WIKIPEDIA

Standard RAID levels

In computer storage, the **standard RAID levels** comprise a basic set of RAID (redundant array of independent disks) configurations that employ the techniques of striping, mirroring, or parity to create large reliable data stores from multiple general-purpose computer hard disk drives (HDDs). The most common types are RAID 0 (striping), RAID 1 (mirroring) and its variants, RAID 5 (distributed parity), and RAID 6 (dual parity). 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.[1]

While most RAID levels can provide good protection against and recovery from hardware defects or defective sectors/read errors (*hard errors*), they do not provide any protection against data loss due to catastrophic failures (fire, water) or *soft errors* such as user error, software malfunction, or malware infection. For valuable data, RAID is only one building block of a larger data loss prevention and recovery scheme – it cannot replace a backup plan.

Contents

RAID 0

RAID 0 (also known as a *stripe set* or *striped volume*) splits ("stripes") data evenly across two or more disks, without parity information, redundancy, or fault tolerance. Since RAID 0 provides no fault tolerance or redundancy, the failure of one drive will cause the entire array to fail; as a result of having data striped across all disks, the failure will result in total data loss. This configuration is typically implemented having speed as the intended goal.[2][3] RAID 0 is normally used to increase performance, although it can also be used as a way to create a large logical volume out of two or more physical disks.[4]

A RAID 0 setup can be created with disks of differing sizes, but the storage space added to the array by each disk is limited to the size of the smallest disk. For example, if a 120 GB disk is striped together with a 320 GB disk, the size of the array will be 120 GB × 2 = 240 GB. However, some RAID implementations allow the remaining 200 GB to be used for other purposes.

The diagram in this section shows how the data is distributed into Ax stripes on two disks, with A1:A2 as the first stripe, A3:A4 as the second one, etc. Once the stripe size is defined during the creation of a RAID 0 array, it needs to be maintained at all times. Since the stripes are accessed in parallel, an n-drive RAID 0 array appears as a single large disk with a data rate n times higher than the single-disk rate.

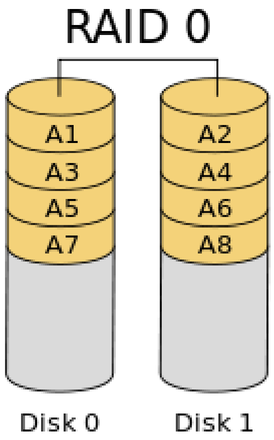

Diagram of a RAID 0 setup

Performance

A RAID 0 array of n drives provides data read and write transfer rates up to n times as high as the individual drive rates, but with no data redundancy. As a result, RAID 0 is primarily used in applications that require high performance and are able to tolerate lower reliability, such as in scientific computing[5] or computer gaming.[6]

Some benchmarks of desktop applications show RAID 0 performance to be marginally better than a single drive.[7][8] Another article examined these claims and concluded that "striping does not always increase performance (in certain situations it will actually be slower than a non-RAID setup), but in most situations it will yield a significant improvement in performance".[9][10] Synthetic benchmarks show different levels of performance improvements when multiple HDDs or SSDs are used in a RAID 0 setup, compared with single-drive performance. However, some synthetic benchmarks also show a drop in performance for the same comparison.[11][12]

RAID 1

RAID 1 consists of an exact copy (or *mirror*) of a set of data on two or more disks; a classic RAID 1 mirrored pair contains two disks. This configuration offers no parity, striping, or spanning of disk space across multiple disks, since the data is mirrored on all disks belonging to the array, and the array can only be as big as the smallest member disk. This layout is useful when read performance or reliability is more important than write performance or the resulting data storage capacity.[13][14]

The array will continue to operate so long as at least one member drive is operational.[15]

