Search

- Home
- About Us
  - Overview
  - Management
  - News and Events
  - Alliances
- Services
  - Overview
  - Product Development
  - ReEngineering & Porting
  - Product Sustenance
  - Testing and QA
- Competencies
  - Software Product Engineering
    - Windows .Net
    - System Software
    - Mobile Applications
    - Embedded Software
  - Datacom & Networking
  - Hardware & Systems
    - BIOS & Firmware
    - Device Driver
  - Embedded & Mobile
- Success Stories
  - Overview
  - Case Studies
  - White Papers
  - Testimonials
- Clients
- Careers
- Contact Us
  - Overview
  - Call me back
  - Request Information

Product Design and Development

Methodology from Calsoft Labs brings people, advanced tools, reusable components, frameworks and processes to develop products in emergent technologies...more»

Inside Whitepaper

CONTACT US NOW

- Call Me Back
- Request Information

- Call or email us at **+1.925.249.3000** or **info@calsoftlabs.com**

PARTNER SOLUTIONS

- Adobe Macromedia Flash Player Licensee Partner
- Phoenix Trusted Alliance Partner
- HP Authorized Development Partner for APDK

Whitepapers

FEBRUARY | 2006

Del.icio.us Digg It

Redundant Array of Inexpensive Disks (RAID)

Introduction

Low cost, high performance, 3.5-inch hard-disk drives dominate the storage systems for all practical purposes. But storage and reliability requirements for the high-end enterprise systems exceed the features available with the single drives. That's were Redundant Array of Inexpensive Disks (RAID) systems are essentially useful with the highend systems.

RAID technology was developed to address the faulttolerance and performance limitations of conventional disk storage. It can offer fault tolerance and higher throughput levels than a single hard drive or group of independent hard drives. While arrays were once considered complex and relatively specialized storage solutions, today they are easy to use and essential for a broad spectrum of client/server applications.

TOP

History of RAID Systems

RAID technology was first defined by a group of computer scientists at the University of California at Berkeley in 1987. The scientists studied the possibility of using two or more disks to appear as a single device to the host system.

Although the array's performance was better than that of large, single-disk storage systems, reliability was unacceptably low. To address this, the scientists proposed redundant architectures to provide ways of achieving storage fault tolerance. In addition to defining RAID levels 1 through 5, the scientists also studied data striping -- a non-redundant array configuration that distributes files across multiple disks in an array. Often known as RAID 0, this configuration actually provides no data protection. However, it does offer maximum throughput for some data-intensive applications such as desktop digital video production.

A number of factors are responsible for the growing adoption of arrays for critical network storage. More and more organizations have created enterprise-wide networks to improve productivity and streamline information flow. While the distributed data stored on network servers provides substantial cost benefits, these savings can be quickly offset if information is frequently lost or becomes inaccessible. As today's applications create larger files, network storage needs have increased proportionately. In addition, accelerating CPU speeds have outstripped data transfer rates to storage media, creating bottlenecks in today's systems.

RAID storage solutions overcome these challenges by providing a combination of outstanding data availability, extraordinary and highly scalable performance, high capacity, and recovery with no loss of data or interruption of user access. By integrating multiple drives into a single array – which is viewed by the network operating system as a single disk drive -- organizations can create cost-effective,mini computersized solutions of up to a terabyte or more of storage.

