HP Disk Array: Mass Storage Fault Tolerance for PC Servers

In the process of offering a new technology to the marketplace the expertise of the user is often not considered. The HP Disk Array offers RAID technology with special installation and configuration features tailored for ease of use.

by Tom A. Skeie and Michael R. Rusnack

The rapid proliferation of PC-based local area network systems and the growing dependence on them has resulted in concern about the reliability of such systems, particularly in the area of data storage. PC-based systems do not have the history or perception of reliability that minicomputer or mainframe systems have built up over the years. Thus, it is no surprise that concern about fault tolerance is becoming a critical issue in the PC community. This concern has led to a great deal of interest and research into external storage systems and redundancy.

The significant improvement in CPU performance in the 1980s (50 to 100% per year) created the need for similar performance improvements in other system components, such as memory and system storage. Improvements in memory performance came mostly from new memory architectures and algorithms rather than in the memory components. Similarly, magnetic hard disk technology, the most commonly used component for system storage, could not provide the needed performance improvements. A technology called RAID (Redundant Array of Inexpensive Disks)[1] was proposed as a method for improving the I/O throughput for system storage. The basic proposal suggested that the I/O bandwidth could be improved by spreading the data over multiple hard disks to obtain some level of concurrency. However, because of the unacceptable trade-off in system storage reliability caused by this data distribution,[2] multiple data-redundancy algorithms (RAID levels) were proposed. A brief tutorial on RAID technology is given on page 74.

The benefits of improved I/O throughput and data protection have made RAID technology widely accepted on computer platforms of all kinds, including PC servers. While many architectures can efficiently implement a RAID storage system, the generic architecture includes the mechanism for implementing one or more of the RAID algorithms (typically a hard disk controller) and a method for communicating and controlling a group of hard disks (see Fig. 1).

The HP Disk Array implements RAID using a hardware-based EISA-to-SCSI controller in which multiple RAID algorithms are implemented in the controller's firmware. A controller implementing RAID is commonly referred to as a disk array controller. For the HP Disk Array controller five hard disks are connected to each of the controller's two SCSI buses giving a total of 8G bytes (assuming RAID level 5) of protected capacity controlled by each controller. Fig. 2 shows the hardware components of the HP Disk Array. In addition to the benefits of RAID, the HP Disk Array offers a wide range of features, such as predictable failure notification, automatic disk-failure detection, and automatic disk-failure recovery. What really differentiates the HP Disk Array from a majority of similar RAID products is the focus on making the complex RAID technology readily available to PC network administrators who may not have an in-depth knowledge about RAID technology, SCSI technology, or the target operating system.

After an investigation of other systems offering RAID technology, we found that it was difficult to install and configure many of them properly and to recover from a disk failure, even for knowledgeable PC network administrators. We saw this as an opportunity to differentiate the HP Disk Array product from other disk array products. New ease-of-use software utilities were designed specifically to simplify the usability of the HP Disk Array product.

The final product includes several complementary ease-of-use features such as:

- A disk array controller optimized for performance with the selected disks
- Hard disks prejumpered, configured, and mounted in hot-plug* enclosures
- A disk cabinet with automatic SCSI ID selection

* Hot-plug in this context is the ability to replace a disk while the system is in operation.

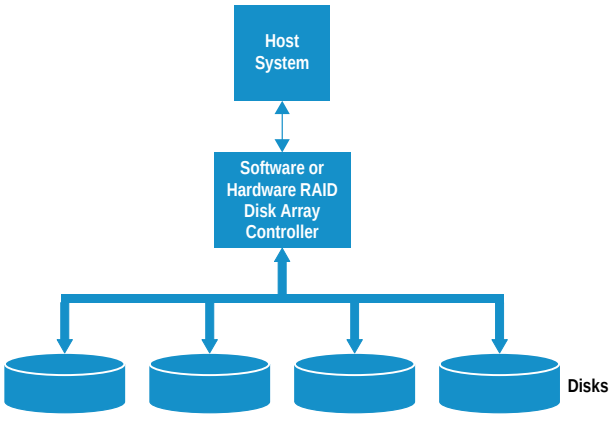

Fig. 1. Generic RAID architecture.

Fig. 2. The components of an HP storage system. The SCSI-II controller, the disk array controller, and the disk module make up the HP Disk Array.

- Software designed specifically for easy configuration and integration of the RAID system into any network operating system.

This article describes the decisions made during the design phase of the HP Disk Array that resulted in making the RAID technology significantly more usable for the typical user without sacrificing valuable product features.

Storage Categories

In a typical computer there are many places to store information either temporarily or permanently. Some of these storage media include RAM, shadow RAM (BIOS), video RAM, read cache, write cache, hard disks, flexible disk, and tape. Whatever the storage medium, computer storage can generally be classified as either primary storage or secondary storage.

Primary storage is often referred to as the system memory, or just the memory. The memory is typically made from semiconductor components and is characterized as being very fast, volatile, and expensive. Computers use memory for immediate and temporary storage. This is the storage containing the data that the CPU operates on and manipulates. Most servers today typically have about 16M bytes to 64M bytes of primary storage.

Secondary storage is often referred to as the system storage, or just the storage. The storage medium is most often magnetic and is slower, nonvolatile, and much cheaper than memory. The storage is therefore used as a permanent repository of information. Servers typically have between 1 Gbyte and 100 Gbytes of storage.

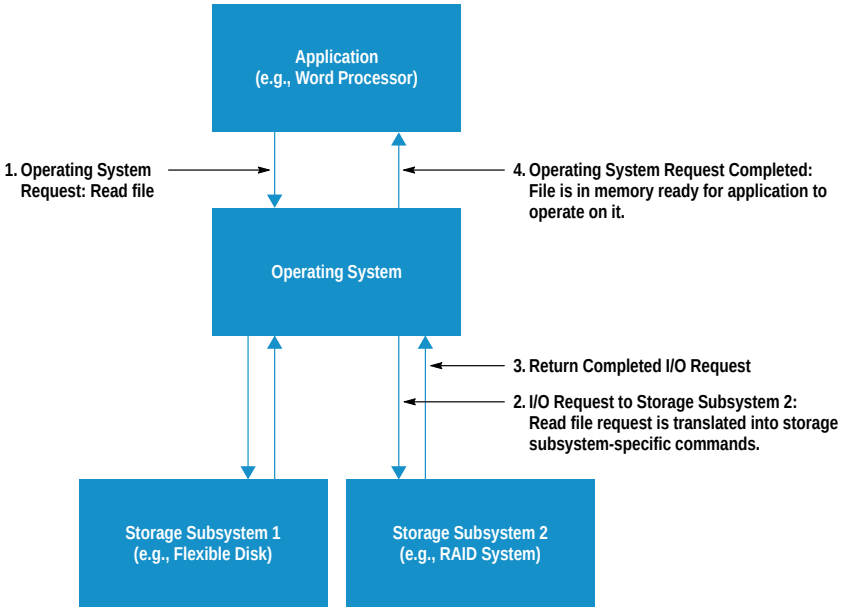

Fig. 3. A typical I/O request flow.