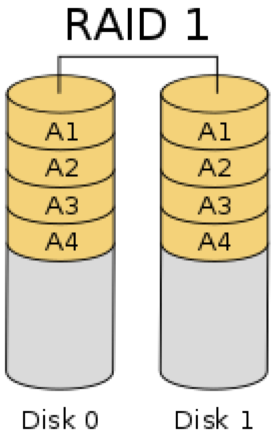

Diagram of a RAID 1 setup

Performance

Any read request can be serviced and handled by any drive in the array; thus, depending on the nature of I/O load, random read performance of a RAID 1 array may equal up to the sum of each member's performance,[a] while the write performance remains at the level of a single disk. However, if disks with different speeds are used in a RAID 1 array, overall write performance is equal to the speed of the slowest disk.[14][15]

Synthetic benchmarks show varying levels of performance improvements when multiple HDDs or SSDs are used in a RAID 1 setup, compared with single-drive performance. However, some synthetic benchmarks also show a drop in performance for the same comparison.[11][12]

RAID 2

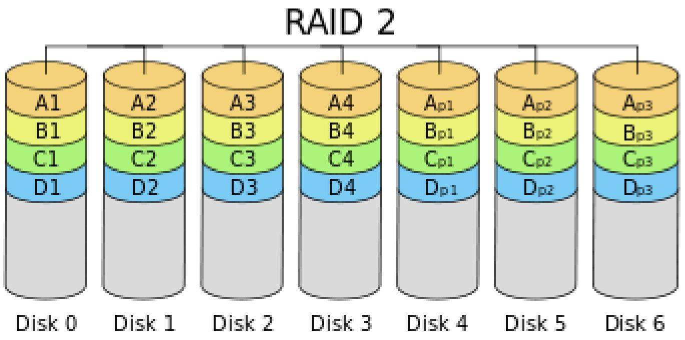

Diagram of a RAID 2 setup

RAID 2, which is rarely used in practice, stripes data at the bit (rather than block) level, and uses a Hamming code for error correction. The disks are synchronized by the controller to spin at the same angular orientation (they reach index at the same time), so it generally cannot service multiple requests simultaneously.[16][17] However, depending with a high rate Hamming code, many spindles would operate in parallel to simultaneously transfer data so that "very high data transfer rates" are possible[18] as for example in the DataVault where 32 data bits were transmitted simultaneously.

With all hard disk drives implementing internal error correction, the complexity of an external Hamming code offered little advantage over parity so RAID 2 has been rarely implemented; it is the only original level of RAID that is not currently used.[16][17]

RAID 3

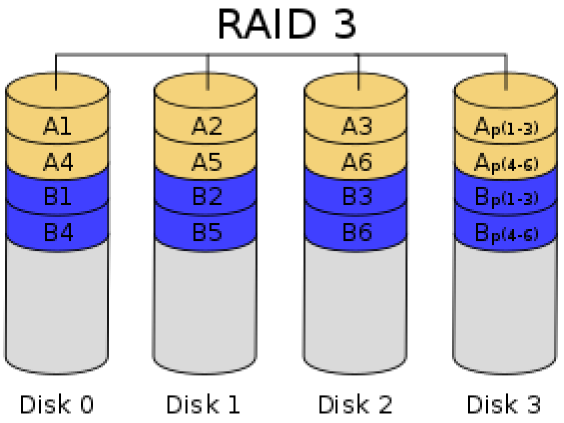

Diagram of a RAID 3 setup of six-byte blocks and two parity bytes, shown are two blocks of data in different colors.

RAID 3, which is rarely used in practice, consists of byte-level striping with a dedicated parity disk. One of the characteristics of RAID 3 is that it generally cannot service multiple requests simultaneously, which happens because any single block of data will, by definition, be spread across all members of the set and will reside in the same physical location on each disk. Therefore, any I/O operation requires activity on every disk and usually requires synchronized spindles.

