DIFFERENT RAID LEVELS

RAID has developed since its original definition in 1987. Due to technical progress some RAID levels are now practically meaningless, whilst others have been modified or added at a later date. This section introduces the RAID levels that are currently the most significant in practice. We will not introduce RAID levels that represent manufacturer- pecific variants and variants that only deviate slightly from the basic forms mentioned in the following.

2.5.1 RAID 0: block-by-block striping

RAID 0 distributes the data that the server writes to the virtual hard disk onto one physical hard disk after another block-by-block (block-by-block striping). Figure 2.9 hows a RAID array with four physical hard disks. In Figure 2.9 the server writes the blocks A, B, C, D, E, etc. onto the virtual hard disk one after the other. The RAID controller distributes the sequence of blocks onto the individual physical hard disks: it writes the first block, A, to the first physical hard disk, the second block, B, to the second physical hard disk, block C to the third and block D to the fourth. Then it begins to write to the first physical hard disk once again, writing block E to the first disk, block F to the second, and so on.

RAID 0 increases the performance of the virtual hard disk as follows: the individual hard disks can exchange data with the RAID controller via the I/O channel significantly more quickly than they can write to or read from the rotating disk. In Figure 2.9 the RAID controller sends the first block, block A, to the first hard disk. This takes some time to write the block to the disk. Whilst the first disk is writing the first block to the physical hard disk, the RAID controller is already sending the second block, block B, to the second hard disk and block C to the third hard disk. In the meantime, the first two physical hard disks are still engaged in depositing their respective blocks onto the physical hard disk. If the RAID controller now sends block E to the first hard disk, then this has written block A at least partially, if not entirely, to the physical hard disk.

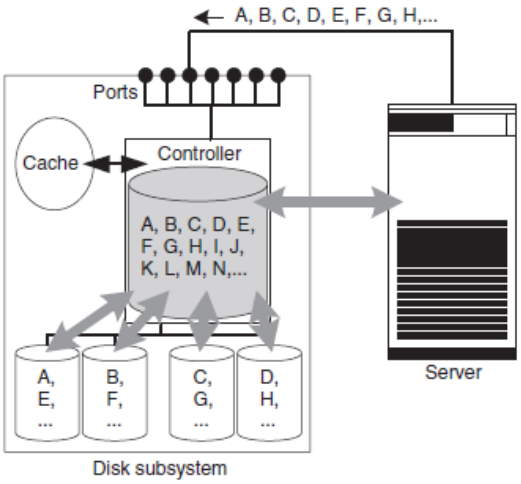

Figure 2.9 RAID 0 (striping): As in all RAID levels, the server sees only the virtual hard disk. The RAID controller distributes the write operations of the server amongst several physical hard disks. Parallel writing means that the performance of the virtual hard disk is higher than that of the individual physical hard disks.

In the example, it was possible to increase the throughput fourfold in 2002: Individual hard disks were able to achieve a throughput of around 50MB/s. The four physical hard disks achieve a total throughput of around 4 × 50MB/s _ 200MB/s. In those days I/O techniques such as SCSI or Fibre Channel achieve a throughput of 160MB/s or 200MB/s. If the RAID array consisted of just three physical hard disks the total throughput of the hard disks would be the limiting factor. If, on the other hand, the RAID array consisted of five physical hard disks the I/O path would be the limiting factor. With five or more hard disks, therefore, performance increases are only possible if the hard disks are connected to different I/O paths so that the load can be striped not only over several physical hard disks, but also over several I/O paths.

RAID 0 increases the performance of the virtual hard disk, but not its fault-tolerance. If a physical hard disk is lost, all the data on the virtual hard disk is lost. To be precise, therefore, the ‘R’ for ‘Redundant’ in RAID is incorrect in the case of RAID 0, with ‘RAID 0’ standing instead for ‘zero redundancy’.

