

DISASTER

RECOVERY

JOURNAL

The Journal Dedicated to Business Continuity Since 1987

The Impact Of DB2 Version 4 On Recovery

By Willie Favero

DB2 is once again the talk of the town with the arrival of Version 4. So it is time to take a look at how the latest release of IBM's Relational Database Management System affects the recovery process. The recovery changes in Version 4 come in two flavors. One is data sharing, an area that will introduce a whole new way of thinking about recovery.

The second change in Version 4 is enhancements to the basic recovery function. These are the functions that everyone will need to use at some point in time and therefore are the focus of this article.

Utility Enhancements

Some changes to the utilities by Version 4 have been as subtle as increasing the number of BSAM buffers used by the utilities from 8 to 20. This single modification can result in an elapsed time saving for the COPY and RECOVER INDEX utilities.

However, as you read on you will soon see that some overdue improvements have been made. The next few paragraphs will focus on the COPY, RECOVER, and MERGECOPY utilities and discusses some of the more significant enhancements to these utilities.

COPY

Before you can think about recovering your data, you need to make some kind of copy. Version 4 has made a number of enhancements to the DB2 copy process. The first is DFDSS Concurrent Copy. Yes, it is true that the concept of using concurrent copy to backup DB2 data was introduced in DB2 Version 3. However, it had a few limitations. Taking advantage of concurrent copy could add to the length and complexity of a recovery. The single greatest problem with concurrent copy was that the process did not register the copy with DB2.

The backup had to be tracked manually outside DB2. This step is not necessary if using DB2's image copies. Once the correct backup is located, it must be restored using DFDSS RESTORE, again outside

DB2. Finally a DB2 recovery job can be run using the LOGONLY keyword introduced in Version 3. All this translated to changing existing procedures and tracking another set of backups.

With Version 4, concurrent copy is a manageable and streamlined alternative. The entire process is now under DB2's control. When the DB2 COPY utility is executed using the new keyword CONCURRENT, concurrent copy is invoked to backup the data and backup is now registered in SYSCOPY. If SHRLEVEL(REFERENCE) is specified with CONCURRENT, a quiesce is performed and a Q is registered in SYSCOPY's ICTYPE when the quiesce completes.

No quiesce is performed for a SHRLEVEL(CHANGE) copy. At successful completion of the concurrent copy, the Q is changed to an F (see figure 1). If the copy fails, the Q is left in SYSCOPY. In addition, because it is a concurrent copy backup, the SYSCOPY column STYPE is set to C (for concurrent). During the restore phase of a subsequent DB2 recovery, SYSCOPY is examined for the most current image copy record.

If the record found contains a C in the STYPE column, the recover utility invokes DFDSS to restore the concurrent copy backup. Once the restore completes, the utility continues with the log apply phase.

Because the utility is working with a concurrent copy backup rather than a DB2 image copy, the apply phase works analogously to a LOGONLY recovery.

Recover applies log records starting with the recovery base RBA (HPGRBRBA) in the file page set header page. The result is a copy that is almost transparent to the user and a recovery that is transparent to the DBA.

There are some restrictions to be aware of if using concurrent copy to produce data backups. If the PAGE or ERRORRANGE keywords are specified by the recovery job, copies made by concurrent copy and registered in SYSCOPY are bypassed.

Also, DSN1COPY, DSN1PRNT, and DSN1COMP will not run against the output from a concurrent copy. These stand-alone utilities require a DB2 image copy as input. They do not work against a concurrent copy dump.

For example, a DSN1COPY with the OBIDXLAT parameter, cannot use a concurrent copy backup to recover a dropped table space. Next, you cannot invoke the concurrent copy function from the DB2I utility panel.

In addition, the batch id specified on the COPY or RECOVER utility job must have the authority to invoke DFDSS concurrent copy or restore. Finally, an incremental image copy is not allowed after running concurrent copy. An incremental image copy requires a full DB2 image copy be run first.

Making concurrent copy easier to use is not the only change to the copy process. The basic DB2 COPY utility has also been improved. COPY in Version 4 no longer turns off the "page dirty bit" for full or incremental image copies. This can significantly improve the elapsed time of the COPY utility.

