

Can you afford multiple disk drive failure?

Why standard RAID systems may not offer enough protection, by Axel Boehme, Tandberg Data GmbH

The loss or corruption of information is the worst-case scenario for today's data storage installations. The smallest electrical or logical malfunction can cause, in just a few seconds, more damage than external influences like fire or water. Although current storage technologies offer protection in various ways against such hazards, they do this, as they have done for many years, on the basis of two simple algorithms: duplication and check-sum generation, or a combination of the two, depending on the technique.

Many of the currently employed basic RAID (Redundant Array of Independent Disks) systems, especially in the low-end market segments, can only compensate for the failure of one disk drive. Although the probability of a simultaneous failure of several drives is much lower, there are other lurking dangers. For example, all the employed disk drives experience much higher levels of stress during the RAID system reorganisation process, following the replacement of the defective drive, because the reorganisation is performed in addition to the normal 'write and read' requests. A second 'borderline' drive then often gives up the ghost and the data inventory is lost.

RAIDn

The ideal storage system technology would therefore adapt itself to the variable data protection needs of the user, instead of its performance capabilities being unnecessarily capped by pre-set limits. There is a patented algorithm that delivers protection against a multiple drive failure far beyond the possibilities of conventional RAID techniques, effectively taking RAID into its 'nth' dimension. This algorithm will be available in the future under the name RAIDn.

The current standard techniques of RAID are RAID 1 (mirroring), RAID 5 (parity summing) and RAID 0 (striping). The latter, however, only increases the speed and not the reliability. Used alone, RAID 1 offers the redundancy of $n-1$ drives (n = total number of drives) and the storage volume of one drive (normally $n = 2$). RAID 5, on the other hand, allows only one redundant drive with a storage volume of $n-1$. The following examination of a few typical combinations of nine or ten disk drives allows RAIDn to be compared with the conventional RAID techniques. The relevant factors are:

- Minimum redundancy (number of defective drives that can definitely be restored)
- Maximum redundancy (number of defective drives that can potentially be restored, dependent on the actual failure)
- Usable storage capacity*
- Ideal read speed*
- Ideal write speed*

(*based on a single hard disk of the same type)

For the purpose of a fairly simple comparison, the examination assumes ideal conditions for the data transfer rates and disregards additional computation overhead.

(A) RAID 1+0 array:

Two-mirrored RAID 0 arrays (striping), each with five drives. One RAID 0 array is defective as soon as one drive is lost, the minimum redundancy is therefore 1, the maximum redundancy is 5 provided that all drives of the same array are affected. The drive capacity is 5, the write speed is 5 and the read speed is 10 (provided that the system makes an optimum use of the parallelism of the read processes on the mirrored drives).

(B) RAID 5+1 array:

Two mirrored RAID 5 arrays, each with five drives. The minimum redundancy is 3 (the data can be restored if at least one RAID 5 array has only one defect) the maximum redundancy is 6 (failure of a complete array and a single drive of the mirror). The capacity is the same as a single RAID 5 array, i.e. 4, the read speed is 10 (if the aforementioned preconditions are fulfilled), the write-speed is 4 (generation of the parity is omitted). The same results are obtained with 1+5 arrays.

(C) RAID 5+5 array:

Three independent RAID 5 arrays, each with three disks, are configured again to a higher level RAID 5 array (RAID 5 does not mean that sets of five disks must be used here). This allows a complete array to be lost without affecting the data integrity. Also possible is any failure of one drive in all three arrays without data loss. The minimum redundancy is therefore 3 (the failure of 0 – 2 – 2 is beyond compensation), the maximum redundancy is 5. The storage capacity is only 4 (with nine disks), the read speed is 9, the write speed is 4. All in all, this RAID variant is the least attractive even though it requires one drive less than the others.

RAIDn array:

The alternative RAIDn array is defined by n = total number of drives and m = number of permissible failures ($n > m > 0$). The storage capacity is then $n-m$, whereby any number of drives up to m can fail without loss of data integrity. The read speed is n and the write speed $n-m$, in each case as the theoretical maximum.

As special features, the RAIDn configurations $m = 1$ produces the well-known RAID 5 array, $m = 0$ produces the RAID 0 array and $m = n-1$ produces the RAID 1 array.

With the same number of used disk drives, the RAIDn algorithm delivers the same minimum redundancy, i.e. compensates any failure of at least two drives and additionally offers 50-75 percent more capacity than a standard RAID.

The minimum redundancy increases to an immense 200-250 percent if the additional capacity is not required, i.e. the compensation of a simultaneous failure of a much higher number of disk drives and data restore is possible.

The write speed of RAIDn is the same as that of the established RAID arrays, the read speeds are, however, much higher, in each case proportional to the corresponding capacities.

The conventional RAID arrays RAID 0, RAID 1, RAID 4, RAID 5 and RAID 0+1 are obtained as special cases of the universal RAIDn and are thus a part of the new algorithm.

RAIDn uses an extremely fast computing algorithm. This differs greatly from the more complex earlier Mariani or Reed Solomon algorithms. The RAIDn technique was initially implemented in the LINUX operating system. It will also be ported to other operating systems, eg Windows, within the scope of further development. RAIDn is a LINUX kernel module that allows all the customary functions for creating and mounting immediately after loading with the insmod utility and the writing and reading of data with dd or similar utilities.

Fully analogue to the standard RAID techniques, RAIDn also offers the possibility of dynamically changing the number of used drives. This means that it is not only possible to increase the capacity of RAIDn by installing additional disk drives but also, if desired, to increase the protection level by defining the new drive as additional redundancy, whilst the system is running. All these expansions occur without interruption or restriction of access to existing data inventories. The rebuild of the RAIDn array, following the replacement of one or several defective disk drives, also occurs without interruption.

Example

Let's suppose that the total number of available disk drives in RAIDn product will have a maximum of $n = 9$, so a practical definition of the possible drive failures will lie between $m = 1$ and $m = 3$. With three possible drive failures, this system will not only offer the same protection level as a mirrored RAID 5 but also require one drive less than RAID 5+1 whilst offering 50 percent more disk drive capacity (six usable drives instead of only four).

With its performance and protection advantages it should be no time before RAIDn has gained strong customer acceptance.

Tandberg Data GmbH are exhibiting at Storage Expo the UK's largest and most important event dedicated to data storage, now in its 4th year, the show features a comprehensive free education programme, and over 90 exhibitors at the National Hall, Olympia, London from 13 - 14 October 2004. <http://www.storage-expo.com>