The relationship between these two types of storage is based on how information is transferred in a typical application.

The I/O Process

When an application such as a word processor wants to open a file, it is usually the operating system that is responsible for completing that request. Depending on where the file is currently located (it could already be in memory), the operating system may generate one or more I/O requests to the storage system to bring the file into memory. This process is illustrated in Fig. 3.

A couple of issues should be pointed out here. First, the application does not know how or where the operating system has stored the file it is requesting. Similarly, the operating system may not know how or where the storage subsystem has stored the data making up the file.

For instance, in a RAID system one I/O request from the operating system may result in multiple transfers in the storage subsystem. Since RAID 1 (see page 74) mirrors or duplicates all information from one hard disk to another, every time the operating system requests a file to be written, the RAID subsystem must write this file twice. If the RAID subsystem is using RAID 5, writing a file may result in multiple reads and writes.

Since several I/O transfers may result from a single request from the operating system, something in the system must have the intelligence to distribute the data properly and know how the data should be retrieved. Obviously, in a modular system such as the one described above many approaches to implementing this intelligence in a RAID storage system can be imagined. The following section explores the most common RAID storage system approaches and some of the advantages and disadvantages of the different approaches.

RAID Architectures

A wide variety of architectures have been explored in the effort to provide better I/O throughput and data protection based on RAID technology. These architectures can be divided into two major categories based on whether the RAID algorithms are executed in the host CPU or a dedicated CPU. When using the host CPU, the architecture is generally thought of as a software solution, and when a dedicated CPU is used, the solution is said to have a hardware architecture. The major difference between these architectures most often amounts to a trade-off between cost and performance.

Software RAID Architectures. Since a software RAID architecture uses the host system CPU to execute, one obvious disadvantage with this approach is the potential for lower overall system performance than with a dedicated CPU as in the hardware RAID architecture. However, in many environments, the host CPU is not used to its full capability. Therefore, using the server CPU to execute RAID algorithms provides the most cost-effective RAID architecture while still providing adequate system performance.

Several operating systems now offer some method of data redundancy by implementing the RAID algorithms as part of the operating system (see Fig. 4a). The advantage of this approach is that there is no additional cost to the user. However, the performance may be affected in CPU-intensive application server environments such as database engines.

Not every RAID level is offered through the operating system, and some operating systems offer no data redundancy scheme at all. This has prompted some storage system vendors to provide custom software that implements the desired RAID scheme. Fig. 4b shows the software architecture in which the the desired RAID level is provided with an intelligent device driver.

Hardware RAID Architectures. In environments where the server is used for CPU-intensive applications, adding the overhead of executing RAID algorithms may significantly

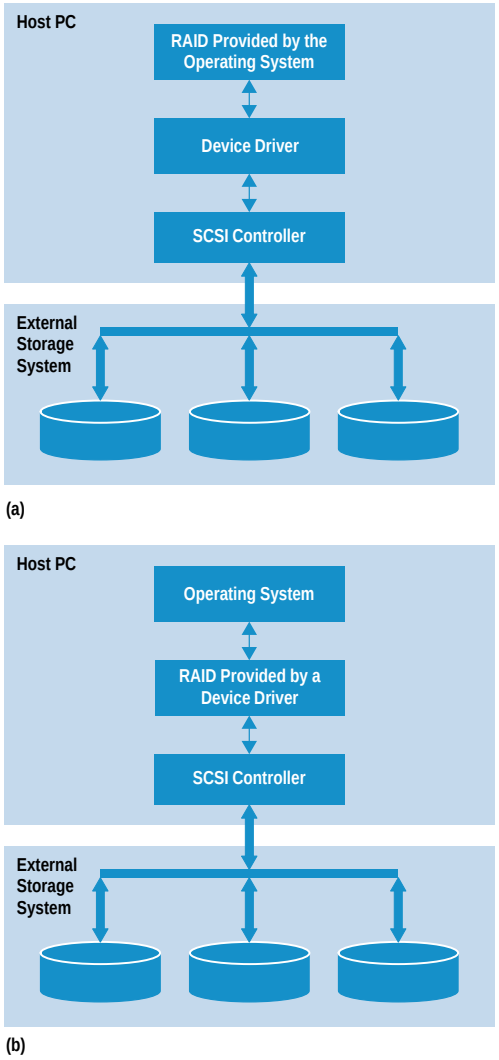

Fig. 4. Software approaches to implementing the RAID technology. (a) Software inside the operating system. (b) Customized software drivers.

An Overview of Raid Technology

The Need for Information Storage

The trend today is to put increasingly larger amounts of an organization's information onto some sort of electronic media for quicker, concurrent, and geographically independent access. The information being put on the electronic media is often critical to the operation of the organization. This trend has created an increasing demand for newer and better methods of electronic storage . One such electronic storage method is data protection via RAID (redundant array of inexpensive disks) technology.

RAID technology finds its roots in a series of papers referred to as the Berkeley papers.[1] Although these papers do not precisely define RAID, they do provide the basis for the architecture of the technology. The need for disk drive redundancy developed as the need for data integrity and system reliability became a growing issue. For example, the consequence of a single point of failure in a disk storage system could result in the failure of the entire computer system. This loss of productivity and often data was deemed unacceptable.

Seven RAID levels are defined. Each level specifies a different disk array configuration and data protection method, and each provides a different level of reliability and performance. Since only a few of these configurations are practical for most online transaction processing systems, file servers, and workstations, RAID levels 0, 1, 3, 5, and 6 are described here.