TOP

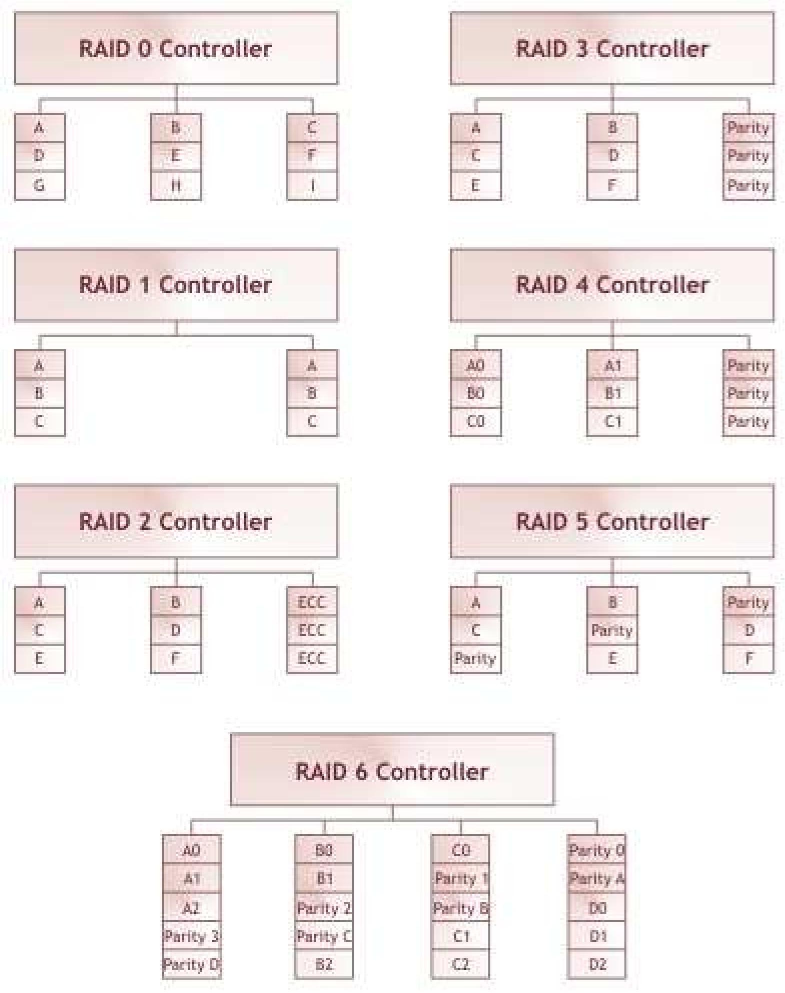
RAID Systems Overview

The dropping cost and rising capacity of high-end 3.5-in. drives plus the falling cost of RAID controllers have brought RAID to low-end servers. Motherboards with integrated SCSI and ATA RAID controllers are readily available.

RAID controllers support one or more standard RAID configurations specified as RAID 0 through RAID 5. Proprietary or experimental configurations have been given designations above RAID 6, and some configurations like RAID 10 are a combination of RAID 1 and RAID 0.

All of the configurations except RAID 0 provide some form of redundancy. RAID controllers with redundancy often support hot swappable drives, allowing the removal and replacement of a failed drive. Typically, information on a replacement drive must be rebuilt from information on the other drives before the system will be redundant and able to handle another drive failure.

To understand how a RAID system works, see the figure below. The letters A through I indicate where data is placed on a disk along with the parity and error-correction code (ECC) support. The diagrams show the minimum disk configurations for each approachwith the exception of RAID 0that can operate with only two drives.



Also known as striping, RAID 0 uses any number of drives. Data is written sequentially by sector or block across the drives. This provides high read and write performance because many operations can be performed simultaneously, even when a sequential block of information is being processed. The downside is the lack of redundancy. Striping is commonly combined with other RAID architectures to provide high performance and redundancy.

Called mirroring or duplexing, RAID 1 has 100% overhead requiring two drives to store information normally stored on only one. Although writing speed is the same as with a single drive, it's possible to have twice the read transfer rate because information is duplicated and the drives can be accessed independently. Also, there's no loss of write performance when a drive is lost and no rebuild delay as with other RAID architectures. RAID 0 is the most expensive approach when it comes to disk usage.

The RAID 2 configuration employs Hamming Code ECC to provide redundancy. It has a simple controller design , but no commercial implementations exist, partly due to the high ratio of ECC disks to data disks.

Striped data with the addition of a parity disk is used by the RAID 3 configuration. It has a high read/write transfer rate, but controller design is complex and difficult to accomplish as software RAID. Plus, the transaction rate of the system is the same as a single-disk drive.

RAID 4 utilizes independent disks with shared parity. This configuration has a very high read transaction and aggregate read rate. Furthermore, it has a low ratio of ECC disks to data disks. Unfortunately, it has a poor write transaction rate and a complex controller design. Most implementations prefer RAID 3 to RAID 4.

