May 6-10, 2007
San Jose Convention Center
San Jose, California, USA

Session: I05, H12

Accelerate Your DB2 Business On Demand Autonomically!

IDUG® 2007 North America

Scott Hayes
DBI (www.Database-Brothers.com)

May 08, 2007 09:20 a.m. – 10:20 a.m. (I05)
May 10, 2007 09:20 a.m. – 10:20 a.m. (H12)

Platform: DB2 for Linux, UNIX, Windows

GoFurther

Agenda

- A bit of history
- What does DB2 automatically tune?
- What is left for a DBA to do?
  - Are we headed for unemployment lines?

GoFurther

May 6-10, 2007
San Jose Convention Center
San Jose, California, USA

IDUG® 2007 North America

Autonomic History

Past performance may not be a valid indicator of future results – however, history is valuable for predicting trends.

GoFurther

IDUG® 2007 North America

Automated DB2 Tuning – History 101

- 1999 – Database-GUYS (DGI) Wise-GUY™ and Pool-GUY™
- DB2 V5 – Self-Tuning LOAD
- DB2 V6 – Index Advisor, Hash Joins, LOAD
- DB2 V8 …

Automatic REORG
Ability to change CFG parameters online!
db2advis (Design Advisor)
Health Monitor
Self Tuning LOAD
Autoconfigure (Configuration Advisor)
Automatic Storage
UTIL_IMPACT_LIMIT Adaptive Utility Throttling
Auto_Runstats (Automatic Statistics Collection and Profiling)
Self Tuning BACKUP
Automatic BACKUP

GoFurther

May 6–10, 2007
San Jose Convention Center
San Jose, California, USA

IDUG® 2007 North America

What does DB2 9 Autonomically Tune?

It slices, it dices, and makes tuning a snap – yours now for only $9.95 if you call within the next 3 minutes…

GoFurther

IDUG® 2007 North America

AUTOMATIC Summary –

db2 "connect to DBNAME"

db2 "get db cfg for DBNAME show detail" | grep -E "AUTOMATIC|SELF_TUNING_MEM|DATABASE_MEMORY|LOCKLIST|MAXLOCKS|PCKCACHESZ|SHEAPTHRES_SHR|SORTHEAP";db2 "select substr(bpname,1,58) as bpname, npages from syscat.bufferpools"

If NPAGES = -2, then automatic memory for that pool is enabled.

GoFurther

Autoconfigure

- Enabled by default for new databases
  - "2 second tuning" better than none / defaults
- Caution – Auto Configure is not multiple instance aware
  - Assumes host is purely database server with one instance
- Addresses many, but not all, inadequate default values
  - Locktimeout still -1 (unlimited)
  - MAXFILOP 64
  - LOGBUFSZ should be at least 128

NUM_IOCLEANERS

- Automatic by default
  - Cannot be changed online
- Based on number of CPUs and partitions
- While better than the V8 default value of "1", beware of aggressive page cleaning causing spikes in transaction elapsed times.
  - Formula "(#CPUs / #Partitions) – 1" may be better, but not less than 1.

NUM_IOSERVERS

- Automatic by default
  - Much better than V8 default value of 3
  - Cannot be changed online
  - Benefits Prefetching – Appropriate for data warehouse but not OLTP!
- Calculated based on parallelism settings of all tablespaces
  - May be too aggressive
    - 1,000 tablespaces with 3X parallelism across 60 spindles = 3,000 IOSERVERS?
      - And default MAXFILOP is 64?
  - Try # spindles + 10% as alternate for DW, -50% if OLTP
    - Monitor Time Waited on Prefetch
    - Monitor Asynchronous Pages Read per Request (APPR)

Attendee Notes - Asynchronous Pages read Per Request (APPR)

- Compute for Database, Bufferpools, and Tablespaces
- APPR = (Asynchronous pool data page reads + Asynchronous pool index page reads) / (Asynchronous data read requests + Asynchronous index read requests)
  - Your Value? _____
- Measure of Prefetch Efficiency, generally should be greater than 10, preferably higher.