2.5.2 RAID 1: block-by-block mirroring

In contrast to RAID 0, in RAID 1 fault-tolerance is of primary importance. The basic form of RAID 1 brings together two physical hard disks to form a virtual hard disk by mirroring the

data on the two physical hard disks. If the server writes a block to the virtual hard disk, the RAID controller writes this block to both physical hard disks (Figure 2.10).

The individual copies are also called mirrors. Normally, two or sometimes three copies of the data are kept (three-way mirror). In a normal operation with pure RAID 1, performance increases are only possible in read operations. After all, when reading the data the load can be divided between the two disks. However, this gain is very low in comparison to RAID 0. When writing with RAID 1 it tends to be the case that reductions in performance may even have to be taken into account. This is because the RAID controller has to send the data to both hard disks. This disadvantage can be disregarded for an individual write operation, since the capacity of the I/O channel is significantly higher than the maximum write speed of the two hard disks put together. However, the I/O channel is under twice the load, which hinders other data traffic using the I/O channel at the same time.

2.5.3 RAID 0+1/RAID 10: striping and mirroring combined

The problem with RAID 0 and RAID 1 is that they increase either performance (RAID 0) or fault-tolerance (RAID 1). However, it would be nice to have both performance and fault-tolerance. This is where RAID 0+1 and RAID 10 come into play. These two RAID levels combine the ideas of RAID 0 and RAID 1

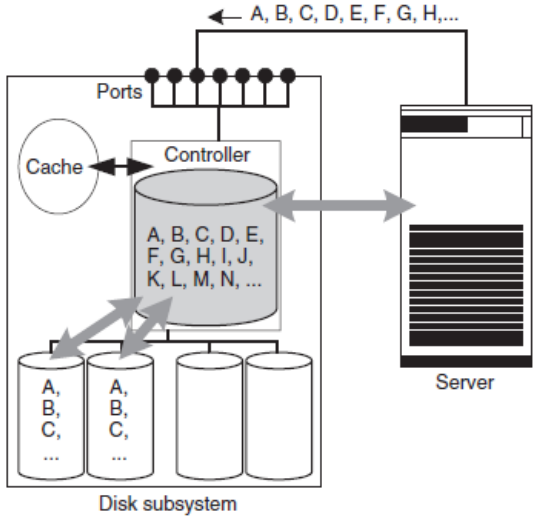

Figure 2.10 RAID 1 (mirroring): As in all RAID levels, the server sees only the virtual hard disk. The RAID controller duplicates each of the server's write operations onto two physical hard disks. After the failure of one physical hard disk the data can still be read from the other disk.

RAID 0+1 and RAID 10 each represent a two-stage virtualisation hierarchy. Figure 2.11 shows the principle behind RAID 0+1 (mirrored stripes). In the example, eight physical hard disks are used. The RAID controller initially brings together each four physical hard disks to form a total of two virtual hard disks that are only visible within the RAID controller by means of RAID 0 (striping). In the second level, it consolidates these two virtual hard disks into a single virtual hard disk by means of RAID 1 (mirroring); only this virtual hard disk is visible to the server.

In RAID 10 (striped mirrors) the sequence of RAID 0 (striping) and RAID 1 (mirroring) is reversed in relation to RAID 0+1 (mirrored stripes). Figure 2.12 shows the principle underlying RAID 10 based again on eight physical hard disks. In RAID 10 the RAID controller initially brings together the physical hard disks in pairs by means of RAID 1 (mirroring) to form a total of four virtual hard disks that are only visible within the RAID controller. In the second stage, the RAID controller consolidates these four virtual hard disks into a virtual hard disk by means of RAID 0 (striping). Here too, only this last virtual hard disk is visible to the server. In both RAID 0+1 and RAID 10 the server sees nly a single hard disk, which is larger, faster and more fault-tolerant than a physical hard disk. We now have to ask the question: which of the two RAID levels, RAID 0+1 or RAID 10, is preferable?