The RAID 5 configuration is a set of independent disks with distributed parity.Similar to RAID 3, the contents of the parity disk are spread across each disk instead of concentrating all of this information on a single disk. RAID 5 has high read performance with a medium write transaction rate speed. It has a good aggregate transfer rate, making it one of the most common approaches to RAID, even with a software RAID solution.

This configuration, however, has one of the most complex controller designs, and the individual block transfer rate is the same as a single disk. Rebuilding a replacement drive is difficult and time consuming. But some controllers implement this feature so that it takes place in the background with a degraded application throughput until rebuilding has been completed.

Essentially, RAID 6 is RAID 5 plus an extra parity drive. One parity is generated across disks while the second parity tends to the data on another disk. The two independent parity drives provide high fault tolerance that permits multiple drive failures to happen.

Controller design is very complex, even compared to RAID 5. Furthermore, RAID 6 has very poor write performance because two parity changes must be made for each write.

Many RAID controllers place disks on independent channels to allow simultaneous access to multiple drives. Likewise, large memory caches significantly improve controller performance.

TOP

RAID Levels

There are several different RAID levels or redundancy schemes, each with inherent cost, performance, and availability (fault-tolerance) characteristics designed to meet different storage needs. No individual RAID level is inherently superior to any other. Each of the five array architectures is well-suited for certain types of applications and computing environments. For client/server applications, storage systems based on RAID levels 1, 0/1, and 5 have been the most widely used.

TOP

RAID Level 0 (Non-Redundant)

A non-redundant disk array, or RAID level 0, has the lowest cost of any RAID organization because it does not employ redundancy at all. This scheme offers the best performance since it never needs to update redundant information but it does not have the best performance. Redundancy schemes that duplicate data, such as mirroring, can perform better on reads by selectively scheduling requests on the disk with the shortest expected seek and rotational delays. Without, redundancy, any single disk failure will result in data-loss. Non-redundant disk arrays are widely used in super-computing environments where performance and capacity, rather than reliability, are the primary concerns.

Sequential blocks of data are written across multiple disks in stripes. The size of a data block, which is known as the stripe width, varies with the implementation, but is always at least as large as a disk's sector size. When it comes time to read back this sequential data, all disks can be read in parallel. In a multi-tasking operating system, there is a high probability that even non-sequential disk accesses will keep all of the disks working in parallel.

- *Minimum number of drives:* 2
- *Strengths:* Highest performance
- *Weaknesses:* No data protection; One drive fails, all data is lost

TOP

RAID Level 1 (Mirrored)

The traditional solution, called mirroring or shadowing, uses twice as many disks as a non-redundant disk array. Whenever data is written to a disk the same data is also written to a redundant disk, so that there are always two copies of the information.When data is read, it can be retrieved from the disk with the shorter queuing, seek and rotational delays. If a disk fails, the other copy is used to service requests. Mirroring is frequently used in database applications where availability and transaction time are more important than storage efficiency.

- *Minimum number of drives:* 2
- *Strengths:* Very high performance; Very high data protection; Very minimal penalty on write performance.
- *Weaknesses:* High redundancy cost overhead; Because all data is duplicated, twice the storage capacity is required.

TOP

RAID Level 2 (Memory Style)

Memory systems have provided recovery from failed components with much less cost than mirroring by using Hamming codes. Hamming codes contain parity for distinct overlapping subsets of components. In one version of this scheme, four disks require three redundant disks, one less than mirroring. Since the number of redundant disks is proportional to the log of the total number of the disks on the system, storage efficiency increases as the number of data disks increases.

If a single component fails, several of the parity components will have inconsistent values, and the failed component is the one held in common by each incorrect subset. The lost information is recovered by reading the other components in a subset, including the parity component, and setting the missing bit to 0 or 1 to create proper parity value for that subset. Thus, multiple redundant disks are needed to identify the failed disk, but only one is needed to recover the lost information.

In you are unaware of parity, you can think of the redundant disk as having the sum of all data in the other disks. When a disk fails, you can subtract all the data on the good disks form the parity disk; the remaining information must be the missing information. Parity is simply this sum modulo 2.