This makes it suitable for applications that demand the highest transfer rates in long sequential reads and writes, for example uncompressed video editing. Applications that make small reads and writes from random disk locations will get the worst performance out of this level.[17]

The requirement that all disks spin synchronously (in a lockstep) added design considerations to a level that provided no significant advantages over other RAID levels, so it quickly became useless and is now obsolete.[16] Both RAID 3 and RAID 4 were quickly replaced by RAID 5.[19] RAID 3 was usually implemented in hardware, and the performance issues were addressed by using large disk caches.[17]

RAID 4

RAID 4 consists of block-level striping with a dedicated parity disk. As a result of its layout, RAID 4 provides good performance of random reads, while the performance of random writes is low due to the need to write all parity data to a single disk.[20]

In diagram 1, a read request for block A1 would be serviced by disk 0. A simultaneous read request for block B1 would have to wait, but a read request for B2 could be serviced concurrently by disk 1.

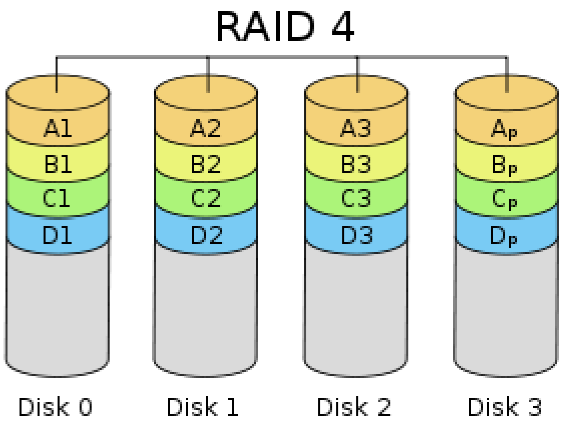

Diagram 1: A RAID 4 setup with dedicated parity disk with each color representing the group of blocks in the respective parity block (a stripe)

RAID 5

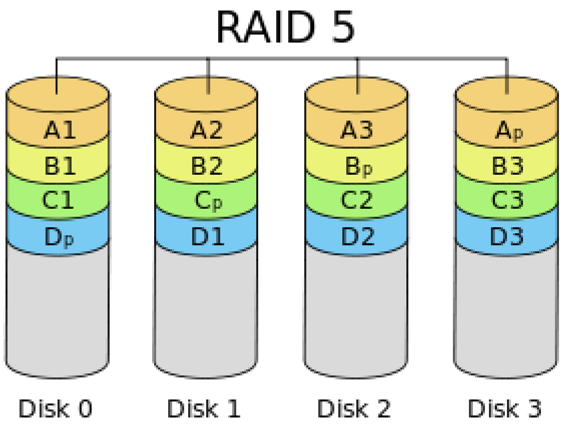

Diagram of a RAID 5 setup with distributed parity with each color representing the group of blocks in the respective parity block (a stripe). This diagram shows left asymmetric algorithm

RAID 5 consists of block-level striping with distributed parity. Unlike in RAID 4, parity information is distributed among the drives. It requires that all drives but one 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.[5] RAID 5 requires at least three disks.[21]

In comparison to RAID 4, RAID 5's distributed parity evens out the stress of a dedicated parity disk among all RAID members. Additionally, write performance is increased since all RAID members participate in the serving of write requests. Although it will not be as efficient as a striping (RAID 0) setup, because parity must still be written, this is no longer a bottleneck.[22]

Since parity calculation is performed on the full stripe, small changes to the array experience *write amplification*: in the worst case when a single, logical sector is to be written, the original sector and the according parity sector need to be read, the original data is removed from the parity, the new data calculated into the parity and both the new data sector and the new parity sector are written.

RAID 6

RAID 6 extends RAID 5 by adding another parity block; thus, it uses block-level striping with two parity blocks distributed across all member disks.[23]

According to the Storage Networking Industry Association (SNIA), the definition of RAID 6 is: "Any form of RAID that can continue to execute read and write requests to all of a RAID array's virtual disks in the presence of any two concurrent disk failures. Several methods, including dual check data computations (parity and Reed-Solomon), orthogonal dual parity check data and diagonal parity, have been used to implement RAID Level 6."[24]