RAID 0

Although it is often debated that since RAID 0 is not redundant and should therefore not be considered as a RAID mode, nearly all RAID solutions include this mode. RAID 0 distributes the data across all the disks in the disk array configuration (see Fig. 1) Since there is no redundancy, the capacity utilization is 100 percent.

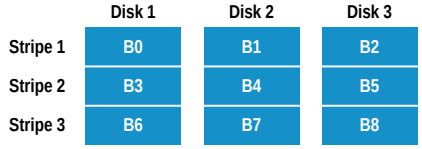

Fig. 1. RAID 0 configuration.

The data blocks in Fig. 1 are broken into three parts and striped across three disks. What this means is that while data is being retrieved on disk 1, the data on disks 2 and 3 can be requested and ready to transfer sooner than if three I/O requests were made.

RAID 1

To answer the need for reliability and data integrity, system managers have often implemented a solution in which write data is mirrored on two separate disk systems. This implementation is referred to as RAID 1. The primary advantage of RAID 1 is its simplicity. RAID 1 provides a slight improvement in read performance over the other implementations. However, write performance is poor because all data is duplicated. The primary disadvantage of RAID 1 is cost, because for every byte of storage used on a system an equal amount of storage must be provided as a mirror. This results in a cost differential of 100% over standard nonredundant mass storage.

In Fig. 2, the data is mirrored on each disk. In the event of a failure on disk 1, the same data is available on disk 2.

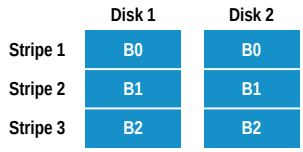

Fig. 2. RAID 1 configuration.

RAID 3

The architecture for RAID 3 is often referred to as a parallel array because of the parallel method that the array controller uses in reading and writing to the disk drives. For RAID 3 it is necessary to provide two or more data disks plus an ECC (error correcting code) disk. Data is dispersed or striped across the data disks, with the ECC disk containing an exclusive-OR of the data from the other disks. Unlike the other RAID solutions, the data is dispersed across the disk in a byte interleave rather than the typical block interleave. With the spindles all synchronized, the data is placed on the same cylinder, head, and sector at the same time. In the case of RAID 3, each drive is connected to a dedicated SCSI channel, which further ensures the performance.

This architecture can handle any single disk failure in the chain. If a data drive fails, data can be recovered from the failed drive by reconstructing the exclusive-OR of the remaining drives and the ECC drive. The advantage of this scheme is that redundancy is achieved at a lower cost (compared to RAID 1). The primary disadvantage is its I/O performance for small amounts of data. When an application requires the transfer of large sequential files, such as graphic images for workstations, this is the best method.

Fig. 3 shows the configuration for RAID 3. The data is striped across drives like the RAID 0 configuration. If a failure occurs, the data is reconstructed based on the parity data on disk 3.

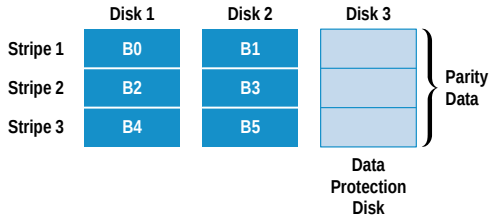

Fig. 3. RAID 3 configuration.

RAID 5

RAID 5 was defined in an effort to improve the write performance of RAID 1 and RAID 3 systems. Like RAID 3, the data blocks are distributed over the disk drives in the system, but unlike RAID 3, the ECC data is also distributed across all the drives (see Fig. 4). With this configuration reads and writes can be performed in parallel.

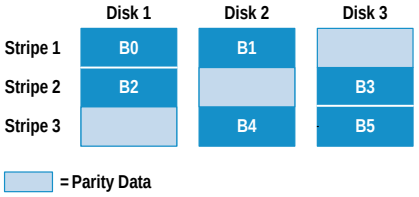

Fig. 4. RAID 5 configuration.

RAID 6

Like RAID 0, RAID 6 is not yet accepted as a standard RAID configuration. This configuration not only mirrors data (like RAID 1) but also stripes the information (see Fig. 5). Because the data is striped, the performance is very similar to that measured in a RAID 0 configuration. The penalty for use of this configuration is that 100 percent more disk space is required.

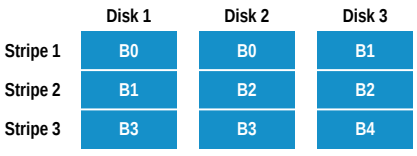

Fig. 5. RAID 6 configuration.

Reference

1. D. Patterson, G. Gibson, and R. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)". *ACM SIGMOD Conference Proceedings*, Chicago Illinois, June 1988, pp. 109-116.

lower the overall system performance. A separate CPU dedicated to executing the RAID algorithms is therefore used in a hardware RAID architecture to offload RAID algorithm execution from the server CPU. Since this architecture requires additional hardware, the major disadvantage here is a higher cost than a software architecture.

The hardware RAID approaches can be categorized as internal or external solutions, referring to whether the RAID hardware resides inside or outside the host server. Internal RAID hardware has to interface directly with a host system bus, such as with an EISA or PCI bus. Disk drives are then connected to the internal RAID hardware directly via one or more SCSI buses (see Fig. 5a). While such a solution may offer superior performance, it does have some subtle disadvantages. Since the RAID hardware must interface directly with the host server hardware and operating system, there is a high level of dependence between the host and the RAID hardware. This dependence affects many aspects of the product's usability, such as the serviceability of failed hardware. For instance, if a problem occurs in the RAID hardware, the entire server must be shut down for repair.

