



## ► Building Reliable Newspaper Systems - Protecting Data

By Richard J. Cichelli, SCS President

This article will help you build or choose more reliable information technology (IT) for your newspaper. If you are concerned about system and software quality, then you are concerned about the functionality, reliability, usability, efficiency, maintainability and portability of the system. Each of these measures of system quality can be divided into many sub-factors, each of which requires careful analysis.

The focus of this article is on an aspect of reliability: specifically, how to prevent data loss. In order to provide a context for data loss reliability measures, I'll use an ad tracking application from the newspaper business as an example. In newspapers, ad tracking systems are used as workflow management systems for the production of display ads and their subsequent archiving as digital ad assets.

Clearly, a reliable ad tracking system will need to keep data safe and available. We will touch on the issue of system availability as it relates to data loss. That is not to say that system availability issues in general are not as critical or complex as data loss issues; it is just that both issues are equally large and complex.

If you want to understand how to prevent data loss, you need to know the top causes for it. Dan Gardner, Vice President of Development at the Renew Data Corporation and a long time industry expert on data recovery provides a top 10 list.

### Top 10 Causes of Data Loss

(from the June 2003 issue of *ComputerWorld Magazine*)

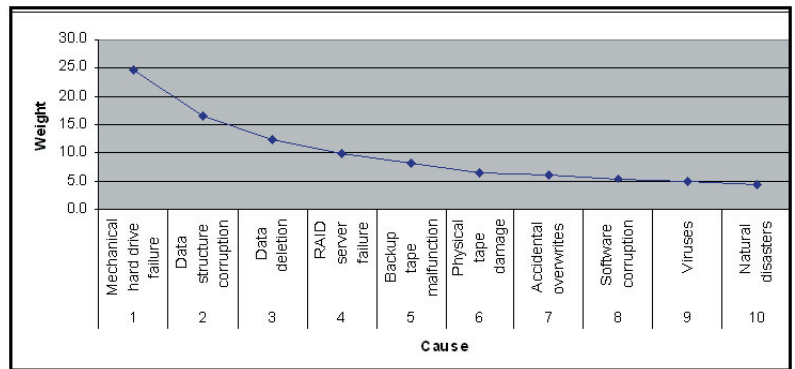
Rank	Cause	Weight*
1	Mechanical hard-drive failure	24.7
2	Data structure corruption	16.5
3	Accidental or intentional data deletion	12.4
4	RAID server failure	9.9

Software Consulting Services, LLC  
630 Selvaggio Drive, Suite 420  
Nazareth, PA 18064  
Sales: 1-800-568-8006  
Fax: 610-746-7900  
E-mail: sales@newspapersystems.com  
www.newspapersystems.com

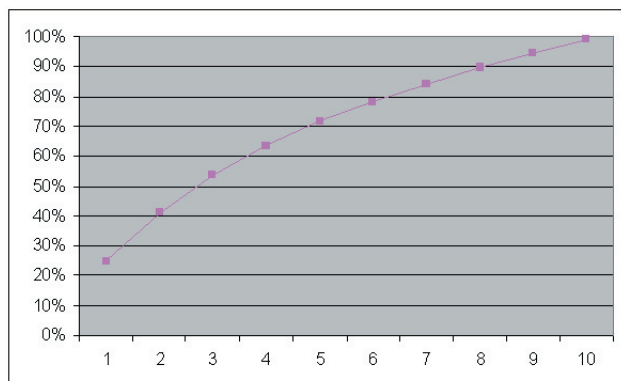
5	Backup tape malfunction	8.2
6	Physical tape damage	6.3
7	Accidental overwrites	6.2
8	Software corruption	5.5
9	Viruses	4.9
10	Natural disasters	4.4

(\*Weights computed using Zipf's Law for ranked items--the probability of many ranked events is inversely proportional to their rank.)

The third column of the table was added to show the likelihood of each of the causes. You can use them as an estimate of the probable likelihood of this type of event as a percentage of all data loss events. As you can see from the table, we would expect the cause of data loss to be due to one of the first three reasons more than 50% of the time. When all ten causes are considered, it is likely that at least 99% of the cases are covered. Using this type of ranking helps focus our attention on the most likely causes of data loss.



Each application has different requirements for reliability. In the early days of SCS, we made microcomputer-based systems for managing intravenous solutions in hospital pharmacies. More often than not, IV solutions were being administered to cancer patients. These substances were so dangerous that those who handled them wore protective suits and gloves and worked with the chemicals under ventilated hoods. The potential downside risk of data corruption for this application included the possibility (a remote one, we hope) of killing patients.



