IBM GLOBAL SERVICES

IBM

Z25

ERP Performance & Availability Tips Using DB2 for z/OS

William Favero

IBM DB2 Information Management Technical Conference

Sept. 20-24, 2004

Las Vegas, NV

Disclaimer

The information contained in this presentation has not been submitted to any formal IBM review and is distributed on an "As Is" basis without any warranty either expressed or implied. The use of this information is a customer responsibility.

The materials in this presentation are also subject to

- enhancements at some future date,
- a new release of DB2, or
- a Programming Temporary Fix (PTF)

IBM MAY HAVE PATENTS OR PENDING PATENT APPLICATIONS COVERING SUBJECT MATTER IN THIS DOCUMENT. THE FURNISHING OF THIS DOCUMENT DOES NOT IMPLY GIVING LICENSE TO THESE PATENTS.

For additional information visit the URL

http://www.ibm.com/legal/copytrade.phtml ***for* "Copyright and trademark information"**

Notes Only

Abstract

If you are dealing with Enterprise Applications like SAP, PeopleSoft or Siebel (or even your everyday client application), there are some basics performance and availability tips that may work for you when working with DB2 for z/OS. This presentation will discuss handling scheduled and unscheduled outages so you can come closer to continuous availability. It will also discuss some of the performance knobs available in DB2 for z/OS to assist you with possibly improving your performance.

Agenda

- Caching
- DSNZPARM
- REORG
- Future “relief”

Caching

Dynamic SQL Processing

- Requires two steps: prepare and then execute
- Prepare can be expensive and includes
  - Parsing SQL statement
  - Validating SQL syntax
  - Search catalog to
    - Validate tables
    - Validate columns
    - Insure proper set of privileges are in place for this SQL statement
  - Create optimized access path
  - Create executable statement

No Caching

This is what dynamic SQL has always had

- Prepare SQL statement ABC
- Commit
- SQL statement ABC must be prepared again before it can be re-executed

Or

- Program A prepares SQL statement ABC
- Program A terminates
- Program is rerun and SQL statement ABC must be prepared again

No Caching

- Continued…
  - Program A prepares SQL statement ABC
  - Program A terminates
  - Program B starts and must prepare the exact same SQL statement ABC

Local Cache

Persistent PREPARE

- **Bind Option -** KEEPDYNAMIC(YES)
  - Prepared SQL statement persist across Commit
  - Will use dynamic statement caching if active

- **Will require application changes** to avoid unnecessary PREPAREs

Notes Only

When the local dynamic statement cache uses KEEPDYNAMIC(YES) in combination with CACHEDYN=YES, the local dynamic statement cache can take up a large amount of virtual storage.

MAXKEEPD

MAX KEPT DYN STMTS

- **System wide limit**
- Specify the total number of prepared, dynamic SQL statements that can be saved past a commit point by applications that run with the KEEPDYNAMIC(YES) bind option
- Can have a significant impact on the DBM1 address space’s storage usage

Persistent PREPARE

- **Dynamic Statement Caching will affect Persistent PREPARE**
- **If Dynamic Caching on:**
  - Prepared statement is cached
  - Cached prepared statement is used
- **If Dynamic Caching off:**
  - Only statement string is saved
  - DB2 prepares statement string after commit

Global Caching

Prepared Statement Caching

- **System wide option determined at installation time**
    - DSNTIP4 - **CACHE DYNAMIC SQL** field
        - Acceptable values: YES, NO
        - Default: NO
        - DSNZP*ARM*: DSN6SPRM CACHEDYN
- **All DML prepared using**
    - SQL PREPARE
    - SQL EXECUTE IMMEDIATE
- **Prepare statement reuse if:**
    - Exact match on SQL statement string
    - Authid must match (*more later*)
    - Some BIND options must be consistent
        - CURRENTDATA, DYNAMICRULES, ISOLATION, SQLRULES, and QUALIFIER

Prepared Statement Caching

- Not available across data sharing members
- Be careful -
  - – Caching statement may add cost to preparing statement
  - – Statement skeletons take up space in the EDM pool
  - – If caching is enabled (CACHEDYN YES), there is a cache search cost
- However…
  - – Can have significant CPU savings for an ERP application

CMTSTAT

DB2 Version 7

- If DSNZPARM CMTSTAT = INACTIVE
  - DB2 writes an accounting record when a transaction commits
  - Connection qualifies to become inactive
- If using KEEPDYNAMIC(YES) bind option
  - DB2 cannot disconnect the connection from the thread (which is what happens when the connection becomes inactive), because the thread contains information about locally cached statements
  - DDF threads always (has to) remain active
- **Result:**
  - Accounting records are not cut at transaction boundaries. Likewise, DDF does not re-establish WLM enclaves at transaction boundaries.

