

RAID

Mark E. Donaldson

RAID (Redundant Array of Inexpensive Disks) is a storage technology that groups multiple hard drives into what appears to be one logical volume. The term RAID was introduced in a late-1987 paper by Patterson, Gibson, and Katz of the University of California-Berkley entitled "A Case for Redundant Arrays of Inexpensive Disks (RAID)." The paper compared RAID to SLED (Single Large Expensive Disk) and described five-disk array architectures, or levels. RAID technology is currently the hottest mass storage topic in the literature.

Disk arrays generally improve system performance by supporting multiple simultaneous read and/or write operations as well as by increasing capacity and providing fault tolerance. The use of multiple drives in an array actually increases the chances that a drive failure will occur. However, the data redundancy of RAID allows the array to tolerate a drive failure. Originally, there were five basic levels of RAID defined. Others, and combinations of levels, have since been added. Below is a basic description of each level of RAID:

RAID Level 0

This form of RAID is not RAID as described in the Berkley paper because there is no data redundancy. Most disk arrays use striping, or distribution of data across multiple drives. RAID 0 implements striping without redundancy and is, therefore, less reliable than a single drive. The only advantage is increased speed.

RAID Level 1

RAID 1 implements mirroring, or shadowing, of disks. Each drive in the system has a copy, or "mirror, of itself. If a drive falls, the duplicate drive keeps working with no lost data or downtime. Since there are two sources of data, the average access time for a read request will be faster than that for a single drive. For a write request, which is almost always preceded by a read, the decrease in read seek time of RAID 1 is offset by the increase in write seek time (since the data has to be written to two disks). A read and two writes takes the same time as a read/write for a single drive. RAID 1, with an optimized controller, has slightly lower overall access times than a single drive.

The main advantage of RAID 1 over other RAID architectures is simplicity. It only requires a dual channel controller or a minimal device driver using one or two controllers to implement. No change to the operating system is needed. RAID 1 is relatively expensive to implement because only half the available disk space is used for data storage. In addition, the necessity of duplicate drives requires more power and more space for the same storage capacity.

RAID Level 2

RAID 2 is an architecture that succeeds in reducing disk overhead (the cost of storage space lost to redundancy) by using Hamming codes to detect and correct errors. Check disks are required in addition to the data disks. The data is striped across the disks along with an Interleaved Hamming code. Because all of the data disks must seek before a read starts, and because for a write the data disks must seek, read data, all drives (including check disks) must seek again, and then the data is written, seek times are very slow compared to a single drive. However, once the seek is completed, data transfer rates are very high. For an array with 8 data drives, the drives will transmit data in parallel. The transfer rate of the array will be 8 times that of a single drive.

RAID 2 is best for reading and writing large data blocks at high data transfer rates. In the microcomputer environment the existing error detection/ correction features result in redundant error isolation data for RAID 2 and make RAID 2 impractical for microcomputers. By letting the drives manage error detection, it is possible to Implement RAID requiring only one check disk for error correction.

RAID

Mark E. Donaldson

RAID Level 3

By assuming that each disk drive in the array can detect and report errors, the RAID system only has to maintain redundancy in the data necessary to correct errors. RAID 3 employs a single check disk (parity disk) for each group of drives. Data is striped across each of the data disks. The check disk receives the XOR (exclusive OR) of all the data written to the data drives. Data for a failed drive can be reconstructed by computing the XOR of all the remaining drives. This approach reduces disk overhead from RAID 1 and 2. For a five-disk array, four of the drives store data; providing 4 GB of data storage in a 5 GB array. RAID 3 also has the same high transfer rates as RAID 2. However, because every data drive is involved in every read or write, a penalty is paid.

RAID 3 can process only one I/O transaction at a time. In addition, the minimum amount of data that can be written or read from a RAID 3 array is the number of data drives multiplied by the number of bytes per sector, referred to as a transfer unit. A typical five-drive array would have four data disks, one parity disk, and might have a 512-byte sector size on each disk. The transfer unit would be 2048 bytes (4 x 512). When a data read is smaller than the transfer unit, the entire unit is read anyway, increasing the length of a read operation. For a data write smaller than the transfer unit, although only a portion of a sector of each disk needs to be modified, the array must still deal with complete transfer units. A complete unit must be read from the array; the data must be rewritten where necessary; and the modified data must be written back to the data disks and the check disk updated. RAID 3 works well in applications that process large chunks of data.

