Have You Really Backed up?

Tape problems

When you desperately need to restore data will it be there? Is your software the right choice for such an important task.

BY ADRIAN KERTON

Your hard drive has crashed, but there's an inner contentment. Not only did you back up the whole system last night, but you keep a spare new hard disk drive at hand for just such an eventuality.

No Problem; just put in the new disk, partition and format it, add the operating system, install the backup programme, (or on Unix and Linux use tar or cpio), and watch the tape drive recover the whole lot whilst you have some coffee. You'll have it all restored before most people get in for work.

You start the restore, but why is the tape drive thrashing around so much? Your heart sinks as you realise the restore is not restoring. An error message appears on the screen. It doesn't matter what the message says, you know you are now in deep trouble. The restore has failed and you know you cannot get at your information!

You eject the tape and put in the previous backup tape, OK it's last week's, but that's better than nothing. Misery sets in as the same thing happens again. The phone rings. Your boss and his boss want to know why their computers are not on line. You start to explain and they make it quite clear that if you cannot get their computers on-line in an hour, you won't have a job at the end of that hour.

Why? What could have brought you to this miserable episode?

Well there are a number of causes and it's probably worth examining them to make sure you don't get caught like this. Analysts have determined that 20-30% of backups fail, and the user doesn't even know it.

It doesn't matter which technology you are using to backup; tape, disk, optical, whatever, there are some golden rules you need to follow. Generally the backup hardware will be dictated by the amount of data you want to backup, but don't be fooled by what at first sight appears to be obvious.

Case 1

You're safe. Your drive has Read After Write (RAW).

Just about all tape drives have read after write capability. This means that there is a read head positioned just after the write head and the tape drive verifies that what it reads is exactly what it has written. If there is drop out on the tape Read After Write will detect it, the backup application will try to write the data again and if there are problems it will move down the tape and write again on a good piece of the tape, so no problem there! Or is there?

Some backup applications rely on the read after write function within the tape drive to serve as the backup verification mechanism, but there are a lot of hurdles in the way of the data trying to get to the tape head.

If the data is going across a network then the problems are magnified as corruption can occur anywhere in the network hardware or software before it gets close to the backup device. Read After Write won't help you if you present corrupt data to the tape drive.

Consider data coming off a disk and going to a SCSI tape drive on the same computer. It travels from the disk, onto the bus to memory and then back from memory to the bus to the SCSI host bus adapter (HBA) where the software driver has to be correctly matched to the operating system. Then through the adapter hardware to the SCSI cable to the tape drive, where the tape drive's firmware needs to match the adapter card driver. Finally through the tape drive hardware.

Within a tape drive, the data presented to the tape is often manipulated in structured ways to ensure that it gets the best distribution of flux transitions on to the tape. This makes sure your data is in the most robust format there can be. It is not unknown for something to go wrong between the data connector and the write head. In such cases Read After Write will report all is well because the read head reads the data that the write head wrote, but this data is not the data you wanted to write! It is corrupt, and if something has gone badly wrong it could be random data.

The result? Garbage on the restore.

Golden Rule

Don't rely on RAW or backup software (including Unix/Linux embedded apps) that relies on Read After Write. **ALWAYS** run a verify pass on your backup. If the backup application does not support verify, ditch it for one that does and do it now before it is too late.

Case 2

You go to restore a data file and find it is not on the backup set.

Why? Because the backup application had a complicated user interface and you misunderstood the include/exclude feature on the backup application, or you mistyped the latest free backup command line programme by one letter. Result you only backed up system files when in fact you wanted to include only data files.

This can easily happen when a new job is created, because once a backup job has been created it does its job each time running in the background, and the administrator forgets all about it. When a new job is required the administrator has to "relearn" the application because

it is used so rarely, sometime only once every two or three years. Often, during the needs for a new backup job the administrator has changed, so the person creating the new backup job has to start from scratch with a package they have never seen before and no one else is around to act as a mentor.

Golden Rule

ALWAYS try a restore from a backup whenever a new backup job has been configured (to a test directory is useful) to make sure the files you want are actually there. You should always do this even if you have run a verify pass, as this will only verify that the files you selected to be backed up are actually there. If you selected the wrong files, verify alone will not help you.

