CONTENTS

INTRODUCTION

Present-day requirements for systemized, network-based storage of data encompass a range of users and applications that would have staggered the imagination twenty years ago.

The meteoric rise of Internet data access became possible as a result of improved network capabilities and then commenced to drive frenetic development of newer and more effective advances in those capabilities themselves.

Global information exchange has exploded to encompass arenas vastly beyond the few select database/mainframe preserves of education and government.

Manageable data storage in the hundreds of gigabytes, which two decades ago would have been reserved for operations on the scale of major military installations, may today be insufficient to the needs of a mid-level novelty retailer.

It is, today, difficult to define a field of endeavor in which RAID systems are not deployed. The two-decade-old principles of large data storage and management have grown and evolved into new forms, yet the basic advantages of dependability, availability, and protection remain key factors in RAID technology's enduring value.

OVERVIEW: WHAT IS RAID?

RAID stands for Redundant Array of Inexpensive Disks.

The basic function of a RAID structure, or array, is to provide a plurality of Hard Disk units for the purpose of massive data storage. Introduction of the concept took place in a 1988 UC Berkeley publication entitled *A Case for Redundant Arrays of Inexpensive Disks*. The intention of this technology is to allow users and operations to take advantage of the large amount of space by storing important data in more secure multiple (redundant) locations, and yet access it as easily and quickly as a single PC-based disk drive.

RAID advantages were originally featured on only expensive server systems, but with the last few years' massive growth in data-intensive enterprise, widespread demand has resulted in a wider and more affordable range of RAID configurations.

With the increased popularity and availability of RAID technology came an explosion in available solutions, each suitable to different user profiles. Additionally, the evolution of technology produced hugely improved storage capabilities, meaning that the disks making up the RAID array were getting larger and larger in terms of capacity, without compromising the limits of physical storage space.

A major challenge is the difficulty of providing access to the stored information and the ability to update (*I/O* for In and Out) at speeds similar to those of single disk systems. Essential to the efficient optimization of RAID technology is prevention of a "bottleneck" scenario. The calculation of this access velocity is referred to as *throughput*.

One basic solution to this difficulty lies in distributing the burden of transfer among all the disks in the array, thus reducing the workload borne by any single disk.

Also of major concern is the *availability* of the stored information. The requirement for available storage, 24 hours a day, 7 days a week dictates the replication of data throughout the array's disks, hence redundancy.

Storage Management requirements are also a significant demand, allowing efficient and uncomplicated access to all the stored data, presenting users with a single access point irrespective of the array's size.

Performance, Availability, Capacity, & Cost

These four factors are the major considerstions to be taken into account when judging the suitabiliry of any RAID configuration with respect to one's individual application needs.

An assessment of the strong points of each individual configuration are complex, and indeed, take up a large part of this paper's contents. This type of assessment involves finding a balance among all the pertinent characteristics in order to ascertain th ideal configuration.

BASIC CONCEPTS

Disk Mirroring

The term *mirroring* refers to the storage of two copies of the same data located on separate hard drives within an array, or within different arrays. When inputting the data, the system writes the information simultaneously to both locations.

A major part of RAID technology's redundancy advantages, mirroring allows continued operation of the system in the event of one drive or array failing. There is little downtime incurred, and restoration can be performed using the mirrored copy. I/O performance may suffer during the reconstruction due to the increased activity between locations. This sacrifice of performance occurs, to a lesser degree, during all write operations.

A possible balance can be found in performance increase during read operations, resulting from the controller focusing on only one of the mirrored locations, freeing up the other for additional tasks. This is a fundamental benefit of RAID technology, referred to as *parallelism*.

Mirroring offers a distinct advantage to data safety concerns, especially in environments where availability is critical.

A downside to mirroring is the requirement for doubled physical storage space, a possible liability for operations whose data reserves are large.

Parity

RAID uses *parity* as an additional redundancy measure. RAID parity is not unlike parity as it is applied in memory technology.

The parity process uses a logic operation to create an additional data element, or *parity bit*. As a function of the binary operation, the parity bit (in this case, much larger than an actual bit) remains intact in any event where one (but no more than one) original data element is lost.

Since the operation can be applied to any number of bits, it can be applied to entire drives, allowing protection without the burden of duplicating the entire data set. This gives parity a distinct advantage over mirroring.