CMTSTAT

- **DB2 Version 8**
- DSNZPARM CMTSTAT = INACTIVE
- KEEPDYNAMIC(YES) still prevents DDF threads from becoming inactive
- **However:**
  - KEEPDYNAMIC(YES) allows DB2 to cut accounting records at transaction boundaries
  - New behavior allows period based WLM goals for threads that commit frequently, but cannot become inactive only because they use KEEPDYNAMIC(YES).
  - Threads that cannot become inactive for other reasons, do not reset the WLM enclave, and period based goals are probably not right for them (as was the case in the past).
- This new behavior is supported for DRDA clients. DB2 for z/OS clients that use the DB2 private protocol are not affected by this change.
- The presence of held cursors or declared temporary tables keeps the threads active and does not allow accounting intervals or WLM enclaves to complete as it did in Version 7

Notes Only

That is, at commit time, when DB2 evaluates whether a DDF connection is eligible for inactivation, if the only reason why it cannot become inactive is the presence of cached dynamic SQL statements due to KEEPDYNAMIC(YES), DDF still cuts an accounting record, and also completes the WLM enclave (as if KEEPDYNAMYIC(YES) is not specified). When a new request arrives from the client system over the connection (that is still active), a new enclave is created and a new accounting interval is started.

EDM Pool

- If Dynamic Statement Caching is turned on, statements can be cached in a data space
- DB2 determines the default size of the data space as follows:
  - If the EDM pool storage size is less than or equal to 40 MB, the EDM data space is 40 MB.
  - If the EDM pool storage size is larger than 40 MB, the EDM data space size equals the EDM pool size.
  - See next slide for overrides

EDM Pool and Data Spaces

- **EDMPOOL DATA SPACE SIZE**
  - Acceptable values: 1-2097152
  - Default: based on calculations
  - DSNZPARM: DSN6SPRM EDMDSPAC
  - Starting point
- **EDMPOOL DATA SPACE MAX**
  - Acceptable values: 0-2097152
  - Default: 1048576
  - DSNZPARM: DSN6SPRM EDMDSMAX
  - Had no meaning if dynamic caching is not turned on

Notes Only

The EDMPOOL DATA SPACE SIZE field specifies the size (in kilobytes) of the data space that can be used by the environmental descriptor manager (EDM) pool. The CLIST calculates a data space size only if you specified YES for CACHE DYNAMIC SQL on panel DSNTIP4, otherwise it is zero.

The EDMPOOL DATA SPACE MAX field specifies the MAXIMUM size (in kilobytes) of the data space that can be used by the environmental descriptor manager (EDM) pool. The CLIST calculates a maximum data space size only if you specified YES for CACHE DYNAMIC SQL on panel DSNTIP4.

This portion of the EDM pool host the global dynamic statement cache only.

EDM Pool

In DB2 Version 8, a data space is no longer used. Storage for the statement cache is above the 2Gb bar.

EDM STATEMENT CACHE for V8

- Acceptable values: 5000 to 1048576
- DSNZP*xxx*: DSN6SPRM EDMSTMTC (new)

Version 8 always allocates at least 5 MB of storage for the dynamic statement cache regardless of the setting of CACHEDYN. Because CACHEDYN is changeable online changeable in V8, DB2 needs an initial size to be allocated at startup time to allow the size to be changed online.

Notes Only

The default value for EDM STATEMENT CACHE is taken from the EDMPOOL STORAGE SIZE field on panel DSNTIPC, or 5000 K, whichever is larger. Specify the size (in KB) of the statement cache that can be used by the EDM. This value is used at DB2 startup time as the minimum value. This value cannot be decreased below the value that is specified at DB2 startup. The CLIST calculates a statement cache size. This storage pool is located above the 2-GB bar. You have a choice of:

v Accepting the value in the Calculated column of panel DSNTIPC; the CLIST calculates this value based on input from previous panels. If a value is in the Override column, you must erase the override value in order to accept the calculated value.