Autonomic Memory Tuning – STMM (Self Tuning Memory Manager)

- DB2's scheme for dynamically adjusting memory between:
  1. Locklist & Maxlocks
  2. Package Cache (pckcachesz)
  3. Sortheap & sheapthresh_shr
  4. Bufferpools
- At least 2 of the 4 memory resources above must be set to automatic --- DB2 robs from Peter to pay Paul

STMM -Considerations & Recommendations for OLTP

- OLTP (Transactional Databases)
  - Start with LOCKLIST & PCKCACHESZ automatic
  - Ensure that physical design (indexes, MQTs, MDC) is optimized and free of defects
    - Use multiple bufferpools to separate random (synchronous) I/O from sequential (asynchronous prefetch) I/O.
      - Investigate, and cure if possible, causes of sequential I/O
      - CATLGBP, TEMPSPCBP, USERSYNCBP, USERASYNCBP
  - Enable AUTOMATIC for SORT and Bufferpools to achieve optimum "fine tuning", then lock down sizes.
  - Monitor Carefully: TX Rates & Elapsed times, CPU Busy, changes in I/O behavior (Sync, Async, Times)
  - *If you thought dynamic SQL introduced unpredictability to your environment, what will STMM do?*

STMM -Considerations & Recommendations for DW

- Data Warehouse Databases
  - Start with LOCKLIST, PCKCACHESZ, SORT and Bufferpool all automatic
  - Optimize physical design (indexes, MQTs, MDC)
    - Use one large bufferpool
      - Possibly use 2nd bufferpool for "hot" lookup tables
      - IBMDEFAULTBP and HOTBP
  - Monitor Carefully: Query Throughput & Elapsed times, CPU Busy, changes in I/O behavior (Sync, Async, APPR, Read/Write Times)

STMM -Considerations & Recommendations for DPF

- Data Warehouse Databases - DPF
  - The bufferpool(s) must not have any entries in sysibm.sysbufferpoolnodes
  - One partition is the tuning "king" and dictates tuning for all partitions.
    - db2 "call sysproc.admin_cmd ('get stmm tuning dbpartitionnum')" [Returns Current Ruling Partition]
    - db2 "call sysproc.admin_cmd ('update stmm tuning dbpartitionnum N') [Changes Ruling Partition to "N"]
  - Use ACTIVATE DATABASE command – all partitions must be active for STMM

STMM – Implementation Details

- DATABASE_MEMORY is AUTOMATIC by default on Windows and AIX, COMPUTED for Linux, Sun, HP
- db2 update db cfg for DBNAME using self_tuning_mem ON (Use OFF to lock current values)
- db2 update db cfg for DBNAME using LOCKLIST AUTOMATIC
- db2 update db cfg for DBNAME using MAXLOCKS AUTOMATIC
- db2 update db cfg for DBNAME using PCKCACHESZ AUTOMATIC
- db2 update db cfg for DBNAME using SORTHEAP AUTOMATIC SHEAPTHRES_SHR AUTOMATIC
- db2 alter bufferpool ***IBMDEFAULTBP*** size AUTOMATIC
- "MANUAL" locks in current sizes and disables automation for that component – Alternatively, Integer values disable automatic tuning and will cause change to the specified value

STMM - Monitoring

- db2 “get db cfg for DBNAME show detail”
  - Shows current and pending sizes
- db2 “get snapshot for database on DBNAME”
  - Shows current, configured, and high water mark memory heap sizes
- db2 “get snapshot for bufferpools on DBNAME”

```
Node number                              = 0
Tablespaces using bufferpool             = 3
Alter bufferpool information:
 Pages left to remove                    = 0
 Current size                            = 1008
 Post-alter size                         = 1008
```

- Be sure to try the new SYSIBMADM.SNAP* administrative views

May 6–10, 2007
San Jose Convention Center
San Jose, California, USA

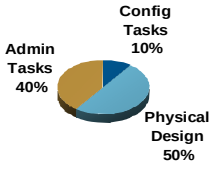
IDUG® 2007 North America

What’s Left for the DBA to do?