A RAID 2 system would normally have as many data disks as the word size of the computer, typically 32. In addition, RAID 2 requires the use of extra disks to store an error-correcting code for redundancy. With 32 data disks, a RAID 2 system would require 7 additional disks for a Hamming-code ECC.

For a number of reasons, including the fact that modern disk drives contain their own internal ECC, RAID 2 is not a practical disk array scheme.

- *Minimum number of drives:* Not used in LAN
- *Strengths:* Previously used for RAM error environments correction (known as Hamming Code) and in disk drives before he use of embedded error correction.
- *Weaknesses:* No practical use; Same performance can be achieved by RAID 3 at lower cost.

TOP

RAID Level 3 (Bit-Interleaved Parity)

One can improve upon memory-style ECC disk arrays by noting that, unlike memory component failures, disk controllers can easily identify which disk has failed. Thus, one can use a single parity rather than a set of parity disks to recover lost information.

In a bit-interleaved, parity disk array, data is conceptually interleaved bit-wise over the data disks, and a single parity disk is added to tolerate any single disk failure. Each read request accesses all data disks and each write request accesses all data disks and the parity disk. Thus, only one request can be serviced at a time. Because the parity disk contains only parity and no data, the parity disk cannot participate on reads, resulting in slightly lower read performance than for redundancy schemes that distribute the parity and data over all disks. Bit-interleaved, parity disk arrays are frequently used in applications that require high bandwidth but not high I/O rates. They are also simpler to implement than RAID levels 4, 5, and 6.

Here, the parity disk is written in the same way as the parity bit in normal Random Access Memory (RAM), where it is the Exclusive Or of the 8, 16 or 32 data bits. In RAM, parity is used to detect single-bit data errors, but it cannot correct them because there is no information available to determine which bit is incorrect. With disk drives, how ever, we rely on the disk controller to report a data read error.Knowing which disk's data is missing, we can reconstruct it as the Exclusive Or (XOR) of all remaining data disks plus the parity disk.

- *Minimum number of drives:* 3
- *Strengths:* Excellent performance for large, sequential data requests.
- *Weaknesses:* Not well-suited for transaction-oriented network applications; Single parity drive does not support multiple, simultaneous read and write requests.

TOP

RAID Level 4 (Block-Interleaved Parity)

The block-interleaved, parity disk array is similar to the bit-interleaved, parity disk array except that data is interleaved across disks of arbitrary size rather than in bits.The size of these blocks is called the striping unit. Read requests smaller than the striping unit access only a single data disk. Write requests must update the requested data blocks and must also compute and update the parity block. For large writes that touch blocks on all disks, parity is easily computed by exclusive-or'ing the new data for each disk. For small write requests that update only one data disk, parity is computed by noting how the new data differs from the old data and applying those differences to the parity block. Small write requests thus require four disk I/Os: one to write the new data, two to read the old data and old parity for computing the new parity, and one to write the new parity. This is referred to as a

read-modify-write procedure. Because a block-interleaved, parity disk array has only one parity disk, which must be updated on all write operations, the parity disk can easily become a bottleneck. Because of this limitation, the block-interleaved distributed parity disk array is universally preferred over the block-interleaved, parity disk array.

- *Minimum number of drives:*3 (Not widely used)
- *Strengths:* Data striping supports multiple simultaneous read requests.
- *Weaknesses:* Write requests suffer from same single parity-drive bottleneck as RAID 3; RAID 5 offers equal data protection and better performance at same cost.

For small writes, the performance will decrease considerably. To understand the cause for this, a one-block write will be used as an example.

- A write request for one block is issued by a program.
- The RAID software determines which disks contain the data, and parity, and which block they are in.
- The disk controller reads the data block from disk.
- The disk controller reads the corresponding parity block from disk.
- The data block just read is XORed with the parity block just read.
- The data block to be written is XORed with the parity block.
- The data block and the updated parity block are both written to disk.

It can be seen from the above example that a one block write will result in two blocks being read from disk and two blocks being written to disk. If the data blocks to be read happen to be in a buffer in the RAID controller, the amount of data read from disk could drop to one, or even zero blocks, thus improving the write performance.