Parameter Markers

```
SELECT *
    FROM
       WHERE col = “abc”
```

No reuse because statements are not exactly the same

```
SELECT *
    FROM
       WHERE col = “def”
```

```
SELECT *
    FROM
       WHERE col = ?
```

Cache reuse

```
SELECT *
    FROM
       WHERE col = ?
```

Notes Only

Using parameter markers with dynamic statement caching, the cost of repeated prepares can be reduced significantly.

Invalidating Statement Cache

Version 8 invalidation of dynamically cached statements after CREATE INDEX

If indexes are known:

- Execute RUNSTATS utility with the options
  - UPDATE NONE REPORT NO in DB2 V8.
    - This combination only invalidate statements in the dynamic statement cache without any data access or computation cost

However, for Enterprise Application solutions, indexes are often created when maintenance is applied so their existence is not obvious to a DBA.

- If a new index is created, the affected cached statements are invalidated for both data sharing and non-data sharing, but not quiesced.
  - Not quiesced means that statements that are used in active transactions can still be employed until these transactions complete.

Invalidating Statement Cache

Additional Version 8 considerations

- Cached statements are invalidated when a new partition is added to a table
- REORG REBALANCE will invalidate statement cache

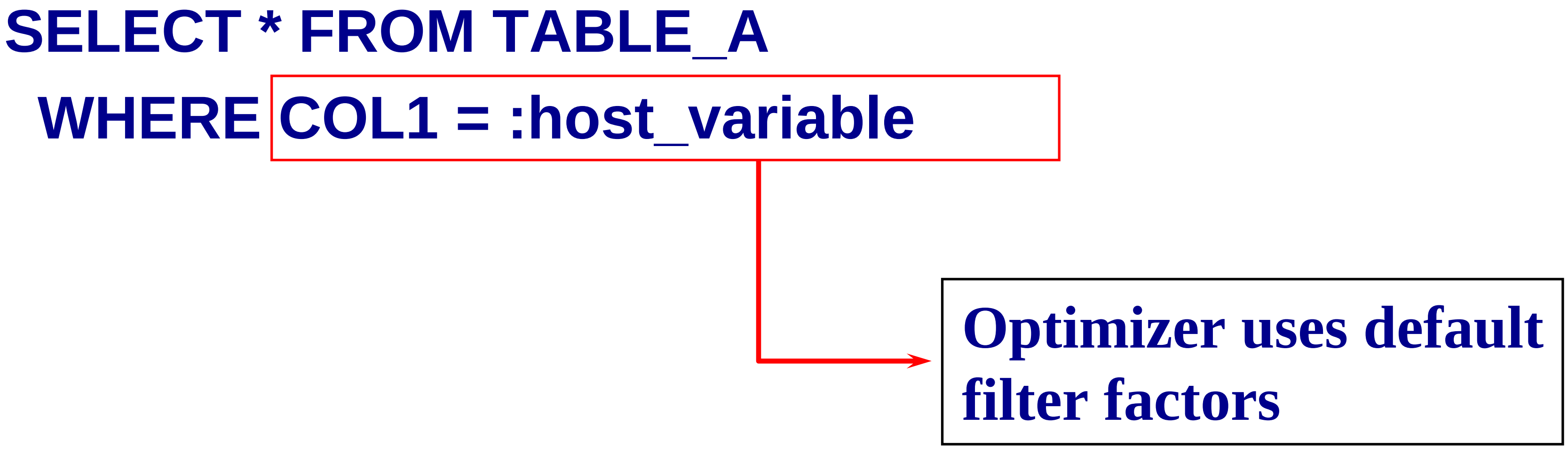
Authorization IDs

- SAP
  - Application server uses:
    - one unique authorization ID
    - one unique SQLID
  - Different application servers can connect with different authorization ID
- PeopleSoft
  - Application server verifies the user is authorized for function requested
  - Manages the connect to DB2 z/OS using:
    - Common authorization ID
    - Common SQLID

