Backup Strategies

QUICK STUDY

DEFINITION

Backup strategies take many different forms, depending on exactly what is backed up (and when, and to where), whether the technology uses compression, how far apart copies are made and other factors.

BY RUSSELL KAY

EVERY IT manager knows the critical importance of regularly backing up computer systems and data . . . and of being able to restore any or all of them in the event of a system outage, hardware failure, natural disaster or other data loss.

For a long time, daily backups were usually done by writing copies of files to magnetic tape. This was typically an overnight batch job that ran when no other regular production work was scheduled. Periodically — perhaps once a week — a full or complete backup of all data and systems would be made.

In a technique called data reduction backup, files were typically made smaller through some form of lossless compression, such as Zip files, before being written to tape. A related option, called a mirror backup, skips the compression step and writes to another disk, allowing backup files to be read and managed using normal system tools.

But the amount of data that organizations use and store has grown rapidly, and with this comes the need to keep systems running for longer periods of time (up to and including around the clock).

Given the ever-decreasing available times during which backup could be done (known as the backup window) and the increasing length of time required to actually perform that backup, corporate IT found itself caught in a bind: It couldn't guarantee to keep the system running unless it had up-to-date backups, but neither could it shut the system down, even partially, so that it could actually do those backups.

Partial Backups

A number of strategies have been developed to resolve this dilemma. The first is partial backups. These depend on the existence of full backups made at regular intervals, and the idea is to save time by backing up only those files that have been changed, knowing that you already have a backup of the other, unchanged files.

To determine which files have changed, the backup software looks at the modification date and time of every file on the system, and if a file's time stamp is later than the last full backup, that file is included in the next differential backup. To restore files, whether individually or the complete system, you have to first restore the latest full backup and then the latest differential backup. Obviously, this type of restore operation is slightly more complicated than one from just a full backup.

But as the number and size of high-activity files keep growing, these differential backups can take nearly as long to complete as a full backup, which is much easier and simpler to restore. So someone got the idea of backing up only those files that had changed since the last differential backup.

This three-step scheme is called incremental backup, and yes, it does reduce the amount of data that has to be backed up. It seems a pretty good trade-off until you actually have to restore something. You first restore the most recent full backup — so far so good — and then restore the latest differential backup; finally, you restore every one of the intervening incrementals since that differential. All this is necessary to make sure that all files are current to the last backup.

Here's a quick example. Say a full backup was done on Saturday and the system crashed the following Friday, with differential backups made on each intervening day. After restoring the full backup, you would then have to restore, in chronological order, the backups from Sunday, Monday, Tuesday, Wednesday, Thursday and Friday. Whew!

In addition to the time needed to make all those restores, just the effort in mounting and remounting all those tapes would be significant. Automated hardware, including tape cartridge libraries and jukeboxes, can help out some with this process, but a differential restore is not a trivial event, especially if your systems are large enough that the full backups are done less often than weekly.

Incremental and differential backups can be combined so that the incremental copy includes all changes made since the last full or differential backup. This requires even more careful record-keeping and tape inventory, but it can make for somewhat faster, if more complex, restores.

Continuous Backup

Another problem with those older backup schemes is that they aren't well suited to transactional and real-time database-driven systems, where it is essential to back up every stage of every transaction, file change, disk write or I/O operation. So far, the best response to this need appears to be continuous data protection.

With CDP, also called continuous or time-based backup, you automatically capture, on disk in a separate location, every version of data that a user saves. With this technique, you can restore data to any given point in time, up to and including the last saved disk write and I/O operation prior to a failure.

An important factor that differentiates CDP from RAID, replication or mirroring is that those techniques only protect against a hardware failure by saving the most recent copy of the data. CDP will also help you recover from data corruption, by letting you find the exact point at which data has become corrupted.

One issue is granularity: How big a chunk of data do you save for each type of application? The whole file, or just the changes? Complete mailboxes, or individual e-mail messages? Database files and indexes, or transaction logs? Most CDP products save just the changed byte or disk block, not the entire file. Change one byte of a 10GB file, and CDP backs up only the changed byte or block. Traditional incremental and differential backups copy the entire files. Thus, CDP frequently needs less space on backup media.

Snapshots

A somewhat different approach that isn't considered full CDP uses a snapshot methodology, recording complete system states at regular intervals (potentially as often as every few minutes). Snapshots rely heavily on pointers to the original volume, which must be intact.

Snapshots can usually be created quickly, and they can be used to restore or re-create point-of-time disk states. But snapshots are not backups, and they must themselves be backed up if they are to support any recovery from disk crashes or other physical failure.

Backup Strategies Compared

BACKUP SCHEME	TYPICAL MEDIUM	ALL OR SOME DATA?	DATA COMPRESSED?	MAJOR ADVANTAGE
Full backup	Tape	All	Yes	Simple
Differential	Tape	Some	Yes	Saves time
Incremental	Tape	Some	Yes	Very fast
Mirror	Disk	All	No	Simple, fast
Snapshots	Disk	Some	No	Very fast
Continuous	Disk	Some	No	Efficient media use

Kay is a Computerworld *contributing writer in Worcester, Mass. You can contact him at russkay@charter.net.*

Are there technologies or issues you'd like to learn about in QuickStudy? Send your ideas to quickstudy@computerworld.com

To find a complete archive of our QuickStudies, go online to **computerworld.com/quickstudies**