It helps if your backup package is easy to use and doesn't have too many bells and whistles to learn. Don't choose a backup package that does everything, unless you really need the extras.

Case 3

You have backed up, run a verify pass and a restore, but 3 months later the restore fails with some error message, that usually says the restore will be aborted. Typically tar or cpio will generate "tape I/O read error" and the restore will be aborted.

Why? The backup application met a bad spot on tape and quite rightly found an error because it couldn't read the data properly. Now you have the first few files from the backup, the bulk of it is still on the tape. This is typical of a backup application that is just a user interface built on top of tar or cpio.

Another problem that might arise is when the backup application uses multi-streaming from different client systems to the same tape. In this technique, data from one client group is interleaved with data from other client groups onto the same tape. This means that any particular client group's data will be divided and spread amongst the data of the other client groups on the backup media. If the backup is large it may have spread over a number of tapes. The danger in this approach is if one block of data cannot be read during a recovery, data from the multiple clients will be lost. Also restores are complex requiring the management of the multiple tape sets just to recover a single client system.

Golden Rule

Choose a backup application that has built in error handling. Surprisingly very few backup applications can satisfactorily accommodate errors during a restore. Check with the software company to understand what they do to ensure the availability of the data.

An application's bells and whistles are no good if the underlying technology cannot deliver the data. Your data is important, so meticulous care should be taken to check a backup software's capabilities to fully understand the level of protection it affords.

Case 4

You backed up with a verify pass, the restore runs perfectly, but then the complaints start rolling in. The data has errors, some files are in error with characters missing.

Why? When a backup application is based on the cpio format, the checksums used to verify the data's accuracy are only calculated on the meta data (data "about" the data block), and does not checksum the actual data. Therefore a cpio verify pass cannot verify the actual data is correct, only that the header information is correct.

Some backup applications verify the backup by conducting test restores on random backup sets. The same issue previously addressed applies here. If the backup data hasn't been 100% verified, users can still experience aborted restores because corrupt data can still be experienced. If the first bit of the restore is bad, the entire backup will be lost even though segments of the backup set proved to be accurate.

It should be noted that a tar archive can be fully verified using a bit by bit check against the disk. This doubles the time it takes to do a complete backup. Nothing must change on the disk between the backup and verify, otherwise errors will be generated and each error will have to be investigated. This approach is impractical because of today's shrinking backup windows.

Some applications note the problems with a backup and record them in the fault log. It is very easy to forget to check the log, particularly if it is someway down the directory chain, so you will not know if your backup has failed.

Golden Rule

Make sure your backup application incorporates the checks and balances to assure that the data you believe you backed-up actually made it to the backup media accurately and can be successfully and accurately recovered. Without this assurance, all other application functionality is window dressing. Make sure your backup application has some sort of notification that alerts you when there is a problem with the backup, usually by email.

Finally

"I don't need backup – I've got RAID."

RAID is fault tolerant, it is not fault free. The Internet is full of tales of RAID arrays that fail. Remember also that users deleting their data is one of the most common causes of lost data, and in that case RAID will not help you. If the building catches fire, the RAID array may be lost, but a tape backup made with a reliable backup package and stored off-site will save the day.

Backing-up data is a simple concept; just move data to a safe place and bring it back when it's needed. In reality, how this work gets done is very complex. The process should not be an "art form," but good science and engineering.

The availability of your data, and your sanity, depends on it. ■

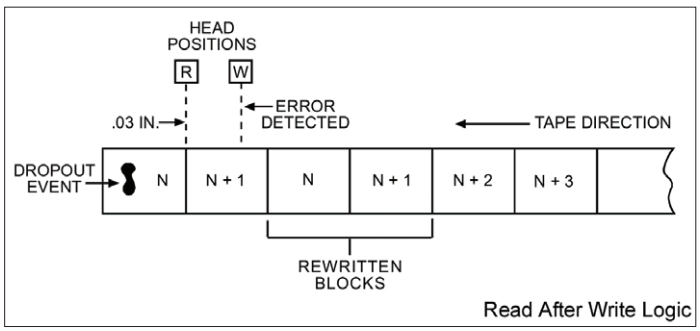

Figure 1: Read After Write Logic on the tape