

Help us provide free content to the world by **donating today!**

RAID

From Wikipedia, the free encyclopedia

RAID — which stands for **Redundant Array of Inexpensive Disks** ^[1], or alternatively **Redundant Array of Independent Disks** (a less specific name, and thus now the generally accepted one ^[2]) — is a technology that employs the simultaneous use of two or more hard disk drives to achieve greater levels of performance, reliability, and/or larger data volume sizes.

The phrase "RAID" is an umbrella term for computer data storage schemes that can divide and replicate data among multiple hard disk drives. RAID's various designs all involve two key design goals: increased data reliability and increased input/output performance. When several physical disks are set up to use RAID technology, they are said to be *in* a *RAID* array. This array distributes data across several disks, but the array is seen by the computer user and operating system as one single disk. RAID can be set up to serve several different purposes.

Contents

- 1 Purpose and basics
- 2 Principles
- 3 Standard levels
- 4 Nested levels
- 5 Non-standard levels
- 6 Implementations
 - 6.1 Operating system based ("software RAID")
 - 6.2 Hardware-based
 - 6.3 Firmware/driver based RAID
 - 6.4 Network-attached storage
 - 6.5 Hot spares
- 7 Reliability terms
- 8 Problems with RAID
 - 8.1 Correlated failures
 - 8.2 Atomicity
 - 8.3 Unrecoverable data
 - 8.4 Write cache reliability
 - 8.5 Equipment compatibility
- 9 History
- 10 See also
- 11 References
- 12 Further reading
- 13 External links

Purpose and basics

Some arrays are "redundant" in a way that writes extra data derived from the original data across the array organized so that the failure of one (sometimes more) disks in the array will not result in loss of data; the bad disk is replaced by a new one, and the data on it reconstructed from the remaining data and the extra data. A redundant array allows less data to be stored. For instance, a 2-disk RAID 1 array loses half of its capacity, and a RAID 5 array with several disks loses the capacity of one disk.

Other RAID arrays are arranged so that they are faster to write to and read from than a single disk.

There are various combinations of these approaches giving different trade offs of protection against data loss, capacity, and speed. RAID levels 0, 1, and 5 are the most commonly found, and cover most requirements.

- RAID 0 (striped disks) distributes data across several disks in a way that gives improved speed and full capacity, but all data on all disks will be lost if any one disk fails.
- RAID 1 (mirrored disks) could be described as a backup solution, using two (possibly more) disks that each store the same data so that data is not lost as long as one disk survives. Total capacity of the array is just the capacity of a single disk. The failure of one drive, in the event of a hardware or software malfunction, does not increase the chance of a failure nor decrease the reliability of the remaining drives (second, third, etc).
- RAID 5 (striped disks with parity) combines three or more disks in a way that protects data against loss of any one disk; the storage capacity of the array is reduced by one disk.
- RAID 6 (less common) can recover from the loss of two disks.
- RAID 7 (experimental) single disk array in development by the Michaelis Project.
- RAID 10 (or 1+0) uses both striping and mirroring.

RAID involves significant computation when reading and writing information. With true RAID hardware the controller does all of this computation work. In other cases the operating system or simpler and less expensive controllers require the host computer's processor to do the computing, which reduces the computer's performance on processor-intensive tasks (see "Software RAID" and "Fake RAID" below). Simpler RAID controllers may provide only levels 0 and 1, which require less processing.

RAID systems with redundancy continue working without interruption when one, or sometimes more, disks of the array fail, although they are vulnerable to further failures. When the bad disk is replaced by a new one the array is rebuilt while the system continues to operate normally. Some systems have to be shut down when removing or adding a drive; others support hot swapping, allowing drives to be replaced without powering down. RAID with hot-swap drives 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 not a good alternative to backing up data. Data may become damaged or destroyed without harm to the drive(s) on which it is stored. For example, part of the data may be overwritten by a system malfunction; a file may be damaged or deleted by user error or

malice and not noticed for days or weeks; and of course the entire array is at risk of catastrophes such as theft, flood, and fire.

