

Standard RAID levels

From Wikipedia, the free encyclopedia

The **standard RAID levels** are a basic set of RAID configurations that employ the techniques of striping, mirroring, or parity to create large reliable data stores from general purpose computer hard disk drives. The most common types today are RAID 0 (striping), RAID 1 and variants (mirroring), RAID 5 (distributed parity) and RAID 6 (dual parity). RAID levels and their associated data formats are standardized by the Storage Networking Industry Association in the Common RAID Disk Drive Format (DDF) standard.^[1]

Contents

- 1 RAID 0
 - 1.1 Performance
- 2 RAID 1
- 3 RAID 2
- 4 RAID 3
- 5 RAID 4
- 6 RAID 5
- 7 RAID 6
 - 7.1 Performance (speed)
 - 7.2 Implementation
 - 7.2.1 Computing parity
- 8 Comparison
- 9 Non-standard RAID levels and non-RAID drive architectures
- 10 Notes
- 11 References
- 12 External links

RAID 0

A **RAID 0** (also known as a *stripe set* or *striped volume*) splits data evenly across two or more disks (striped) without parity information for speed. RAID 0 was not one of the original RAID levels and provides no data redundancy. RAID 0 is normally used to increase performance, although it can also be used as a way to create a large logical disk out of two or more physical ones.

A RAID 0 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 100 GB disk is striped together with a 350 GB disk, the size of the array will be 200 GB (100 GB × 2).

$$\begin{aligned}\text{Size} &= 2 \cdot \min(100\text{ GB}, 350\text{ GB}) \\ &= 2 \cdot 100\text{ GB} \\ &= 200\text{ GB}\end{aligned}$$

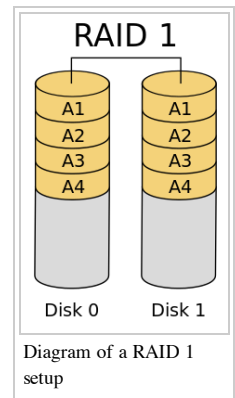
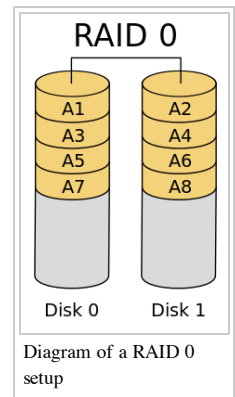
The diagram shows how the data is distributed into A*x* stripes to the disks. Accessing the stripes in the order A1, A2, A3, ... provides the illusion of a larger and faster drive. Once the stripe size is defined on creation it needs to be maintained at all times.

Performance

RAID 0 is also used in areas where performance is desired and data integrity is not very important, for example in some computer gaming systems. However, real-world tests with computer games have shown that RAID-0 performance gains are minimal, although some desktop applications will benefit.^{[2][3]} Another article examined these claims and concluded: "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."^[4]

RAID 1

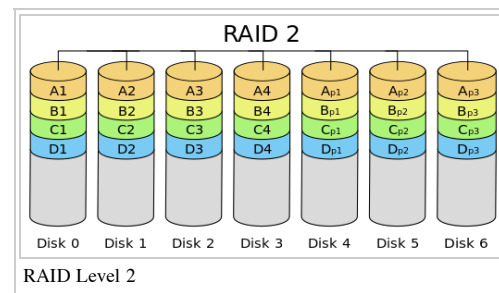
An exact copy (or *mirror*) of a set of data on two disks. This is useful when read performance or reliability is more important than data storage capacity. Such an array can only be as big as the smallest member disk. A classic RAID 1 mirrored pair contains two disks (see reliability geometrically) over a single disk.