Conversely, parity offers a lower index of fault tolerance than mirroring, which, as we have seen, provides a complete duplicate set of information for backup.

Parity data is not necessarily confined to a single drive. The parity information may be written across the complete array, referred to as *distributed parity*.

Striping (discussed later in this paper) with parity can be performed, albeit with a demand for large amounts of computing power to perform the necessary algorithmic operations. Since the parity information must be re-computed at every read or write operation, this solution calls for the implementation of a hardware-based controller solution (discussed later in this paper) to offload the considerable CPU burden. The operations required are extensive enough that a software-based controller (discussed later in this paper) would seriously compromise CPU bandwidth.

Recovery from failure is also a considerably more complex operation than when mirroring is utilized, requiring significantly more time and resources.

Disk Striping

With disk *striping*, parallelism is improved exponentially. The principle of striping involves the mapping of data across multiple physical drives in an array, creating a large virtual drive spread across more than one physical drive.
This represents a significant boost to the array's performance, since the parts can be read (and written) at the same time. Transfer time is also improved dramatically, thus increasing the throughput.
When information enters the RAID system, it is divided into smaller increments before being written to disk.
Striping is executed in two different ways, dependent upon the nature of the application needs. Each uses different methods to distribute the information to be stored, and each has a suitability to different environments.

Block-level stripes

Larger, or *block-level* stripes may be indicated for environments where many simultaneous requests for smaller-sized records are likely to take place.
This type of I/O-intensive operation benefits from entire records being stored in the single stripe. The drives in the array are then able to respond to concurrent requests in an independent fashion.

Byte-level Stripes

Some operations require access to larger records. In this case, smaller, or *byte-level* stripes are probably more suitable. In byte-level striping, data is broken up into individual bytes and written in sequence across the drives.
The writing process sets the bytes across the drives in sequence, irrespective of the quantity of segments written.
For example, if the data is separated into 32 bytes, and the array has 8 disks, the first byte is written to Disk 1, the second to Disk 2, and so on. After reaching the end of the array and writing the eighth section, the ninth will be written back on Disk 1, and the process continues until all the data is written.

Which Stripe Size is Best?

There is no easy answer to this question.
Both methods of striping are better for the distinct environments mentioned.
Increasing and decreasing the size of stripes produces easily observed performance effects. It is suggested that the key to determining an optimum stripe size lies in experimentation with both processes. Beginning with a medium stripe size and increasing and decreasing size will provide differences in performance that, when recorded and analyzed, will inform the decision.

ABOUT RAID LEVELS

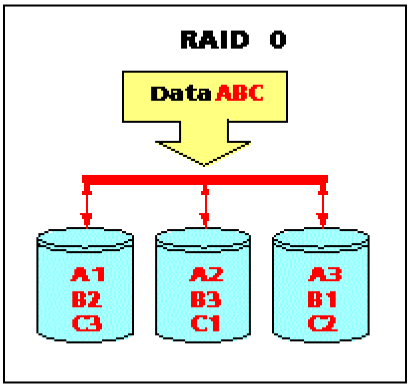
What are RAID Levels?

The UC Berkeley originators described guidelines for different models of RAID configuration, known as *RAID levels*. It should be noted that use of the term "level" does not imply a hierarchical relationship between the various levels. These original levels are assigned designations as RAID Levels 0 through 6.

In addition to the original RAID levels, development has led to the establishment of *Hybrid RAID levels*. These actually combine the original 7 levels' features.

Hybrid RAID Levels are numbered with 2 digits, each of which indicates the original RAID configuration whose combined elements the hybrid employs.

RAID Level 0



Raid Level 0 is the most basic level, in that it incorporates a simple striping array. Its 0 designation indicates that there is no data redundancy, per se.

As explained in the section on striping, 0 level provides tremendous advantages in performance, both reading and writing data, especially compared to the other levels.

As well, since there is no requirement for additional storage space, Level 0 is also the most affordable configuration.

The same absence of overhead that crates these advantages does, however, mean that this level provides no wherewithal for the recovery of lost data.

It is strongly recommended that users of this level reserve its use for non-critical data or implement separate data backup measures.