Plenty. You still have a job.

In an ideal DB2 9 World…

Where will a DBA's time go?



Taming the Autonomic Beast

- Configuration Changes, 10-15% Improvement
- Physical Design Changes, 85-90% Improvement
- Administrative Tasks
  - Online REORG
  - Monitoring and Testing Backups
  - HADR
  - Recovery Testing
  - LBAC Configuration
  - Auditing
  - Security Administration
  - Performance Monitoring and Locking Diagnosis
  - Table Compression Analysis & Implementation

May 6-10, 2007
San Jose Convention Center
San Jose, California, USA

IDUG® 2007 North America

Configuration Changes

Autoconfigure is a good start…

… let's explore some more necessary changes.

Let's Get Busy – Configuration Changes

1. db2 "update db cfg for DBNAME using MAXFILOP 512", then monitor DB Snap to ensure Database Files Closed = 0. Changes online at TX boundary.
2. Catalog Cache Size – not an Automatic participant, but can be changed online
   - Monitor Catalog Cache Hit Ratio (from DB Snap) and increase CATALOGCACHE_SZ until CATHR > 95%
     - CATHR = 100 – ((Catalog cache inserts * 100) / Catalog cache lookups)
3. Verify appropriate value for NUM_IOSERVERS
   - DW - #Spindles + 10%
   - OLTP - #Spindles – 50%

IDUG® 2007 North America

NUM_IOCLEANERS + CHNGPGS_THRESH: Asynchronous Write Percentage (AWP)

- Compute for *Database*, Bufferpools, and Tablespaces
- AWP = (((Asynchronous pool data page writes + Asynchronous pool index page writes) * 100) / (Buffer pool data writes + Buffer pool index writes))
  - Your Value? _____ (use DB Snap)
- Should be > 90%, else need to increase num_iocleaners (up to #CPUs – 1) and/or lower CHNGPGS_THRESH by 5% decrements

GoFurther

IDUG® 2007 North America

Configuration Changes: LOGBUFSZ Online=no AT=no

- Buffers read & write I/O to log files
- Default size 8 is grossly too small for most of today's databases, autoconfigure will "tune"
- Set it to 256-512, higher if you do lots of Rollbacks or have high DML activity
- Must be less than DBHEAP size
- LGRHR (Log Read Hit Ratio) = 100 – ((Number read log IOs * 100) / Log pages read)
  - Your value? _____ (use DB SNAP)
- If LGRHR < 98%, increase LOGBUFSZ further

GoFurther

May 6–10, 2007
San Jose Convention Center
San Jose, California, USA

IDUG® 2007 North America

Physical Design Changes

Memory Tuning will be an exercise in futility if the physical design does not support the workload. Memory shell games will only mask real performance problems…

GoFurther

Workload Analysis > Well Stated Opportunity

- 80+% of Tuning Benefit comes from complete and accurate understanding of the SQL workload and its costs
- What is the most costly, most harmful, SQL during peak periods? Recent periods? Over time?
  - Highest CPU Consumption
  - Highest Sort Time Consumption
  - Highest average Elapsed times
  - Highest Read I/O (rows read)
- Grouping & Cost Aggregation of similarly structured SQL statements is imperative to "True Cost" determination

SQL Equalization & Cost Aggregation

US Patent # 6,772,411

Select c1, c2, c4 from tbl where c5 = '0360' cpu=.1
100's of SQL statements per second…
SQL Snapshot shows 19 different statements!
WRONG ANSWER!