RAID 2

A **RAID 2** 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. Extremely high data transfer rates are possible. This is the only original level of RAID that is not currently used.^{[5][6]}

All hard disks eventually implemented Hamming code error correction. This made RAID 2 error correction redundant and unnecessarily complex. This level quickly became useless and is now obsolete. There are no commercial applications of RAID 2.^{[5][6]}

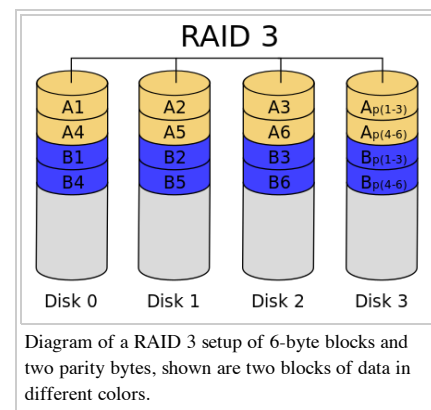


RAID 3

A **RAID 3** uses byte-level striping with a dedicated parity disk. RAID 3 is very rare in practice. One of the characteristics of RAID 3 is that it generally cannot service multiple requests simultaneously. This happens because any single block of data will, by definition, be spread across all members of the set and will reside in the same location. So, 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.^[6]

The requirement that all disks spin synchronously, a.k.a. lockstep, added design considerations to a level that didn't give significant advantages over other RAID levels, so it quickly became useless and is now obsolete.^[5] Both RAID 3 and RAID 4 were quickly replaced by RAID 5.^[7] RAID 3 was usually implemented in hardware, and the performance issues were addressed by using large disk caches.^[6]

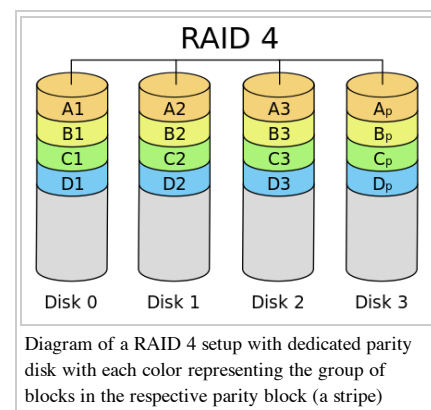


RAID 4

A **RAID 4** uses block-level striping with a dedicated parity disk.

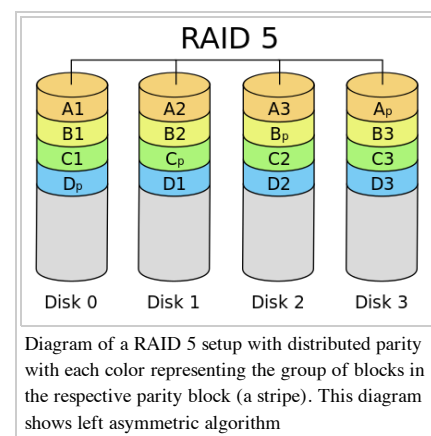
In the example on the right, a read request for block A1 would be serviced by disk 0. A simultaneous read request for block B1 would have to wait, but a read request for B2 could be serviced concurrently by disk 1.

RAID 4 is very uncommon, but one enterprise level company that has previously used it is NetApp. The aforementioned performance problems were solved with their proprietary Write Anywhere File Layout (WAFL), an approach to writing data to disk locations that minimizes the conventional parity RAID write penalty. By storing system metadata (inodes, block maps, and inode maps) in the same way application data is stored, WAFL is able to write file system metadata blocks anywhere on the disk. This approach in turn allows multiple writes to be "gathered" and scheduled to the same RAID stripe—eliminating the traditional read-modify-write penalty prevalent in parity-based RAID schemes.^[8]



RAID 5

A **RAID 5** comprises 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. RAID 5 requires at least three disks.^[9]



RAID 6

RAID 6 extends RAID 5 by adding an additional parity block; thus it uses block-level striping with two parity blocks distributed across all member disks.

Performance (speed)

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. It can be as fast as a RAID-5 system with one fewer drive (same number of data drives).^[10]

Implementation

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."^[11]

Computing parity

Two different *syndromes* need to be computed in order to allow the loss of any two drives. One of them, **P** can be the simple XOR of the data across the stripes, as with RAID 5. A second, independent syndrome is more complicated and requires the assistance of field theory.