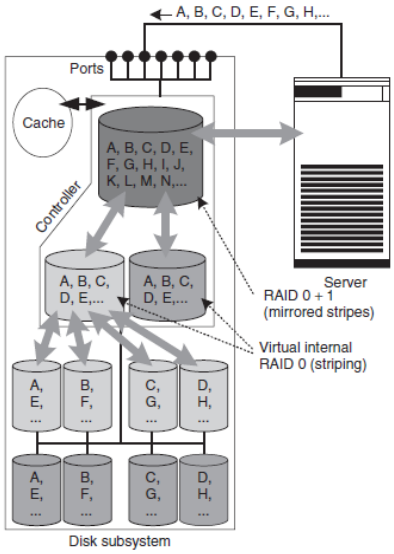

Figure 2.11 RAID 0+1 (mirrored stripes): As in all RAID levels, the server sees only the virtual hard disk. Internally, the RAID controller realises the virtual disk in two stages: in the first stage it brings together every four physical hard disks into one virtual hard disk that is only visible within the RAID controller by means of RAID 0 (striping); in the second stage it consolidates these two virtual hard disks by means of RAID 1 (mirroring) to form the hard disk that is visible to the server.

The question can be answered by considering that when using RAID 0 the failure of a hard disk leads to the loss of the entire virtual hard disk. In the example relating to RAID 0+1

(Figure 2.11) the failure of a physical hard disk is thus equivalent to the effective failure of four physical hard disks (Figure 2.13). If one of the other four physical hard disks is lost, then the data is lost. In principle it is sometimes possible to reconstruct the data from the remaining disks, but the RAID controllers available on the market cannot do this particularly well.

In the case of RAID 10, on the other hand, after the failure of an individual physical hard disk, the additional failure of a further physical hard disk – with the exception of the corresponding mirror – can be withstood (Figure 2.14). RAID 10 thus has a significantly higher fault-tolerance than RAID 0+1. In addition, the cost of restoring the RAID system after the failure of a hard disk is much lower in the case of RAID 10 than RAID 0+1. In RAID 10 only one physical hard disk has to be recreated. In RAID 0+1, on the other and, a virtual hard disk must be recreated that is made up of four physical disks. However, the cost of recreating the defective hard disk can be significantly reduced because a physical hard disk is exchanged as a preventative measure when the number of read errors start to increase. In this case it is sufficient to copy the data from the old disk to the new.

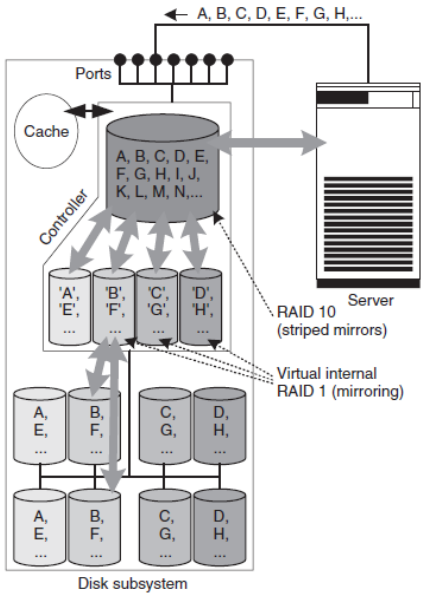

Figure 2.12 RAID 10 (striped mirrors): As in all RAID levels, the server sees only the virtual hard disk. Here too, we proceed in two stages. The sequence of striping and mirroring is reversed in relation to RAID 0+1. In the first stage the controller links every two physical hard disks by means of RAID 1 (mirroring) to a virtual hard disk, which it unifies by means of RAID 0 (striping) in the second stage to form the hard disk that is visible to the server.