Relative Costs

- How the DBA needs to see the SQL Workload:

SQL Statement	Count	TotCPU	CPU%
Select c1, c2, c4 from tbl where c5 = '?'	16	1.6	66.6
Select c1, c2, c4 from tbl where c5 > '?'	2	.6	25.0
Select c1, c2, c4 from tbl where c8 = '?'	1	.2	8.33
Totals:	**19**	**2.4**	**100.00**

Solving Problems > Effective Solutions

- Given a costly SQL statement, 3 possible solutions:

1) Physical Design Change (95%)
   - Add an Index
   - Add/modify Cluster Index
     - Most potent weapon against poor application performance
   - Drop Ineffective/Costly Indexes
     - Low Cardinality, Skewed Distributions, Redundant Indexes
   - Generated Columns with new supporting Index
   - MQT/AST/MDC tables
2) Tweak Catalog Statistics to "fool" optimizer (2%)
   - A temporary and difficult to maintain "solution"
3) Re-write/modify SQL (3%)
   - The DB2 Optimizer Re-writes SQL. Isn't re-writing re-written SQL redundant?

Hidden Physical Design Costs – **obfuscated** by well tuned memory

- A simple SELECT executed with high frequency against a table with only 32 rows consumed 34% of ALL CPU time on an SMP 4-way
- Myth: Small tables don't need indexes
- Realities:
  - Explains don't identify costly SQL against small tables
  - Explains don't consider frequency of execution
  - Only Dynamic SQL Equalization finds high cost SQL
  - Even ONE row tables can benefit from indexes

Which tables have highest Rows Read?

- db2 "select tabname, rows_read from sysibmadm.snaptab order by rows_read desc fetch first 10 rows only"
- Two Possibilities:
  1. The data is very popular!
     - Consider placing the table in its own tablespace on best performing disks. Can also assign tablespace to its own bufferpool.
  2. Expensive TBSCANs are occurring against the table
     - Use SQL Equalization to identify SQL
     - Use IBM Design Advisor for Physical Design solutions (Indexes, MDC, MQT, partitioning)

Heads Up: The Design Advisor does NOT recommend XML Indexes

Which Tables have highest rows written? What indexes are defined?

- db2 "get snapshot for tables on DBNAME"
- db2 "select tabname, rows_written from sysibmadm.snaptab order by rows_written desc fetch first 10 rows only"
- Back at your office, carefully review all indexes on these Top 10 tables to ensure they conform to best practices guidelines (coming up)
- You can query 'SYSCAT.INDEXES', use the command "db2 describe indexes for table *schema.tablename* show detail", or use a GUI (like Control Centre)

Index Design Guidelines

- Indexes with Cardinality = 1 are a performance death sentence. Do not create indexes "just in case"…
- Indexes with Skewed distributions are expensive to maintain on Insert, Update, Delete
- Redundant Indexes are expensive to maintain, consume disk, and provide no value to DB2 – Drop them!
  - IX on C1, C2 <<- Redundant Index
  - IX on C1, C2, C4
- Use composite indexes to replace single column indexes
- For multi-column indexes, place the column that is most frequently known (= predicate) first.
- Use Clustering Indexes to reduce Sort & CPU costs

Composite Indexes Accelerate Your Business

- "SELECT * from TB where C1 = ? And C2 = ? And C3 >= ?" caused an SLA to be missed and service contracts nearly lost
- Myth: Use 3 single column indexes on C1, C2, and C3 individually
- Realities:
  - Index AND-ing can be CPU and I/O expensive
  - A single composite index on columns C1, C2, & C3 is dramatically faster and more efficient
  - The INDEX/Design Advisor favors composite indexes, but identifying the costly SQL is the trick >> SQL EQ!

Which Tablespaces have the slowest Read and Write times (ORMS, OWMS)?

- Ensure tablespace best practices implemented
  - Multiple containers
    - Equal Size
    - Different Devices/Paths
    - PREFETCHSIZE is 3-4X EXTENTSIZE
  - Containers not on OS Paging disks or other very busy spindles
  - RAID storage has "db2set DB2_PARALLEL_IO=* (or tsid list)" enabled
  - DMS Storage can be 5-10% faster, but more difficult to administrate
  - Use Automatic Storage with multiple paths

Attendee Notes - ORMS

- ORMS for the Database Overall:
  - db2 "select (POOL_READ_TIME / (POOL_DATA_P_READS + POOL_INDEX_P_READS + 1) as ORMS from sysibmadm.snapdb where db_name = '*DBNAME*'"
- ORMS for the top 10 (read) slowest tablespaces:
  - db2 "select tbsp_name, (POOL_READ_TIME / (POOL_DATA_P_READS + POOL_INDEX_P_READS + 1) as ORMS from sysibmadm.snaptbsp order by ORMS desc fetch first 10 rows only"
- How do the top 10 TS compare to the database average?