Principles

RAID combines two or more physical hard disks into a single logical unit by using either 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: mirroring, the copying of data to more than one disk; striping, the splitting of data across more than one disk; and 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 the 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 to improve speed, for example a system that delivers video on demand TV programs to many viewers.

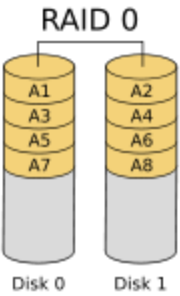
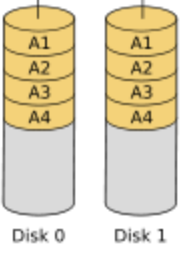

The configuration affects reliability and performance in different ways. The problem with using more disks is that it is more likely that one will go wrong, but by using error checking the total system can be made more reliable by being able to survive and repair the failure. Basic mirroring can speed up reading data as a system can read different data from both the disks, but it may be slow for writing if the configuration requires that both disks must confirm that the data is correctly written. Striping is often used for performance, where it allows sequences of data to be read from multiple disks at the same time. Error checking typically will slow the system down as data needs to be read from several places and compared. 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. PC Format Magazine claims that "in all our real-world tests, the difference between the single drive performance and the dual-drive RAID 0 striped setup was virtually non-existent. And in fact, the single drive was ever-so-slightly faster than the other setups, including the RAID 5 system that we'd hoped would offer the perfect combination of performance and data redundancy"^[3].

Standard levels

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).

Following is a brief summary of the most commonly used RAID levels.^[4]

Level	Description	Minimum # of disks	Image

RAID 0	<p>Striped set without parity/[Non-Redundant Array]. Provides improved performance and additional storage but no fault tolerance. Any disk failure destroys the array, which becomes more likely with more disks in the array. A single disk failure destroys the entire array because when data is written to a RAID 0 drive, the data is broken into fragments. The number of fragments is dictated by the number of disks in the array. The fragments are written to their respective disks simultaneously on the same sector. This allows smaller sections of the entire chunk of data to be read off the drive in parallel, giving this type of arrangement huge bandwidth. RAID 0 does not implement error checking so any error is unrecoverable. More disks in the array means higher bandwidth, but greater risk of data loss.</p>	2	 <p style="text-align: center;">RAID 0</p>
RAID 1	<p>Mirrored set without parity. Provides fault tolerance from disk errors and failure of all but one of the drives. Increased read performance occurs when using a multi-threaded operating system that supports split seeks, very small performance reduction when writing. Array continues to operate so long as at least one drive is functioning. Using RAID 1 with a separate controller for each disk is sometimes called <i>duplexing</i>.</p>	2	 <p style="text-align: center;">RAID 1</p>
RAID 2	<p>Redundancy through Hamming code. Disks are synchronised and striped in very small stripes, often in single bytes/words. Hamming codes error correction is calculated across corresponding bits on disks, and is stored on multiple parity disks.</p>	3	
RAID 3	<p>Striped set with dedicated parity/Bit interleaved parity. This mechanism provides an improved performance and fault tolerance similar to RAID 5, but with a dedicated parity disk rather than rotated parity stripes. The single parity disk is a bottle-neck for writing since every write requires updating the parity data. One minor benefit is the dedicated parity disk allows the parity drive to fail and operation will continue without parity or performance</p>	3	 <p style="text-align: center;">RAID 3</p>

