REORG SHRLEVEL NONE for LOBs deprecated in V10 NFM
  - Will run with RC0 but with a message saying nothing done
  - Convert LOB REORGs to SHRLEVEL REFERENCE
    - (or CHANGE in NFM)

Backup/Recovery Enhancements

- Improve COPY/RECOVER performance & reduce overhead
- Faster PIT recovery
- Allow creation of consistent copies with no outage
- Improved CHANGELIMIT processing
- Improved incremental image copy processing

Flashcopy Support

- Dataset level FlashCopy for utilities
  - COPY
  - REORG inline copy
  - LOAD inline copy
  - FlashCopy of indexes for LOAD, REORG, REORG INDEX, REBUILD INDEX
- Can combine with sequential copy if required
- ZPARMs for global settings & utility parms for local settings
- FlashCopy backups can be used as input to:
  - RECOVER
  - COPYTOCOPY
    - Create sequential copies from FlashCopy
  - DSN1COPY, DSN1PRNT
    - Remove performance issue with DSN1COPY of inline copies
  - Cannot unload from FlashCopy
    - Use COPYTOCOPY and unload from that

Flashcopy Support

- REORG, REBUILD, LOAD SHRLEVEL NONE always produce consistent copies
- COPY, LOAD SHRLEVEL CHANGE produce consistent copies if FLASHCOPY CONSISTENT specified
  - Copy made consistent by backing out uncommitted updates against copy as shadow
- Flashcopies are dataset level but may be copied to single dataset to create sequential copy

COPY

- COPY CHANGELIMIT
  - Delay allocating output dataset until CHANGELIMIT checked
  - &ICTYPE in template will no longer be a “C”, instead will reflect the correct type of image copy
  - Use RTS to decide between incremental or full
- Incremental copies
  - Delay allocating output dataset until pages to be copied are found
  - Insert dummy SYSCOPY record to register empty IIC

PIT Recovery

- New BACKOUT option on RECOVER
  - Roll back on log from current point instead of restoring recovery base and rolling forward
  - Works with PIT consistency, so changes prior to logpoint may be backed out
  - Can only be done once for a given log range

Logging Enhancements

- Provide ability to checkpoint based on both time and number of log records
  - Meaning of CHKFREQ is unchanged
    - Minimum # of log records raised from 200 to 1000
  - New ZPARMs to control new behavior
    - CHKLOGR – number of log records between checkpoints
      - 1000 – 99,999,999
    - CHKMINS – number of minutes between checkpoints
      - 1-1439
    - CHKTYPE SINGLE|BOTH – govern old/new
  - Set by dynamic ZPARM or –SET LOG command
    - -SET LOG change does not persist across restart
  - -DIS LOG command indicates settings and if mode is SINGLE or BOTH

Logging Enhancements

- Dynamic add of active logs
  - New –SET LOG NEWLOG option
  - New active log must be IDCAMS defined & preformatted by DSNJLOGF
  - Only a single log dataset at a time
    - Issue command twice for dual logging
  - Limit is still 93 active log pairs
  - No dynamic delete of active logs
- Pre-emptable backout
  - Pre-V10, abort/backout schedules non-preemptable SRB
    - On single CPU system may give impression of DB2 hang
  - V10: Create enclave at restart for preemptable SRB backout processing

Summary

- Significant availability improvements in DB2 10
- Continue to exploit availability enhancements in current release

Haakon Roberts
haakon@us.ibm.com

Session code: A08