RAID

From Wikipedia, the free encyclopedia

RAID (**redundant array of independent disks**) is a data storage virtualization technology that combines multiple physical disk drive components into a single logical unit for the purposes of data redundancy, performance improvement, or both.[1]

Data is distributed across the drives in one of several ways, referred to as RAID levels, depending on the required level of redundancy and performance. The different schemes, or data distribution layouts, are named by the word RAID followed by a number, for example RAID 0 or RAID 1. Each schema, or RAID level, provides a different balance among the key goals: reliability, availability, performance, and capacity. RAID levels greater than RAID 0 provide protection against unrecoverable sector read errors, as well as against failures of whole physical drives.

Contents

History

The underlying concept of RAID was first spoken of by the co-founders of Geac Computer Corporation, Gus German and Ted Grunau, who referred to it as MF-100.[2]

The term "RAID" was invented by David Patterson, Garth A. Gibson, and Randy Katz at the University of California, Berkeley in 1987. In their June 1988 paper "A Case for Redundant Arrays of Inexpensive Disks (RAID)", presented at the SIGMOD conference, they argued that the top performing mainframe disk drives of the time could be beaten on performance by an array of the inexpensive drives that had been developed for the growing personal computer market. Although failures would rise in proportion to the number of drives, by configuring for redundancy, the reliability of an array could far exceed that of any large single drive.[3][4]

Although not yet using that terminology, the technologies of the five levels of RAID named in the June 1988 paper were used in various products prior to the paper's publication,[5] including the following:

- In 1977, Norman Ken Ouchi at IBM filed a patent disclosing what was subsequently named RAID 4.[6]
- Around 1983, DEC began shipping subsystem mirrored RA8X disk drives (now known as RAID 1) as part of its HSC50 subsystem.[7]
- In 1986, Clark et al. at IBM filed a patent disclosing what was subsequently named RAID 5.[8]
- Around 1988, the Thinking Machines' DataVault used error correction codes (now known as RAID 2) in an array of disk drives.[9] A similar approach was used in the early 1960s on the IBM 353.[10][11]

Industry RAID manufacturers later tended to interpret the acronym as standing for "redundant array of *independent* disks".[12][13][14][15]

Overview

Many RAID levels employ an error protection scheme called "parity", a widely used method in information technology to provide fault tolerance in a given set of data. Most use simple XOR, but RAID 6 uses two separate parities based respectively on addition and multiplication in a particular Galois field or Reed–Solomon error correction.[16]

RAID can also provide data security with solid-state drives (SSDs) without the expense of an all-SSD system. For example, a fast SSD can be mirrored with a mechanical drive. For this configuration to provide a significant speed advantage an appropriate controller is needed that uses the fast SSD for all read operations. Adaptec calls this "hybrid RAID".[17]

Standard levels

A number of standard schemes have evolved. These are called *levels*. Originally, there were five RAID levels, but many variations have evolved, notably several nested levels and many non-standard levels (mostly proprietary). RAID levels and their associated data formats are standardized by the Storage Networking Industry Association (SNIA) in the Common RAID Disk Drive Format (DDF) standard:[18][19]

RAID 0


Storage servers with 24 hard disk drives and built-in hardware RAID controllers supporting various RAID levels

RAID 0 consists of striping, without mirroring or parity. The capacity of a RAID 0 volume is the sum of the capacities of the disks in the set, the same as with a spanned volume. There is no added redundancy for handling disk failures, just as with a spanned volume. Thus, failure of one disk causes the loss of the entire RAID 0 volume, with reduced possibilities of data recovery when compared with a broken spanned volume. Striping distributes the contents of files roughly equally among all disks in the set, which makes concurrent read or write operations on the multiple disks almost inevitable and results in performance improvements. The concurrent operations make the throughput of most read and write operations equal to the throughput of one disk multiplied by the number of disks. Increased throughput is the big benefit of RAID 0 versus spanned volume,[13] at the cost of increased vulnerability to drive failures.

RAID 1

RAID 1 consists of data mirroring, without parity or striping. Data is written identically to two drives, thereby producing a "mirrored set" of drives. Thus, any read request can be serviced by any drive in the set. If a request is broadcast to every drive in the set, it can be serviced by the drive that accesses the data first (depending on its seek time and rotational latency), improving performance. Sustained read throughput, if the controller or software is optimized for it, approaches the sum of throughputs of every drive in the set, just as for RAID 0. Actual read throughput of most RAID 1 implementations is slower than the fastest drive. Write throughput is always slower because every drive must be updated, and the slowest drive limits the write performance. The array continues to operate as long as at least one drive is functioning.[13]

RAID 2

RAID 2 consists of bit-level striping with dedicated Hamming-code parity. All disk spindle rotation is synchronized and data is striped such that each sequential bit is on a different drive. Hamming-code parity is calculated across corresponding bits and stored on at least one parity drive.[13] This level is of historical significance only; although it was used on some early machines (for example, the Thinking Machines CM-2),[20] as of 2014 it is not used by any commercially available system.[21]

RAID 3

RAID 3 consists of byte-level striping with dedicated parity. All disk spindle rotation is synchronized and data is striped such that each sequential byte is on a different drive. Parity is calculated across corresponding bytes and stored on a dedicated parity drive.[13] Although implementations exist,[22] RAID 3 is not commonly used in practice.