RAID Level 1

This level is the simplest form of RAID available with fault tolerance. Level 1 is based on the mirroring concept. Simply, level 1 involves multiple sets of the stored information, written to two or more drives.

In the event of a disk failure, the remaining disk or disks still hold a version of the data, so operations can continue. Restoration of the lost drive is not a problem, since the existing version can be copied. As before, the rewriting process may adversely affect I/O performance.

A strategy sometimes employed to enhance fault tolerance further is to connect separate controllers to both the primary and mirrored drives, thus creating the same doubling for backup in the controller system as exists for the data storage itself.

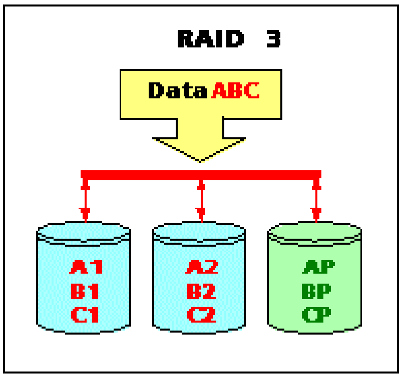
RAID Level 1 is suitable for applications with critical data needs, such as smaller databases that demand superior reliability.

It is not recommended for large databases, since the mirroring solution exerts a heavy physical storage toll.

RAID Level 2

In its original conception, RAID 2 was promoted as a basic striped array with the addition of data protection measures. Its specification in the UC Berkeley paper dictates the use of an error Checking and Correction method (ECC). The method groups data bits and check bits to protect stored data. Modern-day hard drives carry the ECC code embedded in each sector, and this configuration offers little advantage beyond that of RAID 3, so it is rarely used.

RAID Level 3



RAID 3 has some similarity to RAID 2, in that it utilizes a striped array across groups of drives, but it bolsters data protection by the designation of a single *parity drive* in the array. This dedicated drive keeps the parity information for the other data drives. These data drives make use of byte-level striping to spread large record information across the entire array. Level 3 makes use of each sector's embedded ECC to detect and repair errors.

Like all arrays employing parity, in the event of drive failure, RAID 3 utilizes the exclusive OR (XOR) of the parity system to actually regenerate any lost information, providing a favorable level of protection.

Even so, RAID 3 has a drawback in that the location of the parity data on a single drive requires the system to wait for all parity data to be completed before beginning the next write operation, resulting in a bottleneck when writing to the array. Irrespective of I/O advantages in the data disk system, the parity disk is the limiting factor.

RAID 3's utilization of byte-level striping makes it a suitable choice for single-user environments where large files are in use. Maximum implementation of RAID 3's performance usually entails the synchronization of access to the drives, or *drive spindle sync*. Practical use of RAID 3 also calls for hardware-based control.

RAID 3 requires a minimum of 2 disks for data and one for the parity drive. Level 3 only requires the addition of a single extra disk to the data array, making it a more affordable and easier solution than RAID 1.

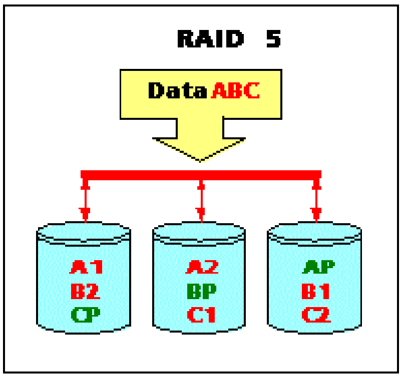
RAID Level 4

RAID 4 is identical to RAID 3, with a single dedicated parity drive and striped data drives. The difference is that unlike Level 3, RAID 4's striping consists of block-level striping, allowing an entire record to be written to each stripe. This makes Level 4 a suitable system for transactional environments, that is, those where many smaller reads are needed at the same time.

As with RAID 3, RAID 4 suffers a write bottleneck due to its single parity disk.

RAID 4's good data recovery performance and affordability are the same as those in RAID 3.

RAID Level 5



RAID 5 is extremely popular, due to its combination of advantages found in other levels while overcoming many of their limitations.

Level 5 consists of a data disk array striped with large block-level striping, as in RAID 4, but in order to overcome the write bottleneck experienced in single parity disk arrays like Levels 3 and 4, Level 5 spreads the parity locations across the entire data disk set, installing a parity segment on each drive. This arrangement is called *distributed parity*, and it greatly enhances the array's I/O performance, since the burden of parity write operations is shared throughout the array.