Performance

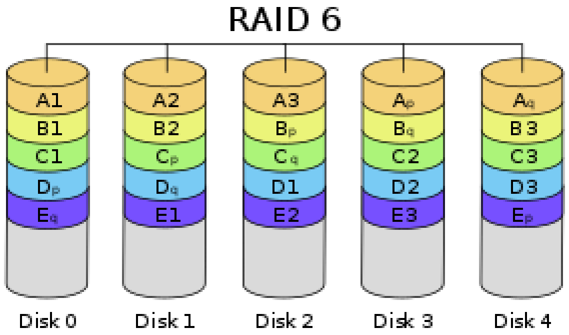

Diagram of a RAID 6 setup, which is identical to RAID 5 other than the addition of a second parity block

RAID 6 does not have a performance penalty for read operations, but it does have a performance penalty on write operations because of the overhead associated with parity calculations. Performance varies greatly depending on how RAID 6 is implemented in the manufacturer's storage architecture—in software, firmware, or by using firmware and specialized ASICs for intensive parity calculations. RAID 6 can read up to the same speed as RAID 5 with the same number of physical drives.[25]

When either diagonal or orthogonal dual parity is used, a second parity calculation is necessary for write operations. This doubles CPU overhead for RAID-6 versus single parity RAID levels. When a Reed Solomon code is used the second parity calculation is unnecessary. Reed Solomon has the advantage of allowing all redundancy information to be contained within a given stripe.

Simplified parity example

Suppose we would like to distribute our data over n chunks. Our goal is to define two parity values $\mathbf{P}$ and $\mathbf{Q}$, known as *syndromes*, resulting in a system of $n+2$ physical drives that is resilient to the loss of any two of them. In order to generate more than a single independent syndrome, we will need to perform our parity calculations on data chunks of size $k>1$. A typical choice in practice is a chunk size $k=8$, i.e. striping the data per-byte. We will denote the base-2 representation of a data chunk D as $d_0d_1...d_{k-1}$, where each d_i is either 0 or 1.

If we are using a small number of chunks $n \leq k$, we can use a simple parity computation, which will help motivate the use of the Reed-Solomon system in the general case. For our first parity value $\mathbf{P}$, we compute the simple XOR of the data across the stripes, as with RAID 5. This is written

$$\mathbf{P} = D_0 \ \oplus \ D_1 \ \oplus \ D_2 \oplus ... \oplus D_{n-1}$$

where $\oplus$ denotes the XOR operator. The second parity value is analogous, but with each data chunk bit-shifted a different amount. Writing $\mathrm{shift}(D) = d_1...d_{k-1}d_0$, we define

$$\mathbf{Q} = D_0 \oplus \mathrm{shift}(D_1) \oplus \mathrm{shift}^2(D_2) \oplus ... \oplus \mathrm{shift}^{n-1}(D_{n-1})$$

In the event of a single drive failure, the data can be recomputed from $\mathbf{P}$ just like with RAID 5. We will show we can also recover from simultaneous failure of 2 drives. If we lose a data chunk and $\mathbf{P}$, we can recover from $\mathbf{Q}$ and the remaining data by using the fact that $D \oplus D = 0$. Suppose on a system of $n=4$ chunks, the drive containing chunk D_3 has failed. We can compute

$$\mathbf{Q} \oplus D_0 \oplus \mathrm{shift}(D_1) \oplus \mathrm{shift}^2(D_2) = \mathrm{shift}^3(D_3)$$

and recover the lost data D_3 by undoing the bit shift. We can also recover from the failure of two data disks by computing the XOR of $\mathbf{P}$ and $\mathbf{Q}$ with the remaining data. If in the previous example, chunk D_0 had been lost as well, we would compute

$$A = \mathbf{P} \oplus D_1 \oplus D_2 = D_0 \oplus D_3$$

$$B = \mathbf{Q} \oplus \mathrm{shift}(D_1) \oplus \mathrm{shift}^2(D_2) = D_0 \oplus \mathrm{shift}^3(D_3)$$

On a bitwise level, this represents a system of $2k$ equations in $2k$ unknowns which uniquely determine the lost data.