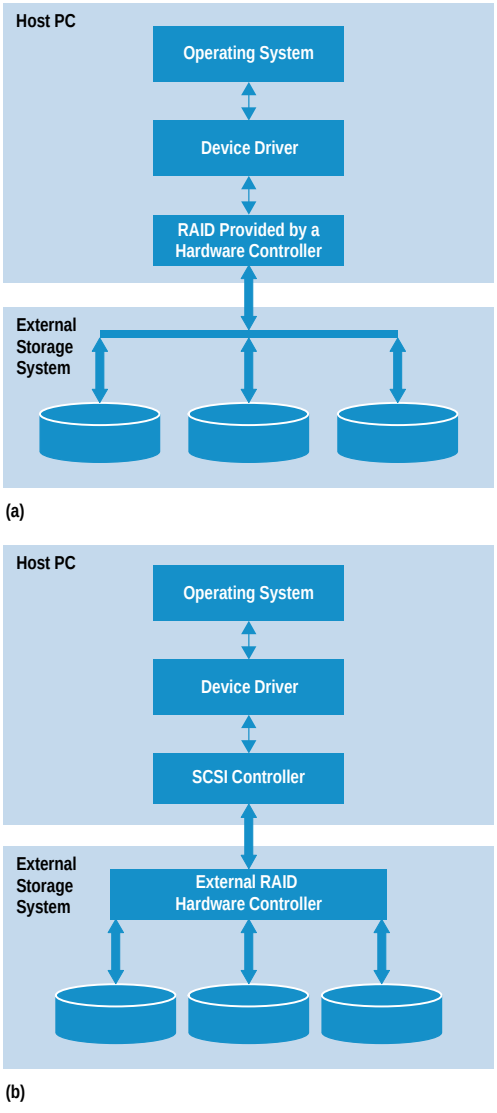

Fig. 5. Hardware approaches to implementing the RAID technology. (a) Inside the host system. (b) External to the host system.

This is not the case with an external RAID hardware solution. Here the RAID hardware is typically residing in the same enclosure as the hard disks, external to the host system (see Fig. 5b) and therefore provides more independence from the host system. In addition to making an external product more serviceable, the independence allows the RAID system manufacturer to more readily support new host hardware systems and new operating systems. Since the external RAID systems require additional hardware, they tend to be more costly and have lower performance than internal solutions.

Many other architectural differences exist in RAID systems that affect the price/performance ratio. For instance, the RAID hardware may use onboard RAM for caching data. Such caching can provide a significant performance improvement, especially when executing RAID level 5 write operations. For a sequential write operation, RAID hardware can calculate the ECC for the stripe without having to perform any read operations if the controller has enough memory to cache one entire stripe. Recall that the amount of data in a stripe depends on two things: the chunk size and the stripe set size. For example, if the chunk size is 64K bytes and there are four disks in the stripe set, the required cache for one stripe would be $4 \times 64K$ bytes = 256K bytes. For the PC server environment, RAID hardware commonly uses between 4M bytes and 64M bytes of onboard RAM shared between read and write operations. Thus, cache can become a significant cost increase to the overall RAID system.

Architectural differences such as the number and type of disk interfaces used (e.g., fast SCSI, fiber, and SSA (Serial Storage Architecture)) can also affect the attributes of a RAID hardware product. For RAID systems that manage large amounts of storage, multiple disk interface buses may be necessary to obtain the best bus utilization. The CPU capabilities will also affect the RAID system price and performance.

The HP Disk Array Architecture

The HP Disk Array product was intended to take advantage of the emerging disk array market for PC servers. It was to be offered as an integrated storage solution for HP's network server product, HP NetServer, as well as a third-party storage product. As discussed briefly above, there are many viable RAID system architectures and features with different end product qualities. Deciding on which architecture and features would best fit the HP Disk Array product was based on the following product requirements:

- Time to market. Because of the dynamics in the PC market, a product (or project) rapidly becomes outdated. If a product is not released within a certain period of time (i.e., "window of opportunity"), it can mean the difference between product success and failure. As mentioned, the HP Disk Array was also to be integrated as part of a PC server developed by a different HP Division. Therefore, two separate programs were dependent on a timely release.

- Operating system support. HP has an established customer base on a wide range of PC server operating systems. While some companies offer new products with limited operating system support, the HP product requirement called for support on all common PC server operating systems. At the time of the first release of the product this included DOS (for boot support), Novell NetWare, IBM OS/2, SCO UNIX®, Microsoft® OS/2, and Banyan VINES.
- Capacity. Based on market research that a typical PC server had a storage need of 1 Gbyte to 5 Gbytes, the HP Disk Array was targeted to offer capacities ranging from 2 Gbytes to 7 Gbytes.
- Performance. Several market research studies found that improving I/O performance was the main reason to purchase a disk array among most PC server users. Performance was therefore an important product requirement.
- Ease of use. Entry-level products should have ease of use as one of their primary attributes. This was not the case with many of the RAID systems we examined. Thus, making it easy for our customers to use RAID technology was an important requirement for the HP Disk Array product.

While all of these product requirements contributed to the product's success, most of them can be considered to be entry-level requirements. These are the criteria necessary to be considered a competitor in this market.

The following sections describe in more detail some of the challenges the R&D team had to deal with to make the HP Disk Array RAID product easy to use.

Installation of a RAID System

Since the PC environment is considered open (that is, hardware and software from one PC manufacturer are expected to operate with any other PC manufacturer's hardware and software), products tend to be very generic when they reach the end user. This often makes the installation and configuration of new hardware and software more troublesome for the user. Installing a new RAID system on a new or existing server is no exception. In fact, it is generally more difficult since the hardware and the RAID system must be configured. The installation of a RAID system involves three major steps: installing and configuring the hardware, configuring the RAID system, and then making the added storage capacity available to the operating system's storage pool by partitioning and adding a file system. This process can be quite intimidating, even for the most advanced user. Since the last installation step is the same (at least very similar) for any new storage, the following sections will focus on the first two steps.

Hardware Installation and Configuration

After unpacking and verifying that all the parts of a disk array system are available, the user must first install and configure the hard disks and then proceed to do the same for the RAID controller.

Installing and Configuring the Hard Disks. Most RAID controllers communicate with the hard disks via the SCSI (Small Computer System Interface) interface protocol. SCSI is a general-purpose interface protocol that allows multiple devices such as disks to communicate in a peer-to-peer fashion on a parallel bus interface. Each device is referenced with a device number, or SCSI ID. If the device numbers are not properly set up (e.g., two devices having the same ID number), the SCSI system will not work.

In addition to setting the SCSI ID, several SCSI interface parameters must also be properly set for each device on the SCSI bus. Among these are settings for synchronous and asynchronous data transfer, data parity check, and maximum data transfer speed. The SCSI devices are configured by using hardware and software switches.

