

# Managing your data performance, availability, and retention

*It happens more than you'd like to believe. A small company upgrades its storage infrastructure and puts all of its data on a high availability RAID array with dual controllers and battery backup for its write buffers. A large IT organization sets up a remote mirror of its critical storage for immediate recovery from a disaster in its local data center. Both companies are thrilled with the performance and availability that their new storage systems deliver, but neither fully considered the fact that having a Storage Area Network (SAN) and/or RAID array, even with remote mirroring, is not the same as having a comprehensive data management strategy.*

*Then the disaster strikes. It's not the one they planned for: the single disk failure that would have left the RAID array intact, or the data center fire that would have caused a failover to the remote site. An improperly patched database management system scribbles all over the customer database. In the case of the large IT organization, the error is instantly replicated on the remote mirror. And there's no backup tape.*

What was the root cause of the problem in these examples? Bad RAID technology? Not necessarily. These organizations didn't have a well-constructed data management policy. Every organization needs a policy for data performance, availability, and retention, and the technology to support it. RAID (and possibly a Storage Area Network) is one tool you'll probably need, but it's not the only one you'll need. Every data management policy must include stable-state backups on media (such as tape) that stores your data not in motion.

## RAID: Just one tool in the data management shed

RAID, a Redundant Array of Inexpensive Disks, was once confined to enterprise data centers. Today you can buy a RAID device for your desktop computer for only a couple hundred dollars. RAID makes everyone feel that their data is safe. If you use RAID, you should consider what it does for you and what it doesn't do for you.

Disk drives, because they rely on moving parts, are the most fallible part of a computing system. RAID techniques allow you to use multiple, low-cost disks to create a disk subsystem that has overall higher reliability and/or performance than a single disk by itself. RAID will typically play an important role in your data performance and availability strategy, but it alone does not constitute a data retention policy.

RAID can give you higher or lower reliability, and higher or lower performance than a single disk, depending on which level RAID you choose. RAID doesn't necessarily give you absolute reliability, or the highest possible performance, so it's useful to know exactly

what RAID does and doesn't do for you so that you can make appropriate choices.

## RAID fundamentals

All of the RAID levels that you've heard about are built using a combination of one or more of four fundamental building blocks. Surprisingly, the definition of RAID is loose enough that some RAID levels don't implement the first word of the acronym, redundant, making it even more important to know what you're getting when you buy RAID.

## Concatenation

Concatenation is the simple act of using more than one disk to create a single logical volume that is larger than a single disk. Concatenation can be used to create volumes of virtually any size, making it possible, for example, to store large databases. If used by itself, concatenation reduces reliability because the failure of any one disk in logical volume means that a substantial portion of your data might be lost. Concatenation is often used in combination with redundancy to provide large amounts of storage with higher reliability.

## Striping

Striping lays out data sequentially across an array of disks. Block-wise striping lays out a sequence of blocks across a set of disks, starting back at the beginning when one block has been placed on each disk. Byte-wise striping similarly lays out sequential bytes across multiple disks. Figure 1 illustrates block-wise striping across two disks.

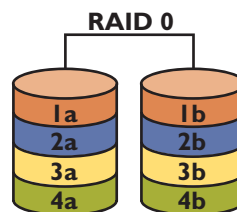


Figure 1: RAID 0: Striping

Striping is known as RAID 0. It improves performance by moving data from each of the disks in an array at the same time. At the lowest level, data can stream from a disk drive only as fast as the drive rotates. If data is striped across five drives, the stripe can deliver data at five times the rate of a single drive. Striping reduces reliability to that of old-fashioned Christmas tree lights, however. One light burns out, and the entire string of lights goes out. If a single drive in a stripe fails, the array cannot function.

Striping alone is used where performance is very important, but reliability isn't. It is used most often in combination with the next two techniques so that its performance benefits are also enhanced with reliability benefits.

## Mirroring

Mirroring, also known as RAID 1, puts exactly the same data on more than one disk so that if any disk fails, the data can be accessed from the remaining disk(s).

Mirroring is typically configured with pairs of disks, which yields half the risk of failure of a single drive. Figure 2 illustrates a mirrored pair of disks where each block is replicated on each disk.

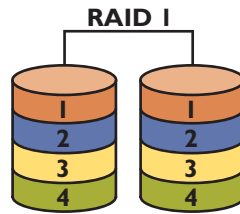


Figure 2: RAID 1: Mirroring

Mirroring is so simple that it is appearing everywhere, including on desktop PCs and many external storage devices. It's known as a "poor man's" RAID because it is so simple, and it is great for relatively small amounts of data.

