Reorganizing the System Catalog in DB2 Version 4

BY CRAIG S. MULLINS

Of the many new features added to DB2 Version 4 one of the most eagerly awaited by database administrators is the ability to expediently reorganize the DB2 catalog and DB2 directory in a systematic manner.

The DB2 catalog is the central repository for DB2 object and user metadata. DB2 is constantly referring to that metadata as it processes applications and queries. The physical condition of the tablespaces and indexes that comprise the DB2 catalog is therefore a major component in overall DB2 subsystem performance. Likewise, the DB2 directory contains internal control structures such as DBDs and skeleton cursor tables which can be accessed only by DB2 itself. The information in the DB2 directory is critical for database access, utility processing, plan and package execution and logging. It is quite obvious that efficient access to information this critical should be of paramount importance.

Figure 1 Reorganization Indicators

COLUMN	CATALOG TABLE	OBJECT	IMPACT
PERCDROP	SYSIBM.SYSTABLEPART	TABLESPACE	+++++
NEAROFFPOS	SYSIBM.SYSINDEXPART	TABLESPACE	+
FAROFFPOS	SYSIBM.SYSINDEXPART	TABLESPACE	++++
CLUSTERRATIO	SYSIBM.SYSINDEXES	INDEX	-----
NEARINDREF	SYSIBM.SYSTABLEPART	INDEX	+
FARINDREF	SYSIBM.SYSTABLEPART	INDEX	++++
LEAFDIST	SYSIBM.SYSINDEXPART	INDEX	+++

Prior to DB2 Version 4, reorganization of the DB2 catalog and DB2 directory was not possible. The only option for any type of "reorganization" activity was to run the RECOVER INDEX utility on DB2 catalog indexes. This rebuilt the indexes, but had no impact on the underlying data housed in the actual physical tablespace.

As of DB2 Version 4 it is permitted to execute the REORG utility on tablespaces and indexes in the DB2 catalog database (DSNDB06) and on specific tablespaces (SCT02, SPT01, and DBD01) in the DB2 directory database (DSNDB01).

When Should the DB2 Catalog and Directory Be Reorganized?

To determine when to reorganize the system catalog, DBAs can use the same basic indicators used to determine whether application tablespaces should be reorganized. Although it has always been a wise course of action to execute RUNSTATS on the DB2 Catalog tablespaces, it becomes even more important now that these tablespaces can be reorganized. The RUNSTATS utility collects statistical information that is used by the optimizer to generate access paths. Additionally, these statistics can be analyzed to determine when a REORG should be run. When RUNSTATS is run for a catalog tablespace, the statistics about that system catalog tablespace are gathered and then stored in the DB2 Catalog tables themselves!

The table contained in Figure 1 should serve as a basic guide to help in determining when to reorganize the system catalog tablespaces and indexes. (Note: The data in Figure 1 can be used for data contained in application tablespaces and indexes as well as in DB2 catalog tablespaces and indexes.)

Figure 2 Reorganization Indicators

When You REORG...	Be Sure to Also REORG...
DSNDB06.SYSDBASE	DSNDB01.DBD01
DSNDB06.SYSPLAN	DSNDB01.SCT02
DSNDB06.SYSPKAGE	DSNDB01.SPT01

Using Figure 1, the column and table name where the statistic can be found is given in the first two columns of the chart. The third column indicates whether the statistic is applicable for a tablespace or an index. The fourth column gives an indication of the impact of the statistic. A plus (+) sign indicates that you should REORG more frequently as the value in that column gets larger. A minus (-) sign indicates that you should REORG more frequently as the value gets smaller. As the number of "+" or "-" signs increases, the need to REORG becomes more urgent. For example, as PERCDROP gets larger, the need to REORG is very urgent, as indicated by the five plus (+) signs.

For the SYSDBASE, SYSVIEWS and SYSPLAN catalog tablespaces, the value for the FAROFFPOS and NEAROFFPOS columns of SYSINDEXPART can be higher than for other tablespaces before they need to be reorganized.