Authorization IDs

- Siebel
  - With native TSO/RACF security, verify user authorized for requested function using
    - Primary user authorization ID
    - Common SQLID (typically the creator of the tables)
    - Common SQLID must be part of a user's secondary authorization ID list and the application issues a SET CURRENT SQLID to that common SQLID
    - Recommended to restrict Siebel users from accessing TSO
  - Siebel External Security Adapter,
    - Maintain user information in an external authentication system, such as an LDAP repository
      - Users log on to the Siebel application
      - External Security Adapter validates the user name and password
      - External Security Adapter finds match and retrieves a generic set of user credentials (username/password)

Reoptimization For Input Variables



SELECT * FROM TABLE_A

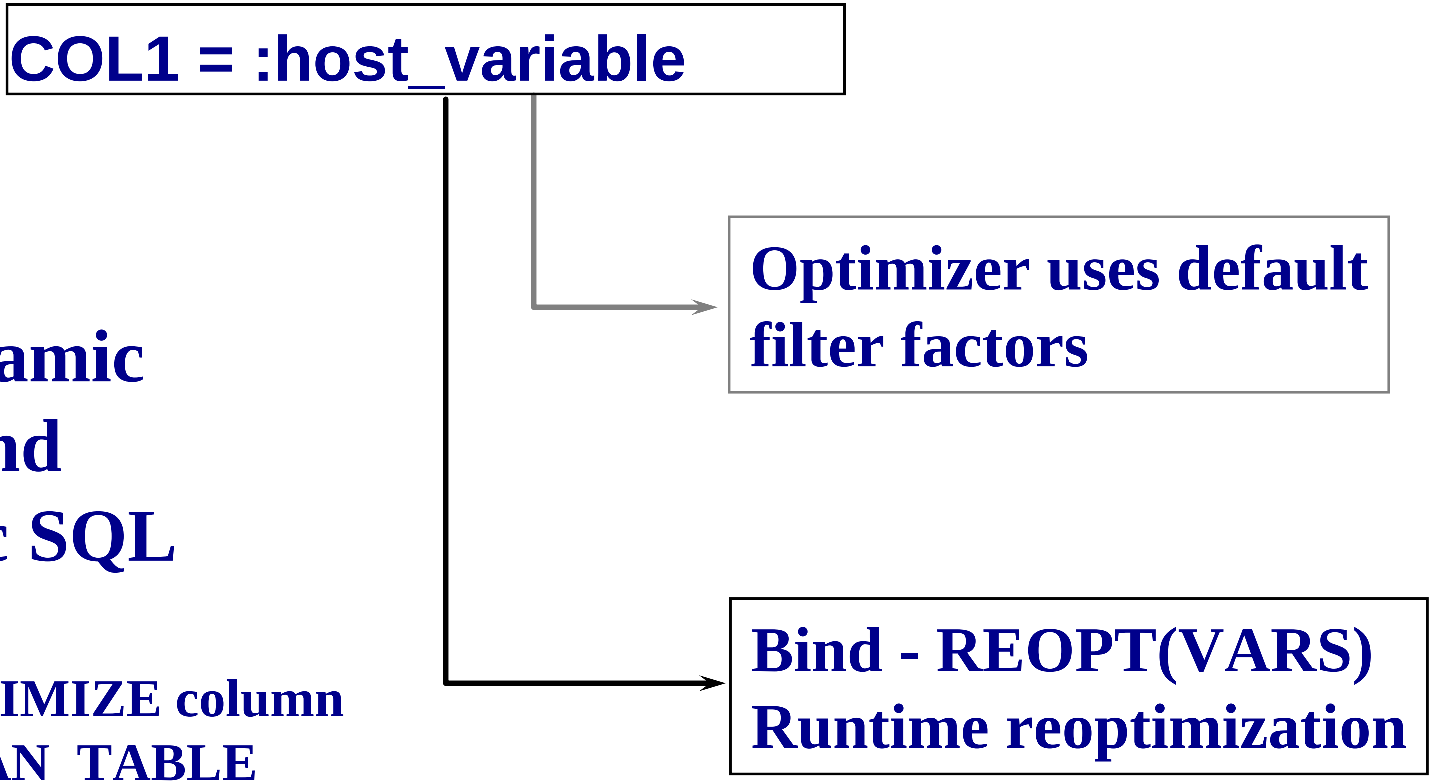
WHERE COL1 = :host_variable

Optimizer uses default filter factors

Dynamic
and
Static SQL

Reoptimization For Input Variables

SELECT * FROM TABLE_A
WHERE COL1 = :host_variable



Optimizer uses default filter factors

Bind - REOPT(VARS)
Runtime reoptimization

Dynamic
and
Static SQL

WHEN_OPTIMIZE column added to PLAN_TABLE

REOPT Keyword

- The BIND option REOPT(ONCE) allows re-optimization an SQL statement based on the host variable value the first time it is executed. This reduces the cost of multiple optimization when REOPT(VARS) was used in the past.
- REOPT(ONCE) reuses the statement stored in the dynamic statement cache.
- REOPT(ONCE) can be EXPLAINed because the statement is in the cache, something that could not be done with REOPT(VARS)

