**RAID 0 : Striped Disk Array without Fault Tolerance**

|  |  |
| --- | --- |
| http://www.aaa-datarecovery.com/images/raid0.gif | |
| RAID Level 0 requires a minimum of 2 drives to implement | |
| http://www.aaa-datarecovery.com/images/advantage.gifAdvantages | http://www.aaa-datarecovery.com/images/disadvantage.gifDisadvantages |
| RAID 0 implements a striped disk array, the data is broken down into blocks and each block is written to a separate disk drive  No parity calculation overhead is involved  Very simple design  Easy to implement | Not a 'True' RAID because it is NOT fault-tolerant  The failure of just one drive will result in all data in an array being lost  Should never be used in mission critical environments |
| A **RAID 0** (also known as a **striped set**) splits data evenly across two or more disks with no parity information for redundancy. It is important to note that RAID 0 was not one of the original RAID levels, and is not redundant. RAID 0 is normally used to increase performance, although it can also be used as a way to create a small number of large virtual disks out of a large number of small 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 120 GB disk is striped together with a 100 GB disk, the size of the array will be 200 GB. Although RAID 0 was not specified in the original RAID paper, an idealized implementation of RAID 0 would split I/O operations into equal-sized blocks and spread them evenly across two disks. RAID 0 implementations with more than two disks are also possible, however the reliability of a given RAID 0 set is equal to the average reliability of each disk divided by the number of disks in the set. That is, reliability (as measured by [mean time to failure](http://en.wikipedia.org/wiki/Failure_rate) (MTTF) or [mean time between failures](http://en.wikipedia.org/wiki/Failure_rate) (MTBF)) is roughly inversely proportional to the number of members—so a set of two disks is roughly half as reliable as a single disk. The reason for this is that the [file system](http://en.wikipedia.org/wiki/File_system) is distributed across all disks. When a drive fails the file system cannot cope with such a large loss of data and coherency since the data is "striped" across all drives. Data can be recovered using special tools. However, it will be incomplete and most likely corrupt.  While the block size can technically be as small as a byte it is almost always a multiple of the hard disk sector size of 512 bytes. This lets each drive seek independently when randomly reading or writing data on the disk. If all the accessed sectors are entirely on one disk then the apparent seek time would be the same as a single disk. If the accessed sectors are spread evenly among the disks then the apparent seek time would be reduced by half for two disks, by two-thirds for three disks, etc. assuming identical disks. For normal data access patterns the apparent seek time of the array would be between these two extremes. The transfer speed of the array will be the transfer speed of all the disks added together.  RAID 0 is useful for set-ups such as large [read-only](http://en.wikipedia.org/wiki/Read-only) [NFS](http://en.wikipedia.org/wiki/Network_File_System) [servers](http://en.wikipedia.org/wiki/Servers) where [mounting](http://en.wikipedia.org/wiki/Mount) many disks is time-consuming or impossible and redundancy is irrelevant. Another use is where the number of disks is limited by the [operating system](http://en.wikipedia.org/wiki/Operating_system). In [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), the number of drive letters for hard disk drives may be limited to 24, so RAID 0 is a popular way to use more than this many disks. It is also a popular choice for gaming systems where performance is desired. However, since data is shared between drives without redundancy, hard drives cannot be swapped out as all disks are dependent upon each other.  Also with RAID 0 if one drive failed you may have lost a lot more than if you had your data spread across separate drives as files with bits missing can be of very limited usefulness and you cannot copy important data manually to multiple drives (unless you have multiple separate arrays).  **Concatenation (JBOD)**  Although a **concatenation** of disks (also called **JBOD**, or "Just a Bunch of Disks") is not one of the numbered RAID levels, it is a popular method for combining multiple physical disk drives into a single virtual one. As the name implies, disks are merely [concatenated](http://en.wikipedia.org/wiki/Concatenate) together, end to beginning, so they appear to be a single large disk.  In this sense, concatenation is akin to the reverse of [partitioning](http://en.wikipedia.org/wiki/Drive_partition). Whereas partitioning takes one physical drive and creates two or more logical drives, JBOD uses two or more physical drives to create one logical drive.  In that it consists of an Array of Inexpensive Disks (no redundancy), it can be thought of as a distant relation to RAID. JBOD is sometimes used to turn several odd-sized drives into one useful drive. Therefore, JBOD could use a 3 GB, 15 GB, 5.5 GB, and 12 GB drive to combine into a logical drive at 35.5 GB, which is often more useful than the individual drives separately.  One advantage JBOD has over RAID 0 is in the case of drive failure. Whereas in RAID 0, failure of a single drive will usually result in the loss of all data in the array, in a JBOD array only the data on the affected drive is lost, and the data on surviving drives will remain readable. | |