RAID 4

RAID 4 consists of block-level striping with dedicated parity. This level was previously used by NetApp, but has now been largely replaced by a proprietary implementation of RAID 4 with two parity disks, called RAID-DP.[23] The main advantage of RAID 4 over RAID 2 and 3 is I/O parallelism: in RAID 2 and 3, a single read/write I/O operation requires reading the whole group of data drives, while in RAID 4 one I/O read/write operation does not have to spread across all data drives. As a result, more I/O operations can be executed in parallel, improving the performance of small transfers.[3]

RAID 5

RAID 5 consists of block-level striping with distributed parity. Unlike RAID 4, parity information is distributed among the drives, requiring all drives but one to be present to operate. Upon failure of a single drive, subsequent reads can be calculated from the distributed parity such that no data is lost. RAID 5 requires at least three disks.[13] RAID 5 implementations are susceptible to system failures because of trends regarding array rebuild time and the chance of drive failure during rebuild (see "Increasing rebuild time and failure probability" section, below).[24] Rebuilding an array requires reading all data from all disks, opening a chance for a second drive failure and the loss of the entire array. In August 2012, Dell posted an advisory against the use of RAID 5 in any configuration on Dell EqualLogic arrays and RAID 50 with "Class 2 7200 RPM drives of 1 TB and higher capacity" for business-critical data.[25]

RAID 6

RAID 6 consists of block-level striping with double distributed parity. Double parity provides fault tolerance up to two failed drives. This makes larger RAID groups more practical, especially for high-availability systems, as large-capacity drives take longer to restore. RAID 6 requires a minimum of four disks. As with RAID 5, a single drive failure results in reduced performance of the entire array until the failed drive has been replaced.[13] With a RAID 6 array, using drives from multiple sources and manufacturers, it is possible to mitigate most of the problems associated with RAID 5. The larger the drive capacities and the larger the array size, the more important it becomes to choose RAID 6 instead of RAID 5.[26] RAID 10 also minimizes these problems.[27]

Nested (hybrid) RAID

In what was originally termed *hybrid RAID*,[28] many storage controllers allow RAID levels to be nested. The elements of a *RAID* may be either individual drives or arrays themselves. Arrays are rarely nested more than one level deep.[29]

The final array is known as the top array. When the top array is RAID 0 (such as in RAID 1+0 and RAID 5+0), most vendors omit the "+" (yielding RAID 10 and RAID 50, respectively).

- RAID 0+1: creates two stripes and mirrors them. If a single drive failure occurs then one of the stripes has failed, at this point you are running effectively as RAID 0 with no redundancy, significantly higher risk is introduced during a rebuild than RAID 1+0 as all the data from all the drives in the remaining stripe has to be read rather than just from 1 drive increasing the chance of an unrecoverable read error (URE) and significantly extending the rebuild window. [30][31][32]
- RAID 1+0: creates a striped set from a series of mirrored drives. The array can sustain multiple drive losses so long as no mirror loses all its drives.[33]
- JBOD RAID N+N: With JBOD (Just a Bunch Of Disks), it is possible to concatenate disks, but also volumes such as RAID sets. With larger drive capacities, write and rebuilding time may increase dramatically (especially, as described above, with RAID 5 and RAID 6). By splitting larger RAID sets into smaller subsets and concatenating them with JBOD, write and rebuilding time may be reduced. If a hardware RAID controller is not capable of nesting JBOD with RAID, then JBOD can be achieved with software RAID in combination with RAID set volumes offered by the hardware RAID controller. There is another advantage in the form of disaster recovery, if a small RAID subset fails, then the data on the other RAID subsets is not lost, reducing restore time.