This system will no longer work applied to a larger number of drives $n > k$. This is because if we repeatedly apply the shift operator k times to a chunk of length k, we end up back where we started! If we tried to apply the algorithm above to a system containing k disks, the second set of equations would be $B = D_0 \oplus \mathbf{shift}^k(D_k) = D_0 \oplus D_k$, which is the same as the first set of equations. This is only half as many as we need to solve for the missing values.

General parity system

It is possible to support a far greater number of drives by choosing the parity function more carefully. The issue we face is to ensure that a system of equations over the finite field $\mathbb{Z}_2$ has a unique solution, so we will turn to the theory of polynomial equations. Consider the Galois field $GF(m)$ with $m = 2^k$. This field is isomorphic to a polynomial field $F_2[x]/(p(x))$ for a suitable irreducible polynomial $p(x)$ of degree k over $\mathbb{Z}_2$. We will represent the data elements D as polynomials $\mathbf{D} = d_{k-1}x^{k-1} + d_{k-2}x^{k-2} + ... + d_1x + d_0$ in the Galois field. Let $\mathbf{D}_0, \ldots, \mathbf{D}_{n-1} \in GF(m)$ correspond to the stripes of data across hard drives encoded as field elements in this manner. We will use $\oplus$ to denote addition in the field, and concatenation to denote multiplication. The reuse of $\oplus$ is intentional: this is because addition in the finite field $\mathbb{Z}_2$ represents to the XOR operator, so computing the sum of two elements is equivalent to computing XOR on the polynomial coefficients.

A generator of a field is an element of the field such that g^i is different for each nonnegative $i < m - 1$. This means each element of the field, except the value 0, can be written as a power of g. A finite field is guaranteed to have at least one generator. Pick one such generator g, and define $\mathbf{P}$ and $\mathbf{Q}$ as follows:

$$\mathbf{P} = \bigoplus_i \mathbf{D}_i = \mathbf{D}_0 \ \oplus \ \mathbf{D}_1 \ \oplus \ \mathbf{D}_2 \oplus \ldots \oplus \mathbf{D}_{n-1}$$

$$\mathbf{Q} = \bigoplus_i g^i \mathbf{D}_i = g^0 \mathbf{D}_0 \ \oplus \ g^1 \mathbf{D}_1 \ \oplus \ g^2 \mathbf{D}_2 \oplus \ldots \oplus g^{n-1} \mathbf{D}_{n-1}$$

As before, the first checksum $\mathbf{P}$ is just the XOR of each stripe, though interpreted now as a polynomial. The effect of g^i can be thought of as the action of a carefully chosen linear feedback shift register on the data chunk.[26] Unlike the bit shift in the simplified example, which could only be applied k times before the encoding began to repeat, applying the operator g multiple times is guaranteed to produce $m = 2^k - 1$ unique invertible functions, which will allow a chunk length of k to support up to $2^k - 1$ data pieces.

If one data chunk is lost, the situation is similar to the one before. In the case of two lost data chunks, we can compute the recovery formulas algebraically. Suppose that $\mathbf{D}_i$ and $\mathbf{D}_j$ are the lost values with $i \neq j$, then, using the other values of D, we find constants A and B:

$$A = \mathbf{P} \ \oplus \ \left(\bigoplus_{\ell:\ \ell \neq i \text{ and } \ell \neq j} D_\ell \right) = D_i \oplus D_j$$

$$B = \mathbf{Q} \ \oplus \ \left(\bigoplus_{\ell:\ \ell \neq i \text{ and } \ell \neq j} g^\ell D_\ell \right) = g^i D_i \oplus g^j D_j$$

We can solve for D_i in the second equation and plug it into the first to find $D_j = (g^{n-i+j} \oplus 1)^{-1} g^{n-i} B \oplus A$, and then $D_i = A \oplus D_j$.

Unlike **P**, The computation of **Q** is relatively CPU intensive, as it involves polynomial multiplication in $F_2[x]/(p(x))$. This can be mitigated with a hardware implementation or by using an FPGA.