Some RAID systems provide the opportunity to load-balance the I/O transfers between multiple SCSI buses. For instance, if the RAID system has seven hard disks and each disk can offer a sustained transfer rate of 3 Mbytes/s, a single SCSI 8-bit (narrow) bus with a maximum transfer rate of 10 Mbytes/s could be saturated (7×3 Mbytes/s = 21 Mbytes/s for commands and data and 0.66×21 Mbytes/s = 14 Mbytes/s for data assuming a typical 33% SCSI command overhead).* Thus, the user must know the approximate characteristics of the SCSI bus and the disk drives to distribute the disk drives properly.

Before hard disks can be used for storing useful data, they have to be low-level formatted. This is a process in which the magnetic medium of the disk drive is divided into generic fixed-size sectors (typically 512 bytes) where data can be stored.

Installing and Configuring the RAID Controller. Setting up the RAID controller is perhaps the most difficult and definitely the most critical part of the hardware installation. During this process the user has to power down the server, take the cover off the server PC, and install the RAID controller into an empty slot. If not handled properly, static electricity could cause the entire server to malfunction after this operation. Further, if the controller has hardware jumpers, they must be configured before the controller is installed in the server.

Once the system is put back together again, a configuration program, such as the EISAConfig program for an EISA system, is run to assign host system resources to the controller (e.g., IRQ and BIOS addresses). Since the RAID controller has one or more SCSI interfaces in addition to EISA, it is very common to configure the controller's SCSI parameters with the EISAConfig utility.

RAID System Configuration

Once the hardware is installed and configured, the RAID system configuration can begin. To the user this is probably the greatest challenge in the entire installation process because the RAID technology introduces many new terms and concepts such as stripe set size, chunk size, and RAID level. The user must understand how the selection of a parameter will affect the system in terms of data protection, system performance, and capacity utilization.

Fig. 6 shows the dependency relationships between some of the configurable RAID system parameters. Note that Fig. 6 is not meant as a complete picture, but only as an illustration of the complexity involved in determining the values to assign to these parameters. The operating system is the only

* The maximum number of devices on a narrow SCSI bus is eight, and since the disk controller is considered to be one device, a maximum of seven hard disks can be connected to the bus.

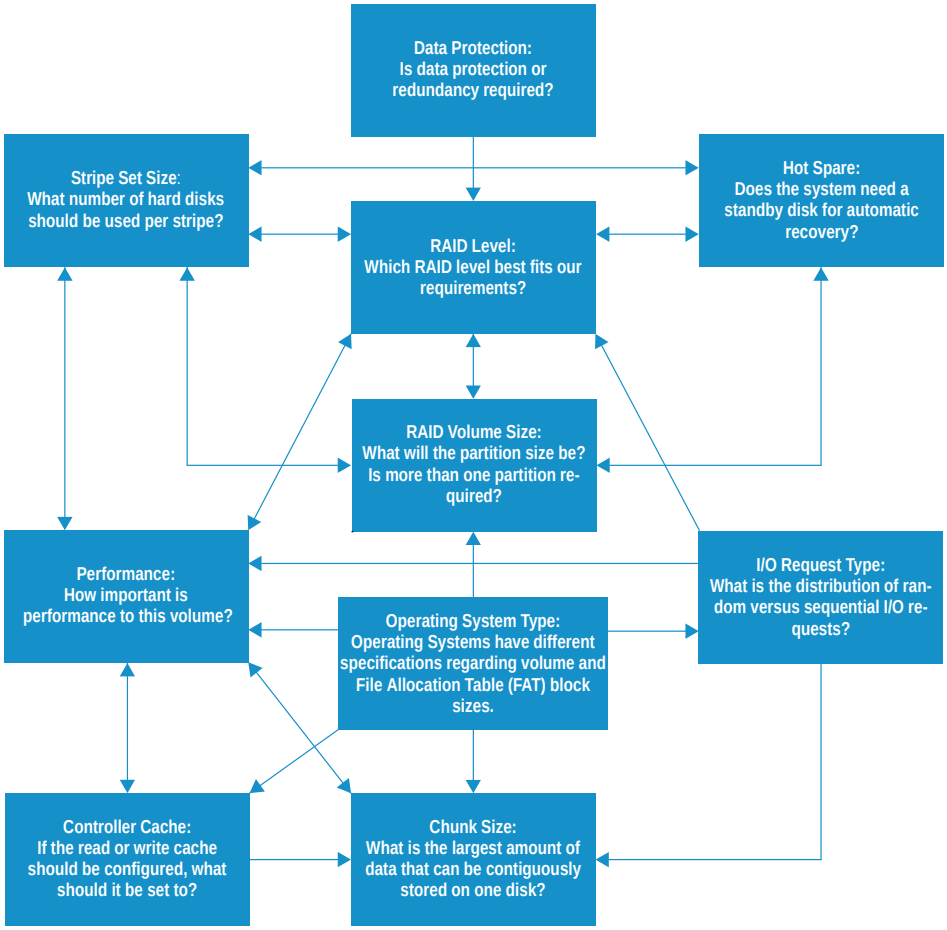

Fig. 6. Dependency relationships between RAID parameters.

entity in Fig. 6 that has no dependencies because its configuration is typically fixed by the time the storage system is installed. Without a thorough understanding of these dependencies, the user will not know what effect the value assigned to one parameter will have on the remaining installation. The following sections look at some of these parameters and their dependencies in more detail.

- Stripe set size. Before any other selections such as logical volume size and RAID level can be set, a decision must be made on how the hard disks should be grouped. A stripe set is the collection of disks that make up an array that implements a RAID level that typically uses a disk striping technique.[3] Disk striping involves spreading data over multiple disks in an interleaved pattern to improve performance (see Fig. 7). Selection of a stripe set greatly affects the remaining configuration parameters such as the RAID level and logical volume size. The behavior of the array will also be affected, especially in terms of its performance. Stripe set size has the following dependencies:
  - Performance. If the hard disk enclosure has multiple SCSI channels with hard disks distributed over multiple channels, it may be desirable to have multiple stripe sets to obtain better performance of the overall system. Also, depending on the nature of the data and the frequency of distribution, multiple stripe sets may be desirable.
  - RAID level. Each RAID level has its own constraints or requirements. For example, RAID level 1 can only be used with an even number of hard disks, while RAID level 5 can be used with an even or odd number of hard disks.