RAID 5 is sometimes referred to as a *Rotating Parity Array*.

While write efficiency is greatly improved over Level 4, parity calculations must still be performed for each write, and this still creates a slowdown. Optimization of a RAID 5 array can be achieved by adjusting the stripe size to best deal with the environment's needs.

Data protection is still optimal, with the full regenerative capabilities of parity and the fact that parity blocks are isolated from their associated data blocks. Cost effectiveness of Level 5 is also good, comparable to Levels 3 and 4. Hardware control is likewise a necessity.

As with RAID 4, Level 5 is ideally suited to applications where large numbers of users are accessing information simultaneously. It is not recommended for environments where write speed is critical.

Its optimal combination of redundancy, performance, and storage efficiency make it one of the most widely favored configurations currently in use.

RAID Level 6

While not as popular as Level 5, RAID 6 does offer some advantages.

RAID 6 improves Level 5's distributed parity edge by doubling the parity calculations and storing two independent sets of parity results on different drives. Consequently, RAID 6 carries the maximum level of reliability of any non-hybrid configuration, since even if a drive and its first parity segment fail, the second set of parity information remains intact, allowing recovery.

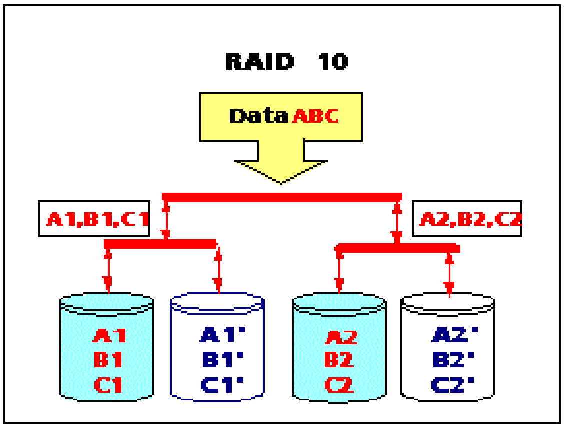
Level 6 limitations become apparent during rebuilds, when I/O performance is compromised more than Level 5, as well as during write operations, since the doubled parity calculations consume twice the system resources.

These factors, in addition to the cost burden necessitated by the extra space of the additional parity drives, limit the use of RAID 6 to systems where accessibility of information is of highest priority.

Hybrid RAID Levels

In the years since the establishment of RAID principles, many system developers have implemented array configurations that combine favorable aspects of the original levels. These hybrid levels can frequently go further in meeting individual environments' needs while minimizing the drawbacks.

RAID Level 10



RAID 10 is the most basic hybrid level, combining, as the designation implies, features of RAID 1 and RAID 0. The combination utilizes the mirroring found in RAID 1 with Level 0's data striping.

Different methodologies are employed to create the combination. RAID 0+1 stripes the data across mirrored drive sets. This configuration may be referred to as a *stripe of mirrors*.

The converse implementation is RAID 1+0, in which the information is striped across numerous drives, and the entire array is mirrored by one or more mirror arrays. This is called, predictably, a *mirror of stripes*.

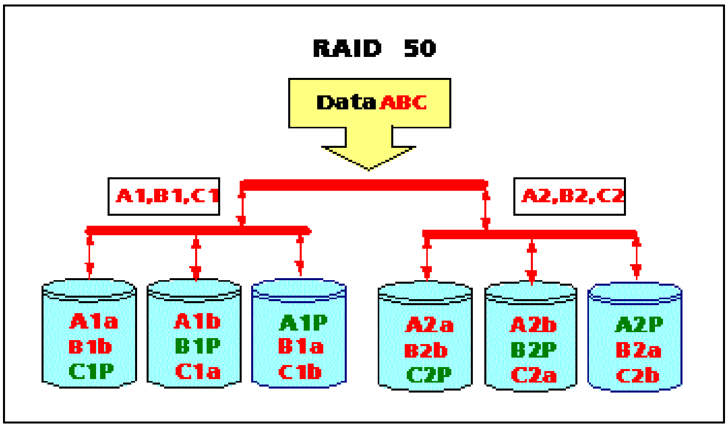
Level 10's unique advantages include the data transfer strength of striped arrays and high accessibility of mirroring. As with all non-parity arrays, recovery performance is very good, since data need only be copied from existing mirrored data.