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 specialized needs of a small niche group. Such configurations include the following:

- Linux MD RAID 10 provides a general RAID driver that in its "near" layout defaults to a standard RAID 1 with two drives, and a standard RAID 1+0 with four drives; however, it can include any number of drives, including odd numbers. With its "far" layout, MD RAID 10 can run both striped and mirrored, even with only two drives in `f2` layout; this runs mirroring with striped reads, giving the read performance of RAID 0. Regular RAID 1, as provided by Linux software RAID, does not stripe reads, but can perform reads in parallel.[33][34][35]

- Hadoop has a RAID system that generates a parity file by xor-ing a stripe of blocks in a single HDFS file.[36]
- BeeGFS, the parallel file system, has internal striping (comparable to file-based RAID0) and replication (comparable to file-based RAID10) options to aggregate throughput and capacity of multiple servers and is typically based on top of an underlying RAID to make disk failures transparent.

Implementations

The distribution of data across multiple drives can be managed either by dedicated computer hardware or by software. A software solution may be part of the operating system, part of the firmware and drivers supplied with a standard drive controller (so-called "hardware-assisted software RAID"), or it may reside entirely within the hardware RAID controller.

Software-based

Software RAID implementations are provided by many modern operating systems. Software RAID can be implemented as:

- A layer that abstracts multiple devices, thereby providing a single virtual device (e.g. Linux kernel's md and OpenBSD's softraid)
- A more generic logical volume manager (provided with most server-class operating systems, e.g. Veritas or LVM)
- A component of the file system (e.g. ZFS, GPFS or Btrfs)
- A layer that sits above any file system and provides parity protection to user data (e.g. RAID-F)[37]

Some advanced file systems are designed to organize data across multiple storage devices directly, without needing the help of a third-party logical volume manager:

- ZFS supports equivalents of RAID 0, RAID 1, RAID 5 (RAID-Z), RAID 6 (RAID-Z2) and a triple-parity version RAID-Z3. As it always stripes over top-level vdevs, it supports equivalents of the 1+0, 5+0, and 6+0 nested RAID levels (as well as striped triple-parity sets) but not other nested combinations. ZFS is the native file system on Solaris and illumos, and is also available on FreeBSD and Linux. Open-source ZFS implementations are actively developed under the OpenZFS umbrella project.[38][39][40][41][42]
- GPFS, initially developed by IBM for media streaming and scalable analytics, supports declustered RAID protection schemes up to n+3. A particularity is the dynamic rebuilding priority which runs with low impact in the background until a data chunk hits n+0 redundancy, in which case this chunk is quickly rebuilt to at least n+1. On top, GPFS supports metro-distance RAID 1.[43]
- Btrfs supports RAID 0, RAID 1 and RAID 10 (RAID 5 and 6 are under development).[44][45]
- XFS was originally designed to provide an integrated volume manager that supports concatenating, mirroring and striping of multiple physical storage devices.[46] However, the implementation of XFS in Linux kernel lacks the integrated volume manager.[47]

Many operating systems provide RAID implementations, including the following:

- Apple's macOS and macOS Server support RAID 0, RAID 1, and RAID 1+0.[48][49]
- FreeBSD supports RAID 0, RAID 1, RAID 3, and RAID 5, and all nestings via GEOM modules and ccd.[50][51][52]
- Linux's md supports RAID 0, RAID 1, RAID 4, RAID 5, RAID 6, and all nestings.[53] Certain reshaping/resizing/expanding operations are also supported.[54]
- Microsoft's server operating systems support RAID 0, RAID 1, and RAID 5. Some of the Microsoft desktop operating systems support RAID. For example, Windows XP Professional supports RAID level 0, in addition to spanning multiple drives, but only if using dynamic disks and volumes. Windows XP can be modified to support RAID 0, 1, and 5.[55] Windows 8 and Windows Server 2012 introduces a RAID-like feature known as Storage Spaces, which also allows users to specify mirroring, parity, or no redundancy on a folder-by-folder basis.[56]
- NetBSD supports RAID 0, 1, 4, and 5 via its software implementation, named RAIDframe.[57]
- OpenBSD supports RAID 0, 1 and 5 via its software implementation, named softraid.[58]

If a boot drive fails, the system has to be sophisticated enough to be able to boot off the remaining drive or drives. For instance, consider a computer whose disk is configured as RAID 1 (mirrored drives); if the first drive in the array fails, then a first-stage boot loader might not be sophisticated enough to attempt loading the second-stage boot loader from the second drive as a fallback. The second-stage boot loader for FreeBSD is capable of loading a kernel from such an array.[59]

Firmware- and driver-based

A SATA 3.0 controller that provides RAID functionality through proprietary firmware and drivers

Software-implemented RAID is not always compatible with the system's boot process, and it is generally impractical for desktop versions of Windows. However, hardware RAID controllers are expensive and proprietary. To fill this gap, inexpensive "RAID controllers" were introduced that do not contain a dedicated RAID controller chip, but simply a standard drive controller chip with proprietary firmware and drivers. During early bootup, the RAID is implemented by the firmware and, once the operating system has been more completely loaded, the drivers take over control. Consequently, such controllers may not work when driver support is not available for the host operating system.[60] An example is Intel Matrix RAID, implemented on many consumer-level motherboards.[61][62]