## Parity

Parity is used in conjunction with striping in order to increase the reliability of the stripe. For each data block in a stripe (assuming block-wise striping), a parity block is created that consists of a bit-wise sum of each of the blocks with the carry bits thrown away. For example, if the first bit of each of three blocks is 1, 1, 0, the parity bit is 0. (Binary  $1+1+0=10$ , and only the low-order 0 is retained). If any one of the three blocks is lost, each parity bit can be used to figure out what the missing bit was.

Where mirroring requires, at minimum, a doubling of the number of disks, using parity to increase performance requires only an incremental increase in the number of disks, with the caveat that the stripe with parity can withstand the failure of only one of its disks. This comes at a cost, however, which is that each write operation requires parity to be recalculated using data from the entire set of disks.

Byte-wise striping with parity is known as RAID 3, and block-wise

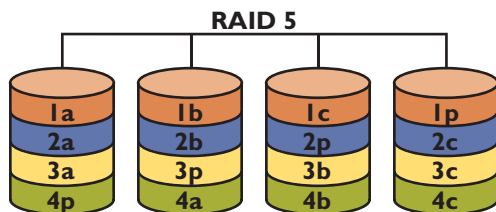


Figure 3: RAID 5: Block-level striping with parity

striping with parity is known as RAID 5. Figure 3 illustrates a RAID 5 array with three data disks and one parity disk, which allows for the failure of any one of the four disks at the cost of only one disk for parity. Disks are labeled with a numeric stripe number and an alphabetic block number, with 'p' denoting the parity block.

If a single disk fails, the parity bits are used to reconstruct the contents of the failed disk. This recalculation time creates a window of time during which a second disk failure can result in the loss of the entire array's worth of data. One technique for minimizing this time window is having a hot standby disk in the

array that can be used to store the reconstructed data, rather than having to wait for an administrator to figure out that a failure has occurred and that a new disk needs to be plugged into the array. Figure 3 illustrates another technique, which distributes the parity data evenly across the array so that a failure does not require immediate restoration of all data blocks.

## RAID in the real world

All RAID levels are based on the four building blocks of concatenation, striping, mirroring, and parity. RAID 0+1 is a mirror of two stripes, as illustrated in Figure 4. RAID 1+0 (10) is a stripe of two mirrors, as Figure 5 illustrates. RAID 5+0 (50) is a stripe of RAID 5 arrays (Figure 6) and RAID 10+0 is a stripe of RAID 10 arrays (Figure 7). Striping gives you performance, while mirroring and parity give you reliability. Different combinations of these building blocks make different tradeoffs between performance,

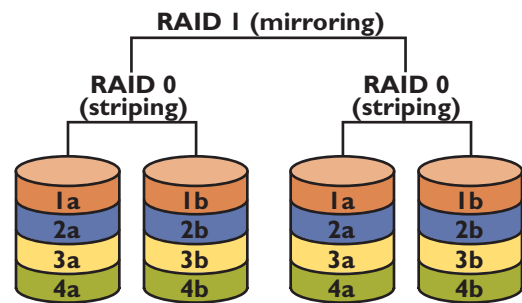


Figure 4: RAID 0+1: Mirror of two stripes

availability, and cost. Not surprisingly, each one is good for storing

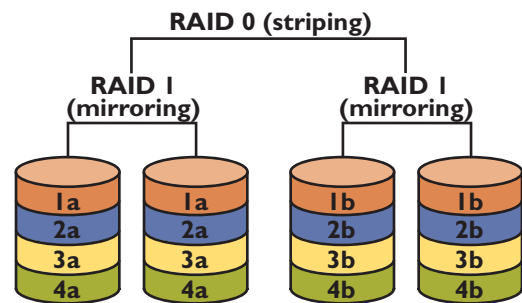


Figure 5: RAID 1+0 (10): Stripe of two mirrors

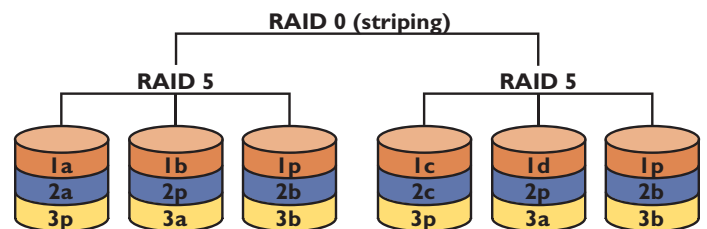


Figure 6: RAID 5+0 (50): Stripe of two RAID 5 arrays

different types of data. Some of the most popular RAID levels and their common uses are illustrated in Table 1 on page 3.