In addition to the guidelines in Figure 1, consider catalog and directory reorganization in the following situations:

- To reclaim space and size tablespaces appropriately when DB2 catalog and directory data sets are not using a significant portion of their allocated disk space (PRIQTY).
- When it is necessary to move the DB2 catalog and directory to a different device.
- When the DB2 catalog and directory data sets contain a large number of secondary extents.

Synchronizing System Catalog Reorganization

It is a more difficult prospect to determine when the DB2 directory tablespaces should be reorganized. The RUNSTATS utility does not maintain statistics for these "tablespaces" like it can for the DB2 Catalog.

However, it is possible to base the reorganization of the DB2 directory tablespaces on the reorganization schedule of the DB2 catalog tablespaces. In fact, in certain situations, it is imperative that specific DB2 directory tablespaces are reorganized when a "companion" DB2 catalog tablespace is reorganized. The chart contained in Figure 2 provides information on keeping the DB2 catalog and DB2 directory tablespaces "in sync."

The tablespaces in Figure 2 are logically related and DB2 recommends that you reorganize them at the same time to keep them synchronized.

Reorganizing the DB2 Catalog

There are 11 system catalog tablespaces and six directory tablespaces (refer to Figures 3 and 4). DB2 has different rules for different sets of these tablespaces. There are three groupings of tablespaces. Tablespaces:

- Cannot be reorganized at all;
- Can be reorganized using normal REORG procedures; and
- Can be reorganized using special REORG procedures.

There are only two tablespaces in the first grouping of tablespaces which cannot be reorganized at all: DSNDB01.SYSUTILX and DSNDB01.SYSLGRNG. Do not attempt to reorganize these tablespaces as DB2 will not permit it.

The second grouping of tablespaces are those that the REORG utility processes as it would any other tablespace:

- DSNSB06.SYSCOPY;
- DSNSB06.SYSGPAUT;
- DSNSB06.SYSPKAGE;
- DSNSB06.SYSSTATS;
- DSNSB06.SYSSTR;
- DSNSB06.SYSUSER;
- DSNSB01.SCT02; and
- DSNSB01.SPT01.

The third, and final grouping of tablespaces, must be processed differently than other tablespaces:

- DSNDB06.SYSDBASE;
- DSNDB06.SYSDBAUT;
- DSNDB06.SYSGROUP;
- DSNDB06.SYSPLAN;
- DSNDB06.SYSVIEWS; and
- DSNDB01.DBD01.

These six tablespaces require special "handling and care." Because they have a different internal configuration than most other tablespaces, a different calculation is required for the size of the unload data set (SYSREC) used during the REORG utility. These tablespaces contain internal links. Links are internal pointers that tie the information in their tables together hierarchically. A link can be thought of as a type of parent-child relationship. Due to these links, the BUILD and SORT phases of the REORG utility are not executed.

The WORKDDN, SORTDATA, SORTDEVT, SORTNUM options are ignored when reorganizing these tablespaces.

Also, the REORG utility cannot be restarted from the last checkpoint when used against these six tablespaces. Instead, it must be restarted from the beginning of the PHASE. Also, as mentioned before, a different set of steps must be executed during reorganization for these tablespaces.

Steps To REORG the Six "Special" Tablespaces

The following steps should be used when reorganizing the six "different" tablespaces (DSNDB06.SYSDBASE, DSNDB06.SYSDBAUT, DSNDB06.-SYSGROUP, DSNDB06.SYSPLAN, DSNDB06.SYSVIEWS, and DSNDB01.DBD01):

- Calculate size of unload data set (SYSREC);

  The SYSREC data set for the "special" tablespaces has a different format than the other tablespaces. This causes a special calculation to be required to determine its size. The equation to use is: DATA SET SIZE IN BYTES = (28 + LONGROW) * NUMROWS