Because some minimal hardware support is involved, this implementation approach is also called "hardware-assisted software RAID",[63][64][65] "hybrid model" RAID,[65] or even "fake RAID".[66] If RAID 5 is supported, the hardware may provide a hardware XOR accelerator. An advantage of this model over the pure software RAID is that—if using a redundancy mode—the boot drive is protected from failure (due to the firmware) during the boot process even before the operating systems drivers take over.[65]

Integrity

Data scrubbing (referred to in some environments as *patrol read*) involves periodic reading and checking by the RAID controller of all the blocks in an array, including those not otherwise accessed. This detects bad blocks before use.[67] Data scrubbing checks for bad blocks on each storage device in an array, but also uses the redundancy of the array to recover bad blocks on a single drive and to reassign the recovered data to spare blocks elsewhere on the drive.[68]

Frequently, a RAID controller is configured to "drop" a component drive (that is, to assume a component drive has failed) if the drive has been unresponsive for eight seconds or so; this might cause the array controller to drop a good drive because that drive has not been given enough time to complete its internal error recovery procedure. Consequently, using RAID for consumer-marketed drives can be risky, and so-called "enterprise class" drives limit this error recovery time to reduce risk. Western Digital's desktop drives used to have a specific fix. A utility called WDTLER.exe limited a drive's error recovery time. The utility enabled TLER (time limited error recovery), which limits the error recovery time to seven seconds. Around September 2009, Western Digital disabled this feature in their desktop drives (e.g. the Caviar Black line), making such drives unsuitable for use in RAID configurations.[69] However, Western Digital enterprise class drives are shipped from the factory with TLER enabled. Similar technologies are used by Seagate, Samsung, and Hitachi. Of course, for non-RAID usage, an enterprise class drive with a short error recovery timeout that cannot be changed is therefore less suitable than a desktop drive.[69] In late 2010, the Smartmontools program began supporting the configuration of ATA Error Recovery Control, allowing the tool to configure many desktop class hard drives for use in RAID setups.[69]

While RAID may protect against physical drive failure, the data is still exposed to operator, software, hardware, and virus destruction. Many studies cite operator fault as the most common source of malfunction,[70] such as a server operator replacing the incorrect drive in a faulty RAID, and disabling the system (even temporarily) in the process.[71]

An array can be overwhelmed by catastrophic failure that exceeds its recovery capacity and, of course, the entire array is at risk of physical damage by fire, natural disaster, and human forces, while backups can be stored off site. An array is also vulnerable to controller failure because it is not always possible to migrate it to a new, different controller without data loss.[72]

Weaknesses

Correlated failures

In practice, the drives are often the same age (with similar wear) and subject to the same environment. Since many drive failures are due to mechanical issues (which are more likely on older drives), this violates the assumptions of independent, identical rate of failure amongst drives; failures are in fact statistically correlated.[13] In practice, the chances for a second failure before the first has been recovered (causing data loss) are higher than the chances for random failures. In a study of about 100,000 drives, the probability of two drives in the same cluster failing within one hour was four times larger than predicted by the exponential statistical distribution—which characterizes processes in which events occur continuously and independently at a constant average rate. The probability of two failures in the same 10-hour period was twice as large as predicted by an exponential distribution.[73]

Unrecoverable read errors during rebuild

Unrecoverable read errors (URE) present as sector read failures, also known as *latent sector errors* (LSE). The associated media assessment measure, *unrecoverable bit error* (UBE) rate, is typically guaranteed to be less than one bit in 10^{15} for enterprise-class drives (SCSI, FC, SAS or SATA), and less than one bit in 10^{14} for desktop-class drives (IDE/ATA/PATA or SATA). Increasing drive capacities and large RAID 5 instances have led to the maximum error rates being insufficient to guarantee a successful recovery, due to the high likelihood of such an error occurring on one or more remaining drives during a RAID set rebuild.[13][74] When rebuilding, parity-based schemes such as RAID 5 are particularly prone to the effects of UREs as they affect not only the sector where they occur, but also reconstructed blocks using that sector for parity computation. Thus, an URE during a RAID 5 rebuild typically leads to a complete rebuild failure.[75]