	penalty.		
RAID 4	Block level parity. Identical to RAID 3, but does block-level striping instead of byte-level striping. In this setup, files can be distributed between multiple disks. Each disk operates independently which allows I/O requests to be performed in parallel, though data transfer speeds can suffer due to the type of parity. The error detection is achieved through dedicated parity and is stored in a separate, single disk unit.	3	
RAID 5	Striped set with distributed parity. Distributed parity requires all drives but one to be present to operate; drive failure requires replacement, but the array is not destroyed by a single drive failure. Upon drive failure, any subsequent reads can be calculated from the distributed parity such that the drive failure is masked from the end user. The array will have data loss in the event of a second drive failure and is vulnerable until the data that was on the failed drive is rebuilt onto a replacement drive.	3	
RAID 6	Striped set with dual distributed Parity. Provides fault tolerance from two drive failures; array continues to operate with up to two failed drives. This makes larger RAID groups more practical, especially for high availability systems. This becomes increasingly important because large-capacity drives lengthen the time needed to recover from the failure of a single drive. Single parity RAID levels are vulnerable to data loss until the failed drive is rebuilt: the larger the drive, the longer the rebuild will take. Dual parity gives time to rebuild the array without the data being at risk if one drive, but no more, fails before the rebuild is complete.	4	

Nested levels

Many storage controllers allow RAID levels to be nested: the elements of a *RAID* may be either individual disks or RAID's themselves. Nesting more than two deep is unusual.

As there is no basic RAID level numbered larger than 9, nested RAID's are usually

unambiguously described by concatenating the numbers indicating the RAID levels, sometimes with a "+" in between. For example, RAID 10 (or RAID 1+0) consists of several level 1 arrays of physical drives, each of which is one of the "drives" of a level 0 array striped over the level 1 arrays. It is not called RAID 01, to avoid confusion with RAID 1, or indeed, RAID 01. When the top array is a RAID 0 (such as in RAID 10 and RAID 50) most vendors omit the "+", though RAID 5+0 is clearer.

- RAID 0+1: striped sets in a mirrored set (minimum four disks; 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 drives fail on both sides of the mirror the data on the RAID system is lost.
- RAID 1+0: mirrored sets in a striped set (minimum four disks; 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. In a failed disk situation, RAID 1+0 performs better because all the remaining disks continue to be used. The array can sustain multiple drive losses so long as no mirror loses all its drives.
- RAID 5+0: stripe across distributed parity RAID systems.
- RAID 5+1: mirror striped set with distributed parity (some manufacturers label this as RAID 53).

Non-standard levels

Many configurations other than the basic numbered RAID levels are possible, and many companies, organizations, and groups have created their own non-standard configurations, in many cases designed to meet the specialised needs of a small niche group. Most of these non-standard RAID levels are proprietary.

Some of the more prominent modifications are:

- Storage Computer Corporation uses *RAID 7*, which adds caching to RAID 3 and RAID 4 to improve I/O performance.
- EMC Corporation offered *RAID 5* as an alternative to RAID 5 on their Symmetrix systems (which is no longer supported on the latest releases of Enginuity, the Symmetrix's operating system).
- The ZFS filesystem, available in Solaris, OpenSolaris, FreeBSD and Mac OS X, offers *RAID-Z*, which solves RAID 5's write hole problem.
- NetApp's Data ONTAP uses RAID-DP (also referred to as "double", "dual" or "diagonal" parity), which is a form of RAID 6, but unlike many RAID 6 implementations, does not use distributed parity as in RAID 5. Instead, two unique parity disks with separate parity calculations are used. This is a modification of RAID 4 with an extra parity disk.
- Accusys *Triple Parity* (RAID TP) implements three independent parities by extending RAID 6 algorithms on its FC-SATA and SCSI-SATA RAID controllers to tolerate three-disk failure.
- Linux MD RAID10 (RAID10) implements a general RAID driver that defaults to a standard RAID 1+0 with 4 drives, but can have any number of drives, including an odd

number. MD RAID10 can run striped and mirrored with only 2 drives with the f2 layout (mirroring with striped reads, normal Linux software RAID 1 does not stripe reads, but can read in parallel) ^[5].

