R.A.I.D. – A Primer

"Redundant Array of Independent Drives"

By Paul D. McDonald, MBA
Overland Park, Kansas

Abstract

This paper is for anyone looking to design hardware for a computer system that handles a large amount of data.

RAID is an acronym that stands for" Redundant Array of Independent Drives" or sometimes "Disks." It is a term for data storage schemes that divide and/or replicate data among multiple hard drives. RAID can be designed to provide increased data reliability or increased I/O performance, though one goal may compromise the other.

A number of standard schemes have evolved which are referred to as "levels." There were five RAID levels originally conceived, but many more variations have evolved, notably several nested levels and many non-standard levels (mostly proprietary).

Overview

RAID combines multiple physical hard disks into a single logical unit either by using special hardware or software. Hardware solutions often are designed to present themselves to the attached system as a single hard drive and the operating system is unaware of the technical workings. Software solutions are typically implemented in the operating system, and again would present the RAID drive as a single drive to applications.

There are three key concepts in RAID:

1. **Mirroring**, the copying of data to more than one disk;
2. **Striping**, the splitting of data across more than one disk;
3. **Error correction**, where redundant data is stored to allow problems to be detected and possibly fixed (known as "fault tolerance").

Different RAID levels use one or more of these techniques depending on system requirements. The main aims of using RAID are to improve reliability, important for protecting information that is critical to a business, for example a database of customer orders; or where speed is important, for example a system that delivers on-demand TV programs to many viewers.

The configuration affects reliability and performance in different ways. The problem with using more disks is that it is more likely that one will go wrong, but by using error checking the total system can be made more reliable by being able to survive and repair the failure.

Important Terms

RAID

Redundant Array of Independent Drives (or Disks). The term is used to describe data storage schemes among multiple hard drives.

Mirroring

A mirror in computing is a direct copy of a data set. On the Internet, a mirror site is an exact copy of another Internet site. Mirror sites are most commonly used to provide multiple sources of the same information, and are of particular value as a way of providing reliable access to large downloads. Mirroring is a type of file synchronization.

Striping

The term striping refers to the segmentation of logically sequential data, such as a single file, so that segments can be written to multiple physical devices (usually disk drives) in a round-robin fashion. This technique is useful if the processor is capable of reading or writing data faster than a single disk can supply or accept it. While data is being transferred from the first disk, the second disk can locate the next segment.

Basic Mirroring

Basic mirroring can speed up reading data as a system can read different data from both the disks, but it may be slow for writing if it insists that both disks must confirm that the data is correctly written.

Striping

Striping is often used for performance, where it allows sequences of data to be read off multiple disks at the same time.

Error Checking

Error checking typically will slow the system down as data needs to be read from several places and compared.

RAID Design—a "Compromise"

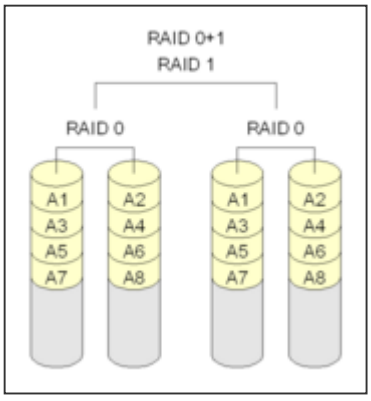
The design of RAID systems is therefore a compromise and understanding the requirements of a system is important. Modern Disk arrays typically provide the facility to select the appropriate RAID configuration.

RAID systems can be designed to keep working when there is failure - disks can be hot swapped and data recovered automatically whilst the system keeps running. Other systems have to be shut down whilst the data is recovered. RAID is often used in High availability systems, where it is important that the system keeps running as much of the time as possible.

RAID is typically used on servers but can be used on workstations. The latter is especially true in storage-intensive computers such as those used for video and audio editing.

Nested RAID levels

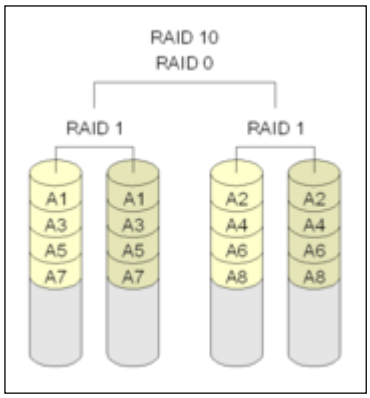
Many storage controllers allow RAID levels to be nested. That is, one RAID can use another as its basic element, instead of using physical drives. It is instructive to think of these arrays as layered on top of each other, with physical drives at the bottom.