Cost benefits are, however, affected by the need for at least two complete sets of the data.

RAID 10 is still very popular due to its comparatively simple implementation combined with high performance and superior redundancy.

Enterprise environments frequently consider the 50% storage increase an acceptable expense, given this configuration's throughput advantages and excellent fault tolerance.

RAID Levels 30 and 50



In Levels 30 and 50, parity operations are combined with disk striping. Basically, RAID 30 or RAID 50 comprise arrays with information striped across two RAID 3 or RAID 5 configurations.

Benefits from use of these levels depend on the stripe size employed in combination with the specific parity type. The independent parity access found in RAID 30 (as in RAID 3) serves high-throughput needs, while RAID 50's distributed parity (from RAID 5) satisfies high transfer rate considerations.

Naturally, the presence of parity operations entails a loss of performance in rebuilding a failed drive. As well, costs are somewhat higher with the requirement for an additional dedicated drive (Level 30) or dedicated shared space (Level 50).

CHOOSING A RAID LEVEL

As mentioned, the selection of an appropriate RAID level calls for a detailed assessment of the environment's specific needs, as well as cost, space, and personnel considerations.

The following table outlines the comparative features of the most popular RAID levels.

RAID Level	Capacity	Data Reliability	Read/Write Performance	Rebuild Performance	Suggested Applications
RAID 0	100%	N/A	Very good	N/A	Non-critical Data
RAID 1	50%	Excellent	R: Very good W: Good	Good	Critical Information
RAID 3	(n–1)/n	Good	R: Good W: Fair	Fair	Single-user, large-file processing, eg: video, image processing
RAID 5	(n–1)/n	Good	R: Good W: Fair	Poor	Database, transaction-based applications
RAID 6	(n-2)/n	Excellent	R: Very good W: Poor	Poor	Critical Information w/ minimal cost
RAID 10	50%	Excellent	R: Very good W: Good	Good	Critical Information w/ better performance
RAID 30/50	(n–1)/n	Excellent	R: Very good W: Good	Fair	Critical Information w/ fair performance

**n= disk drive complement*

RAID IMPLEMENTATION

As with the various RAID levels, implementation of the array into the system takes many different forms. The RAID array, irrespective of configuration, must have its operations managed effectively. The method of management is referred to as *control*, and the system component responsible is called the *controller*.

Again, the selection of an appropriate method can only be made after considering the application's needs and the demands likely to be put on it, as well as cost and the relative complexity of the entire system.

Software-based Control

This method is normally found only in simpler, entry-level servers or workstations. Microsoft Windows NT/ 2000, Novell Netware, and Linux, for example, are networked Operating Systems that incorporate integrated software-based control of certain RAID levels including 0, 1, and 5.

The major advantage inherent in software-based control is cost.

Since the OS is already in place, no additional expense need be borne by the user to achieve control of the RAID system.

It does, however, present serious potential for performance disadvantages, especially if data recovery is required. As a software element, this form of control necessarily uses CPU bandwidth, and in complicated or processing-intensive operations such as those in a rebuild operation, the amount of microprocessor resource utilized can be considerable.

As well, some software-controlled arrays will be unable to boot in the event of a failure in the array's boot drive. In most cases, software controlled implementations require the presence of a separate boot drive, a drive not part of the array.

An additional limitation is this method's inability to support hybrid RAID levels.

Hardware-based Control

Two popular types of hardware-based RAID control are *Embedded RAID control* and *PCI card-based RAID control*.