IFCIDs

Tracing the full SQL statement, you have two options:

- If the statement is in the dynamic statement cache, you can use IFCID 317 to retrieve the full SQL statement text through the READS interface of the IFI.
- If the statement is not in the dynamic statement cache, you can use the new IFCID 350. Unlike IFCID 63, it contains the full SQL statement text.

EXPLAIN STMTCACHE

EXPLAIN STMTCACHE - Clause added in Version 8

- STMTID id-host-variable or integer-constant
  - Specifies that the cached statement associated with the statement ID contained in host variable id-host-variable or specified by integer-constant is to be explained.
  - The statement ID is an integer that uniquely determines a statement that has been cached in dynamic statement cache. The statement ID can be retrieved through IFI monitor facilities from IFCID 316 or 124 and is shown in some diagnostic IFCID trace records such as 172, 196, and 337.
    - For every row that the EXPLAIN statement inserts into the plan table, statement table, or function table, the QUERYNO column contains the value of the statement ID.

EXPLAIN STMTCACHE

EXPLAIN STMTCACHE - Clause added in Version 8

- STMTTOKEN token-host-variable or string-constant
  - The STMTTOKEN parameter takes either a host variable or a string constant. This statement token string is associated with the cached statement by the application program that originally prepares and inserts the statement into the cache. The application does this by using the RRSAF SET_ID function or by using the sqlseti API from a remotely-connected program
  - For every row inserted into the plan table, statement table, or function table by the EXPLAIN statement, the column STMTTOKEN (which is newly added column) is given the value of the statement token, and the column QUERYNO is given the value of the statement ID for the cached statement with the statement token.

Notes Only

When DB2 explains a statement from the statement cache, it writes the current access path that is used by that statement in the PLAN_TABLE. The statement is NOT going through access path selection again when the STMTCACHE option is used on the EXPLAIN statement (This is different from normal EXPLAIN processing of dynamic SQL statements). Going through access path selection again would defeat the purpose, as we try to find out what the *current access path* of a statement in the cache is. Determining the access path again may result in a different access path, as statistics for example may have changed in the meantime.

Buffer Pools

- Use data type and data access characteristics to spread across pools
- If z/OS, consider data spaces
  - If DB2 Version 8, there are no more data spaces
- Isolate sort in its own pool
- Monitor

DSNZPARM

NPGTHRSH

- **NPGTHRSH** system parameter
  - – 0 standard cost optimization, default.
  - – N if NPAGES < n, then prefer index access,
    - recommended value 10.
  - – to force index when NPAGES=-1, specify NPGTHRSH=502
  - – -1 always prefer index access, not recommended
- Advantages: very efficient and easy to exploit
- Disadvantages:
  - – Less efficient than standard optimization in some cases
  - – Nevertheless, using the feature is much lesser of evils

*This feature is a system wide option

Notes Only

The subsystem parameter NPGTHRSH, in macro DSN6SPRM, lets you specify that DB2 uses special access path selection for tables under a given size. The default value is 0 which means no special access path selection is used.

VOLATILE

CREATE/ALTER TABLE …. VOLATILE

- For cases where statistics do not accurately describe tables contents
- Index access is used if possible
  - List prefect is disables
- NOT VOLATILE is the default
- Affects only the table it is defined to

SMSDCFL / SMSDCIX

- Add to IDCAMS DATACLASS Clause
  - SMSDCFL: SMS data class name keyword for all table space data sets
  - SMSDCIX: SMS data class name keyword for all index space data sets
- APAR PQ32414 (V6)

Notes Only

The SMSDCFL is the SMS data class name keyword for all table space data sets and the SMSDCIX is the SMS data class name for all index space data sets. These 2 new keywords are optional. If specified, it will be added into the ADMF initialization parameter block as part of the ZPARM. If either one is specified, all corresponding table space and/or index space data sets will be allocated as a SMS-managed DB2 STOGROUP defined VSAM LDSs.