- Infrant (Now part of Netgear) *X-RAID* offers dynamic expansion of a RAID5 volume without having to backup/restore the existing content. Just add larger drives one at a time, let it resync, then add the next drive until all drives are installed. The resulting volume capacity is increased without user downtime.
- BeyondRAID created by Data Robotics and used in the Drobo series of products, implements both mirroring and striping simultaneously or individually dependent on disk and data context. BeyondRAID is more automated and easier to use than many standard RAID levels. It also offers instant expandability without reconfiguration, the ability to mix and match drive sizes and the ability to reorder disks. It is a block-level system and thus file system agnostic although today support is limited to NTFS, HFS+, FAT32, and EXT3. It also utilizes thin provisioning to allow for single volumes up to 16TB depending on the host operating system support.

Implementations

The distribution of data across multiple drives can be managed either by dedicated hardware or by software. When done in software the software may be part of the operating system or it may be part of the firmware and drivers supplied with the card.

Operating system based ("software RAID")

Software implementations are now provided by many operating systems. A software layer sits above the (generally block-based) disk device drivers and provides an abstraction layer between the logical drives (RAIDs) and physical drives. Most common levels are RAID 0 (striping across multiple drives for increased space and performance) and RAID 1 (mirroring two drives), followed by RAID 1+0, RAID 0+1, and RAID 5 (data striping with parity) are supported.

Microsoft's server operating systems support 3 RAID levels; RAID 0, RAID 1, and RAID 5. Some of the Microsoft desktop operating systems support RAID such as Windows XP Professional which supports RAID level 0 in addition to spanning multiple disks but only if using dynamic disks and volumes.

Apple's Mac OS X Server supports RAID 0, RAID 1, and RAID 1+0. ^[6]

FreeBSD supports RAID 0, RAID 1, RAID 3, and RAID 5.

NetBSD supports RAID 0, RAID 1, RAID 4 and RAID 5 (and any nested combination of those like 1+0) via its software implementation, named raidframe.

OpenSolaris and Solaris 10 supports RAID 0, RAID 1, RAID 5, and RAID 6 (and any nested combination of those like 1+0) via ZFS with limited support on the system hard drive (RAID 1 only). Through SVM, Solaris 10 and earlier versions support RAID 0, RAID 1, and RAID 5 on both system and data drives

The software must run on a host server attached to storage, and server's processor must dedicate processing time to run the RAID software. This is negligible for RAID 0 and RAID 1, but may be significant for more complex parity-based schemes. Furthermore all the busses

between the processor and the disk controller must carry the extra data required by RAID which may cause congestion.

Another concern with operating system-based RAID is the boot process, it can be difficult or impossible to set up the boot process such that it can failover to another drive if the usual boot drive fails and therefore such systems can require manual intervention to make the machine bootable again after a failure. Finally operating system-based RAID usually uses formats specific to the operating system in question so it cannot generally be used for partitions that are shared between operating systems as part of a multi-boot setup.

Most operating system-based implementations allow RAIDs to be created from partitions rather than entire physical drives. For instance, an administrator could divide an odd number of disks into two partitions per disk, mirror partitions across disks and stripe a volume across the mirrored partitions to emulate a RAID 1E configuration. Using partitions in this way also allows mixing reliability levels on the same set of disks. For example, one could have a very robust RAID-1 partition for important files, and a less robust RAID-5 or RAID-0 partition for less important data. (Some controllers offer similar features, e.g. Intel Matrix RAID.) Using two partitions on the same drive in the same RAID is, however, dangerous. If, for example, a RAID 5 array is composed of four drives 250 + 250 + 250 + 500 GB, with the 500-GB drive split into two 250 GB partitions, a failure of this drive will remove two partitions from the array, causing all of the data held on it to be lost.

Hardware-based

Hardware RAID controllers use different, proprietary disk layouts, so it is not usually possible to span controllers from different manufacturers. They do not require processor resources, the BIOS can boot from them, and tighter integration with the device driver may offer better error handling.