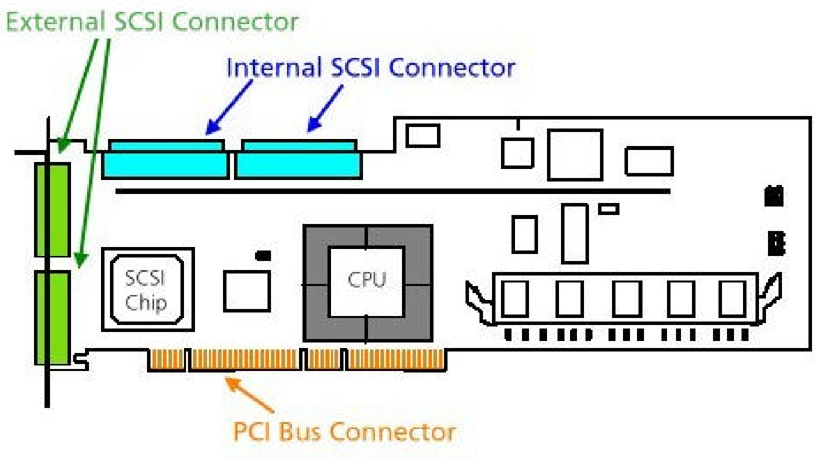
Embedded RAID Control

In this case, the operating appliance is carried on the system board (also known as *RAID on Mother Board*), similar to embedded audio and embedded network control.

While incurring additional expense compared to a software solution, this form of controller is still significantly more cost effective than a full independent hardware controller component.

Features available are likely, however to be somewhat limited compared to card-based control. Some embedded controllers do feature I/O management processors, similar to those available on some card-based controllers, able to assume the processing burden of intensive RAID operations such as parity calculations, as well as being employed in non-RAID operations.

PCI card-based RAID control



Card-based controllers provide the maximum level of flexibility and scalability for system expansion, justifying a large part of its more expensive characteristic. Additionally, card-based control can be *duplexed*, or doubled, providing significant improvements in fault tolerance. Available features in this type of control vary with price. Some card controllers are offered as reasonably priced alternatives to embedded control, and some carry a formidable array of high-performance capabilities, with countless variations between the two.

Some of the more advanced features available include:

- Built-in XOR processor for dedicated parity support
- Multiple channels to accommodate system expansion
- Battery supported cache memory for enhanced I/O performance

The most flexible form of control, PCI card controllers allow users to not only expand the system as needed, but also replace the controller itself when more advanced features become available.

About Caching

Caching refers to the controller-managed capability for advance storage on chip of what the system considers to be upcoming information, thus attempting to ease some of the performance deficits inherent in Read/Write operations. Any controller's ability to execute and mange caching is an important consideration, but the user must identify the appropriate caching type for their own individual needs.

Read-ahead caching

In this method, the data outlined in the Read request is read, but a portion of the subsequent information on the disk is also read, and stored in the controller's cache memory. If the controller's projection is warranted, and the subsequent information is indeed requested in the Read operation, the information is available through the much faster system I/O bus access, the actual data disk need not be read, and performance is greatly improved.

Depending as it does on the likelihood that the next successive section of data will be the next requested section of data, read-ahead caching is a very efficient performance boost for applications involving the access of large sequential records, such as multimedia files.

Conversely, it is unsuited to environments in which a more random read selection is probable, such as database or transactional operations.

Some controllers can be configured to initiate or disable read-ahead caching, and some carry the capability to intelligently activate the mode if conditions are suitable.

Write caching

Write caching can allay the write penalty present in some RAID types. RAID 4 and RAID 5 are especially suited to its application. Write caching takes two forms, *write-through* and *write-back* caching. In write-through, the controller must wait for the entire write to be completely written to the disk prior to beginning the next request.

Write-back, however, allows significantly higher performance gains, since the controller will commence the next requested write operation when the information is cached, but before it is actually written to the disk. Thus higher throughput is achieved.

Write-back caching does, however, carry a certain liability to data integrity. If system power or operation is interrupted prior to the actual writing of the data to the disk, the information is lost. The threat is compounded in parity-equipped arrays since both the pertinent data and the parity update information will be lost.

In this situation, power backup such as a UPS or battery backup for the controller is normally employed. Additionally (as mentioned later), some high performance controllers carry a built-in battery backup just for the cache, and the write is completed upon resumption of the power supply.

IDE RAID Card

Originally, RAID connection through IDE was considered too low-performance, compared to SCSI, to be considered a viable alternative for users.

Recent years have seen a dramatic upswing in the usability and popularity of

IDE RAID, however, thanks to both the rapidly increasing performance capabilities of IDE drives and the nearly unforeseen growth of smaller, yet more data-intensive computing environments.