**RAID 1 : Mirroring and Duplexing**

|  |  |
| --- | --- |
| http://www.aaa-datarecovery.com/images/raid1.gif | |
| RAID Level 1 requires a minimum of 2 drives to implement | |
| http://www.aaa-datarecovery.com/images/advantage.gifAdvantages | http://www.aaa-datarecovery.com/images/disadvantage.gifDisadvantages |
| One Write or two Reads possible per mirrored pair  100% redundancy of data means no rebuild is necessary in case of a disk failure, just a copy to the replacement disk  Simplest RAID storage subsystem design | Highest disk overhead of all RAID types (100%) - inefficient  Typically the RAID function is done by system software, loading the CPU/Server and possibly degrading throughput at high activity levels. Hardware implementation is strongly recommended. |
| **RAID 1** creates an exact copy (or **mirror**) of a set of data on two or more disks. This is useful for set-ups where [redundancy](http://en.wikipedia.org/wiki/Redundancy) is more important than using all the disks' maximum [storage](http://en.wikipedia.org/wiki/Storage) [capacity](http://en.wikipedia.org/wiki/Capacity). The array can only be as big as the smallest member disk, however. An ideal RAID 1 set contains two disks, which increases reliability by a factor of two over a single disk, but it is possible to have many more than two copies. Since each member can be addressed independently if the other fails, reliability is a linear multiple of the number of members. To truly get the full redundancy benefits of RAID 1, independent disk controllers are recommended, one for each disk. Some refer to this practice as **splitting** or **duplexing.**  When reading both disks can be accessed independently. Like RAID 0 the average seek time is reduced by half when randomly reading but because each disk has the exact same data the requested sectors can always be split evenly between the disks and the seek time remains low. The transfer rate would also be doubled. For three disks the seek time would be a third and the transfer rate would be tripled. The only limit is how many disks can be connected to the controller and its maximum transfer speed. Most IDE RAID 1 cards use a broken implementation and only read from one disk so their read performance is that of a single disk. Some older RAID 1 implementations would read both disks simultaneously and compare the data to catch errors. The error detection and correction on modern disks makes this no longer necessary. When writing, the array acts like a single disk as all writes must be written to all disks.  RAID1 has many administrative advantages. For instance, in some 365\*24 environments, it is possible to "Split the Mirror": declare one disk as inactive, do a backup of that disk, and then "rebuild" the mirror. This procedure is less critical in the presence of the "snapshot" feature of some filesystems, in which some space is reserved for changes, presenting a static point-in-time view of the filesystem. Alternatively, a set of disks can be kept in much the same way as traditional backup tapes are.  Also, one common practice is to create an extra mirror of a volume (also known as a **Business Continuance Volume** or **BCV**) which is meant to be split from the source RAID set and used independently. In some implementations, these extra mirrors can be split and then incrementally re-established, instead of requiring a complete RAID set rebuild. | |