Nested RAIDs are usually signified by joining the numbers indicating the RAID levels into a single number, sometimes with a '+' in between. For example, RAID 1+0 (or RAID 10) conceptually consists of multiple level 1 arrays stored on physical drives with a level 0 array on top, striped over the level 1 arrays. In the case of RAID 0+1, it is most often called RAID 0+1 as opposed to RAID 01 to avoid confusion with RAID 1. However, when the top array is a RAID 0 (such as in RAID 10 and RAID 50), most vendors choose to omit the '+', though RAID 5+0 is more informative.

Here are some common nested RAID levels:

- **RAID 0+1**: Striped Set + Mirrored Set (4 disk minimum; Even number of disks) provides fault tolerance and improved performance but increases complexity. The key difference from RAID 1+0 is that RAID 0+1 creates a second striped set to mirror a primary striped set. The array continues to operate with one or more drives failed in the same mirror set, but if two or more drives fail on different sides of the mirroring, the data on the RAID system is lost.
- **RAID 1+0** *(sometimes called RAID 10)*: Mirrored Set + Striped Set (4 disk minimum; Even number of disks) provides fault tolerance and improved performance but increases complexity. The key difference from RAID 0+1 is that RAID 1+0 creates a striped set from a series of mirrored drives. The array can sustain multiple drive losses as long as no two drives lost comprise a single pair of one mirror.



- **RAID 5+0**: A stripe across distributed parity RAID systems
- **RAID 5+1**: A mirror striped set with distributed parity (some manufacturers label this as RAID 53)

RAID implementations

The distribution of data across multiple drives can be managed either by dedicated hardware or by software. Additionally, there are hybrid RAIDs that are partially software and hardware-based solutions.

Software RAID

Software implementations are provided by most operating systems. A software layer sits above the (generally block based) disk device drivers and provides an abstraction layer between the logical drives (RAID arrays) and physical drives. Software RAID is typically limited to RAID 0 (striping across multiple drives for increased space and performance), RAID 1 (mirroring two drives) and RAID 5 (data striping with parity).

In a multi-threaded operating system (such as Linux, FreeBSD, Mac OS X, Windows NT/2000/XP/Vista and Novell NetWare) the operating system can perform overlapped I/O, allowing multiple read or write requests to be initiated without waiting for completion on each request. This is the capability that makes RAID 0+1 possible in an operating system. However, most operating systems do not support RAID 0+1 striping or mirroring with parity, due to the substantial processing demands of calculating parity.

Software implementations require some very small amount of processing time, which is provided by the main CPU in the host system. Since SCSI, PATA, and SATA drives all support asynchronous read/write, any multi-threaded operating system can support non-parity RAID on multiple hard drives with only a one percent increase in CPU overhead[citation needed].

Software implementations can exceed the performance levels of hardware-based RAID due to the high-performance of modern CPUs[citation needed]. Since the software must run on a host server attached to storage, the processor (as mentioned above) on that host must dedicate processing time to run the RAID software. Like hardware-based RAID, if the server experiences a hardware failure, the attached storage could be inaccessible for a period of time.

Software implementations can allow RAID arrays to be created from partitions rather than entire physical drives.

Hardware RAID

A hardware implementation of RAID requires at a minimum a special-purpose RAID controller. On a desktop system, this may be a PCI expansion card, or might be a capability built in to the motherboard. In industrial applications the controller and drives are provided as a stand alone enclosure. The drives may be IDE/ATA, SATA, SCSI, SSA, Fibre Channel, or any combination thereof. The using system can be directly attached to the controller or, more commonly, connected via a SAN. The controller hardware handles the management of the drives, and performs any parity calculations required by the chosen RAID level.

Most hardware implementations provide a non-volatile read/write cache which, depending on the I/O workload, will improve performance. Cached RAID controllers are most commonly used in industrial applications.

Hardware implementations provide guaranteed performance, add no overhead to the local CPU complex and can support many operating systems, as the controller simply presents a logical disk to the operating system.

Hardware implementations also typically support hot swapping, allowing failed drives to be replaced while the system is running.