In prior releases, DB2 not only had to turn the dirty bits off in each page it copied, it had to write those pages. This meant the utility had to run as an update program. Because it no longer modified the dirty bit, Version 4 saves the one write for every page it reads.

The only pages that need to be updated and rewritten are the space map pages. COPY has a second improvement affecting only incremental image copies. In the past, DB2 used the modified page bit in the space map page to determine which pages needed to be copied. It then performed a random I/O for that page. In Version 4, an incremental image copy uses list prefetch to read the modified pages.

RECOVER TABLESPACE

Now let us turn our attention to the RECOVER TABLESPACE utility. Each DB2 subsystem is defined as either the local site or the recovery site at installation time. The current site is controlled through the SITETYP keyword on the DSN6SPRM macro in DSNZPARMS. The current site can be set to RECOVERYSITE or LOCALSITE. THE DB2 COPY utility can create copies for either site by specifying COPYDDN to define the local copies and RECOVERDDN to define the recovery site copies. In releases prior to Version 4, the DB2 RECOVER utility always used the current setting of SITETYP. This determined whether to use the copy registered to DB2 as the local site copy (SITETYP=LOCALSITE), or the recovery site copy (SITETYP=RECOVERYSITE), during a table space recovery. With Version 4, two new keywords have been added to the RECOVER utility, LOCALSITE and RECOVERYSITE. Specifying either of these keywords overrides DB2's use of current site when choosing a copy during a recovery.

Now, if an I/O error occurs on a copy input at the local site and the recovery site copies are still on-site and available for use, RECOVER can be rerun with the new RECOVERYSITE keyword and use the recovery site copies.

There is an additional keyword added to RECOVER, TOLOGPOINT. TOLOGPOINT is used to recover to a prior point in time in a data sharing environment. TOLOGPOINT has the same intent as the TORBA keyword. The difference is that data sharing will no longer use RBAs, so an alternative to an RBA is needed. Once DB2 is enabled to a data sharing group, it uses a value derived from a timestamp called the log record sequence number (LRSN) to identify log records. Therefore, to recover to a prior point in time, one can use TOCOPY or the new TOLOGPOINT specifying the LRSN of the recovery point. In a non-data sharing environment, an RBA can be specified using the TOLOGPOINT keyword. For the first time, RECOVER can leave a table space in check pending if that table space contains a table with a Check Constraint defined on it. If a table has been created or altered to use check constraints and the table space is recovered to point prior to the constraint being added, the table space is left in check pending status. Why? When the constraint was added to the table, the data in the table was checked to make sure it did not violate the new constraint.

This could have happened immediately when the constraint was added to the table, or by the CHECK DATA utility if the alter placed the table in check pending. When the point in time recovery is run, the constraints still exist in the DB2 catalog.

The data used as past of the partial recovery is from a point before the data was checked for constraint violations. The result is a check pending condition when the recovery completes.

Finally, a new precaution to add to the DB2 Catalog recovery documentation. The DBD for DSNDB06 is now defined in DBD01 rather than in a DB2 DSECT.

This allows a DB2 user to modify certain attributes of the Catalog. An example is creating user defined indexes on the Catalog. Because the Catalog DBD is now part of DBD01, it is important to always recover the Catalog and Directory to same point in time to keep them in sync.

RECOVER INDEX

A work data set can be eliminated from the RECOVER INDEX utility to improve the utilities overall performance. Simply remove the DD statement referenced by WORKDDN option.

This keyword defaults to SYSUT1. Removing this DD statement allows the recover index utility to avoid writing and reading from the work data set. This option should only be used when the recover

index job can be restarted from the very beginning.

This is significant because RECOVER INDEX also has enhanced restart capacities. In prior releases, RECOVER INDEX was always restarted from the beginning of the job. It can now be restarted at the beginning of the UNLOAD, SORT, or BUILD phase.

In addition, the BUILD is also restarted after the completion of the last index built.

MERGECOPY

MERGECOPY can specify COPYDDN or RECOVERDDN to define up to 4 outputs, 2 for local site and two for recovery site processing.

This option is only valid when producing a new full image copy using the NEWCOPY YES keyword. In previous releases of DB2, MERGECOPY could only combine the incremental for the current site, whether it was the local site or recovery site.

Down Level

Page Detection

With more users turning to DFDSS's concurrent copy, full pack backups, and other non-standard methods for copying DB2 data, a mechanism is needed to insure that data restored outside DB2 is the correct information.