**RAID 2 : Hamming Code ECC**

|  |  |
| --- | --- |
| http://www.aaa-datarecovery.com/images/raid2.gif | |
| Each bit of data word is written to a data disk drive (4 in this example: 0 to 3). Each data word has its Hamming Code ECC word recorded on the ECC disks. On Read, the ECC code verifies correct data or corrects single disk errors. | |
| http://www.aaa-datarecovery.com/images/advantage.gifAdvantages | http://www.aaa-datarecovery.com/images/disadvantage.gifDisadvantages |
| 'On the fly' data error correction  Extremely high data transfer rates possible  The higher the data transfer rate required, the better the ratio of data disks to ECC disks | Very high ratio of ECC disks to data disks with smaller word sizes - inefficient  Entry level cost very high - requires very high transfer rate requirement to justify.  No commercial implementations exist. |

**RAID 3 : Parallel transfer with parity**

|  |  |
| --- | --- |
| http://www.aaa-datarecovery.com/images/raid3.gif | |
| The data block is subdivided ('striped') and written on the data disks. Stripe parity is generated on Writes, recorded on the parity disk and checked on Reads. RAID Level 3 requires a minimum of 3 drives to implement | |
| http://www.aaa-datarecovery.com/images/advantage.gifAdvantages | http://www.aaa-datarecovery.com/images/disadvantage.gifDisadvantages |
| Very high Read data transfer rate  Very high Write data transfer rate  Disk failure has an insignificant impact on throughput  Low ratio of ECC (Parity) disks to data disks means high efficiency | Transaction rate equal to that of a single disk drive at best (if spindles are synchronized)  Controller design is fairly complex  Very difficult and resource intensive to do as a 'software' RAID |

**RAID 4 : Independent Data disks with shared Parity disk**

|  |  |
| --- | --- |
| http://www.aaa-datarecovery.com/images/raid4.gif | |
| Each entire block is written onto a data disk. Parity for same rank blocks is generated on Writes, recorded on the parity disk and checked on Reads. RAID Level 4 requires a minimum of 3 drives to implement | |
| http://www.aaa-datarecovery.com/images/advantage.gifAdvantages | http://www.aaa-datarecovery.com/images/disadvantage.gifDisadvantages |
| Very high Read data transaction rate  Low ratio of ECC (Parity) disks to data disks means high efficiency  High aggregate Read transfer rate  Low ratio of ECC (Parity) disks to data disks means high efficiency | Quite complex controller design  Worst Write transaction rate and Write aggregate transfer rate  Difficult and inefficient data rebuild in the event of disk failure  Block Read transfer rate equal to that of a single disk |