Hybrid RAID

Hybrid RAID implementations have become very popular with the introduction of inexpensive RAID controllers, implemented using a standard disk controller and then implementing the RAID in the controllers BIOS extension (for early boot-up/real mode operation) and the operating system driver (for after the system switches to protected mode). Since these controllers actually do all calculations typically proprietary to a given RAID controller manufacturer and typically cannot span multiple controllers. The only advantages over software RAID are that the BIOS can boot from them, and the tighter integration with the device driver may offer better error handling.

Both hardware and software implementations may support the use of hot spare drives, a pre-installed drive which is used to immediately (and almost always automatically) replace a drive that has failed. This reduces the mean time to repair period during which a second drive failure in the same RAID redundancy group can result in loss of data. It also prevents data loss when multiple drives fail in a short period of time, as is common when all drives in an array have undergone very similar use patterns, and experience wear-out failures.

Reliability of RAID configurations

Failure rate

The mean time to failure (MTTF) or the mean time between failures (MTBF) of a given RAID may be lower or higher than those of its constituent hard drives, depending on what type of RAID is employed.

Mean time to data loss (MTTDL)

In this context, the average time before a loss of data in a given array.

Mean time to recovery (MTTR)

In arrays that include redundancy for reliability, this is the time following a failure to restore an array to its normal failure-tolerant mode of operation. This includes time to replace a failed disk mechanism as well as time to re-build the array (i.e. to replicate data for redundancy).

Unrecoverable bit error rate (UBE)

This is the rate at which a disk drive will be unable to recover data after application of cyclic redundancy check (CRC) codes and multiple retries. This failure will present as a sector read failure. Some RAID implementations protect against this failure mode by remapping the bad sector, using the redundant data to retrieve a good copy of the data, and rewriting that good data to the newly mapped replacement sector. The UBE rate is typically specified at 1 bit in 1015 for enterprise class disk drives (SCSI, FC, SAS) , and 1 bit in 1014 for desktop class disk drives (IDE, ATA, SATA). Increasing disk capacities and large RAID 5 redundancy groups have led to an increasing inability to successfully rebuild a RAID group after a disk failure because an unrecoverable sector is found on the remaining drives. Double protection schemes such as RAID 6 are attempting to address this issue, but suffer from a very high write penalty.

Atomic Write Failure

Also known by various terms such as torn writes, torn pages, incomplete writes, interrupted writes, non-transactional, etc. This is a little understood and rarely mentioned failure mode for redundant storage systems that do not utilize transactional features.

Database researcher Jim Gray wrote "Update in Place is a Poison Apple" during the early days of relational database commercialization. However, this warning largely went unheeded and fell by the wayside upon the advent of RAID, which many software engineers mistook as solving all data storage integrity and reliability problems.

Many software programs update a storage object "in-place"; that is, they write a new version of the object on to the same disk addresses as the old version of the object. While the software may also log some delta information elsewhere, it expects the storage to present "atomic write semantics," meaning that the write of the data either occurred in its entirety or did not occur at all.

However, very few storage systems provide support for atomic writes, and even fewer specify their rate of failure in providing this semantic. Note that during the act of writing an object, a RAID storage device will usually be writing all redundant copies of the object in parallel, although overlapped or staggered writes are more common when a single RAID processor is responsible for multiple drives. Hence an error that occurs during the process of writing may leave the redundant copies in different states, and furthermore may leave the copies in neither the old nor the new state.

The little known failure mode is that delta logging relies on the original data being either in the old or the new state so as to enable backing out the logical change, yet few storage systems provide an atomic write semantic on a RAID disk.

Since transactional support is not universally present in hardware RAID, many operating systems include transactional support to protect against data loss during an interrupted write. Novell Netware, starting with version 3.x, included a transaction tracking system. Microsoft introduced transaction tracking via the journaling feature in NTFS.

To mitigate this problem, some high-end RAID cards use a battery-backed write cache. If an "atomic" write only partially completes because of power failure, the controller flushed the unwritten data to disk when the power is restored. Some provide the capability of testing the battery periodically (however, this leaves the system without a fully charged battery for several hours). This solution still has potential failure cases: the battery may have worn out, the power may be off for too long, the disks could be moved to another controller, the controller itself could fail in the middle of a write, etc. However, on a well-maintained machine it probably prevents corruption in most incidents.

About the Author

Paul D. McDonald, MBA, is a SAS Certified Professional. Visit his website at http://www.spikeware.net/.

Special Thanks to

All Wikipedia contributors who provided the main source of information and cool diagrams for this article!

Wikipedia contributors, "RAID," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=RAID&oldid=174744556