IDE provides inexpensive, yet high performance RAID redundancy and high-capacity storage for users whose operations would have been considered prohibitively small previously. A key advantage is IDE RAID's ease of installation, thanks to built-in intelligent capability.

IDE RAID card configurations are especially suitable for low-end servers or high-end workstations. They normally provide support for 4 (hard disk) drives through two or four channels. Some IDE RAID cards can support four disks through four distinct channels. This one channel-per-drive interface offers users increased I/O performance.

With support for RAID levels 0, 1, and 0+1, and hot spare drives, IDE is quickly becoming one of the most popular forms of RAID implementation. Additionally, some IDE RAID cards carry a special XOR processor on board, which provides support for RAID 5.

RAID AND AVAILABILITY

While availability can be supported by the choice of RAID level employed, additional measures should be take in other areas of system configuration and management, so as to provide the highest possible level of protection.

Hot-spare Disks

Hot-spare disk drives offer a distinct improvement in data availability when supported by the controller.

These are disks that have the capability to be automatically implemented without shutting down system operations, thereby avoiding costly system downtime.

Hot spare disks can be configured to provide global backup to the entire data array, or dedicated backup to individual drives.

Upon drive failure, the controller can rebuild lost data located on the failed drive, writing it to the new hot-spare disk to quickly replace the failed drive. This capability, carried on in the background, again without interrupting operations, is called *automatic failover*.

After implementing a replacement disk, once normal operations are restored, the system can then go on to prepare additional hot spare disks, replacing those already pressed into service.

A replacement hot-spare disk must, necessarily, carry a capacity equal to or greater than the disk it is replacing.

Hot-swap Disk

These are disks utilizing a specially constructed connector allowing replacement without interrupting system power.

The system may carry a hot-swap tray, accepting disks to be hot-swapped. An available feature in some hot-swap trays is an adapting connector that accepts standard disks but converts the connection to hot-swap capability.

In the event of a drive failure, drives can be replaced quickly without powering down, and the new replacement drive can be assigned to the RAID rebuild, or be designated as a hot-spare.

Online RAID Capacity Expansion

Certain RAID controllers provide the capability to perform *Online RAID Capacity Expansion*.

In this case, users are able to expand available disk space without shutting down the system. All necessary operations to add storage capacity, including the striping of disks as desired, take place with the system up and running.

While this capability can be extremely valuable to systems facing the possible need for expansion in an environment where availability is a prime concern, it should be noted that in most cases, a decrease in performance will occur when the expansion operation is in progress.

RAID AND FAULT TOLERANCE

Finally, although this discussion has focused on RAID technology, users should be aware that true fault tolerance cannot rely only on the RAID array, irrespective of its characteristics.

In the final analysis, no matter how many levels of data protection are provided by the array, RAID defends the system from data loss associated with disk failure only.

Power interruptions, component failures, inadvertent deletion, environmental hazards such as fire, flood, and earthquake, as well as occurrences of malicious damage, can all jeopardize the integrity of stored data. These and many other potential dangers are well beyond the scope of RAID protection.

For this reason, backup operations for all critical data, outside of any system-based fault tolerance measures, are absolutely crucial and should be performed regularly. Media for this type of backup can include tape backup systems, or other means located entirely outside of the normal storage system.

System Redundancy

Components and system capability can provide benefits of fault tolerance to entire systems.

These elements include:

- **UPS (Uninterrupted Power Supply)-**Backing up the entire system, enabling continuous operation in the event of local power outage
- **Redundant hot-swappable power supply-**Accommodates failure of system power, first by providing redundant backup, secondly by allowing replacement without powering down
- **Redundant hot-swappable fans-**Accommodates failure of cooling system as with power supplies, first by providing redundant backup, secondly by allowing replacement without powering down
- **Redundant controller-**Provides full fault tolerance to control of the RAID array operations
- **Server clustering-**Perhaps the optimum level of fault tolerance, complete server backup is provided by a clustered group

FOR MORE INFORMATION

For more detailed information regarding RAID Technology, please refer to:

RAID Advisory Board (RAB):

http://www.raid-advisory.com/

Mylex Corportation:

http://www.mylex.com/

Adaptec, Inc.:

http://www.adaptec.com/

American Megatrends Inc.

http://www.ami.com/