A hardware implementation of RAID requires at least a special-purpose RAID controller. On a desktop system this may be a PCI expansion card, PCI-e expansion card or built into the motherboard. Controllers supporting most types of drive may be used - IDE/ATA, SATA, SCSI, SSA, Fibre Channel, sometimes even a combination. The controller and disks may be in a stand-alone disk enclosure, rather than inside a computer. The enclosure may be directly attached to a computer, or connected via 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 read/write cache, which, depending on the I/O workload, will improve performance. In most systems the write cache is non-volatile (i.e. battery-protected), so pending writes are not lost on a power failure.

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.

Firmware/driver based RAID

Operating system-based RAID cannot easily be used to protect the boot process and is generally impractical on desktop versions of Windows (as described above). Hardware RAID

controllers are expensive. To fill this gap, cheap "RAID controllers" were introduced that do not contain a RAID controller chip, but simply a standard disk controller chip with special firmware and drivers. During early stage bootup the RAID is implemented by the firmware; when a protected-mode operating system kernel such as Linux or a modern version of Microsoft Windows is loaded the drivers take over.

These controllers are described by their manufacturers as RAID controllers, and it is rarely made clear to purchasers that the burden of RAID processing is borne by the host computer's central processing unit, not the RAID controller itself, thus introducing the aforementioned CPU overhead. Before their introduction, a "RAID controller" implied that the controller did the processing, and the new type has become known in technically knowledgeable circles as "fake RAID" even though the RAID itself is implemented correctly.

Network-attached storage

While not directly associated with RAID, Network-attached storage (NAS) is an enclosure containing disk drives and the equipment necessary to make them available over a computer network, usually Ethernet. The enclosure is basically a dedicated computer in its own right, designed to operate over the network without screen or keyboard. It contains one or more disk drives; multiple drives may be configured as a RAID.

Hot spares

Both hardware and software RAIDs with redundancy may support the use of *hot spare* drives, a drive physically installed in the array which is inactive until an active drive fails, when the system automatically replaces the failed drive with the spare, rebuilding the array with the spare drive included. This reduces the mean time to repair (MTTR), though it doesn't eliminate it completely. A second drive failure in the same RAID redundancy group before the array is fully rebuilt will result in loss of the data; rebuilding can take several hours, especially on busy systems.

Rapid replacement of failed drives is important as the drives of an array will all have had the same amount of use, and may tend to fail at about the same time rather than randomly. RAID 6 without a spare uses the same number of drives as RAID 5 with a hot spare and protects data against simultaneous failure of up to two drives, but requires a more advanced RAID controller.

Reliability terms

Failure rate

The mean time to failure (MTTF) or the mean time between failure (MTBF) of a given RAID is the same as those of its constituent hard drives, regardless of 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.^[7] Mean time to data loss of a given RAID may be higher or lower than that of its constituent hard drives, depending upon what type of RAID is employed.

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.

Write cache reliability

Some RAID systems use RAM write cache to increase performance. A power failure can result in data loss unless this sort of disk buffer is supplemented with a battery to ensure that the buffer has enough time to write from RAM back to disk.

Atomic write failure

Also known by various terms such as torn writes, torn pages, incomplete writes, interrupted writes, non-transactional, etc.

Problems with RAID

Correlated failures

The theory behind the error correction in RAID assumes that failures of drives are independent. Given these assumptions it is possible to calculate how often they can fail and to arrange the array to make data loss arbitrarily improbable.

In practice, the drives are often the same ages, with similar wear. Since many drive failures are due to mechanical issues which are more likely on older drives, this violates those assumptions and failures are in fact statistically correlated. In practice then, the chances of a second failure before the first has been recovered is not nearly as unlikely as might be supposed, and data loss can in practice occur at significant rates.^[8]

Atomicity

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"^[9] 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.

While the battery-backed write cache may partially solve the problem, it is applicable only to a power failure scenario.

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. NetApp WAFL file system solves it by never updating the data in place, as does ZFS.

Unrecoverable data

This can 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 (Unrecoverable Bit Error) rate is typically specified at 1 bit in 10^{15} for enterprise class disk drives (SCSI, FC, SAS), and 1 bit in 10^{14} for desktop class disk drives (IDE/ATA/PATA, 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.