**The top ten causes of data loss account for 99% of all data loss events.**

Now we serve newspapers, and, while losing an ad is a critical problem, it's not that critical.

The standard platform for the SCS/Track™ ad tracking system consists of a pair of beefy servers used to store and manage ads and a collection of PCs or Macintoshes on which ads are built using ad layout software like QuarkXPress®, Multi-Ad Creator® or InDesign®. The most demanding sites follow this architecture and also have remote sites where caching servers support local ad builders. These are used to reduce bandwidth requirements.

On the SCS/Track servers, information about ads is stored in database tables. These tables point to folders in which the actual ads reside. Ad documents contain images, and because of this, the storage requirements for ad documents is several orders of magnitude larger than that needed to store their metadata (i.e., their descriptive data).

SCS/Track handles both workflow management and archiving. The workflow management data needs to be fast and available because it holds the current ads that are destined to appear in an upcoming edition. The ad archive is used in the sales process and also for ad pick-ups. Newspapers can produce editions without the archive, but having an archive online makes doing business much easier.

Before we go into Mr. Gardner's data loss causes, keep in mind that the most probable, traditional reason for ads to be unavailable is simply that, once their ad schedule was complete, they were discarded. If you examine what newspapers do with ads after they run, you will find that they might be kept around on the production system for as little as two weeks to as much as thirteen months. Some newspapers archive expired ads to DVD or CD.

The value of a digital ad asset management system arises out of the need for improved customer relationship management and ad building productivity. When these factors outweigh the costs of the technology, then better digital ad asset management becomes a desired, cost-justified option. High-capacity, high-performance Serial ATA (SATA) disk drives significantly reduce the cost of archival storage, making digital ad asset management much more affordable.

Ideally, one would want to save all of the ads that have appeared in the newspaper. To save what is produced by a single ad builder requires about 10GB per year. We recommend storing ads for a period of two to three years so a 30-person ad services department would yield an accumulation of ad data totaling approximately 300GB per year or a full terabyte for three years of storage.

High-capacity Serial ATA disk drives reduce the cost of a terabyte of disk space to about \$2,200. Lower cost storage means you can buy lots more of it. However, with so much data, the impact of a data loss event poses a serious IT problem. It is within this context that we will address the individual items in the "Causes for Data Loss" table.

### **Mechanical Hard-Drive Failure**

Commodity, component-based Intel servers ordinarily are equipped with SCSI disk drives and RAID 5 controllers. (RAID 5 joins multiple disk drives into a configuration in which parity is used to recover from a single drive failure. A RAID 1 configuration pairs disk drives for redundancy through mirroring. RAID 1 configurations produce the fastest speeds while RAID 5 configurations produce the largest capacity at the least cost while still having redundancy.)

How do SCS/Track servers fail-soft when there is a hard drive failure? SCS/Track servers are paired. Each server is running the application with the primary server interacting with the ad builders. As the primary runs, it passes data off to the secondary (and, perhaps, additional servers) where they update local databases and ad document copies. Should the primary experience an event that takes it down, the secondary takes over.

The long-term reliability of servers is greatly enhanced if they are protected from unnecessary heat, vibration and static electricity. The most likely

points-of-failure include hard drives, power supplies and fans. Reliable servers have dual power supplies and redundant fans. It is a good idea to keep servers in computer rooms where the temperature is controlled (usually at a range between 68 and 72 degrees Fahrenheit) and the humidity is relatively high (usually 65% so as to prevent the damage from static electricity). The AC power should be protected with an uninterruptible power supply. Rapid changes in temperature or humidity (more than one degree or one percent per hour) should be avoided.



Hard drive failure is a fact of life. Drives are rated using what is called mean-time between failures (MTBF). The current norm for SCSI drives is around 1.2 million hours, while parallel ATA/IDE drives are rated at 620,000 hours. The newest Seagate SATA drives are rated at 620,000

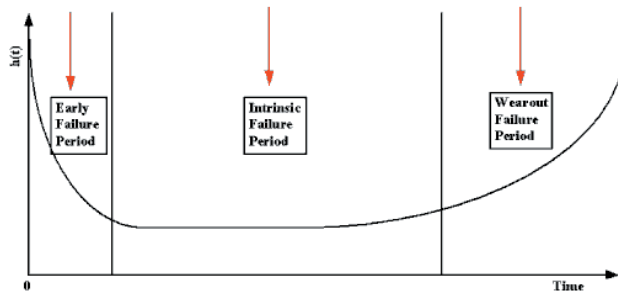
MTBF, and those from Maxtor are rated at 1.1 million hours MTBF. Is it reasonable to substitute a less reliable SATA drive for a SCSI drive? Considering RAID redundancy and a MTBF range between 68 and 136 years and annual failure rates less than 0.75%, all current drive technologies provide more than adequate reliability.