  NUMROWS is the number of rows to be contained in the data set and LONGROW is the length of the longest row in the tablespace. The value for LONGROW can be determined by running the following SQL statement:

```
SELECT
      MAX(RECLENGTH)
FROM
      SYSIBM.SYSTABLES
```

```
WHERE
      DBNAME = 'DSNDB06'
AND TSNAME = 'name of
      tablespace to REORG'
AND CREATOR = 'SYSIBM';
```

- Ensure incompatible operations are not executing;
- Start database DSNDB01 and DSNDB06 for read-only access;
- Run QUIESCE and DSN1CHKR utilities;
- Take full image copy of entire DB2 catalog and directory tablespaces;
- Start DSNDB01 and DSNDB06 for utility access;
- Execute REORG utility;
- Take full image copy of entire DB2 catalog and directory tablespaces; and
- Start tablespace and associated indexes for read/write access.

Steps To REORG Regular Tablespaces

The following steps should be used when reorganizing the remaining, "regular" system catalog and directory tablespaces:

- Calculate size of unload data set (SYSREC) using the normal calculation:
  DATA SET SIZE IN BYTES = LONGROW * NUMROWS
  In this case it is unnecessary to add the additional 28 bytes to the length of the longest row. This is because these system catalog tablespaces do not utilize links;
- Ensure that incompatible operations are not concurrently executing (see the next section for an explanation of incompatible operations);
- Start the tablespace and its associated indexes for read only access;
- Run CHECK INDEX on all indexes associated with the tablespace that is being reorganized;
- Take a full image copy of the entire DB2 catalog and directory tablespaces;
- Start the tablespace and its associated indexes for utility access;
- Execute the REORG utility;
- Take a full image copy of the entire DB2 catalog and directory tablespaces; and
- Start the tablespace and any associated indexes for read/write access.

These steps should be familiar to you because they closely follow the steps executed during the reorganization of an application data tablespace. There are several additional required steps added as precautions because of the critical nature of the DB2 catalog and directory.

Figure 3 DB2 Catalog Tablespaces

Database Name: DSNDB06	
Tablespaces:	
SYSCOPY	contains image copy information (1 table)
SYSDBASE	contains database object information (14 tables)
SYSDBAUT	contains database and database authority information (2 tables)
SYSGPAUT	contains resource authority information (1 table)
SYSGROUP	contains storage group information (2 tables)
SYSPLAN	contains plan information (5 tables)
SYSPKAGE	contains package and stored procedure information (8 tables)
SYSSTATS	contains optimization statistics (5 tables)
SYSSTR	contains translation and check constraint information (3 tables)
SYSUSER	contains user authority information (1 table)
SYSVIEWS	contains view information (4 tables)

Figure 4 DB2 Directory Tablespaces

Database Name: DSNDB01	
Tablespaces:	
DBD01	contains database descriptor information (1 table)
SCT01	contains skeleton cursor table information (1 table)
SPT02	contains skeleton package table information (1 table)
SYSLGRNG	contains recovery log range information (1 table)
SYSUTILX	contains utility processing information (2 tables)

Catalog Reorganization Restrictions

In addition to the procedures outlined previously, there are several restrictions on the manner in which the REORG TABLESPACE utility can be used with the system catalog tablespaces. First, recall that the SYSUTILX and SYSLGRNG tablespaces in the DB2 Directory cannot be reorganized. Furthermore, when reorganizing the DB2 Catalog (DSNDB06) and DB2 Directory (DSNDB01) tablespaces the following options cannot be used:

- The UNLOAD ONLY option is not permitted; and
- The LOG YES option is not permitted as image copies are explicitly required following a catalog and/or directory reorganization.

Also, the reorganization of these two specific tablespaces are treated differently in the manner in which they are tracked by DB2. Generally, DB2 will record the reorganization of any tablespace in the SYSIBM.SYSCOPY system catalog table. However, DB2 records the reorganization of the DSNSB06.SYSCOPY and DSNDB01.DBD01 tablespaces in the log instead.