Double-protection parity-based schemes, such as RAID 6, attempt to address this issue by providing redundancy that allows double-drive failures; as a downside, such schemes suffer from elevated write penalty—the number of times the storage medium must be accessed during a single write operation.[76] Schemes that duplicate (mirror) data in a drive-to-drive manner, such as RAID 1 and RAID 10, have a lower risk from UREs than those using parity computation or mirroring between striped sets.[27][77] Data scrubbing, as a background process, can be used to detect and recover from UREs, effectively reducing the risk of them happening during RAID rebuilds and causing double-drive failures. The recovery of UREs involves remapping of affected underlying disk sectors, utilizing the drive's sector remapping pool; in case of UREs detected during background scrubbing, data redundancy provided by a fully operational RAID set allows the missing data to be reconstructed and rewritten to a remapped sector.[78][79]

Increasing rebuild time and failure probability

Drive capacity has grown at a much faster rate than transfer speed, and error rates have only fallen a little in comparison. Therefore, larger-capacity drives may take hours if not days to rebuild, during which time other drives may fail. The rebuild time is also limited if the entire array is still in operation at reduced capacity.[80] Given an array with only one redundant drive (which applies to RAID levels 3, 4 and 5, and to "classic" two-drive RAID 1), a second drive failure would cause complete failure of the array. Even though individual drives' mean time between failure (MTBF) have increased over time, this increase has not kept pace with the increased storage capacity of the drives. The time to rebuild the array after a single drive failure, as well as the chance of a second failure during a rebuild, have increased over time.[24]

Some commentators have declared that RAID 6 is only a "band aid" in this respect, because it only kicks the problem a little further down the road.[24] However, according to the 2006 NetApp study of Berriman et al., the chance of failure decreases by a factor of about 3,800 (relative to RAID 5) for a proper implementation of RAID 6, even when using commodity drives.[81] Nevertheless, if the currently observed technology trends remain unchanged, in 2019 a RAID 6 array will have the same chance of failure as its RAID 5 counterpart had in 2010.[74][81]

Mirroring schemes such as RAID 10 have a bounded recovery time as they require the copy of a single failed drive, compared with parity schemes such as RAID 6, which require the copy of all blocks of the drives in an array set. Triple parity schemes, or triple mirroring, have been suggested as one approach to improve resilience to an additional drive failure during this large rebuild time.[81]

Atomicity: including parity inconsistency due to system crashes

A system crash or other interruption of a write operation can result in states where the parity is inconsistent with the data due to non-atomicity of the write process, such that the parity cannot be used for recovery in the case of a disk failure (the so-called RAID 5 write hole).[13] The RAID write hole is a known data corruption issue in older and low-end RAIDs, caused by interrupted destaging of writes to disk.[82]

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.[83]

Write-cache reliability

There are concerns about write-cache reliability, specifically regarding devices equipped with a write-back cache, which is a caching system that reports the data as written as soon as it is written to cache, as opposed to when it is written to the non-volatile medium. If the system experiences a power loss or other major failure, the data may be irrevocably lost from the cache before reaching the non-volatile storage.[84]

See also

- Network-attached storage (NAS)
- Non-RAID drive architectures
- Redundant array of independent memory
- S.M.A.R.T.

References

1. Arpaci-Dusseau, Remzi H.; Arpaci-Dusseau, Andrea C. (2014), *Operating Systems: Three Easy Pieces [Chapter: RAID]* (PDF), Arpaci-Dusseau Books
2. Parents of Invention: The Development of Library Automation Systems in the Late 20th Century. (https://books.google.com/books?id=xUT_d4tN7AgC&lpg=PA103&dq=ted%20grunau&pg=PA103#v=onepage&q=ted%20grunau&f=false) By Christopher Syed https://books.google.com/books?id=xUT_d4tN7AgC&lpg=PA103&dq=ted%20grunau&pg=PA103#v=onepage&q=tec
3. Patterson, David; Gibson, Garth A.; Katz, Randy (1988). *A Case for Redundant Arrays of Inexpensive Disks (RAID)* (PDF). SIGMOD Conferences. Retrieved 2006-12-31.
4. Frank Hayes (November 17, 2003). "The Story So Far". Computerworld. Retrieved November 18, 2016. "Patterson recalled the beginnings of his RAID project in 1987. […] 1988: David A. Patterson leads a team that defines RAID standards for improved performance, reliability and scalability."
5. Randy H. Katz (October 2010). "RAID: A Personal Recollection of How Storage Became a System" (PDF). *eecs.umich.edu*. IEEE Computer Society. Retrieved 2015-01-18. "We were not the first to think of the idea of replacing what Patterson described as a slow large expensive disk (SLED) with an array of inexpensive disks. For example, the concept of disk mirroring, pioneered by Tandem, was well known, and some storage products had already been constructed around arrays of small disks."
6. US patent 4092732 (https://worldwide.espacenet.com/textdoc?DB=EPODOC&IDX=US4092732), Norman Ken Ouchi, "System for Recovering Data Stored in Failed Memory Unit", issued 1978-05-30
7. "HSC50/70 Hardware Technical Manual" (PDF). DEC. July 1986. pp. 29, 32. Retrieved 2014-01-03.
8. US patent 4761785 (https://worldwide.espacenet.com/textdoc?DB=EPODOC&IDX=US4761785), Brian E. Clark, et al., "Parity Spreading to Enhance Storage Access", issued 1988-08-02