One of the cool advantages (pun intended) of SATA technology is its use of thin cables. Unlike the wide cables of SCSI technology, these thin cables make chassis airflow work better and thus systems run cooler. Cooler is better.

It should be noted that a higher MTBF does not necessarily mean that the drive provides more reliable capacity. The SCSI drive rating of 1.2 million hours of MTBF is for 146GB drives while the 1.1 million hour MTBF for SATA drives applies to drives with capacities of 250GB. On a cost or per gigabyte rating, the 250GB SATA drive is far more reliable per gigabyte than the 146GB SCSI drive.

A second factor in understanding equipment reliability is to understand the profile of failures over time. Most equipment failures occur either shortly after the initial purchase or toward their end of life. If you draw a graph showing failure rates on a vertical axis and time on the horizontal axis, a characteristic curve is produced. It is called a "bathtub curve" because it resembles the line formed if you were to cut a bathtub in half end to end. This means that failures are least frequent during a server's mid-life.

**The Bathtub Curve**



<http://www.itl.nist.gov/div898/handbook/apr/section1/apr124.htm>

How do you predict server lifetime? Dell allows you to buy an initial three-year warranty on their servers and then allows you to extend this warranty for an additional two years. After that, they expect per call maintenance on the servers. I would say a good rule of thumb is to expect a server to last seven years. It is important to remember that computing equipment is a rapidly evolving technology, so much so that what you spend one dollar on in 2003, you are likely to spend only one penny on in 2015. As computer technology becomes more affordable, you will want to change the business model that you use when purchasing.

### Data Structure Corruption

There are data structures and algorithms used to manage disk space. Not only are there files with your data in them but also data structures which keep track of the metadata (name and location, size, history, etc.) of each file. There are also data structures that keep track of what disk space is available. In general, the data about files, etc. is stored in what are called file allocation tables. If a file allocation table gets messed up, then you lose track of where the files are, where the free disk space is, etc. With a bad file allocation table, you may overwrite and destroy data wherever it might reside on disk drives.

To support reliable, fast access to file allocation tables for today's large disk subsystems requires sophisticated data structures and algorithms. Unix®, Linux®, Windows® and other operating systems now use B-tree data structures or some close variant of B-trees. B-tree data structures and algorithms provide rapid, reliable record management. They are very efficient for accessing and updating but require careful management to be reliable.

Current operating systems include what are called journaling file systems. Journaling file systems maintain consistency when the processing is



interrupted, even when there are data in file and disk buffers. Journaling file systems became even more essential as disk space increased. Recovering with a journaling file system can take less than one minute, even when the disk subsystem is hundreds of gigabytes in size. Running FSCK or a similar program to recover a file allocation table without the journaling file system would take hours. Journaling file systems are standard with nearly all Linux distributions.

### **Accidental or Intentional Data Deletion**

There are moments when you decide what you've done in the past is no longer adequate for what you want to do in the future. One of the most memorable phone calls we have ever received from an SCS/Track customer came from a mid-market newspaper with very heavy production requirements. This was a number of years ago when the servers we supplied were Dell's then high-end machines running SCO Unix with Savware's Stand-By mirroring system software. This software assured that what happened on one server would be mirrored onto a second server. Should the primary server go down, the secondary server would take over for it and continue running the application. We did everything in our power to make these robust solutions.

Now for the matter of the fateful phone call. It seems that, at this newspaper, the lead operator had worked an extra shift. In preparation for leaving, he checked the server free disk space and noticed that things were getting tight. In those days, servers had six 9GB disk drives in a RAID 5 configuration, which yielded 45GB of usable disk space. (Now our minimum server for SCS/Track has 20 times this space.) The tired operator figured an easy way to free space was to clear out the temp directory. In an attempt to do this, he typed the "rm \*.\*" command. He was, of course, a privileged user. Unfortunately, the directory that he was in was the root directory. The operating system happily obliged by starting to delete every file in the system. This was supposed to be mirrored to the backup server using the Savware software. The "rm" command happily removed every single file until it came to destroying something that it needed to run. At that point, it halted. The operator realized what he had done and called for emergency support services. Over a tense period of hours, we were able to deduce that the fail-over machine had not been mirroring for some period of time and that the backup tapes had most of the work in progress. We were able

to get the newspaper back up and running but not without considerable angst and an intense dislike for mirroring software solutions and operator-initiated storage management activity.