The discussion so far has focused on abstract combinations of disks and what RAID levels they yield. In the real world, availability, performance, and cost are influenced by how the disks are deployed:

- RAID can be created with hardware or software; hardware is typically faster and more expensive than software. Some of the higher RAID levels are implemented using both hardware and software. For example, RAID 50 might be configured from a set of hardware RAID 5 devices with software-controlled striping across them.
- Redundant controllers, power supplies, fans, and interfaces can help reduce the number of single points-of-failure that can bring down a hardware RAID array.
- Battery backed-up write caches can speed performance

impact of losing the data is not very high. Unless your application is financial trading, in which case you might be required to maintain an audit trail that includes every access to the Web application. One organization's Web logs are disposable, while another's are business critical.

- Your financial information probably doesn't need the fastest devices you have available. But it had better use one of the more reliable RAID levels so that you don't lose anything between backups, and so that you meet the IRS data retention requirements.
- Some applications need performance above all, and RAID levels using striping are likely to give better performance. Some applications need reliability above all, and RAID variants using parity, such as RAID 5 and its variants, are likely to meet your data availability needs.

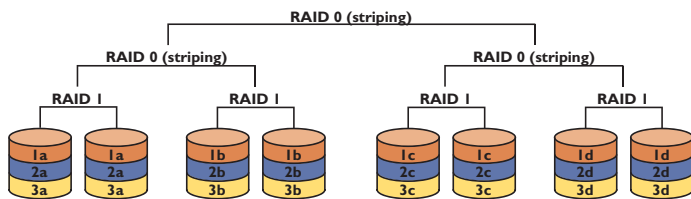


Figure 7: RAID 10+0 (100): Stripe of RAID 10 arrays

by acknowledging disk write completion when the data is transferred to nonvolatile memory, not when it is actually written to disk. If power fails during the disk write, it can be restarted using the cached copy once power is restored.

- Hot-swappable disks, power supplies, fans, and controllers allow RAID arrays to be serviced without turning their power off, and hot standby drives allow RAID controllers to press a spare disk into service without having to wait for the system administrator.

Some organizations go beyond RAID and increase their data's availability by mirroring it to a remote site. Remote replication allows the data to be accessed from the local data center through a slower, wide-area network pipe in the event that the local RAID array fails. It also allows the server infrastructure to be replicated at the remote site in the event of a disaster that takes out the entire data center. If the company's services need to be up and running in only minutes following such a failure, geographic failover can be used to reroute requests to the remote data center in the event of a disaster at the local one.

## Integrating RAID into your organization

Now that you know everything about RAID, you're ready to pick one of the technologies from Table 1 and you're done, right? Not exactly. Different RAID levels are useful for different types of data, so it isn't as simple as choosing a single 'best' RAID level and using it for all of your data.

All of your data is different, and your data can tell you how it likes to be stored. Your applications also can tell you how they like to retrieve it. Between the two of them, you're off to a good start in determining which RAID level to use for which of your data. For example:

- Your Web logs probably don't need to be maintained on the most reliable storage, where cost is no object. The business

## RAID implementations

When you go out to buy a RAID device, you'll find it implemented in many forms. You can buy USB or Firewire RAID boxes for your desktop; direct-attached RAID boxes with SCSI interfaces; network-attached storage devices that serve RAID storage over a network; and virtually every Storage Area Network (SAN) implementation supports RAID. SANs give the flexibility of network-attached storage with the performance of direct-attached devices. Logical-unit 'masking' and various virtualization techniques allow different servers and applications to securely access their own dedicated portions of an enterprise-class storage system. These systems allow data centers to manage a large, centrally located pool of storage for all of their applications. And they often have built-in capabilities such as point-in-time snapshots and remote mirroring.

## RAID is not a data retention policy

One problem that some organizations get into is thinking that RAID is how they implement their data retention policy. Nothing can be further from the truth. Consider what happens when any one of a number of scenarios such as the one that introduced this article occur:

- Your data center is in a floodplain and the 100-year flood occurs
- Your RAID controller has silently failed and has been writing incorrect parity bits
- A bug in your software corrupts your data
- Two disks from the same production lot find themselves in your RAID 5 array, and they fail within minutes of each other, too soon for parity to be re-calculated on a hot standby disk
- An administrator mistakenly deletes your root directory, deleting all files underneath it
- A disgruntled employee removes drives from your array as she goes home from her last day at work

- A hacker modifies your data in a way that you don't discover until six months later

All of these issues highlight the fact that RAID is only a tool, and not the only tool that you need to effectively manage your data. RAID is a tool for managing availability and performance. A comprehensive data management policy is what you need to manage data retention. You need both of them to steer clear of accidents, including the 100-year flood.