To deal with this, the Galois field $GF(m)$ is introduced with $m = 2^k$, where $GF(m) \cong F_2[x]/(p(x))$ for a suitable irreducible polynomial $p(x)$ of degree k . A chunk of data can be written as $d_{k-1}d_{k-2}...d_0$ in base 2 where each d_i is either 0 or 1. This is chosen to correspond with the element $d_{k-1}x^{k-1} + d_{k-2}x^{k-2} + ... + d_1x + d_0$ in the Galois field. Let $D_0, ..., D_{n-1} \in GF(m)$ correspond to the stripes of data across hard drives encoded as field elements in this manner (in practice they would probably be broken into byte-sized chunks). If g is some generator of the field and \oplus denotes addition in the field while concatenation denotes multiplication, then **P** and **Q** may be computed as follows (n denotes the number of data disks):

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

$$\mathbf{Q} = \bigoplus_i g^i D_i = g^0 D_0 \oplus g^1 D_1 \oplus g^2 D_2 \oplus ... \oplus g^{n-1} D_{n-1}$$

For a computer scientist, a good way to think about this is that \oplus is a bitwise XOR operator and g^i is the action of a linear feedback shift register on a chunk of data. Thus, in the formula above,^[12] the calculation of **P** is just the XOR of each stripe. This is because addition in any characteristic two finite field reduces to the XOR operation. The computation of **Q** is the XOR of a shifted version of each stripe.

Mathematically, the *generator* is an element of the field such that g^i is different for each nonnegative i satisfying $i < n$.

If one data drive is lost, the data can be recomputed from **P** just like with RAID 5. If two data drives are lost or a data drive and the drive containing **P** are lost, the data can be recovered from **P** and **Q** or from just **Q**, respectively, using a more complex process. Working out the details is extremely hard with field theory. Suppose that D_i and D_j are the lost values with $i \neq j$. Using the other values of D , constants A and B may be found so that $D_i \oplus D_j = A$ and $g^i D_i \oplus g^j D_j = B$:

$$A = \bigoplus_{\ell: \ell \neq i \text{ and } \ell \neq j} D_\ell = \mathbf{P} \oplus D_0 \oplus D_1 \oplus ... \oplus D_{i-1} \oplus D_{i+1} \oplus ... \oplus D_{j-1} \oplus D_{j+1} \oplus ... \oplus D_{n-1}$$

$$B = \bigoplus_{\ell: \ell \neq i \text{ and } \ell \neq j} g^\ell D_\ell = \mathbf{Q} \oplus g^0 D_0 \oplus g^1 D_1 \oplus ... \oplus g^{i-1} D_{i-1} \oplus g^{i+1} D_{i+1} \oplus ... \oplus g^{j-1} D_{j-1} \oplus g^{j+1} D_{j+1} \oplus ... \oplus g^{n-1} D_{n-1}$$

Multiplying both sides of the equation for B by g^{n-i} and adding to the former equation yields $(g^{n-i+j} \oplus 1)D_j = g^{n-i}B \oplus A$ and thus a solution for D_j , which may be used to compute D_i .

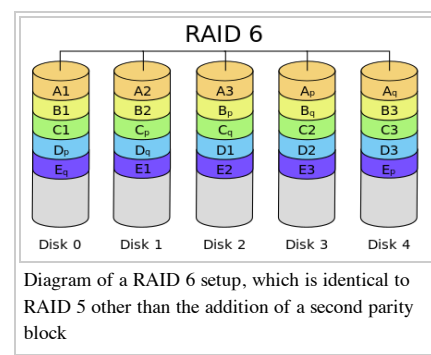
The computation of **Q** is CPU intensive compared to the simplicity of **P**. Thus, a RAID 6 implemented in software will have a more significant effect on system performance, and a hardware solution will be more complex.