Since SCS/Track and all of the software that comes with it are written and maintained by Software Consulting Services, LLC, we implemented a new platform-independent scheme for keeping servers in sync. We use a technology called replication. With replication, changes on one server are reflected on the second server. Transactions applied to the database on the primary server are applied on the second server. The nice thing about this high-level "mirroring" is that it assures that database updates are validated twice. It is much better than low-level block mirroring, which can simply move a problem from one server to another.

By managing storage from within the application, SCS/Track has additional fail-safe facilities. For example, deletions on the primary server are scheduled for somewhat delayed initiation on the secondary server, thus giving time to recover from an accidental data deletion.

There was a recent incident at a new SCS/Track customer site where a 45MB ad with many color graphics was deleted off the primary server. The site was able to copy the ad easily and quickly from the secondary server as soon as they knew that it hadn't been deleted there yet.

### **RAID Server Failure**

We use RAID controllers when we build servers. In general, they offer improved performance and reliability. Linux and other operating systems that we support have RAID software built in; however, we prefer hardware RAID controllers for speed. We prefer to use SCSI RAID controllers from LSI/Logic and SATA RAID controllers from 3Ware. The high-end SCSI controllers feature 128MB of cache and battery backup. With our applications, Linux tunes itself to provide large buffer areas. The 3Ware SATA controllers that we use are very well engineered. They take advantage of the 8MB of cache on each of the SATA drives. They also provide a dedicated channel from the controller to each drive. SATA drives configured in a RAID 1 perform as well as SCSI drives in a RAID 5.

We build servers that fail-soft through replication rather than shared devices.

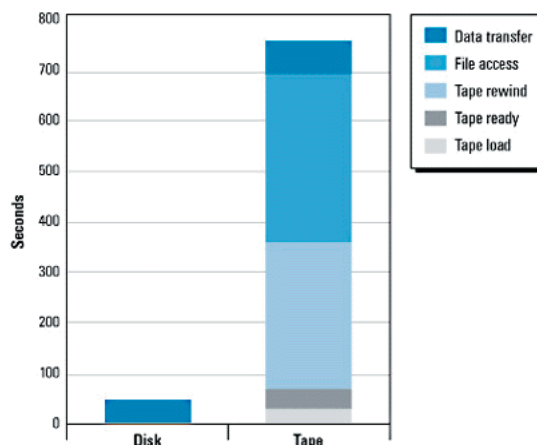
Four years ago, our first fail-over technology for Linux was based on dual servers with a shared SCSI disk drive tower. What seemed like a very reliable and redundant solution proved instead to be tricky to manage. The status of a SCSI array needs to be known in order to manage it. Usually the RAID controller and the disk drives both have this information written on them. If you were to replace a RAID controller card, it can be set to initialize the state of the drives from data stored on the drives. With the shared SCSI tower, we had the state of the drives stored on the drives and each of the two RAID controllers. Switching to the second server meant switching to a second controller card. Failure to accept the status information from the drives would cause the second server's controller card to corrupt the disk subsystem. During fail-overs, this proved to be a source of operator error. From this experience, we've learned to avoid shared equipment solutions. Shared subsystems tend to be more costly and system-dependent than non-shared solutions. Configuring servers to be fast, reliable and cheap is tricky.

### Backup Tape Malfunction

When we first started expanding the storage available on ad tracking systems to support digital ad asset management, we ran into issues of tape backup. While previously a 50/100GB tape drive provided adequate backup storage for six 9GB drives, when we started shipping servers with 300+ GB, using single tape cartridges became impractical. Our first attempt at the solution was to integrate auto-changer tape cartridge technology into the servers. To support the auto-changer, we put SCSI controllers in each server and plugged the cable for the auto-changer into the primary server with the expectation that it would be unplugged and plugged into the secondary one, should this be necessary.

Tape backup technology performs at the rate of about 50-100GB per hour. Having 300GB or more yielded a six-hour backup process. It soon became apparent that the tape cartridge systems were not the way to go. One of the technology advancements that led to this conclusion is that it is simply cheaper and faster to do backup to disk drives. The one advantage the tape has over disk drives is the ability to take it off-site easily. You would want to do this for your disaster recovery plan.

As an alternative, we've worked out a new scheme with SCS/Track. We still do disk-to-disk backup



[http://www1.us.dell.com/content/topics/global.aspx/power/en/ps2q03\\_emc?c=us&cs=555&l=en&s=biz](http://www1.us.dell.com/content/topics/global.aspx/power/en/ps2q03_emc?c=us&cs=555&l=en&s=biz)

with our replication technology. The ads that are being built, i.e., those that are current ads in the workflow, we backup to tape. This does not require more than 70GB and fits easily on the 50/100 QICs that we provide.