Comparison

The following table provides an overview of some considerations for standard RAID levels. In each case:

- Array space efficiency is given as an expression in terms of the number of drives, *n*; this expression designates a fractional value between zero and one, representing the fraction of the sum of the drives' capacities that is available for use. For example, if three drives are arranged in RAID 3, this gives an array space efficiency of $1 - 1/n = 1 - 1/3 = 2/3 \approx 67\%$; thus, if each drive in this example has a capacity of 250 GB, then the array has a total capacity of 750 GB but the capacity that is usable for data storage is only 500 GB.

Level	Description	Minimum number of drives[b]	Space efficiency	Fault tolerance	Read performance	Write performance
					as factor of single disk	
RAID 0	Block-level striping without parity or mirroring	2	1	None	n	n
RAID 1	Mirroring without parity or striping	2	$\frac{1}{n}$	$n - 1$ drive failures	n[a][15]	1[c][15]
RAID 2	Bit-level striping with Hamming code for error correction	3	$1 - \frac{1}{n} \log_2 (n + 1)$	One drive failure[d]	Depends	Depends
RAID 3	Byte-level striping with dedicated parity	3	$1 - \frac{1}{n}$	One drive failure	$(n - 1)$	$(n - 1)$[e]
RAID 4	Block-level striping with dedicated parity	3	$1 - \frac{1}{n}$	One drive failure	$(n - 1)$	$(n - 1)$[e]
RAID 5	Block-level striping with distributed parity	3	$1 - \frac{1}{n}$	One drive failure	n[e]	single sector: $\frac{1}{4}$ full stripe: $(n - 1)$[e]
RAID 6	Block-level striping with double distributed parity	4	$1 - \frac{2}{n}$	Two drive failures	n[e]	single sector: $\frac{1}{6}$ full stripe: $(n - 2)$[e]

System Implications

In measurement of the I/O performance of five filesystems with five storage configurations—single SSD, RAID 0, RAID 1, RAID 10, and RAID 5 it was shown that F2FS on RAID 0 and RAID 5 with eight SSDs outperforms EXT4 by 5 times and 50 times, respectively. The measurements also suggest that RAID controller can be a significant bottleneck in building a RAID system with high speed SSDs.[27]

Nested RAID

Combinations of two or more standard RAID levels. They are also known as RAID 0+1 or RAID 01, RAID 0+3 or RAID 03, RAID 1+0 or RAID 10, RAID 5+0 or RAID 50, RAID 6+0 or RAID 60, and RAID 10+0 or RAID 100.

Non-standard variants

In addition to standard and nested RAID levels, alternatives include non-standard RAID levels, and non-RAID drive architectures. Non-RAID drive architectures are referred to by similar terms and acronyms, notably JBOD ("just a bunch of disks"), SPAN/BIG, and MAID ("massive array of idle disks").

Notes

a. Theoretical maximum, as low as single-disk performance in practice

b. Assumes a non-degenerate minimum number of drives

c. If disks with different speeds are used in a RAID 1 array, overall write performance is equal to the speed of the slowest disk.

d. RAID 2 can recover from one drive failure or repair corrupt data or parity when a corrupted bit's corresponding data and parity are good.