**RAID 5 : Independent Data disks with distributed parity blocks**

|  |  |
| --- | --- |
| http://www.aaa-datarecovery.com/images/raid5.gif | |
| Each entire data block is written on a data disk; parity for blocks in the same rank is generated on Writes, recorded in a distributed location and checked on Reads.RAID Level 5 requires a minimum of 3 drives to implement | |
| http://www.aaa-datarecovery.com/images/advantage.gifAdvantages | http://www.aaa-datarecovery.com/images/disadvantage.gifDisadvantages |
| Highest Read data transaction rate Medium Write data transaction rate Low ratio of ECC (Parity) disks to data disks means high efficiency Good aggregate transfer rate | Disk failure has a medium impact on throughput Most complex controller design Difficult to rebuild in the event of a disk failure (as compared to RAID level 1) Individual block data transfer rate same as single disk |
| A **RAID 5** uses [block](http://en.wikipedia.org/wiki/Block)-level striping with [parity](http://en.wikipedia.org/wiki/Parity) data distributed across all member disks. RAID 5 is one of the most popular RAID levels, and is frequently used in both hardware and software implementations. Virtually all [storage arrays](http://en.wikipedia.org/wiki/Storage_array) offer RAID 5. As with RAID 0, RAID 5 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 used to build a RAID 5 together with two 100 GB disks, each disk will donate 100 GB to the array for a total of 200 GB of storage. 100 GB are used for parity information, and the excess 20 GB from the larger disk are ignored.  Every time a data "block" (sometimes called a "chunk") is written on a disk in an array, a parity block is generated within the same stripe. (A block or chunk is often composed of many consecutive sectors on a disk, sometimes as many as 256 sectors. A series of chunks [a chunk from each of the disks in an array] is collectively called a "stripe".) If another block, or some portion of a block is written on that same stripe, the parity block (or some portion of the parity block) is recalculated and rewritten. The disk used for the parity block is staggered from one stripe to the next, hence the term "distributed parity blocks". This means, of course, that the controller software becomes more complex.  Interestingly, the parity blocks are not read on data reads, since this would be unnecessary overhead and would diminish performance. The parity blocks are read, however, when a read of a data sector results in a [cyclic redundancy check](http://en.wikipedia.org/wiki/Cyclic_redundancy_check) (CRC) error. In this case, the sector in the same relative position within each of the remaining data blocks in the stripe and within the parity block in the stripe are used to reconstruct the errant sector. The CRC error is thus hidden from the main computer. Likewise, should a disk fail in the array, the parity blocks from the surviving disks are combined mathematically with the data blocks from the surviving disks to reconstruct the data on the failed drive "on the fly".  This is sometimes called Interim Data Recovery Mode. The computer knows that a disk drive has failed, but this is only so that the operating system can notify the administrator that a drive needs replacement; applications running on the computer are unaware of the failure. Reading and writing to the drive array continues seamlessly, though with some performance degradation. The difference between RAID 4 and RAID 5 is that, in Interim data recovery mode, RAID 5 might be slightly faster than RAID 4, because, when the CRC and parity are in the disk that failed, the calculation does not have to be performed, while with RAID 4, if one of the data disks fails, the calculations have to be performed with each access.  In RAID 5, where there is only one parity block per stripe, the failure of a second drive results in total data loss. The maximum number of drives is theoretically unlimited, but it is common practice to keep the maximum to 14 or fewer for RAID 5 implementations which have only one parity block per stripe. The reason for this restriction is that there is a greater likelihood of two drives in an array failing in rapid succession when there is greater number of drives. As the number of disks in a RAID 5 increases, the MTBF for the array as a whole can even become lower than that of a single disk. This happens when the likelihood of a second disk failing out of (N-1) dependent disks, within the time it takes to detect, replace and recreate a first failed disk, becomes larger than the likelihood of a single disk failing.  One should be aware that many disks together increase heat, which lowers the real-world MTBF of each disk. Additionally, a group of disks bought at the same time may reach the end of their [bathtub curve](http://en.wikipedia.org/wiki/Failure_rate#Bathtub_curve) together, noticeably lowering the effective MTBF of the disks during that time.In implementations with greater than 14 drives, or in situations where extreme redundancy is needed, RAID 5 with dual parity (also known as RAID 6) is sometimes used, since it can survive the failure of two disks. | |

**RAID 6 : Independent Data disks with two independent distributed parity schemes**

|  |  |
| --- | --- |
| http://www.aaa-datarecovery.com/images/raid6.gif | |
| Each entire data block is written on a data disk; parity for blocks in the same rank is generated on Writes, recorded in a distributed location and checked on Reads.RAID Level 5 requires a minimum of 3 drives to implement | |
| http://www.aaa-datarecovery.com/images/advantage.gifAdvantages | http://www.aaa-datarecovery.com/images/disadvantage.gifDisadvantages |
| RAID 6 is essentially an extension of RAID level 5 which allows for additional fault tolerance by using a second independent distributed parity scheme (two-dimensional parity)  Data is striped on a block level across a set of drives, just like in RAID 5, and a second set of parity is calculated and written across all the drives; RAID 6 provides for an extremely high data fault tolerance and can sustain multiple simultaneous drive failures  Perfect solution for mission critical applications | - Very complex controller design - Controller overhead to compute parity addresses is extremely high - Very poor write performance - Requires N+2 drives to implement because of two-dimensional parity scheme |