Attendee Notes - OWMS

- OWMS for the Database Overall:
  - db2 "select (POOL_WRITE_TIME / (POOL_DATA_WRITES + POOL_INDEX_WRITES + 1) as OWMS from sysibmadm.snapdb where db_name = '*DBNAME*'"
- OWMS for the top 10 (write) slowest tablespaces:
  - db2 "select tbsp_name, (POOL_WRITE_TIME / (POOL_DATA_WRITES + POOL_INDEX_WRITES + 1) as OWMS from sysibmadm.snaptbsp order by OWMS desc fetch first 10 rows only"
- How do the top 10 TS compare to the database average?

Which Tablespaces are Synchronously Read? Asynchronously Read?

- Tablespaces that are highly Synchronously Read contain tables with good indexes that support queries and avoid costly scans. These tablespaces should be placed into a USERSYNCBP Bufferpool.
- Tablespaces with significant Asynchronous reads have tables that are being prefetched – An indexing opportunity may or may not exist. Place these tablespaces into a USERASYNCBP Bufferpool. This bufferpool should be large enough to facilitate effective prefetching (APPR), with memory preference given to USERSYNCBP.

Attendee Notes – Sync Read Percent (SRP)

- For the Database:
  - select 100 - (((pool_async_data_reads + pool_async_index_reads) * 100) / (pool_data_p_reads + pool_index_p_reads + 1)) as SRP from sysibmadm.snapdb where DB_NAME = '*DBNAME*'
- For Tablespaces:
  - select tbsp_name, 100 – (((pool_async_data_reads + pool_async_index_reads) * 100) / (pool_data_p_reads + pool_index_p_reads + 1)) as TSSRP from sysibmadm.snaptbsp order by TSSRP desc
    - Lists tablespaces from most randomly (Synchronously) read to least. SRP should be 70%+ to participate in USERSYNCBP.

When to REORG online (INPLACE)?

- When Table (Overflows * 100 / Rows_Read) > 3% and "significant activity" is present
- db2 "select 'REORG TABLE ' || TABSCHEMA || '.' || TABNAME || ' INPLACE ALLOW WRITE ACCESS;' from sysibmadm.snaptab where (ROWS_READ > 999) AND (((OVERFLOW_ACCESSES * 100) / (ROWS_READ + 1) > 3)"
- Reorgchk? Maybe once or twice a year.
- Remember to REORG indexes after table REORG completes

May 6–10, 2007
San Jose Convention Center
San Jose, California, USA

IDUG® 2007 North America

In Summary…

DB2 9 helps automate configuration tuning, but – you still have a job to do!

When is your tuning job done?

- **OLTP:**
- Rows Read/TX/TB < 10
- DB BP Sync Reads > 90%
- BP, Pkg Cache, & Catlg Cache hit ratios > 95%
- There are no bad apples
  - No Slow TS (ORMS, OWMS)
  - No SQL > 10% CPU
  - No SQL > 50% SLA time
  - No SQL w/ Rows Read/Rows Fetched > 100 (IXEFF)
  - No Files Closed
  - No Lock or Token Waits
- Phone Rage Ends

- **Data Warehouse:**
- Prefetch is Effective (APPR > 10 for each TS)
- No Slow TS (ORMS, OWMS)
- TEMPSPACE defined where data isn't – has 3-6 containers
- DB BP Sync Reads > 25%
- Catlg Cache Hit > 95%
- No Files Closed
- SQL having Frequency>1 uses
  - MQTs / ASTs
  - MDC tables
  - Effective Indexes
- Phone Rage Ends

Session: I05, H12

Accelerate Your DB2 Business On Demand Autonomically!

Thank you for attending!

Scott Hayes

DBI

Scott.Hayes@Database-Brothers.com