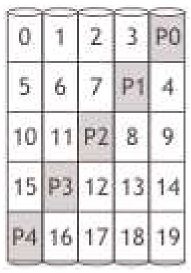
TOP

RAID Level 5 (Block-Interleaved Distributed Parity)

The block-interleaved distributed-parity disk array eliminates the parity disk bottleneck present in the block-interleaved parity disk array by distributing the parity uniformly over all of the disks. An additional, frequently overlooked advantage to distributing the parity is that it also distributes data over all of the disks rather than over all but one. This allows all disks to participate in servicing read operations in contrast to redundancy schemes with dedicated parity disks in which the parity disk cannot participate in servicing read requests. Block-interleaved distributed-parity disk array have the best small read, large write performance of any redundancy disk array. Small write requests are somewhat inefficient compared with redundancy schemes such as mirroring however, due to the need to perform read-modify-write operations to update parity. This is the major performance weakness of RAID level 5 disk arrays.

The exact method used to distribute parity in block-interleaved distributed-parity disk arrays can affect performance. Following figure illustrates left-symmetric parity distribution.

Each square corresponds to a stripe unit. Each column of squares corresponds to a disk. P0 computes the parity over stripe units 0, 1, 2 and 3; P1 computes parity over stripe units 4, 5, 6, and 7 etc.



A useful property of the left-symmetric parity distribution is that whenever you traverse the striping units sequentially, you will access each disk once before accessing any disk device. This property reduces disk conflicts when servicing large requests.

- *Minimum number of drives:* 3
- *Strengths:* Best cost/performance for transaction-oriented networks; Very high performance, very high data protection; Supports multiple simultaneous reads and writes; Can also be optimized for large, sequential requests.
- *Weaknesses:* Write performance is slower than RAID 0 or RAID 1.

TOP

RAID Level 6 (P+Q Redundancy)

Parity is a redundancy code capable of correcting any single, self-identifying failure.As large disk arrays are considered, multiple failures are possible and stronger codes are needed. Moreover, when a disk fails in parity-protected disk array, recovering the contents of the failed disk requires successfully reading the contents of all non-failed disks. The probability of encountering an uncorrectable read error during recovery can be significant. Thus, applications with more stringent reliability requirements require stronger error correcting codes.

Once such scheme, called P+Q redundancy, uses Reed-Solomon codes to protect against up to two disk failures using the bare minimum of two redundant disk arrays. The P+Q redundant disk arrays are structurally very similar to the blockinterleaved distributes-parity disk arrays and operate in much the same manner. In particular, P+Q redundant disk arrays also perform small write operations using a read-modify-write procedure, except that instead of four disk accesses per write requests, P+Q redundant disk arrays require six disk accesses due to the need to update both the `P' and `Q' information.

TOP

RAID Level 10 (Striped Mirrors)

RAID 10 is now used to mean the combination of RAID 0 (striping) and RAID 1 (mirroring). Disks are mirrored in pairs for redundancy and improved performance, and then data is striped across multiple disks for maximum performance.

RAID 10 uses more disk space to provide redundant data than RAID 5. However, it also provides a performance advantage by reading from all disks in parallel while eliminating the write penalty of RAID 5. In addition, RAID 10 gives better performance than RAID 5 while a failed drive remains un-replaced. Under RAID 5, each attempted read of the failed drive can be performed only by reading all of the other disks. On RAID 10, a failed disk can be recovered by a single read of its mirrored pair.

- *Minimum number of drives:* 4
- *Strengths:* Highest performance, highest data protection (can tolerate multiple drive failures).
- *Weaknesses:* High redundancy cost overhead; Because all data is duplicated, twice the storage capacity

is required; Requires minimum of four drives.

TOP

Compound RAID Levels

There are times when more then one type of RAID must be combined, in order to achieve the desired effect. In general, this would consist of RAID-0, combined with another RAID level. The primary reason for combining multiple RAID architectures would be to get either a very large, or a very fast, logical disk. The list below contains a few examples. It is not the limit of what can be done.