Log DSNZPARM Values

- CHKFREQ - Checkpoint Freq
  - Default 50,000
- FREQUENCY TYPE – specifies units for checkpoint frequency
  - Consider using time rather than log records for data sharing for a more uniform recovery

Long Running UR Warnings

- **In DB2 V6 warning based on total system load (**URCHKTH)
  - Be careful, this can be misleading
  - Still useful though
    - Identify long running, low update activity processed
- **Parameter UR LOG WRITE CHECK (**URLGWTH)
  - number of log records written by uncommitted UR before warning
  - monitored independently from existing parameter UR CHECK FREQ
  - when reached a console message issued and IFCID 313 cut
- **Recommendation**
  - CHKFREQ between 5 and 10 minutes
  - URCHKTH = 1
  - URLGWTH = 100K

Notes Only

The URLGWTH allows more precise control over when the warning about non-committing, long running units of recoveries is issued. Instead of being based on the overall system activity it has a UR scope, i.e. the message is issued for a UR if that particular UR generated a user specified number of log records without committing. The old scheme is still useful, but can be deceiving in the case when many URs are doing changes and generate in total lots of log records, but none of the URs should be considered long running. It is useful to detect long running URs that do not generate too many log records.

Other DSNZPARM Values

- SYNCVAL - Monitoring
  - Default NO

Notes Only

Specify whether DB2 statistics recording is synchronized with some part of the hour. You can specify that the DB2 statistics recording interval is synchronized with the beginning of the hour (0 minutes past the hour) or any number of minutes past the hour up to 59. If NO is specified, no synchronization is done. NO is the default.

EXTSEC

- **Extended Security (EXTSEC)**
- **YES**
  - Detailed reason codes are returned to a DRDA level 3 client when a DDF connection request fails because of security errors
  - RACF users can change their passwords using the DRDA change password function
    - This support is only for DRDA requesters that have implemented support for changing passwords.
- **NO**
  - Returns generic error codes to the clients
  - Prevents RACF users from changing their passwords

YES is the recommendation.

This allows properly enabled DRDA clients to determine the cause of security failures without requiring DB2 operator support. It also allows RACF users on properly enabled DB2 clients to change their passwords.

IDTHTOIN

- Consider using IDTHTOIN=0
  - IDTHTOIN default is 120 (seconds) in DB2 V8. For SAP, this parameter should be:
    - Set to 0 (zero) to ensure that idle threads are not automatically canceled by DB2.
      - This parameter is only relevant if CMTSTAT is set to INACTIVE
    - KEEPDYNAMIC(YES) will cause DBATs to remain active and therefore making them susceptible to idle thread timeouts

DSMAX

- **In DB2 Version 8**
  - **DSMAX default is 10,000 and max is 100,000**
  - **z/OS 1.5 (and above) data set control blocks no longer below 16Mb line**
    - **z/OS 1.3 & 1.4 are still below the line**

REORG

Why and When to REORG

- Why?
  - Extent consolidation
  - Re-adding free space
  - Compression
  - Optimize data access
    - Efficient sequential prefetch
    - Index leaf page order
    - Free space to aid clustering
  - Alternative to Unloads and Deletes
- When?
  - Use Real Time Stats* to determine changes since last REORG
    - (Next slide)

*Details in DB2 UDB for z/OS Version 7 Administration Guide, SC26-9931-02, Appendix G

When to REORG

- ***Table space***
  - Last REORG time
  - Rows inserted
  - Rows deleted
  - Rows updated
  - Not perfectly chunked LOB inserts
  - Unclustered row inserts
  - Mass deletes or table drops
  - Overflow records relocated 'near' pointer record
  - Overflow records relocated 'far' from pointer record

Notes Only

TOTALROWS is the total rows in table space.   That, combined with the number of inserts, updates and deletes, can be used to determine if a REORG of the table space is necessary.