Our archives can be up to two terabytes in each server. As ads age and are automatically assigned to the archive, we write them incrementally to DVD media. This provides the possibility for off-site storage of the archive. Tape provides off-site storage for the rest. We've come to believe that inexpensive SATA disk drive technology turns tape archives, libraries and auto-changers into things of the past. We have eliminated tape as an area for concern about data loss.

### Physical Tape Damage

This cause for data loss was eliminated with the switch from tape archiving to disk archiving.

### Accidental Overwrites

We put accidental overwrites and operator error in the same category. When we found out that it was more likely for someone dealing with an emergency or recovery situation to make mistakes, we simplified procedures by automating them as much as possible and making them work with the same protocol that the application works with. The SCS/Track application now reports how storage is being used. Since we know the purpose of each file type, we can also diagnose situations in which files are much larger than they typically should be for their file type. Further, we think we've eliminated nearly all cases in which a fail-over situation creates an even more complicated recovery situation and thus even greater opportunities for operator error.



The approach of having the system managed from within the application, not with some third-party RDBMS, follows from our desire to have our application be platform-independent. A platform-independent turn-key application greatly reduces the need for IT expertise in managing the system. Not only are the day-to-day storage management tasks of the system managed by the system, but so are the emergency situations.

### Software Corruption

Steven Mills, Senior Vice President and Group Executive of IBM, runs the company's 13.6 billion dollar software business. In a recent ComputerWorld interview, Mr. Mills is quoted as saying, "Microsoft, for reasons...of their technology, which was never designed for this type of interconnective, Internet environment, combined with the fact that they are an obvious target, has raised all these concerns." We consider a reliable operating system platform essential to building reliable systems. Clearly, Linux is such a platform and Windows is not yet. The question is, of course, whether or not Microsoft will ever be able to produce a reliable operating system with its closed corporate culture for producing proprietary system software.

### Viruses

Symantec has reported that, in 2003, nearly 4,000 different weaknesses in Microsoft's operating system code have been exploited by virus writers and other hackers. This compares with zero for the Macintosh OS X operating system and eleven for all Unix and Linux systems.

In an article discussing Microsoft's latest attempts to lessen the security failures of their operating systems, the Wall Street Journal noted that the MSBlast and SoBig viruses "disrupted millions of computers worldwide this year."

While SCS offers its systems on Windows platforms as well as Linux, Unix, Solaris® and OS X®, we advise our customers that, for maintainability, Windows servers are more vulnerable and should be avoided.

### Natural Disasters

Planning for disaster recovery is something that the IT community has not done well. We've had our ability to respond to disasters tested a number of times. One customer had their entire facility burn to the ground. They are a weekly newspaper that printed off-site. SCS staffers drove through

the night with a spare server and were able to get the newspaper back into operation at a temporary location and publishing on time. The customer had the foresight (or good fortune) to have a current off-site backup tape. This is a story of overcoming great adversity.

"In August, for instance, CSX Corp. was forced by the Blaster computer worm to temporarily stop its freight trains. That, in turn, forced Amtrak, which uses CSX track for some of its long-distance passenger trains, to delay its service. Some critics of the software industry say it's just a matter of time before a software flaw results in a real catastrophe and loss of life."

*The Wall Street Journal*  
November 17, 2003

The 9/11 attack also tested our systems in a situation of great adversity. Four major publications are produced with SCS technology in Manhattan within view of Ground Zero. These include the *New York Daily News*, the *Time4 Media* magazines, *El Diario la Prensa* and *The Village Voice*. All of them published during this crisis without interruption. Although disaster recovery wasn't on our minds at the time we set these sites up, it was fortunate that the systems were as resilient as they were. It is interesting to note that, while some of our customers couldn't enter their facilities, they were able to produce the newspapers through remote access to our systems.

We see building reliable systems as an obligation we have as we serve a free and independent press.

### Conclusion

In a related white paper entitled "Five-Level Redundancy," the operational aspects of secure and resilient data management are discussed. Through data replication, we've minimized the impact of drive or server failure, data structure corruption and unwarranted data deletion. We've also minimized the possibility of data loss due to tape malfunction and physical tape damage by using online disk-to-disk backup and offline DVD storage. We've reduced accidental overwrites and operator error by automating system processes, maintenance and emergency procedures.

We seek low-cost 5-nines availability (i.e., 99.999% available or less than one minute of down time per year). In pursuit of this goal, we use new

commodity equipment simply and redundantly to reduce the likelihood of a data loss event by 99% over prior typical server configurations.

The evidence so far indicates that we have achieved this goal.

SCS builds trusted newspaper systems.