## The need for data retention

The fact that you can integrate RAID into your organization and still lose all of your data points to the fact that you need to have a data management policy that includes not only data availability and performance, but data retention as well. Just as your applications and your data place requirements on performance and availability, your organization, legal context, and security considerations place demands on data retention. For example:

- The law dictates how long you must retain business-related tax information, personal information on your employees, and, if you're involved in the health-care supply chain, your patients. You need to store this data in such a way that it can't be tampered with, and also so that it can be retrieved without error if and when it becomes necessary. The only foolproof way to store a stable backup is on magnetic or optical media where the bits are no longer moving and can't be modified. Beware that even tape doesn't solve all of your needs: If you don't periodically read and re-write your tapes, you run the risk of the data on them deteriorating.
- Your bank may dictate that when a customer returns an item, you credit the original credit card account on which the purchase was made. How long do you retain such data? You may wish to retain it forever, but the longer you retain it, the more of it you have, and the greater your liability if those

credit card numbers are stolen. A data retention policy dictates not only the time that you need to hold your data, but also when and if you must securely destroy it. Destroying personal information and credit-card numbers on a planned basis may help to limit your liability.

- Your business presence may depend on your Web site, and you need to protect against a defacement or other subtle change that might not immediately be noticed. To protect

your business in cases such as this, your data retention policy might require more frequent backups so that you can return to any one of a number of points in time before the tampering occurred. You also need a baseline that can help in the incident analysis itself.

- Your business-critical documents, such as engineering plans, transaction records, and product-pricing strategies, are your crown jewels. They need to be handled as such, on reliable storage, with a sufficient number of stable backup copies so that you can restore a consistent set of them at any point in time.

## Choices for data migration

Now you're convinced that you need stable offline backups in which your data is at rest stored in a secure, off-site location where it can't be tampered with. What happens if your e-commerce site goes down and you lose thousands of dollars per hour in revenue while you wait for the truck to fetch the backup tapes from your secure repository?

What you need is an orderly progression, or migration, of data from its instantly available, online state, to the offline backup tape stored in an undisclosed bunker. Your business and your applications may benefit by having a choice between the two extremes, known as nearline storage.

Nearline storage, for example a remote mirror of your online storage, and/or an online snapshot of your data taken at periodic intervals, allows you to more quickly restore your operations in the event of a catastrophic failure. A remote mirror allows you to access your data, although somewhat slowly, over a wide-area network connection to your remote data center. A remote mirror with geographic failover allows a remote site to take over operations within seconds. A local, online snapshot allows you to recover from the errant program or administrator that wipes out your data.

Nearline storage also gives you a way to access those old customer records or patient histories from past transactions. Data migrated to a tape library or optical jukebox can be recovered in minutes, rather than the hours that would be required for a migration from its off-site location.

## Creating your comprehensive data-management policy

All of your data is different, and there is a wealth of technology to support the storage characteristics that your organization, application, and your data require. A comprehensive data management policy is a well-thought-out plan that supports your data availability, performance, and retention requirements, and which directs your organization to exercise your plan to be sure that those stable, offline backups really hold the data they're required to. Here are some of the basic steps you'll need to take in implementing your own policy.

### RETENTION FOR SECURITY

Applied Trust recommends that, at a minimum, you retain an initial full backup of every system, annual full backups, and full backups immediately following major upgrades, for the life of each system, to aid in later forensic analysis. Alternating months of full backups should be stored in an off-site, fireproof vault at least five miles from the primary site (such as at a partner facility, or data media management company such as Iron Mountain).

## Classify your data

A good place to start is by creating a list of all of your data, and recognizing that different data needs to be treated differently. Characterize your data according to the different requirements placed upon it:

- **Application requirements.** Different applications make different demands on their data. Online transaction processing requires a storage mechanism with good write performance. Catalog queries can sacrifice write performance in order to achieve better read performance.
- **Growth requirements.** Does your data grow at an astronomical rate, or is it fairly static? If your customer database doubles in size every six months, consider a RAID configuration that supports this level of scalability.
- **Business requirements.** If you have a storage failure, what down time can you tolerate? How fast do you need to be able to access it again in case of failure? Your Web logs probably don't need to be recovered immediately, even if they are required for audit purposes. Your customer database probably needs to be up and running more quickly. If you choose to have a remote mirror, can you tolerate operating at a degraded performance level as you access your data over a limited-bandwidth network connection?
- **Legal requirements.** What laws dictate how you handle your data? Corporate tax information has different retention requirements than personal tax information. Adult medical records have different retention requirements than pediatric records. Take care to note whether the retention requirement exceeds the expected lifetime of your data on your offline media, such as tape.
- **Security requirements.** What storage will you need if you have to perform a forensic analysis of a network intrusion that occurred six months ago? What storage do you want to destroy on a scheduled basis in order to protect personal information and credit-card data?