(((REORGINSERTS+REORGDELETES+REORGUPDATES)*100)/TOTALROWS)>20

When to REORG

- ***Index space***
  - Last REORG time
  - Index entries inserted
  - Index entries 'appended'
  - Index entries deleted
  - Index entries pseudo-deleted
  - Index mass deletes
  - 'Near-off' index page splits
  - 'Far-off' index page splits
  - Index levels added or removed

Online REORG – Good But…

- Read/Write access during the REORG process
- However…
  - Be aware of workload
  - Is concurrent activity well behaved
    - Look for non-committing read-only transactions
    - Look for non-committing update transactions
  - Take advantage of FASTSWITCH option
  - Be aware of BUILD2 phase
- Use selectively
- And then there is Version 8

Version 8 and REORG

- DPSI – Data-Partitioned Secondary Indexes
  - Eliminates BUILD2 phase of Online REORG
  - Eliminates contention between LOAD PART jobs executing on different partitions
- Option to REORG partitions only in REORG Pending state
- Automatic partition rebalancing

Future Relief

What Will Version 8 Do for Us?

- **64 bit virtual storage**
- **Schema evolution**
- **Multi-row fetch & insert**
- **Multiple DISTINCT Clauses**
- **Longer Table Names & Column Names**
- **Additional statistics**
- **Convert Column Type**
- **Altering CLUSTER Option**
- **Adding Columns to Index**
- **Index-only Access Path for VARCHAR**
- **Adding New Partitions**
- **Separate Clustering from Partitioning**
- **...**

Short Prepare

DSC Short Prepare Enhancements

- **V7 Problem**
  - When a matching statement is found in cache, it is copied into thread related storage before it can be used
  - If no storage a GETMAIN is performed
  - To keep size of local pool limited initiate storage contraction regularly

GETMAIN/FREEMAIN make up large part of cost of short prepare

- **V8 Solution**
  - Use a number of shared storage pools (assigned via hashing to minimize contention) to copy statements in
  - Potentially fewer storage required (not each thread needs its own pool)
  - Using "best fit" algorithm to find space in assigned pool
  - Result: Fewer GETMAIN/FREEMAIN operations, faster short prepare

Joins

Complex joins

- DB2 Version 8 increases the limit on the number of tables that can be used in a Join operation to 225.

Reduced Lock Contention

Reduced lock contention on volatile tables

Volatile (or cluster) tables, used primarily by Enterprise Application vendors like SAP, are tables that contain groups (or clusters) of rows which logically belong together. Within each cluster, rows are meant to be accessed in the same sequence every time. Lock contention occurs when DB2 chooses different access paths for different applications operating on the same cluster table. In the absence of support for cluster tables in DB2, users have to either change system-wide parameters that will affect all tables, or change statistics for each such table to ease the lock contention.

Cluster tables are referred to as volatile tables in DB2. Adding a new keyword, VOLATILE, to the CREATE TABLE and ALTER TABLE statements signifies to DB2 which tables should be treated as volatile tables. For volatile tables, index access is chosen whenever possible, regardless of the efficiency of the available index(es). That is, a table scan is not chosen in preference to an inefficient index.

