DATA INTEGRITY AND RECOVERY

AUTHOR: TERESA BRZOZOWSKI

CAROLIAN SYSTEMS INTERNATIONAL INC.
3397 American Drive, Unit 5, Mississauga, Ontario L4V 1T8 CANADA

Database failures and resultant recovery efforts cost HP3000 users thousands of dollars every day in lost processing time and inconvenience. While this paper provides a discussion of IMAGE failure types and various methods of recovery, it also intends to educate the reader as to how some basic database design and implementation procedures can act as proverbial "ounces of prevention" - protecting you and your company from having to needlessly exist and suffer with logically and physically broken databases.

IMAGE FAILURE MODES

A quick review of IMAGE failure modes reveals that such common occurrences as system failures or hangs, disc media failures, datacomm line failures or application failures can all bring database processing to a screeching halt. Failures can also result in physical or logical damage to the database, with physical failures resulting from having bad data on disc (filesys), broken chain pointers or inconsistent root files. Logical failures can result because of missed updates, puts or deletes or missing delete flags. Whatever the cause, there are several standard and some new ways to repair the damage.

CLASSIC IMAGE RECOVERY

A standard method of IMAGE recovery is to restore your most recent copy of your database and forward recover using DBRECOV. However, there are some major deficiencies with this method of recovery. First and foremost, the process is extremely time consuming as it keeps users away from productive processing.

Also, DBRECOV uses a technique of recovery known as staging, whereby the restored DB is updated from the log file via staging files. The problem with staging is that large numbers of transactions can be ignored if an "end" is not found, resulting in these transactions not being applied to the database. The result can be a great deal of time and effort spent on forward recovery, with no guarantee that the recovery will be complete.

INTRINSIC LEVEL RECOVERY - ILR

If a failure occurs during an actual DB intrinsic such as updating, ILR can ensure physical integrity of your database by undoing the intrinsic. The problem with "undoing" is that with IMAGE databases, in some instances the log file and the database may not agree! Improvements made to TurboIMAGE have alleviated this problem.

TURBOIMAGE RECOVERY

With Turbo, ILR will complete the intrinsic call so that the logfile and the database agree, as opposed to just "undoing" it. Turbo allows you to forward recover with DBRECOV as does IMAGE, but it also allows you to initiate a rollback recovery.

Rollback recovery is a more timely method of recovery as it eliminates the need for a DBrestore and to reapply logged transactions to the database. Rollback recovery allows users to bring their current database up, and back out the last incomplete transactions, while complete transactions are left in place.

The use of ILR and Rollback recovery will generally ensure that more data is recovered than is possible with roll forward techniques. This is due to the fact that ILR with rollback recovery requires physical logging.

Physical logging ensures that the changes to the database are recorded and written to the log file as they occur. This prevents the log record from remaining in memory where they can be lost in the event of a failure.

Despite the time savings that can be realized with Turbo's newer recovery features, neither these or IMAGE recovery procedures are of assistance with another common occurrence that results in logically broken databases program aborts.

RECOVERING FROM PROGRAM ABORTS

Programming bugs, user errors and datacomm line failures are just a few of the occurrences that can result in a database becoming logically corrupt. To date, HP3000 sites have had to live with the fact that once their databases have become logically corrupt, that they have to endure this inconvenience until a full recovery procedure can be initiated.

PROBLEMS ASSOCIATED WITH ABORT RECOVERY

Again user downtime is the penalty that must be paid as users have to terminate, partial transactions are deleted or completed, and then users are allowed to access the machine again. However, if strong locking is not in place, the transaction interaction that has been rolled out can inadvertently undo a completed call. The real solution to this dilemma is to have a "net change" rollback. This is currently unavailable, as a "net change" rollback requires a detailed and intimate knowledge of the application.

SOLUTIONS - HOW TO MINIMIZE RECOVERY HEADACHES

The benefit of such facilities as DBRECOV and Rollback recovery can be greatly enhanced if you implement the following safeguards.

1. Turn on logging - despite persistent misconceptions, logging **does not** significantly degrade the performance of your machine. If you are not logging you have precluded yourself from virtually all methods of recovery.

2. Use Begins and Ends - Without DBbegins and DBends, by definition no logical transactions exist. Therefore, database logical integrity is impossible to determine. The best that can be done is to provide for an audit trail of physical transactions.
3. Strong Locking - Some method of strong locking should be implemented. Without strong locking, a transaction can interact with another transaction before it has completed, thereby making the result of a rollback recovery questionable.
4. Turn on ILR - Turning on ILR will ensure that your database will always be physically intact.

Implementation of these four key points is crucial if you are to ensure database integrity and ease of recovery for your company. They can result in tremendous reductions in user downtime and the time is spent on recovery procedures. There is however, another alternative method of database recovery, that when implemented with the aforementioned safeguards, will render downtime due to the initiating of recovery, or the existence of logically corrupt databases due to program aborts, a thing of the past.

AN ALTERNATIVE - DYNAMIC "ROLLBACK"

A facility that provides a dynamic rollback, will actually undo an aborted transaction as the abort occurs. This "real time removal" of an aborted transaction will result in your database always being logically intact. Without the existence of incomplete transactions in your database, it would also unnecessary to have to take the system down to initiate a cleanup.

Such a utility does exist, and is actually one product for the HP3000 from the Carolian Systems Research and Development group. Known as INTACT, this product provides these major capabilities which have been previously unaddressed and unavailable to HP3000 users.

your database, it would also be unnecessary to have to take the system down to initiate a cleanup.

For the HP3000 classics, Carolian Systems has developed a real-time roll-out utility called INTACT. Hewlett-Packard is also seriously investigating this data integrity issue for the Spectrum series, however, dates for the release of such a capability have not yet been released.