The aim is to emulate a very expensive disks buying a bunch of less expensive ones: to increase the performance, the size and the reliability of storage systems, several independent disks are considered as a single one. Data are striped across the disk and accessed in parallel, thus gaining high parallelism in accessing them and load balancing.

Two techniques must be implemented on RAID disks.

1. Data striping: data are written sequentially according to a cyclic algorithm (round robin), it allows reading and writing in parallel.
    a. Stripe unit: the amount of data written on a single disk,
    b. Stripe width: the number of disks considered by the algorithm, which can be less than the total amount of disks.
2. Redundancy: it is necessary as the probability of failure rises with the growing of the number of disks, it consists of error-correcting codes, that are stored on different disks than the data and can be used to reconstruct the data after a failure; the main drawback is the slowing down of the writing operations, as these values must be computed and stored along with the data.

There are many types of architectures, the choice should be done accordingly to the required features.

1. RAID 0

Data are written on a single logical disk and then split by the controller among the several physical disks. It has the lowest costs and the best write performance of all the levels, but the failure of a single disk will cause the loss of data, so it is used where performance and capacity are very important while data reliability is not an issue.

2. RAID 1

Data are duplicated, two physical disks contain the same data; it has a high reliability but read and write performance are not bad, as data can be accessed in parallel without the ned of computing parity bits; the main drawback is the cost, as only 50% of the capacity can be used. In theory, data could be copied on more than one disk, but this solution is never used due to the prohibited costs.

If several disks are available, RAIDs can be combined: x+y means that there are n*m disks in total, then RAID x is applied to groups of n disks, that are treated like a single one onto which is applied RAID y. Two very used configurations are 0+1 and 1+0, the former places redundancy at a higher level, thus becoming less fault tolerant as there are only two groups of RAID1: if two disks on different groups fail, the controller cannot realize that their copies could still be found on the other RAID0 level and so data are lost. Performances and storage capacity are the same.

3. RAID 2

The parity is calculated for several subset of overlapped data, the number of disks where parity bits are stored is proportional to the logarithm of the disks that contain data. When a block fails, several of the parity blocks will have inconsistent value, the failed component is the one held in common by each incorrect subset.

4. RAID 3

Here the failed disk is assumed to be known, which is realistic as the controller is usually able to detect the failed component. There is one disk to store parity bits, while all the others contain data, meaning that only one request can be served at a time.

5. RAID 4

It is similar to level 3, the main difference is that parity is calculated for strips that have the same position in all the disks and then stored in that aimed for redundancy. The fact that there is only one disk to store parity bits can easily become the bottleneck: all the write must access it, so they cannot be parallelised. This level is able to recover from the failure of one disk.

6. RAID 5

It is exactly identical to level 4, but the parity blocks are uniformly distributed over the disk, thus avoiding the bottleneck on the parity disk and allowing load balancing among the data disks. It loses data if more than one disk fails.

7. RAID 6

This level is able to recover from the failure of two disks, as it uses two redundancy schemes P and Q which are independent. On the other hand, it needs a disk more than level 5 to be implemented and a greater computational overhead, so it must be used only for very critical applications.
However, its efficiency grows with the number of disks, as does the probability of having two failures, so it becomes even almost mandatory when a high number of disks is present.
The two values are computed as $P = \sum_i D_i$ and $Q = \sum_i g^i D_i$ with $i \neq 1$ (i is usually equal to 2)
The repairing technique depends on which disk (or disks) have failed:

a. One data disk: $D_i = P - \sum_k D_k$;
b. One parity bit: simply recompute it, data are still available;
c. One data disk and the Q block: the data are reconstructed as $D_i = P - \sum_k D_k$ and then used to recompute Q;
d. One data disk and the P block: data are reconstructed as $D_i = \left(Q - \sum_k g^k D_k\right)/g^i$ and then used to reconstruct P;
e. Two data disks D_i and D_k: a system of equation must be solved, calling $P^* = \sum_t D_t$ with $t \neq i, j$ and $Q^* = \sum_t g^t D_t$ with $t \neq i, j$ the values of the parity bits computed without the broken disks, the two equations are $P_{OLD} = P^* + D_i + D_j$ and $Q_{OLD} = Q^* + g^i D_i + g^k D_k$.

However, these techniques work only for machines with infinite precision, that do not exist. The solution is to use a special algebra, called Galois Fields, that includes only the integer powers of prime numbers and allows to perform all the previous mentioned operations using numbers that can fit into a byte.

8. RAID 7: not standardized yet, we will not see it

Storage systems

There are three main types of storage systems

1. Direct Attached Storage (DAS)

It is a system directly attached to a server or a workstation, it has a limited scalability, a complex management and low performances and it is difficult to read files on other machines. It can be an internal drive as well as an external disk, connected with a "point to point" network.

2. Network Attached Storage (NAS)

A NAS unit is a computer connected to a network that provides only file-based storage to other services in the network, it has its own IP address and the system is very scalable. It is designed as a self-contained solution to share files over the network, so its performance depends mainly on its speed and congestion.
It is used for low-volume access to a large amount of storage by many users.

3. Storage Area Network (SAN)

They are remote storage units that are connected to the PC with a specific network technology, it does not provide the file system and it is seen by the operative system simply as a disk, differently than NAS that is visible as a server.
It is used for petabytes of storage and multiple, simultaneous access to files (Netflix), it is highly scalable.

As said, a special interface is needed to access it: the TCP/IP stack over ethernet leads to many issues, that is not needed for the specific purpose, so a specific protocol called **fibre channel** is implemented. It is a high-speed network technology that is well established in the open system environment. They are accessed using the standard ethernet using an appliance called NAS head.