Some Extra Reading

- SG24-2003-00 - High Availability Considerations DB2 for OS/390 with SAP R/3
- SG24-2078-01 - Database Administration Experiences: SAP R/3 on DB2 for OS/390
- GG24-2233-00 - DB2 for OS/390 Application Design for High Performance
- SG24-2244-00 - DB2 for OS/390 Capacity Planning
- SG24-5156-01 - Planning to Install PeopleSoft with DB2 of OS/390
- SG24-5343-00 - SAP on DB2 for OS/390: Disaster Recovery
- SG24-5486-00 - DB2 for OS/390 and Continuous Availability
- SG24-5681-00 - SAP Business Information Warehouse on OS/390
- SG24-5690-00 - SAP on DB2 for OS/390: Database Availability Considerations
- SG24-5840-00 - SAP on DB2 for OS/390: Application Servers on OS/390
- SG24-6093-00 - PeopleSoft V8 on zSeries Using Sysplex Data Sharing and Enterprise Storage Systems
- SG24-6129-00 - DB2 for z/OS and OS/390 Version 7 Selected Performance Topics
- SG24-6121-00 - DB2 UDB Server for OS/390 and z/OS Version 7 Presentation Guide
- SG24-6287-00 - SAP on DB2 for OS/390 and z/OS: System Cloning
- SG24-6300-00 - DB2 for z/OS Application Programming Topics
- **SG24-6418-00** - Squeezing the Most Out of Dynamic SQL with DB2 for z/OS and OS/390
- SG24-6549-00 - Implementing PeopleSoft 8.4 on zSeries
- SG24-6836-00 - SAP on DB2 UDB for OS/390 and z/OS: High Availability Solution using System Automation
- SG24-6847-00 - SAP on DB2 UDB for OS/390 and z/OS: Implementing Application Servers on Linux for zSeries
- SG24-6868-00 - Siebel 7 with DB2 for z/OS: Database Implementation and Administration Guide
- SG24-6871-00 - DB2 UDB for z/OS V8 - Technical Preview
- SG24-6914-00 - SAP on DB2 Universal Database for OS/390 and z/OS: Multiple Components in One Database (MCOD)
- SG24-6950-00 - SAP on DB2 for z/OS and OS/390: High Availability and Performance Monitor with Data Sharing
- SG24-6952-00 - Distributed Functions of DB2 for z/OS and OS/390
- SG24-6079-00 - DB2 for z/OS Version 8: Everything You Ever Wanted to Know, ... and More
- SG24-7078-00 - Installing Siebel 7.5.3 for z/OS and OS/390
- SG24-7083-00 - DB2 for z/OS Stored Procedures: Through the CALL and Beyond
- **SG24-7088-00** - DB2 UDB for z/OS V8: Through the Looking Glass and What SAP Found There
- REDP0162 - DB2 for z/OS and OS/390 Version 7 Selected Performance topics
- REDP0131 - SAP R/3 on DB2 for OS/390: DB2 Features That Benefit SAP
- REDP3717 - mySAP Business Suite Managed by IBM Tivoli System Automation for Linux

Web Resources

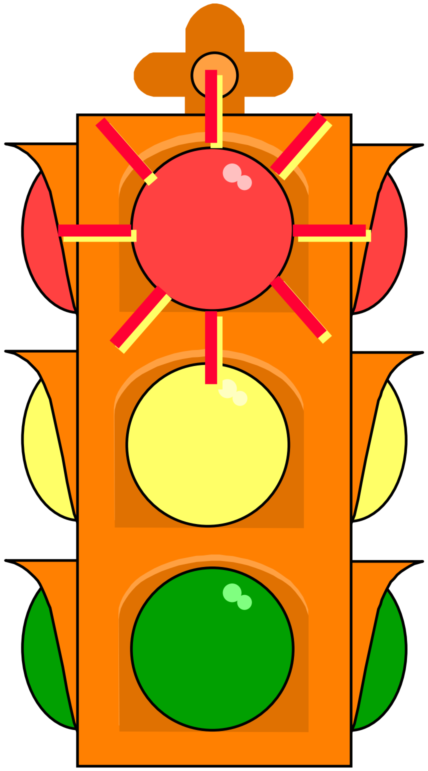
- DB2 for OS/390 - www.ibm.com/software/db2zos
- DB2 Estimator - www.ibm.com/software/data/db2/os390/estimate/
    - » Version 8 of Estimator is now available for download
- ESS - www.ibm.com/storage/hardsoft/diskdrls/technology.htm
- DB2 support/services - www.ibm.com/software/data/db2/zos/support.html
- ODBC - www.ibm.com/software/data/db2/os390/odbc/
- ODBC static profiling - www.ibm.com/software/data/db2/udb/staticcli/
- J2EE/JDBC specifications - java.sun.com/products/jdbc/
- DB2 Java Site - www.ibm.com/software/data/db2/java/
- DB2 for z/OS JDBC - www.ibm.com/software/data/db2/os390/jdbc.html
- DB2 SQLJ - www-3.ibm.com/software/data/db2/java/sqlj/
- DB2 for OS/390 SQLJ - www-3.ibm.com/software/data/db2/os390/sqlj.html

Thank You
for Attending!
Willie

ERP Performance & Availability Tips Using DB2 for z/OS

Session: **Z25**

The End



William Favero

IBM Software Group

IBM Software Services for Information Management

DB2 for z/OS Information Management Services

IBM Certified IT Software Specialist

IBM Certified Solutions Expert - DB2 Version 7 for OS/390

IBM Certified DBA - DB2 V8 for z/OS

Questions: wfavero@us.ibm.com