RAID-1+0

RAID Level 1+0 (also called RAID-10) is the result of RAID-0 applied to multiple RAID-1 arrays. This will create a very fast, stable array. In this array, it is possible to have multiple disk failures, without loosing any data, and with a minimum performance impact. To recover from a failed disk, it is necessary to replace the failed disk, and rebuild that disk from its mirror. For two-drive failures, the probability of survival is 66% for a 4-disk array, and approaches 100% as the number of disks in the array increases.

RAID-0+1

RAID Level 0+1 is the result of RAID-1 applied to multiple RAID-0 arrays. This will create a very fast array. If the RAID-0 controllers (hardware or software) are capable of returning an error for data requests to failed drives, then this array has all the abilities of RAID-10. If an entire RAID-0 array is disabled when one drive fails, this becomes only slightly more reliable then RAID-0.

To recover from a failed disk, it is necessary to replace the failed disk, and rebuild the entire RAID-0 array from its mirror. This requires much more disk I/O than is required to recover from a disk failure in RAID-10. It should be noted that some enterprise-level RAID controllers are capable of tracking which drives in a RAID-0 array have failed, and only rebuilding that drive. These controllers are very expensive.

For two-drive failures, the probability of survival is 33% for a 4-disk array, and approaches 50% as the number of disks in the array increases. This RAID level is significantly less reliable than RAID-1+0. This is because the structure is inherently less reliable in a multi-disk failure, combined with the longer time to reconstruct after a failure (due to a larger amount of data needing to be copied).

The longer time increases the probability of a second disk failing before the first disk has been completely rebuilt.

RAID-3+0

RAID Level 3+0 is the result of RAID-0 applied to multiple RAID-3 arrays. This will improve the performance of a RAID-3 array, and allow multiple RAID-3 arrays to be dealt with as a single logical device. RAID-3+0 has reliability similar to RAID-3, with improved performance. This type of array is most commonly found when combining multiple hardware RAID devices into a single logical device.

RAID-5+0

RAID Level 5+0 (also called RAID-53 for some unknown reason) is the result of RAID-0 applied to multiple RAID-5 arrays. This will improve the performance of a RAID-5 array, and allow multiple RAID-5 arrays to be dealt with as a single logical device. The reliability of this type of array is similar to that of a RAID-1+0 array, but it has the performance impacts of RAID-5. This type of array is most commonly found when combining multiple hardware RAID devices into a single logical device.

TOP

Types of RAID Systems

There are three primary array implementations: software-based arrays, bus-based array adapters/controllers, and subsystem-based external array controllers. As with the various RAID levels, no one implementation is clearly better than another - - although software-based arrays are rapidly losing favor as high-performance, lowcost array adapters become increasingly available. Each array solution meets different server and network requirements, depending on the number of users, applications, and storage requirements.

It is important to note that all RAID code is based on software. The difference among the solutions is where that software code is executed -- on the host CPU (softwarebased arrays) or offloaded to an on-board processor (bus-based and external array controllers).

TOP

Software based RAID

Primarily used with entry-level servers, software-based arrays rely on a standard host adapter and execute all I/O commands and mathematically intensive RAID algorithms in the host server CPU. This can slow system performance by increasing host PCI bus traffic, CPU utilization, and CPU interrupts. Some NOSs such as NetWare and Windows NT include embedded RAID software. The chief advantage of this embedded RAID software has been its lower cost compared to higher-priced RAID alternatives. However, this advantage is disappearing with the advent of lower-cost, bus-based array adapters. The major advantages are Low-Cost & it requires only a standard controller.