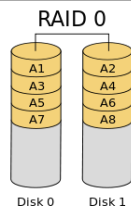
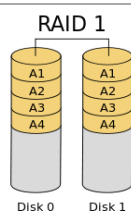
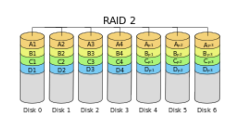
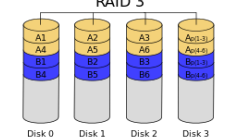
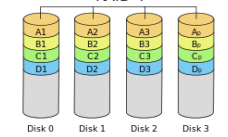
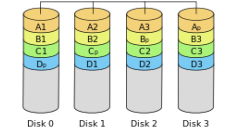
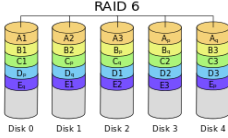
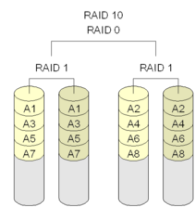
Comparison

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.
- Array failure rate is given as an expression in terms of the number of drives, n , and the drive failure rate, r (which is assumed identical and independent for each drive) and can be seen to be a Bernoulli trial. For example, if each of three drives has a failure rate of 5% over the next three years, and these drives are arranged in RAID 3, then this gives an array failure rate over the next three years of:

$$\begin{aligned} 1 - (1 - r)^n - nr(1 - r)^{n-1} &= 1 - (1 - 5\%)^3 - 3 \times 5\% \times (1 - 5\%)^{3-1} \\ &= 1 - 0.95^3 - 0.15 \times 0.95^2 \\ &= 1 - 0.857375 - 0.135375 \\ &= 0.00725 \\ &\approx 0.7\% \end{aligned}$$



Level	Description	Minimum # of drives ^[a]	Space efficiency	Fault tolerance	Array failure rate ^[b]	Read performance	Write performance	Figure
RAID 0	Block-level striping without parity or mirroring	2	1	0 (none)	$1 - (1 - r)^n$	n×	n×	
RAID 1	Mirroring without parity or striping	2	1/n	n-1 drives	r^n	n× ^[c]	1×	
RAID 2	Bit-level striping with dedicated Hamming-code parity	3	$1 - 1/n \cdot \log_2(n-1)$	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.	(Varies)	(Varies)	(Varies)	
RAID 3	Byte-level striping with dedicated parity	3	$1 - 1/n$	1 drive	$1 - (1 - r)^n - nr(1 - r)^{n-1}$	(n-1)×	(n-1)× ^[d]	
RAID 4	Block-level striping with dedicated parity	3	$1 - 1/n$	1 drive	$1 - (1 - r)^n - nr(1 - r)^{n-1}$	(n-1)×	(n-1)× ^[d]	
RAID 5	Block-level striping with distributed parity	3	$1 - 1/n$	1 drive	$1 - (1 - r)^n - nr(1 - r)^{n-1}$	(n-1)× ^[d]	(n-1)× ^[d]	
RAID 6	Block-level striping with double distributed parity	4	$1 - 2/n$	2 drives	$1 - (1 - r)^n - nr(1 - r)^{n-1} - \binom{n}{2}r^2(1 - r)^{n-2}$	(n-2)× ^[d]	(n-2)× ^[d]	
RAID 1+0	Mirroring without parity, and block-level striping	4	2/n	1 drive / span ^[e]		(n/spans)×	(n/spans)×	
Level	Description	Minimum # of	Space efficiency	Fault tolerance	Array failure rate ^[b]	Read performance	Write performance	Figure


Non-standard RAID levels and non-RAID drive architectures

Alternatives to the above designs include nested RAID levels, non-standard RAID levels, and non-RAID drive architectures. Non-RAID drive architectures are referred to by similar acronyms, notably SLED, Just a Bunch of Disks, SPAN/BIG, and MAID.