9. US patent 4899342 (https://worldwide.espacenet.com/textdoc?DB=EPODOC&IDX=US4899342), David Potter et. al., "Method and Apparatus for Operating Multi-Unit Array of Memories", issued 1990-02-06 See also The Connection Machine (1988) (http://www.svisions.com/sv/raid-patent.html)
10. "IBM 7030 Data Processing System: Reference Manual" (PDF). *bitsavers.trailing-edge.com*. IBM. 1960. p. 157. Retrieved 2015-01-17. "Since a large number of bits are handled in parallel, it is practical to use error checking and correction (ECC) bits, and each 39 bit byte is composed of 32 data bits and seven ECC bits. The ECC bits accompany all data transferred to or from the high-speed disks, and, on reading, are used to correct a single bit error in a byte and detect double and most multiple errors in a byte."
11. "IBM Stretch (aka IBM 7030 Data Processing System)". *brouhaha.com*. 2009-06-18. Retrieved 2015-01-17. "A typical IBM 7030 Data Processing System might have been comprised of the following units: [...] IBM 353 Disk Storage Unit – similar to IBM 1301 Disk File, but much faster. 2,097,152 (2^21) 72-bit words (64 data bits and 8 ECC bits), 125,000 words per second"
12. "Originally referred to as Redundant Array of Inexpensive Disks, the concept of RAID was first developed in the late 1980s by Patterson, Gibson, and Katz of the University of California at Berkeley. (The RAID Advisory Board has since substituted the term Inexpensive with Independent.)" Storage Area Network Fundamentals; Meeta Gupta; Cisco Press; ISBN 978-1-58705-065-7; Appendix A.
13. Chen, Peter; Lee, Edward; Gibson, Garth; Katz, Randy; Patterson, David (1994). "RAID: High-Performance, Reliable Secondary Storage". *ACM Computing Surveys*. **26**: 145–185. doi:10.1145/176979.176981.
14. Donald, L. (2003). "MCSA/MCSE 2006 JumpStart Computer and Network Basics" (2nd ed.). Glasgow: SYBEX.
15. Howe, Denis (ed.). *Redundant Arrays of Independent Disks from FOLDOC*. *Free On-line Dictionary of Computing*. Imperial College Department of Computing. Retrieved 2011-11-10.
16. Dawkins, Bill and Jones, Arnold. "Common RAID Disk Data Format Specification" (http://www.snia.org/tech_activities/standards/curr_standards/ddf/SNIA-DDFv1.2.pdf) *[Storage Networking Industry Association]* Colorado Springs, 28 July 2006. Retrieved on 22 February 2011.
17. "Adaptec Hybrid RAID Solutions" (PDF). *Adaptec.com*. Adaptec. 2012. Retrieved 2013-09-07.
18. "Common RAID Disk Drive Format (DDF) standard". *SNIA.org*. SNIA. Retrieved 2012-08-26.
19. "SNIA Dictionary". *SNIA.org*. SNIA. Retrieved 2010-08-24.
20. Andrew S. Tanenbaum. *Structured Computer Organization 6th ed*. p. 95.
21. Hennessy, John; Patterson, David (2006). *Computer Architecture: A Quantitative Approach, 4th ed*. p. 362. ISBN 978-0123704900.
22. "FreeBSD Handbook, Chapter 20.5 GEOM: Modular Disk Transformation Framework". Retrieved 2012-12-20.
23. White, Jay; Lueth, Chris (May 2010). "RAID-DP:NetApp Implementation of Double Parity RAID for Data Protection. NetApp Technical Report TR-3298". Retrieved 2013-03-02.
24. Newman, Henry (2009-09-17). "RAID's Days May Be Numbered". *EnterpriseStorageForum*. Retrieved 2010-09-07.
25. Peltoniemi, Mikko (2012-08-07). "New RAID level recommendations from Dell". Retrieved 2012-12-01.
26. "Why RAID 6 stops working in 2019". ZDNet. 22 February 2010.
27. Scott Lowe (2009-11-16). "How to protect yourself from RAID-related Unrecoverable Read Errors (UREs). Techrepublic.". Retrieved 2012-12-01.
28. Vijayan, S.; Selvamani, S.; Vijayan, S (1995). "Dual-Crosshatch Disk Array: A Highly Reliable Hybrid-RAID Architecture". *Proceedings of the 1995 International Conference on Parallel Processing: Volume 1*. CRC Press. pp. I–146*ff*. ISBN 0-8493-2615-X.
29. "What is RAID (Redundant Array of Inexpensive Disks)?". *StoneFly.com*. Retrieved 2014-11-20.
30. "Why is RAID 1+0 better than RAID 0+1?". *aput.net*. Retrieved 2016-05-23.
31. "RAID 10 Vs RAID 01 (RAID 1+0 Vs RAID 0+1) Explained with Diagram". *www.thegeekstuff.com*. Retrieved 2016-05-23.
32. "Comparing RAID 10 and RAID 01 | SMB IT Journal". *www.smbitjournal.com*. Retrieved 2016-05-23.
33. Jeffrey B. Layton: "Intro to Nested-RAID: RAID-01 and RAID-10" (http://www.linux-mag.com/id/7928?hq_e=el&hq_m=1151565&hq_l=36&hq_v=3fa9646c7f), Linux Magazine, January 6, 2011
34. "Performance, Tools & General Bone-Headed Questions". tldp.org. Retrieved 2013-12-25.
35. "Main Page - Linux-raid". osdl.org. 2010-08-20. Retrieved 2010-08-24.
36. "Hdfs Raid". Hadoopblog.blogspot.com. 2009-08-28. Retrieved 2010-08-24.
37. "RAID over File System". Retrieved 2014-07-22.
38. "ZFS -illumos". illumos.org. 2014-09-15. Retrieved 2016-05-23.
39. "Creating and Destroying ZFS Storage Pools - Oracle Solaris ZFS Administration Guide". Oracle Corporation. 2012-04-01. Retrieved 2014-07-27.
40. "20.2. The Z File System (ZFS)". *freebsd.org*. Retrieved 2014-07-27.
41. "Double Parity RAID-Z (raidz2) (Solaris ZFS Administration Guide)". Oracle Corporation. Retrieved 2014-07-27.
42. "Triple Parity RAIDZ (raidz3) (Solaris ZFS Administration Guide)". Oracle Corporation. Retrieved 2014-07-27.
43. Deenadhayalan, Veera (2011). "General Parallel File System (GPFS) Native RAID" (PDF). *UseNix.org*. IBM. Retrieved 2014-09-28.
44. "Btrfs Wiki: Feature List". 2012-11-07. Retrieved 2012-11-16.