e. Assumes hardware capable of performing associated calculations fast enough

References

1. "Common raid Disk Data Format (DDF)" (http://www.snia.org/tech_activities/standards/curr_standards/ddf/). *SNIA.org*. Storage Networking Industry Association. Retrieved 2013-04-23.
2. "RAID 0 Data Recovery" (https://www.datarecovery.net/RAID/raid-0-data-recovery.html). *DataRecovery.net*. Retrieved 2015-04-30.
3. "Understanding RAID" (http://www.cru-inc.com/data-protection-topics/understanding-raid/). *CRU-Inc.com*. Retrieved 2015-04-30.
4. "How to Combine Multiple Hard Drives Into One Volume for Cheap, High-Capacity Storage" (http://lifehacker.com/5986883/how-to-combine-multiple-hard-drives-into-one-volume-for-cheap-high-capacity-storage). *LifeHacker.com*. 2013-02-26. Retrieved 2015-04-30.
5. Chen, Peter; Lee, Edward; Gibson, Garth; Katz, Randy; Patterson, David (1994). "RAID: High-Performance, Reliable Secondary Storage". *ACM Computing Surveys*. **26** (2): 145–185. CiteSeerX 10.1.1.41.3889 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.3889). doi:10.1145/176979.176981 (https://doi.org/10.1145%2F176979.176981).
6. de Kooter, Sebastiaan (2015-04-13). "Gaming storage shootout 2015: SSD, HDD or RAID 0, which is best?" (http://www.gameplayinside.com/optimize/gaming-storage-shootout-2015-ssd-hdd-or-raid-0-which-is-best/). *GamePlayInside.com*. Retrieved 2015-09-22.
7. "Western Digital's Raptors in RAID-0: Are two drives better than one?" (http://www.anandtech.com/storage/showdoc.aspx?i=2101). *AnandTech.com*. AnandTech. July 1, 2004. Retrieved 2007-11-24.
8. "Hitachi Deskstar 7K1000: Two Terabyte RAID Redux" (http://www.anandtech.com/storage/showdoc.aspx?i=2974). *AnandTech.com*. AnandTech. April 23, 2007. Retrieved 2007-11-24.
9. "RAID 0: Hype or blessing?" (https://tweakers.net/reviews/515/1/raid-0-hype-or-blessing-pagina-1.html). *Tweakers.net*. Persgroep Online Services. August 7, 2004. Retrieved 2008-07-23.
10. "Does RAID0 Really Increase Disk Performance?" (http://www.hardwaresecrets.com/does-raid0-really-increase-disk-performance/). *HardwareSecrets.com*. November 1, 2006.
11. Larabel, Michael (2014-10-22). "Btrfs RAID HDD Testing on Ubuntu Linux 14.10" (https://www.phoronix.com/scan.php?page=article&item=btrfs_raid01_linux316&num=1). Phoronix. Retrieved 2015-09-19.
12. Larabel, Michael (2014-10-29). "Btrfs on 4 × Intel SSDs In RAID 0/1/5/6/10" (https://www.phoronix.com/scan.php?page=article&item=btrfs_4way_ssdraid&num=1). Phoronix. Retrieved 2015-09-19.
13. "FreeBSD Handbook: 19.3. RAID 1 – Mirroring" (http://www.freebsd.org/doc/handbook/geom-mirror.html). *FreeBSD.org*. 2014-03-23. Retrieved 2014-06-11.
14. "Which RAID Level is Right for Me?: RAID 1 (Mirroring)" (http://www.adaptec.com/en-us/_common/compatibility/_education/raid_level_compar_wp.htm#2.2). *Adaptec.com*. Adaptec. Retrieved 2014-01-02.
15. "Selecting the Best RAID Level: RAID 1 Arrays (Sun StorageTek SAS RAID HBA Installation Guide)" (https://docs.oracle.com/cd/E19691-01/820-1847-20/appendixf.html#50515995_74175). *Docs.Oracle.com*. Oracle Corporation. 2010-12-23. Retrieved 2014-01-02.
16. Vadala, Derek (2003). *Managing RAID on Linux. O'Reilly Series* (https://books.google.com/?id=RM4tahggCVcC&pg=PA6&dq=raid+2+implementation#v=onepage&q=raid%202%20implementation) (illustrated ed.). O'Reilly. p. 6. ISBN 9781565927308.