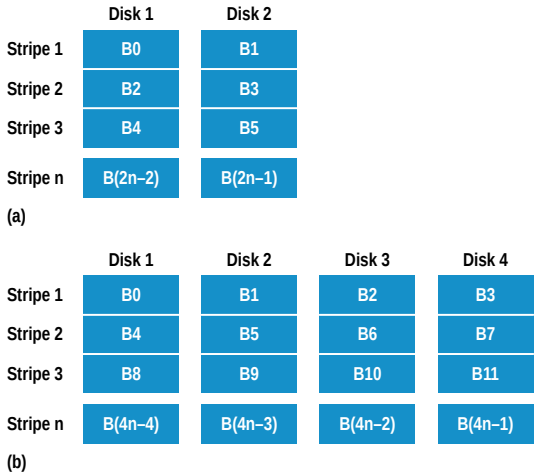

Fig. 7. Two examples of disk striping and different stripe set sizes. Data is spread over disks in equal chunks (or blocks) of data. A chunk is typically between 4K bytes and 64K bytes. By using this scheme of spreading the data among multiple disks, a controller can overlap operations to the disks and thereby improve the overall data access. For example, in (a) two chunks (e.g., B0 and B1) could be read in the same time it takes to read one chunk. If the system bandwidth allows, it is clear that the stripe set in (b) will have more concurrent data access than the stripe set in (a).

- Hot spare. Depending on the number of hard disks available, it may or may not be possible to have a hot spare (standby hard disk) that will automatically replace a failed hard disk in the stripe set. Further, unless all logical volumes defined in the stripe set are configured to be redundant (RAID levels 1, 5, or 6), a hot spare will not serve its intended purpose in the event of a disk failure.

- RAID level. Choosing the RAID level is one of the central parameters in the RAID system installation. It has a lot of interdependencies, including:
  - Data Protection. If the storage system is to withstand a disk failure, some redundancy scheme such as RAID 1 or 5 must be selected. If no protection is required, RAID 0 may be the best choice offering best capacity utilization and good performance.
  - Performance. Selecting a RAID level will have a significant impact on the I/O performance. For instance, if the I/O request types are random writes, RAID 5 is the lowestperformance selection one can make and RAID 6 would be a better choice.
  - RAID volume size. RAID levels 1 and 6 will consume 50% of the available storage capacity, while RAID 5 costs one disk out of the total number of disks in the stripe set. For example, if the stripe set consists of five disks, RAID 5 will offer four disks or 80% of usable capacity for data storage.
- RAID volume size. Some RAID systems allow multiple partitions within one stripe set. This allows the user to customize the storage by having a mix of volume sizes, RAID levels, and caching schemes. Some dependencies for the volume size include:
  - RAID level. As already mentioned, the RAID level dictates what remains of usable capacity.
  - Operating System. Some operating systems, such as OS/2, cannot use partitions larger than 8G bytes. Therefore, it may be necessary to make multiple partitions because of such operating system limitations.
- Chunk size. Selecting the proper amount of contiguous data stored on one disk (chunk size*), is probably the most difficult task in configuring a RAID system. It is easy to explain what chunk size is, but challenging to explain the implications of selecting different chunk sizes. Chunk size mainly affects the performance of the storage system, but the difficulty lies in predicting how the performance will be affected. The following considerations are important in selecting chunk size:
  - Operating system. Based on the I/O request size, the chunk size should be set to some multiple of the request size. For Novell NetWare, this poses an interesting dilemma because the I/O request size can be selected for each volume during installation of the file system. Other operating systems, such as SCO UNIX, have a fixed I/O request size.
  - Controller cache. Cache use will also affect how the chunk size should be set. For example, it is often desirable to fit full stripes in cache for RAID 5 writes.
  - I/O request type. The more sequential the requests, the more benefit larger chunk sizes will have.

The HP Disk Array Configuration and Installation

It is obvious from the discussion above that configuring a RAID system is not a trivial task. This section describes some of the features the HP Disk Array provides to make it simple for users to install and configure a RAID system.

To simplify hardware installation of the HP Disk Array, the hard disks come preformatted and preconfigured from the factory. Since HP is able to control the hard disks used with the RAID controller, it is possible to optimize all configurable parameters at the factory. Any configurable parameter that cannot enhance the product but could instead lead to less than optimized configurations has been removed. Even the SCSI ID selection has been automated. By selecting the

* Chunk size is also referred to as block size.

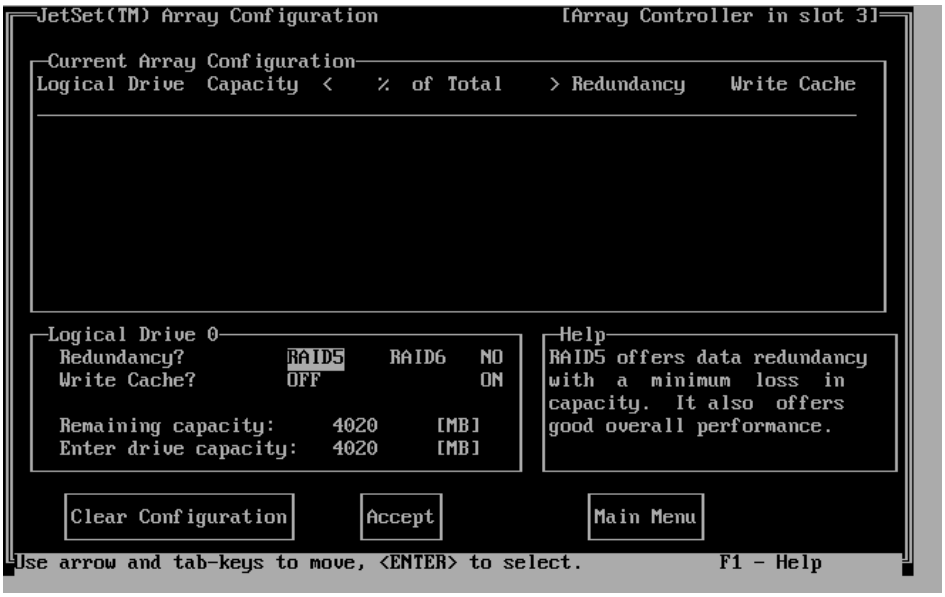

Fig. 8. JetSet configuration screen.