Write cache reliability

The disk system can acknowledge the write operation as soon as the data is in the cache, not waiting for the data to be physically written. However, any power outage can then mean a significant data loss of any data queued in such cache.

Often a battery is protecting the write cache, mostly solving the problem. If a write fails because of power failure, the controller may complete the pending writes as soon as restarted. 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. Some disk systems provide the capability of testing the battery periodically, however this leaves the system without a fully charged battery for several hours.

An additional concern about write cache reliability exists, and that is that a lot of them are write-back cache; a caching system which reports the data as written as soon as it is written to cache, as opposed to the non-volatile medium^[10]. The safer cache technique is write-through, which reports transactions as written when they are written to the non-volatile medium.

Equipment compatibility

The disk formats on different RAID controllers are not necessarily compatible, so that it may not be possible to read a RAID on different hardware. Consequently a non-disk hardware failure may require using identical hardware, or a data backup, to recover the data. Software RAID however, such as implemented in the Linux kernel, alleviates this concern, as the setup is not hardware dependent, but runs on ordinary disk controllers. Additionally,

Software RAID1 disks can be read like normal disks, so no RAID system is required to retrieve the data.

History

Norman Ken Ouchi at IBM was awarded a 1978 U.S. patent 4,092,732^[11] titled "System for recovering data stored in failed memory unit." The claims for this patent describe what would later be termed RAID 5 with full stripe writes. This 1978 patent also mentions that disk mirroring or duplexing (what would later be termed RAID 1) and protection with dedicated parity (that would later be termed RAID 4) were prior art at that time.

The term RAID was first defined by David A. Patterson, Garth A. Gibson and Randy Katz at the University of California, Berkeley in 1987. They studied the possibility of using two or more drives to appear as a single device to the host system and published a paper: "*A Case for Redundant Arrays of Inexpensive Disks (RAID)*" in June 1988 at the SIGMOD conference.^[12]

This specification suggested a number of prototype *RAID levels*, or combinations of drives. Each had theoretical advantages and disadvantages. Over the years, different implementations of the RAID concept have appeared. Most differ substantially from the original idealized RAID levels, but the numbered names have remained. This can be confusing, since one implementation of RAID 5, for example, can differ substantially from another. RAID 3 and RAID 4 are often confused and even used interchangeably.

Their paper formally defined RAID levels 1 through 5 in sections 7 to 11:

- First level RAID: mirrored drives
- Second level RAID: Hamming code for error correction
- Third level RAID: single check disk per group
- Fourth level RAID: independent reads and writes
- Fifth level RAID: spread data/parity over all drives (no single check disk)

See also

- Disk Data Format (DDF)
- Disk array controller
- Redundant Array of Inexpensive Nodes
- Vinum volume manager

References

- ↑ Patterson, David; Garth A. Gibson, Randy Katz (1988). "A Case for Redundant Arrays of Inexpensive Disks (RAID)". *SIGMOD Conference*: pp 109–116. retrieved 2006-12-31
- ↑ "Because the term inexpensive is relative and can be misleading, the proper meaning of the acronym is now generally accepted as Redunant Array of Independent Disks" - Chapter 7, page 302, The Essentials of Computer Organization and Architecture; Linda Null and Julia Lobur; ISBN 0-76370444-X
- ↑ Pickstock, Simon (June 2008). "The truth about RAID" (in English) (Magazine). *PC Format Magazine* (214): 35. United Kingdom: Future Publishing. "In all our real-world tests, the difference between the single drive performance and the dual-drive RAID 0 striped setup was virtually non-existent. And in fact, the single drive was ever-so-slightly faster than the other setups, including the RAID 5 system that we'd hoped would offer the perfect