45. "Btrfs Wiki: Changelog". 2012-10-01. Retrieved 2012-11-14.
46. Philip Trautman; Jim Mostek. "Scalability and Performance in Modern File Systems". *linux-xfs.sgi.com*. Retrieved 2015-08-17.
47. "Linux RAID Setup – XFS". *kernel.org*. 2013-10-05. Retrieved 2015-08-17.
48. "Mac OS X: How to combine RAID sets in Disk Utility". Retrieved 2010-01-04.
49. "Apple Mac OS X Server File Systems". Retrieved 2008-04-23.
50. "FreeBSD System Manager's Manual page for GEOM(8)". Retrieved 2009-03-19.
51. "freebsd-geom mailing list - new class / geom_raid5". Retrieved 2009-03-19.
52. "FreeBSD Kernel Interfaces Manual for CCD(4)". Retrieved 2009-03-19.
53. "The Software-RAID HowTo". Retrieved 2008-11-10.
54. "mdadm(8) - Linux man page". *Linux.Die.net*. Retrieved 2014-11-20.
55. "Using Windows XP to Make RAID 5 Happen". Tomshardware.com. Retrieved 2010-08-24.
56. Sinofsky, Steven. "Virtualizing storage for scale, resiliency, and efficiency". Microsoft.
57. Metzger, Perry (1999-05-12). "NetBSD 1.4 Release Announcement". *NetBSD.org*. The NetBSD Foundation. Retrieved 2013-01-30.
58. "OpenBSD softraid man page". *OpenBSD.org*. Retrieved 2015-10-05.
59. "FreeBSD Handbook". *Chapter 19 GEOM: Modular Disk Transformation Framework*. Retrieved 2009-03-19.
60. "SATA RAID FAQ". Ata.wiki.kernel.org. 2011-04-08. Retrieved 2012-08-26.
61. Red Hat Enterprise Linux - Storage Administrator Guide - RAID Types (https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Storage_Administration_Guide/s1-raid-approaches.html)
62. Charlie Russel; Sharon Crawford; Andrew Edney (2011). *Working with Windows Small Business Server 2011 Essentials*. O'Reilly Media, Inc. p. 90. ISBN 978-0-7356-5670-3.
63. Warren Block. "19.5. Software RAID Devices". *freebsd.org*. Retrieved 2014-07-27.
64. Ronald L. Krutz; James Conley (2007). *Wiley Pathways Network Security Fundamentals*. John Wiley & Sons. p. 422. ISBN 978-0-470-10192-6.
65. Hardware RAID vs. Software RAID: Which Implementation is Best for my Application? Adaptec Whitepaper (http://www.adaptec.com/nr/rdonlyres/14b2fd84-f7a0-4ac5-a07a-214123ea3dd6/0/4423_sw_hwraid_10.pdf)
66. Gregory Smith (2010). *PostgreSQL 9.0: High Performance*. Packt Publishing Ltd. p. 31. ISBN 978-1-84951-031-8.
67. Ulf Troppens, Wolfgang Mueller-Friedt, Rainer Erkens, Rainer Wolafka, Nils Haustein. Storage Networks Explained: Basics and Application of Fibre Channel SAN, NAS, ISCSI, InfiniBand and FCoE. John Wiley and Sons, 2009. p.39
68. Dell Computers, Background Patrol Read for Dell PowerEdge RAID Controllers, By Drew Habas and John Sieber, Reprinted from Dell Power Solutions, February 2006 http://www.dell.com/downloads/global/power/ps1q06-20050212-Habas.pdf
69. "Error Recovery Control with Smartmontools". Retrieved 2011. Check date values in: `|access-date=` (help)