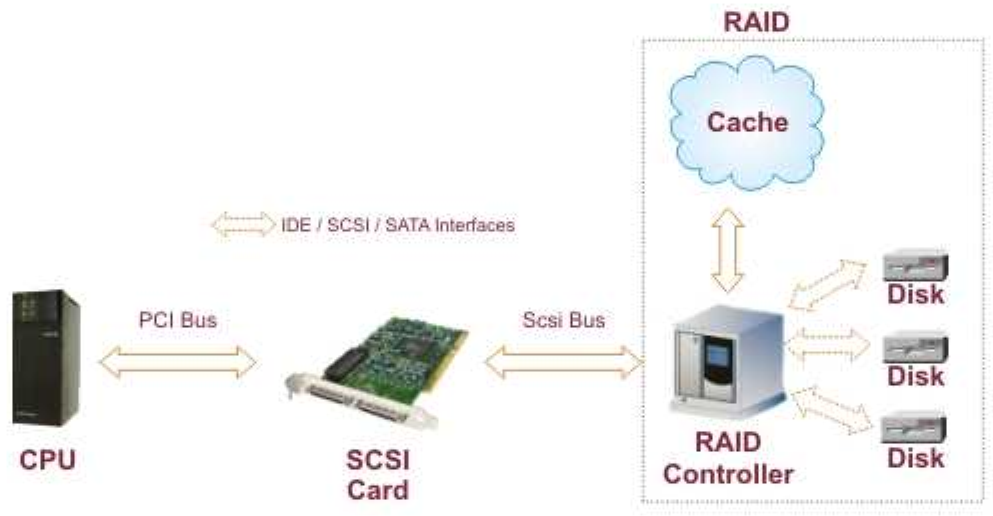
TOP

Hardware based RAID

Unlike software-based arrays, bus-based array adapters/controllers plug into a host bus slot (typically a 133 MByte (MB)/sec PCI bus) and offload some or all of the I/O commands and RAID operations to one or more secondary processors. Originally used only with mid- to high-end servers due to cost, lower-cost bus-based array adapters are now available specifically for entry-level server network applications.

In addition to offering the fault-tolerant benefits of RAID, bus-based array adapters /controllers perform connectivity functions that are similar to standard host adapters. By residing directly on a host PCI bus, they provide the highest performance of all array types. Bus-based arrays also deliver more robust faulttolerant features than embedded NOS RAID software.

Advantages are data protection & performance benefits of RAID and more robust fault-tolerant features and increased performance versus software-based RAID.



External Hardware RAID Card

Intelligent external array controllers bridge between one or more server I/O interfaces and Single / multiple-device channels. These controllers feature an onboard microprocessor, which provides high performance and handles functions such as executing RAID software code and supporting data caching.

External array controllers offer complete operating system independence, the highest availability, and the ability to scale storage to extraordinarily large capacities (up to a terabyte and beyond). These controllers are usually installed in networks of stand-alone Intel-based and UNIX-based servers as well as clustered server environments.

Advantages are OS independent and to build super high-capacity storage systems for high-end servers.

TOP

RAID Performance Modeling

As RAID architectures are being developed, their performance is analyzed using either simulation or analytical methods. Simulators are generally handcrafted, which entails code duplication and potentially unreliable systems. Equally problematic is that RAID architectures, once defined, are difficult to prove correct without extensive simulations.

Analytical Modeling

Analytical models of RAID operations generally utilize drive characteristics to model typical operations. Key factors such as Seek Time (time to move to a specified track), Rotational Latency (time to rotate disk to required sector) and Transfer Time (time to read or write to disk) are used to derive mathematical expressions of the time taken for different operations. Such expressions are usually independent of any particular hardware, and are derived from analysis of typical actions performed during a given RAID operation. Most analytical models of RAID are based on other models of the underlying disks driving the array. Using these drive characteristics, queuing models can be introduced to model the arrival of tasks at each disk in a RAID, as such tasks are handed out by the RAID controller. Combining these queued disks with arrival rates for a single disk in normal operation provides a starting point from which more complicated RAID systems can be modeled.

Simulation

The most notable simulation tool for RAID systems is the previously mentioned RAIDframe system. The system offers multiple ways to test any RAID architecture using the previously mentioned, generic DAG representation. The simulator is able to use utilization data gathered from real world systems to feed in to the simulator as well as accepting real requests from outside systems. It can also be used to generate a Linux software driver to test architecture in a real world system. One downside to this approach is that, since all RAID operations are performed in software, the controller software can easily become a bottleneck (up to 60% of total processing due to software) and hence skew the results.