- combination of performance and data redundancy.”
4. ^ SNIA Dictionary (<http://www.snia.org/education/dictionary>)
 5. ^ Main Page - Linux-raid (<http://linux-raid.osdl.org/>)
 6. ^ "Apple Mac OS X Server File Systems (<http://www.apple.com/server/macosx/technology/filesystem.html>) ". Retrieved on 2008-04-23.
 7. ^ Jim Gray and Catharine van Ingen, "Empirical Measurements of Disk Failure Rates and Error Rates" (http://research.microsoft.com/research/pubs/view.aspx?msr_tr_id=MSR-TR-2005-166) , MSTR-2005-166, December 2005
 8. ^ Disk Failures in the Real World: What Does an MTTF of 1,000,000 Hours Mean to You? (<http://www.usenix.org/events/fast07/tech/schroeder.html>) Bianca Schroeder and Garth A. Gibson
 9. ^ Jim Gray: The Transaction Concept: Virtues and Limitations (<http://www.informatik.uni-trier.de/~ley/db/conf/vldb/Gray81.html>) (Invited Paper) VLDB 1981 (<http://www.informatik.uni-trier.de/~ley/db/conf/vldb/vldb81.html#Gray81>) : 144-154
 10. ^ "Definition of write-back cache at SNIA dictionary (<http://www.snia.org/education/dictionary/w/>) ".
 11. ^ U.S. Patent 4,092,732 (<http://patft.uspto.gov/netacgi/nph-Parser?patentnumber=4092732>)
 12. ^ Patterson, David; Garth A. Gibson, Randy Katz (1988). "A Case for Redundant Arrays of Inexpensive Disks (RAID)". *SIGMOD Conference*: pp 109–116. retrieved 2006-12-31

Further reading

- Charles M. Kozierok (2001-04-17). "Redundant Arrays of Inexpensive Disks (<http://pcguide.com./ref/hdd/perf/raid/index.htm>) ". *The PC Guide*. Pair Networks.

External links

- Different Types of RAID (<http://www.howtofriends.com/raid/>)
- "RAID 6 Essentials" tutorial (<http://www.brainshark.com/winchestersystemsinc1/vu?pi=707421860>)
- RAID (<http://www.dmoz.org/Computers/Hardware/Storage/Subsystems/RAID/>) at the Open Directory Project
- Introduction to RAID (<http://www.ibizdir.com/business-research/web-hosting/61-introducing-raid.html>)
- Working RAID illustrations (<http://www.lascon.co.uk/d008005.htm>)
- RAID system explained in detail (<http://www.eukhost.com/forums/f15/raid-explained-4063/>)
- RAID Levels — Tutorial and Diagrams (<http://www.acnc.com/raid.html>)
- Logical Volume Manager Performance Measurement (<http://www.shub-internet.org/brad/FreeBSD/vinum.html>)
- Animations and Descriptions from Dell to help Learn about RAID Levels 0, 1, 5, 10, and 50 (http://www.dellcommunity.com/supportforums/board/message?board.id=pes_harddrive&thread.id=25335)
- Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems (<http://www.cs.utk.edu/~plank/plank/papers/CS-96-332.pdf>)
- Parity Declustering for Continuous Operation in Redundant Disk Arrays (<http://www.pdl.cmu.edu/PDL-FTP/Declustering/ASPLOS.pdf>)
- An Optimal Scheme for Tolerating Double Disk Failures in RAID Architectures (<http://www.cs.berkeley.edu/~culler/cs252-s02/papers/p245-blaum.pdf>)
- Informations about Raid Systems - German (<http://www.raid-controller.info>)

Retrieved from "<http://en.wikipedia.org/wiki/RAID>"

Categories: RAID

Hidden categories: Articles needing additional references from September 2007 • All articles with unsourced statements • Articles with unsourced statements since October 2008

- This page was last modified on 8 October 2008, at 14:38.
 - All text is available under the terms of the GNU Free Documentation License. (See **Copyrights** for details.)
- Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a U.S. registered 501(c)(3) tax-deductible nonprofit charity.