70. These studies are: Gray, J (1990), Murphy and Gent (1995), Kuhn (1997), and Enriquez P. (2003).
71. Patterson, D., Hennessy, J. (2009), 574.
72. "The RAID Migration Adventure". Retrieved 2010-03-10.
73. 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
74. Harris, Robin (2010-02-27). "Does RAID 6 stop working in 2019?". *StorageMojo.com*. TechnoQWAN. Retrieved 2013-12-17.
75. J.L. Hafner, V. Deenadhaylan, K. Rao, and J.A. Tomlin. "Matrix methods for lost data reconstruction in erasure codes. USENIX Conference on File and Storage Technologies, p15-30, Dec. 13-16, 2005.
76. Miller, Scott Alan (2016-01-05). "Understanding RAID Performance at Various Levels". *Recovery Zone*. StorageCraft. Retrieved 2016-07-22.
77. Art S. Kagel (March 2, 2011). "RAID 5 versus RAID 10 (or even RAID 3, or RAID 4)". *miracleas.com*. Retrieved October 30, 2014.
78. M.Baker, M.Shah, D.S.H. Rosenthal, M.Roussopoulos, P.Maniatis, T.Giuli, and P.Bungale. 'A fresh look at the reliability of long-term digital storage." EuroSys2006, Apr. 2006.
79. "L.N. Bairavasundaram, GR Goodson, S. Pasupathy, J.Schindler. "An analysis of latent sector errors in disk drives". Proceedings of SIGMETRICS'07, June 12-16,2007." (PDF).
80. Patterson, D., Hennessy, J. (2009). *Computer Organization and Design*. New York: Morgan Kaufmann Publishers. pp 604-605.
81. Leventhal, Adam (2009-12-01). "Triple-Parity RAID and Beyond. ACM Queue, Association of Computing Machinery". Retrieved 2012-11-30.
82. " "Write hole" in RAID5, RAID6, RAID1, and other arrays". ZAR team. Retrieved 15 February 2012.
83. 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
84. "Definition of write-back cache at SNIA dictionary".

External links

- RAID (http://dmoztools.net/Computers/Hardware/Storage/Subsystems/RAID) at DMOZ
- "Empirical Measurements of Disk Failure Rates and Error Rates" (http://research.microsoft.com/research/pubs/view.aspx?msr_tr_id=MSR-TR-2005-166), by Jim Gray and Catharine van Ingen, December 2005
- The Mathematics of RAID-6 (https://www.kernel.org/pub/linux/kernel/people/hpa/raid6.pdf), by H. Peter Anvin
- Does Fake RAID Offer Any Advantage Over Software RAID? (http://superuser.com/questions/245928/does-fake-raid-offer-any-advantage-over-software-raid) – Discussion on superuser.com
- Comparing RAID Implementation Methods (ftp://ftp.dell.com/app/3q03-Dum.pdf) – Dell.com
- BAARF: Battle Against Any Raid Five (http://www.miracleas.com/BAARF/BAARF2.html) (RAID 3, 4 and 5 versus RAID 10)
- A Clean-Slate Look at Disk Scrubbing (https://www.usenix.org/legacy/event/fast10/tech/full_papers/oprea.pdf)

Wikimedia Commons has media related to ***Redundant array of independent disks***.

Categories: RAID