DB2 Version 4 provides that mechanism with down level page detection. Before explaining how this was implemented, let us take a look at the problem.

Using figure 2 as an example, a full pack backup is taken at the end of day one. Day two is then just a normal day with end users making modifications to their DB2 data and maybe even running some utilities like LOAD or REORG. Sometime during day three a DASD error occurs.

In this case, a DASD analyst is called in rather than the DB2 DBA. The DASD person notices that a full pack backup exists for the volume and uses it to correct the device error. When the users come on-line on day three, their data looks the same as it did on day one. All work performed on day two has been lost.

How does Version 4 prevent this from happening? DB2 Version 4 has a new DSNZPARM parameter SYSPDFRQ on the macro DSN6SYSP.

This value sets the interval of update for the levelid (HPGLEVEL and HPGPLVL), new values in the header page. By default this interval is set to 5 checkpoints. If SYSPDFRQ is set to 0 (zero) pageset Down level detection is disabled.

At each interval the levelid is updated with the current RBA or the current LRSN if data sharing has been enabled.

At close, or pseudo close, all of the table space pages are written back to DASD. The header page on DASD now reflects the last update to the levelid.

At that time, the levelid from the header page is also written to the DB2 log and to SYSLGRNX. The next time the table space is opened for an SQL statement or utility, the levelid (HPGLEVEL) in the header page of the table space is compared to the levelid recorded in SYSLGRNX.

If they are not equal, DB2 will not allow access to the table space until the condition is corrected. During a DB2 restart, the levelid recorded on the log is compared to the levelid in the header page.

Again, if they are not equal, DB2 will not allow access to the table space. Restart could use the previous

level-id (HPGPLVL) because restart assumes the current level-id was not successfully written because of system failure.

There are a few utilities that avoid down level page detection. They are RECOVER, LOAD, REPAIR, and DSN1PRNT. RECOVER or LOAD REPLACE should be used to correct the problem.

If necessary, REPAIR can modify the levelid in the table space header page as a last resort.

For diagnostics, DSN1PRNT can be used to determine the current levelid. DB2 will not accept a -START DB ACCESS(FORCE) to circumvent the error.

Conclusion

There have been many other changes made to DB2 in Version 4 that will help reduce the elapsed time of a recovery, including SYSLGRNX replacing SYSLGRNG. SYSLGRNX uses two new indexes to eliminate long search times of hash synonym chains.

This will reduce the time needed to find update log ranges. A logical page list (LPL) has been added.

Now if a page is logically in error or the page problem is undetermined, it is recorded in the LPL.

If an LPL error can not be resolved, then the page is moved to the error range list. This could reduce the number of recoveries needed because of page errors caused by things like "device not ready" logical errors.

There are also those small changes that often go unnoticed like defaults. In the past when DB2 was installed the "Recording Max," the number of archives recorded in the BSDS before wrapping around, defaulted to 500. In Version 4, it now defaults to 1000. This is a much safer default.

However, it could throw your copies and archives out of sync. Then there are the improvements that may just eliminate the need for a recovery.

The new WITH RESTRICT ON DROP for an SQL ALTER or CREATE table statement falls into that category. If the clause is added to a table description, that table cannot be dropped until the clause is removed. If the table cannot be dropped, that means that the table space containing the table cannot be dropped, and the database containing the table space that....

I think you get the picture. Even the archive log process has been improved. The archive logs can now be written to two different device types.

In addition, there is a new extension to the -ARCHIVE LOG command, CANCEL OFFLOAD, that allows an offload to be canceled with no affect on DB2.

Version 4 is definitely another giant step forward for DB2 for MVS/ESA. The tasks that revolve around a recovery, something everyone dreads and no one ever has enough time to complete, especially benefit from this latest release.

Willie Favero works for BMC Software in their education and services group where he conducts education and consulting for the company.

This article adapted from Vol. 9#1.

[*DR World Main Index*](#) / [*Return to DRJ's Homepage*](#)

Disaster Recovery World© 1999, and Disaster Recovery Journal© 1999, are copyrighted by Systems Support, Inc. All rights reserved. Reproduction in whole or part is prohibited without the express written permission form Systems Support, Inc.