Finally, in many 24 X 7 environments, it may be necessary to reorganize the system catalog and dictionary while it is being accessed. However, because of the central nature of the system catalog and directory to the operation of DB2, there are restrictions on concurrent activity during catalog reorganization. The restrictions are:

- ALTER, DROP and CREATE statements cannot be executed during the reorganization of any DB2 catalog or DB2 directory tablespace with the exception of SYSIBM.SYSSTR and SYSIBM.SYSCOPY;
- The BIND and FREE commands cannot be issued when the following tablespaces are being reorganized: SYSIBM.SYSDBAUT, SYSIBM.SYSDBASE, SYSIBM.SYSGPAUT, SYSIBM.SYSPKAGE, SYSIBM.SYSPLAN, SYSIBM.SYSSTATS, SYSIBM.SYSUSER and SYSIBM.SYSVIEWS;
- No DB2 utility can be running while SYSIBM.SYSCOPY, SYSIBM.SYSDBASE, SYSIBM.SYSDBAUT, SYSIBM.SYSSTATS and/or SYSIBM.SYSUSER are being reorganized;
- No plan or package may be executed during the reorganization of SYSIBM.SYSPLAN and SYSIBM.SYSPKAGE; and
- The GRANT and REVOKE statements cannot be issued when REORG is being run on SYSIBM.SYSDBASE, SYSIBM.SYSDBAUT, SYSIBM.SYSGPAUT, SYSIBM.SYSPKAGE, SYSIBM.SYSPLAN and/or SYSIBM.SYSUSER.

Conclusion

The ability to reorganize the DB2 catalog and directory tablespaces provides the DBA with a potent new tool for their system tuning arsenal. If you have not yet started to execute RUNSTATS on the system catalog tablespaces, begin to do so immediately. Likewise, it is a smart move to begin analyzing these statistics even before you migrate to DB2 Version 4. This will enable you to determine whether your system catalog will need to be reorganized even before you move to DB2 Version 4. Good luck and happy REORGing.

Craig S. Mullins is a senior technical advisor and team leader for analyst relations at PLATINUM technologies inc. His duties include interfacing with industry analyst groups, authoring white papers, technical support via the Internet and speaking at user groups and industry conferences. He is also a member of the editorial board for EDGE magazine.

Sharing DB2 Applications Code: Bridging CICS, IMS and TSO

— Continued from page 30. —

required routines which perform DB2 COMMIT or ROLLBACK processing or DL/I checkpoint/restart calls. The entry point is WHEREAMI and returns a 4-Character string containing the values or "TSO," "IMS" or "CICS."

A final note about the control program was that COBOL-II was considered for writing the code. There were two problems: the possibility of mixing COBOL-II and VS/COBOL in an execution environment; and the problem of passing a variable parameter list to the interface. The COBOL-II implementation of POINTERs makes it difficult to determine the count of parameters passed into a COBOL program and therefore to decide exactly what to pass along to the real IBM support modules. Therefore, the code is in Assembler Language.

Final Comments

The requirement to efficiently implement and support application systems is one of the most critical issues in mainframe based systems. Processes must be streamlined and simplified to be more reliable and understandable in support of this effort. Furthermore, many systems evolving from older IMS to CICS systems further complicate the development environment.

The methodology described previously makes it possible to develop and implement standard, reusable DB2 code across the three major DB2 application development environments. This, and other similar techniques, make development and transition of systems more consistent because they are based on proven components. Support functions can be implemented in the existing environment and migrated to the new environment without modification. This means systems are more reliable while costing less to develop and support.

Fred Schuff is a system design and integration consultant in the Philadelphia area. He has 20 years of experience with large IBM systems and has been working with multi-platform environments for the past five years.

Charles Alexander has been a programmer for the past 25 years, specializing in mainframe application development and design. Charles is currently an independent consultant on assignment at SUN Company in Philadelphia, Pa.