Fig. 9. A RAID configuration in which a DOS partition must be created.

slot in the cabinet, the SCSI ID is also automatically determined. For the RAID controller, the configurable parameters have been optimized as default settings. The SCSI bus use has also been optimized by matching the characteristics of the known disk drives that can be used with the product. The user does not have to understand any performance or configuration aspects of SCSI.

After the hardware has been installed, the user must configure the disk array. A bootable flexible disk is provided that prompts the user to select the language of choice. The installation process will then allow the user to configure the array.

The array configuration utility used with the HP Disk Array is called JetSet. Some of the features of JetSet utility include:

- Sensing and guiding. JetSet will sense if the array is unconfigured, if new hard disks have been added, or if a recovery operation is required. It will then guide the user to perform all the necessary steps to complete the operation.
- Limit choices. JetSet will only present options that make sense for the current operation. For example, a hot spare hard disk does not make sense for a RAID 0 volume, and the option should not even be offered to the user. This approach also limits clutter on the screen.
- Standard defaults. If the user presses the **Enter** key continuously through a few option screens, the array will be configured to the most common defaults. The user has to make different selections to deviate from these defaults.
- Context-sensitive help. JetSet offers extensive context-sensitive help throughout all menus. The goal here is not to require the user to consult a manual during the configuration process.
- Limit new terminology. JetSet is designed to keep the language as simple as possible and not use engineering jargon . A new technology typically brings new terminology. However, many new terms can often be explained with more familiar terms.
- JetSet for all operating systems. To further simplify the learning curve for the user, JetSet is made available in the same form for all supported operating systems. This provides the benefits of simplified software control and user's manual.

A sample of the main JetSet configuration screen is shown in Fig. 8. The user is only prompted to make decisions relevant to the configuration process.

Once the RAID system has been configured and initialized, it is ready for use by the server's operating system for data storage. If the RAID system is the only storage on the server, it can be made bootable by creating a DOS partition and assigning a DOS file system. This is the case for Novell NetWare as shown in Fig. 9.

This is an important point because some RAID systems will not allow the creation of multiple partitions within one stripe set. This increases the cost of the system because the user must either assign a separate hard disk apart from the RAID

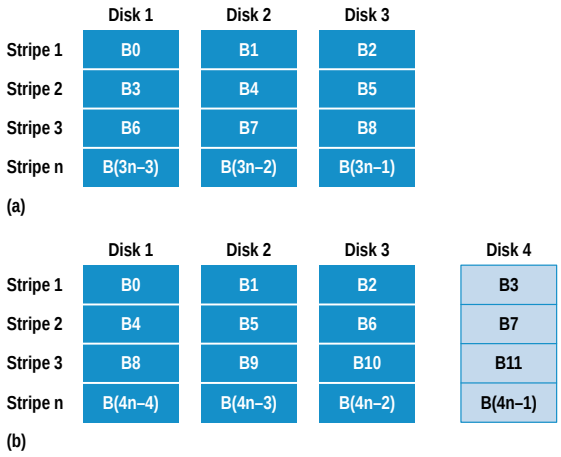

Fig. 10. (a) Data layout in a RAID 0 system. (b) Data layout after adding a new disk.

system as the boot device, or if the RAID system is the only storage system on that server, the DOS bootable partition would be a minimum of a single disk size (1 to 2 Gbytes) and if protected, would take that number times two. The HP Disk Array allows eight volumes for a stripe set using any mix of capacity, RAID level, and caching strategy.

Balancing ease of use and a rich feature set can be very challenging because most manufacturers of products such as RAID systems want to make the products general enough so that they can be used on any host system in any operating system environment with the most features of any available product (a marketing dream). Such generality also challenges the simplicity and ease of use of the system. The HP Disk Array is an example of how few trade-offs were made.

Adding Capacity to a RAID System

Adding capacity to an existing RAID configuration is a well-known dilemma with this technology. Since the data is spread over many disks, a new disk cannot simply be added without shuffling the data around. In Fig. 10 data is spread over three hard disks in an interval of block size B. Each hard disk has n blocks and therefore the array has n stripes. To be able to add capacity to the existing stripe set, some of the data shown in Fig. 10a has to be moved onto the additional disk such that the block layout ends up looking like Fig. 10b. This example is specific for RAID 0.

Adding capacity becomes more complex when the system is configured as a redundant RAID level. Most RAID systems require a complete backup, reconfiguration, and restore for a capacity increase. Algorithms are being developed to allow a capacity increase for an existing stripe set, but are beyond the scope of this discussion.

The HP Disk Array offers a new and simple scheme for adding capacity called *emergency capacity*. Rather than solving the complex problem described above, emergency capacity simply adds a new stripe set to the RAID system (see Fig. 11).

The caveat here is that the new stripe set does not have any disk array characteristics such as added performance and data redundancy. In fact, it is simply a JBOD (just a bunch of disks) configuration. However, it meets the customer's need of immediate capacity expansion. At a more convenient

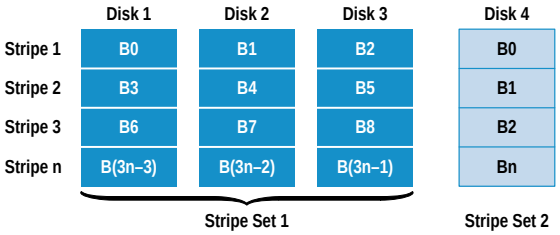

Fig. 11. Adding emergency storage capacity in an HP Disk Array configuration.

time, the user can perform the proper operation for capacity expansion.

Recovering From a Failure

Another major reason why organizations invest in RAID systems is protection against system downtime. According to the March 1994 issue of Byte Magazine, a survey of 450 information service executives at 1000 major companies found computer downtime cost an average of U.S.$78,191 per hour and occurred, on average, nine times a year. A typical outage cost U.S.$300,000 including the cost of recovering or reconstructing the data. However, failures can and do occur, making notification of a failure essential. Some failures do not cause any disruption in the I/O processing (performance may suffer some), while other failures will bring down the entire server.