When you undertake this exercise, what you'll find is that you have several clusters of data that need to be handled in different ways. If you lose your Web and system logs, you'll lose the ability to analyze them later for security issues — but your business won't grid to a halt if this happens. Your crown jewels need to be stored on stable, offline storage with enough of a series of backups that you can recover a consistent set of them at any point in time that is reasonable, which may include reading and re-writing backup tapes on a periodic basis. System files probably have their own category, as do financial and personal data that must meet legal requirements.

## Establish responsibility

Make sure that the responsibility for data management (including backups/retention) in your organization is clearly defined, and individuals are held responsible for both vigilance and execution. Don't push responsibility down to a level in your organization that does not have visibility.

## Test

Test, test, and test again. Whatever online, nearline, and offline data management tools you choose, make sure they actually work. Test to be sure that they are effective and can recover your data, and do so on a regular basis. Have a third party validate your plans and their effectiveness. There's nothing like a third party to discover that your recovery procedures aren't adequately documented.

## Avoid the trap

Don't rely on a single technology or vendor for all of your data-storage needs. Now that you clearly understand the different types of data your organization depends on, and the limitations of each technology to meet all of your needs, don't fall into the trap of believing that RAID provides more than it does. RAID is a data availability and performance mechanism. Your overall data management policy will incorporate your data retention requirements so that you retain the data that your organization works so hard to create. ■

RAID Level	Description	Common Uses	Tradeoffs
0	<i>Striping</i> No redundancy, high performance. Reliability decreases as you add disks. Capacity increases as you add disks. Fast reads and writes.	Good for high performance where availability isn't as important. Often used for small application cache files.	
1	<i>Mirroring</i> Adds redundancy, but not performance, over single-drive levels. Can "split the mirror" by taking one mirrored drive offline to make a consistent backup. Fast reads and writes.	Great for small amounts of data because of its simplicity. 2x cost is considerable for large amounts of data. Usually considered the "better than nothing" option.	
3	<i>Byte-level striping with parity</i> Increases performance by breaking every block across multiple drives. Uses a separate parity disk that can be used to regenerate data for any disk that fails. All disks work in tandem for very fast read and write bandwidth at the expense of latency.	Often used for video file storage.	
5	<i>Block-level striping with parity</i> Low-cost for redundancy, involves a single parity disk that can be used to reconstruct the data for any disk that fails. Reconstruction time can be improved through a hot spare drive, and by distributing parity across all drives, so not all of the data needs to be reconstructed if a disk fails.	Provides reliability with high performance, good for applications with more reads than writes because writing requires parity generation. Often used for filesystems that house the Operating System (Windows, Linux, etc.).	
0+1	<i>Mirror of two stripes</i> Uses RAID 0 and 1 hierarchically to achieve high reliability and high performance. Can tolerate multiple drive failures as long as they are all contained in one of the two mirrors.	Overcomes write performance penalty; important for transactional databases with frequent writes.	
1+0 (10)	<i>Stripe of two mirrors</i> Uses RAID 1 and 0 hierarchically to achieve high reliability and performance at the cost of replicating every disk. Can tolerate multiple disk failures as long as both halves of a mirror do not fail.	Preferred for high-capacity, transactional databases with frequent writes.	
5+0 (50)	<i>Stripe of two RAID 5 arrays</i> Stripes data across multiple RAID 5 arrays for higher performance than RAID 5. One failure can occur in each RAID 5 array. Write performance still suffers because of parity generation.	Provides a low-cost way to grow storage, allowing the number of failures allowed to grow without the cost of mirroring. Good for large databases with more read than write activity.	
10+0 (100)	<i>Stripe of RAID 10 arrays</i> Another technique for supporting very large storage requirements. Can handle multiple disk failures given some constraints.	Good for very large databases; good read and write performance.	

Key Performance Reliability Cost

This article was reprinted from the Q3 2006 issue of *The Barking Seal*, a publication of Applied Trust. You can subscribe to *The Barking Seal* online at <http://www.appliedtrust.com/barkingseal>. "Applied Trust" and the "Seal on a Rock" Applied Trust Engineering logo are registered service marks of Applied Trust. All other trademarks are registered to their respective owners.