RAID Level 4

RAID 4 addresses the problems associated with bit-striping a transfer block of data across the array. As in RAID 3, one drive in the array is reserved as the check disk. This architecture, however, utilizes block or sector striping to the drives, resulting in read transactions involving only one drive and timing comparable to a single drive. In addition, multiple read requests can be handled at the same time. However, since every write accesses the parity disk, only one write at a time is possible. RAID 4 is most useful in an environment where the ratio of reads to writes is very high.

RAID Level 5

Because RAID levels 2 through 4 each use a dedicated check disk, only one write transaction is possible at any time. RAID 5 overcomes the write bottleneck by distributing the error correcting codes (ECC) across each of the disks in the array. Therefore, each disk in the array contains both data and check-data. Distributing check-data across the array allows reads and writes to be done in parallel. Data recovery and seek times are comparable to RAID 4.

RAID Level 6

To further improve the fault tolerance of RAID Level 5, the same Berkeley researchers who developed the initial five RAID levels proposed one more, now known as RAID Level 6. This level adds a second parity drive to the RAID level 5 array. The chief benefit is that any two drives in the array can fail without the loss of data. This enables an array to remain in active service while an individual physical drive is being repaired yet still remain fault tolerant. In effect, a RAID Level 6 array with a single failed physical disk becomes a RAID Level 5 array. The drawback of the RAID Level 6 design is that it requires two parity blocks to be written during every write operation. Its write performance is extremely low, although read performance can achieve levels on par with RAID Level 5.

RAID Level 10

RAID

Mark E. Donaldson

Some arrays employ multiple RAID technologies. RAID Level 10 represents a layering of RAID Levels 0 and 1 to combine the benefits of each. (Sometimes RAID Level 10 is called RAID Level O&I to more specifically point at its origins). To improve input/output performance, RAID Level 10 employs data striping, splitting data blocks between multiple drives. Moreover, the Array Management Software can further speed read operations by filling multiple operations simultaneously from the two mirrored arrays (at times when both halves of the mirror are functional, of course). To improve reliability, the RAID level uses mirroring so that the striped arrays are exactly duplicated. This technology achieves the benefits of both of its individual layers. Its chief drawback is cost. As with simple mirroring, it doubles the amount of physical storage needed for a given amount of logical storage.

RAID Level 53

This level represents a layering of RAID Level 0 and RAID Level 3-the incoming data is striped between two RAID Level 3 arrays. The capacity of the RAID Level 53 array is the total of the capacity of the individual underlying RAID Level 3 arrays. Input/output performance is enhanced by the striping between multiple arrays. Throughput is improved by the underlying RAID Level 3 arrays. Because the simple striping of the top RAID Level 0 layer adds no redundant data, reliability falls. RAID Level 3 arrays, however, are inherently so fault tolerant that the overall reliability of the RAID Level 53 array far exceeds that of an individual hard disk drive. As with a RAID Level 3 array, the failure of a single drive will not adversely affect data integrity.

Disk Array Implementations

Actual disk array implementations are not always as simple or straightforward as described above. Some manufacturers combine features of different RAID levels to create a hybrid, as in RAID 0/1. RAID implementations that are extremely fault-tolerant provide redundancy beyond that of the drives. Additional redundancy is accomplished by providing a system with redundant drive controllers, redundant power supplies, redundant SCSI controllers, and so on. Some manufacturers offer "hot swappability," the ability to replace a failed drive (or other hardware units) without shutting the system down. Other manufacturers offer a spare drive that is automatically put into use rebuilding the failed drive as soon as the system senses a failure.

Still other RAID manufacturers offer only software-based RAID. The RAID architecture is contained in software that the customer implements with his own hardware. And finally, some RAID systems offer more than one level of RAID in the same package to handle mixed applications more efficiently.