17. Marcus, Evan; Stern, Hal (2003). *Blueprints for high availability* (https://books.google.com/?id=D_jYqFoJVEAC&pg=RA2-PA167&dq=raid+2+implementation#v=onepage&q=raid%202%20implementation) (2, illustrated ed.). John Wiley and Sons. p. 167. ISBN 9780471430261.
18. The RAIDbook, 4th Edition, The RAID Advisory Board, June 1995, p.101
19. Meyers, Michael; Jernigan, Scott (2003). *Mike Meyers' A+ Guide to Managing and Troubleshooting PCs* (https://books.google.com/?id=9vfQKUT_BjgC&pg=PT348&dq=raid+2+implementation#v=onepage&q=raid%202%20implementation) (illustrated ed.). McGraw-Hill Professional. p. 321. ISBN 9780072231465.
20. Natarajan, Ramesh (2011-11-21). "RAID 2, RAID 3, RAID 4 and RAID 6 Explained with Diagrams" (http://www.thegeekstuff.com/2011/11/raid2-raid3-raid4-raid6/). *TheGeekStuff.com*. Retrieved 2015-01-02.
21. "RAID 5 Data Recovery FAQ" (http://www.vantagetech.com/faq/raid-5-recovery-faq.html). *VantageTech.com*. Vantage Technologies. Retrieved 2014-07-16.
22. Koren, Israel. "Basic RAID Organizations" (http://www.ecs.umass.edu/ece/koren/architecture/Raid/basicRAID.html). *ECS.UMass.edu*. University of Massachusetts. Retrieved 2014-11-04.
23. "Sun StorageTek SAS RAID HBA Installation Guide, Appendix F: Selecting the Best RAID Level: RAID 6 Arrays" (https://docs.oracle.com/cd/E19494-01/820-1260-15/appendixf.html#50548797_51002). *Docs.Oracle.com*. 2010-12-23. Retrieved 2015-08-27.
24. "Dictionary R" (http://www.snia.org/education/dictionary/r/). *SNIA.org*. Storage Networking Industry Association. Retrieved 2007-11-24.
25. Faith, Rickard E. (13 May 2009). "A Comparison of Software RAID Types" (http://alephnull.com/benchmarks/sata2009/raidtype.html).
26. Anvin, H. Peter (May 21, 2009). "The Mathematics of RAID-6" (https://www.kernel.org/pub/linux/kernel/people/hpa/raid6.pdf) (PDF). *Kernel.org*. Linux Kernel Organization. Retrieved November 4, 2009.
27. Park, Chanhyun; Lee, Seongjin; Won, Youjip (2014). *An Analysis on Empirical Performance of SSD-Based RAID* (https://scholar.google.co.kr/scholar?cluster=1955840554275208892). *Information Sciences and Systems*. **2014**. pp. 395–405. doi:10.1007/978-3-319-09465-6_41 (https://doi.org/10.1007%2F978-3-319-09465-6_41). ISBN 978-3-319-09464-9.

Further reading

- "Learning About RAID" (https://web.archive.org/web/20090220092914/http://support.dell.com/support/topics/global.aspx/support/entvideos/raid?c=us&l=en&s=gen). *Support.Dell.com*. Dell. 2009. Archived from the original (http://support.dell.com/support/topics/global.aspx/support/entvideos/raid?c=us&l=en&s=gen) on 2009-02-20. Retrieved 2016-04-15.
- Redundant Arrays of Inexpensive Disks (RAIDs) (http://pages.cs.wisc.edu/~remzi/OSTEP/file-raid.pdf), chapter 38 from the *Operating Systems: Three Easy Pieces* book by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau

External links

- IBM summary on RAID levels (https://www.ibm.com/support/knowledgecenter/POWER8/p8ebk/raidlevelsummary.htm)
- RAID 5 parity explanation and checking tool (http://www.dtidata.com/resourcecenter/2008/05/08/raid-configuration-parity-check/)
- RAID Calculator for Standard RAID Levels and Other RAID Tools (https://www.icc-usa.com/raid-calculator/)
- Sun StorEdge 3000 Family Configuration Service 2.5 User’s Guide: RAID Basics (https://docs.oracle.com/cd/E19168-01/817-3337-18/appa_raid_basic.html)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Standard_RAID_levels&oldid=921902072"

This page was last edited on 18 October 2019, at 17:07 (UTC).