Notes

- a. $\wedge a \wedge b$ Assumes a non-degenerate minimum number of drives
- b. $\wedge a \wedge b$ Assumes independent, identical rate of failure amongst drives
- c. \wedge Theoretical maximum, as low as 1x in practice
- d. $\wedge a \wedge b \wedge c \wedge d \wedge e \wedge f$ Assumes hardware is fast enough to support
- e. \wedge RAID 10 can only lose one drive per span up to the max of $n/2$ drives.

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. [^] "Western Digital's Raptors in RAID-0: Are two drives better than one?" (<http://www.anandtech.com/storage/showdoc.aspx?i=2101>). AnandTech. July 1, 2004. Retrieved 2007-11-24.
3. [^] "Hitachi Deskstar 7K1000: Two Terabyte RAID Redux" (<http://www.anandtech.com/storage/showdoc.aspx?i=2974>). AnandTech. April 23, 2007. Retrieved 2007-11-24.
4. [^] "RAID 0: Hype or blessing?" (<http://tweakers.net/reviews/515/1/raid-0-hype-or-blessing-pagina-1.html>). Tweakers.net. August 7, 2004. Retrieved 2008-07-23.
5. [^] ^{*a b c*} Derek Vadala (2003). *Managing RAID on Linux. O'Reilly Series* (<http://books.google.com/?id=RM4tahggCVc&pg=PA6&dq=raid+2+implementation#v=onepage&q=raid%20%20implementation>) (illustrated ed.). O'Reilly. p. 6. ISBN 9781565927308.
6. [^] ^{*a b c d*} Evan Marcus, Hal Stern (2003). *Blueprints for high availability* (http://books.google.com/?id=D_jYqFoJVEAC&pg=RA2-PA167&dq=raid+2+implementation#v=onepage&q=raid%20%20implementation) (2, illustrated ed.). John Wiley and Sons. p. 167. ISBN 9780471430261.
7. [^] Michael Meyers, Scott Jernigan (2003). *Mike Meyers' A+ Guide to Managing and Troubleshooting PCs* (http://books.google.com/?id=9vfQKUT_BJgC&pg=PT348&dq=raid+2+implementation#v=onepage&q=raid%20%20implementation) (illustrated ed.). McGraw-Hill Professional. p. 321. ISBN 9780072231465.
8. [^]  "NetApp DNA" (http://partners.netapp.com/go/techontap/matl/NetApp_DNA.html). (password-protected)
9. [^] Chen, Peter; Lee, Edward; Gibson, Garth; Katz, Randy; Patterson, David (1994). "RAID: High-Performance, Reliable Secondary Storage" (<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.3889>). *ACM Computing Surveys* **26**: 145–185.
10. [^] Rickard E. Faith (13 May 2009). *A Comparison of Software RAID Types* (<http://alephnull.com/benchmarks/sata2009/raidtype.html>).
11. [^] "Dictionary R" (<http://www.snia.org/education/dictionary/r/>). Storage Networking Industry Association. Retrieved 2007-11-24.
12. [^] Anvin, H. Peter (21 May 2009). "The mathematics of RAID-6" (<http://www.kernel.org/pub/linux/kernel/people/hpa/raid6.pdf>). Retrieved November 4, 2009.

External links

- RAID Calculator for Standard RAID Levels and Other RAID Tools (<http://www.icc-usa.com/raid-calculator.asp>)
- IBM summary on RAID levels (<http://www-1.ibm.com/support/docview.wss?uid=swg21149421>)
- RAID 5 parity explanation and checking tool (<http://www.dtdata.com/resourcecenter/2008/05/08/raid-configuration-parity-check/>)
- Animations and details on RAID levels 0, 1, and 5 (<http://support.dell.com/support/topics/global.aspx/support/entvideos/raid?c=us&l=en&s=gen>)
- The Open-E Blog, "How does RAID 5 work? The Shortest and Easiest explanation ever!" (<http://blog.open-e.com/how-does-raid-5-work/>)

Retrieved from "http://en.wikipedia.org/w/index.php?title=Standard_RAID_levels&oldid=609377163"

Categories: RAID

- This page was last modified on 20 May 2014 at 13:00.
 - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy.
- Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.