**RAID 7 : Optimized Asynchrony for High I/O Rates as well as High Data Transfer Rates**

|  |  |
| --- | --- |
| http://www.aaa-datarecovery.com/images/raid7.gif | |
| Fully implemented process oriented real time operating system resident on embedded array control microprocessor. RAID 7 is a registered trademark of Storage Computer Corporation. | |
| http://www.aaa-datarecovery.com/images/advantage.gifAdvantages | http://www.aaa-datarecovery.com/images/disadvantage.gifDisadvantages |
| Overall write performance is 25% to 90% better than single spindle performance and 1.5 to 6 times better than other array levels Host interfaces are scalable for connectivity or increased host transfer bandwidth Small reads in multi user environment have very high cache hit rate resulting in near zero access times No extra data transfers required for parity manipulation | One vendor proprietary solution Extremely high cost per MB Very short warranty Not user serviceable Power supply must be UPS to prevent loss of cache data |

**RAID 0+1 : High Data Transfer Performance**

|  |  |
| --- | --- |
| http://www.aaa-datarecovery.com/images/raid01.gif | |
| RAID Level 0+1 requires a minimum of 4 drives to implement | |
| http://www.aaa-datarecovery.com/images/advantage.gifAdvantages | http://www.aaa-datarecovery.com/images/disadvantage.gifDisadvantages |
| RAID 0+1 is implemented as a mirrored array whose segments are RAID 0 arrays  RAID 0+1 has the same fault tolerance as RAID level 5  RAID 0+1 has the same overhead for fault-tolerance as mirroring alone  Excellent solution for sites that need high performance but are not concerned with achieving maximum reliability | RAID 0+1 is NOT to be confused with RAID 10. A single drive failure will cause the whole array to become, in essence, a RAID Level 0 array  Very expensive / High overhead  Very limited scalability at a very high inherent cost  All drives must move in parallel to proper track lowering sustained performance |

**RAID 10 : Very High Reliability combined with High Performance**

|  |  |
| --- | --- |
| http://www.aaa-datarecovery.com/images/raid10.gif | |
| RAID Level 10 requires a minimum of 4 drives to implement | |
| http://www.aaa-datarecovery.com/images/advantage.gifAdvantages | http://www.aaa-datarecovery.com/images/disadvantage.gifDisadvantages |
| RAID 10 is implemented as a striped array whose segments are RAID 1 arrays  RAID 10 has the same fault tolerance as RAID level 1  RAID 10 has the same overhead for fault-tolerance as mirroring alone  Excellent solution for sites that would have otherwise gone with RAID 1 but need some additional performance boost | Very expensive / High overhead  All drives must move in parallel to proper track lowering sustained performance  Very limited scalability at a very high inherent cost |

**RAID 53 : High I/O Rates and Data Transfer Performance**

|  |  |
| --- | --- |
| http://www.aaa-datarecovery.com/images/raid53.gif | |
| RAID Level 53 requires a minimum of 5 drives to implement | |
| http://www.aaa-datarecovery.com/images/advantage.gifAdvantages | http://www.aaa-datarecovery.com/images/disadvantage.gifDisadvantages |
| RAID 53 should really be called 'RAID 03' because it is implemented as a striped (RAID level 0) array whose segments are RAID 3 arrays  RAID 53 has the same fault tolerance as RAID 3 as well as the same fault tolerance overhead  High data transfer rates are achieved thanks to its RAID 3 array segments  Maybe a good solution for sites who would have otherwise gone with RAID 3 but need some additional performance boost | Very expensive to implement  All disk spindles must be synchronized, which limits the choice of drives  Byte striping results in poor utilization of formatted capacity |

http://www.aaa-datarecovery.com/raid\_tutorial.htm