Another simulation tool presented in the literature (and available for academic use) is the RAID-Sim application. It is built on top of a simulator for single disks, known as DiskSim, developed at the University of Michigan. Unlike RAIDframe, RAIDSim is labeled as very difficult to use, and has no generic interface. One other tool the authors are aware of is Sim-SAN, an environment for simulating and analyzing Storage Area Networks in which RAID usually serves as the underlying storage.

TOP

Parity & Fault Tolerance

The concept behind RAID is relatively simple. The fundamental premise is to be able to recover data on-line in the event of a disk failure by using a form of redundancy called parity. In its simplest form, parity is an addition of all the drives used in an array. Recovery from a drive failure is achieved by reading the remaining good data and checking it against parity data stored by the array. Parity is used by RAID levels 2, 3, 4, and 5. RAID 1 does not use parity because all data is completely duplicated (mirrored). RAID 0, used only to increase performance, offers no data redundancy at all.

RAID technology does not prevent drive failures. However, RAID does provide insurance against disk drive failures by enabling real-time data recovery without data loss. The fault tolerance of arrays can also be significantly enhanced by choosing the right storage enclosure. Enclosures that feature redundant, hot-swappable drives, power supplies, and fans can greatly increase storage subsystem uptime based on a number of widely accepted measures:

Mean Time to Data Loss (MTDL). The average time before the failure of an array component causes data to be lost or corrupted.

Mean Time between Data Access / Availability (MTDA). The average time before non-redundant components fail, causing data inaccessibility without loss or corruption.

Mean Time To Repair (MTTR). The average time required to bring an array storage subsystem back to full fault tolerance.

Mean Time Between Failure (MTBF). Used to measure computer component average reliability/life expectancy. MTBF is not as well-suited for measuring the reliability of array storage systems as MTDL, MTTR or MTDA because it does not account for an array's ability to recover from a drive failure. In addition, enhanced enclosure environments used with arrays to increase uptime can further limit the applicability of MTBF ratings for array solutions.

TOP

Our Implementation

In one of our driver projects related to Storage Area Networks (SAN), we have implemented the RAID-01 & RAID-10. The RAID-01 (or RAID 0+1) is a mirrored pair (RAID-1) made from two stripe sets (RAID-0); hence the name RAID 0+1, because it is created by first creating two RAID-0 sets and adding RAID-1. If you lose a drive on one side of a RAID-01 array, then lose another drive on the other side of that array before the first side is recovered, you will suffer complete data loss. It is also important to note that all drives in the surviving mirror are involved in rebuilding the entire damaged stripe set, even if only a single drive was damaged. Performance during recovery is severely degraded during recovery unless the RAID subsystem allows adjusting the priority of recovery. However, shifting the priority toward production will lengthen recovery time and increase the risk of the kind of the catastrophic data loss mentioned earlier.

RAID-10 (or RAID 1+0) is a stripe set made up from N mirrored pairs. Only the loss of both drives in the same-mirrored pair can result in any data loss and the loss of that particular drive is 1/Nth as likely as the loss of some drive on the opposite mirror in RAID-01. Recovery only involves the replacement drive and its mirror so the rest of the array performs at 100% capacity during recovery. Also since only the single drive needs recovery bandwidth requirements during recovery are lower and recovery takes far less time reducing the risk of catastrophic data loss.

The most appropriate RAID configuration for a specific file system or database table space must be determined based on data access patterns and cost versus performance tradeoffs. RAID-0 offers no increased reliability. It can, however, supply performance acceleration at no increased storage cost. RAID-1 provides the highest performance for redundant storage, because it does not require readmodify- write cycles to update data, and because multiple copies of data may be used to accelerate read-intensive applications. Unfortunately, RAID-1 requires at least double the disk capacity of RAID-0. Also, since more than two copies of the data exist; RAID-1 arrays may be constructed to endure loss of multiple disks without interruption. Parity RAID allows redundancy with less total storage cost. The readmodify- write it requires, however, will reduce total throughput in any small write operations (read-only or extremely read-intensive applications are fine). The loss of a single disk will cause read performan ce to be degraded while the system reads allother disks in the array and re-computes the missing data. Additionally, it does not support losing multiple disks, and cannot be made redundant.

TOP

- Home
- Contact us
- Privacy Policy
- Link To Us
- Site map