Failure Notification. Before any manual restoration process can begin the system administrator needs to know that a problem has occurred. The failure notification method is therefore a very important part of the RAID storage system. The notification should be immediate (i.e., within minutes of the failure), visible, and understandable to the administrator.

Many approaches are used for failure notification. Standards such as SNMP (Simple Network Management Protocol) and

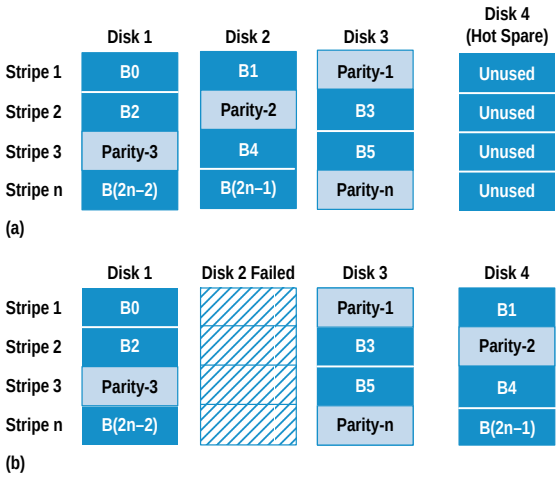

Fig. 12. Failure recovery. (a) RAID system with a hot spare before the failure. (b) After disk 2 has failed and the hot spare is brought online.

DMI (Desktop Management Interface) are emerging. However, since standards have a tendency to take a long time before they are complete enough for the industry to use, industry tends to take its own direction.

Failure Recovery. As mentioned, some failures are recoverable. One such recoverable failure is a single hard disk that is part of a redundant RAID stripe set. The storage system may have the ability to recover by itself by rebuilding on a hot spare (standby hard disk). The failed disk should be replaced to make it the new hot spare. If the system does not have a hot spare, the failed disk must be replaced before rebuilding can take place.

Fig. 12a shows the storage configuration before a failure and Fig. 12b shows what happens after a failure on disk 2 and the completion of using the hot spare (disk 4) to rebuild.

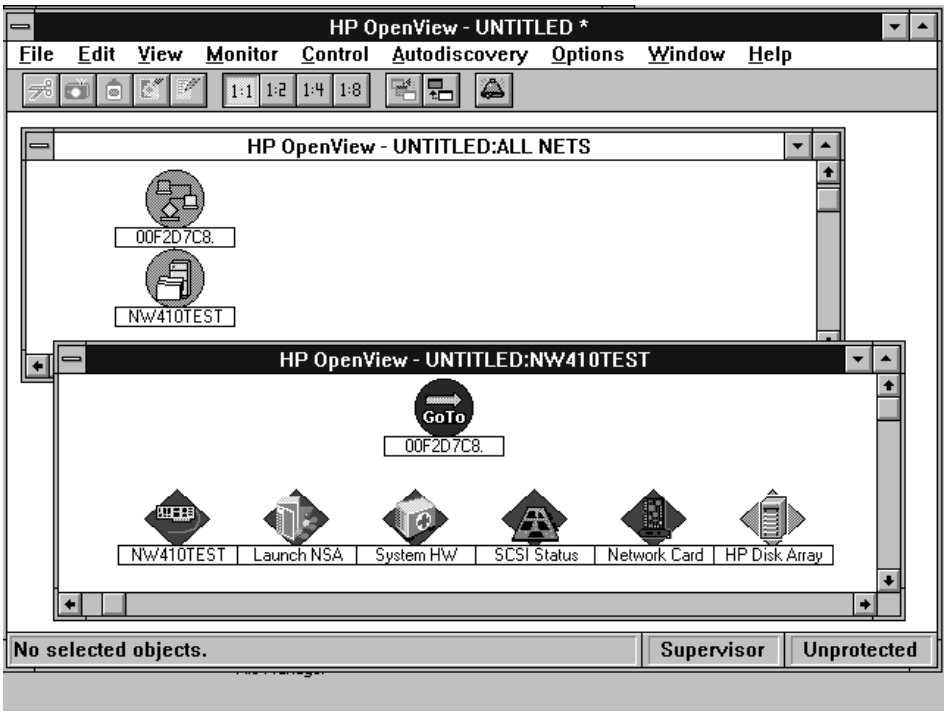

Fig. 13. HP NetServer Assistant/ OpenView screen for remote servicing.

The HP Disk Array takes failure notification one step further. In addition to offering notification of failures that have already happened, it offers a predictive failure notification so that for some failures, the user will be warned about failures that are about to happen before the system is degraded.

It is essential to get notified about failures at some accessible place. If the server is not physically close to where the system administrator is located, audible alarms and messages on the server are not very useful. The HP Disk Array offers an integrated warning mechanism using SNMP traps to an HP NetServer Assistant/OpenView remote console. This allows the system administrator to view error logs and failures from any location (see Fig. 13).

The HP Disk Array offers automatic failure detection and restoration of redundancy with the use of a hot spare disk drive. If no hot spare is used, the online version of the JetSet utility is required to start the restoration process. Upon loading, the online JetSet utility will notify the user about which disk has failed and needs to be replaced before the restoration can begin.

Conclusion

In developing products based on emerging technologies, it is the responsibility of the designers to make sure that these technologies are quickly accepted and available to the typical user. However, in the effort to bring a product to market in a timely manner, the end user is often not prioritized. The result is a product that reflects the complexity of the technology it is built on.

The overall goal for the HP Disk Array design team was to provide RAID technology to the average PC network administrator. Defining what an average PC network administrator meant and what such a user knew about RAID technology took center stage throughout the development. The resulting product is easy enough to use for a novice user, but still provides options and information for the more advanced user.

References

1. D. Patterson, G. Gibson, and R. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)". *ACM SIGMOD Conference Proceedings*, Chicago, Illinois, June 1988, pp. 109-116

2. M. Schulze, G. Gibson, R. Katz, and D. Patterson, "How Reliable Is a RAID?," *COMPCON Spring 1989*, San Francisco, California, February 1989.

3. *The RAIDBook: A Source Book for RAID Technology*, The RAID Advisory Board, Lino Lakes, Minnesota, 1993.