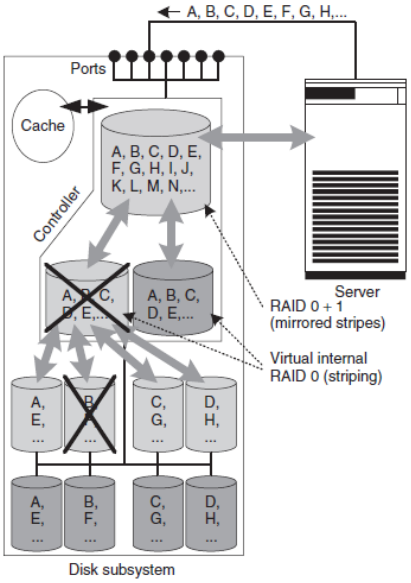

Figure 2.13 The consequences of the failure of a physical hard disk in RAID 0+1 (mirrored stripes) are relatively high in comparison to RAID 10 (striped mirrors). The failure of a physical hard disk brings about the failure of the corresponding internal RAID 0 disk, so that in effect half of the physical hard disks have failed. The recovery of the data from the failed disk is expensive.

However, things look different if the performance of RAID 0+1 is compared with the performance of RAID 10. In Section 5.1 we discuss a case study in which the use of RAID 0+1 is advantageous. With regard to RAID 0+1 and RAID 10 it should be borne in mind that the two RAID procedures are often confused. Therefore the answer ‘We use RAID 10!’ or ‘We use RAID 0+1’ does not always provide the necessary clarity. In discussions it is better to ask if mirroring takes place first and the mirror is then striped or if striping takes place first and the stripes are then mirrored.

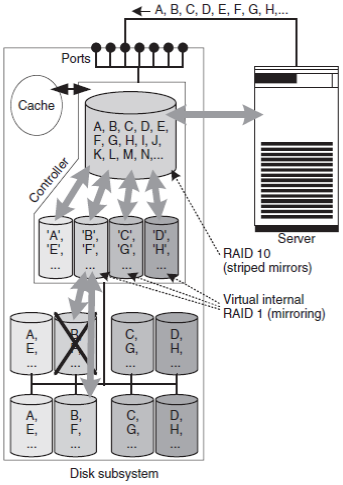

Figure 2.14 In RAID 10 (striped mirrors) the consequences of the failure of a physical hard disk are not as serious as in RAID 0+1 (mirrored stripes). All virtual hard disks remain intact. The recovery of the data from the failed hard disk is simple.

2.5.4 RAID 4 and RAID 5: parity instead of mirroring

RAID 10 provides excellent performance at a high level of fault-tolerance. The problem with this is that mirroring using RAID 1 means that all data is written to the physical hard disk

twice. RAID 10 thus doubles the required storage capacity. The idea of RAID 4 and RAID 5 is to replace all mirror disks of RAID 10 with a single parity hard disk. Figure 2.15 shows the principle of RAID 4 based upon five physical hard disks. The server again writes the blocks A, B, C, D, E, etc. to the virtual hard disk sequentially. The RAID controller stripes the data blocks over the first four physical hard disks. Instead of mirroring all data onto the further four physical hard disks, as in RAID 10, the RAID controller calculates a parity block for every four blocks and writes this onto the fifth physical hard disk.

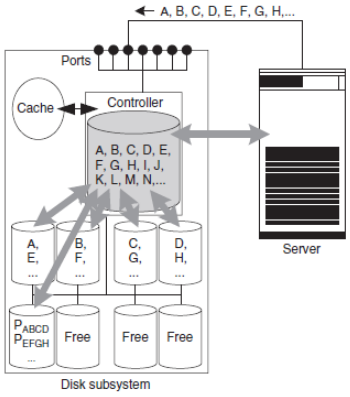

Figure 2.15 RAID 4 (parity disk) is designed to reduce the storage requirement of RAID 0+1 and RAID 10. In the example, the data blocks are distributed over four physical hard disks by means of RAID 0 (striping). Instead of mirroring all data once again, only a parity block is stored for each four blocks.

For example, the RAID controller calculates the parity block PABCD for the blocks A, B, C and D. If one of the four data disks fails, the RAID controller can reconstruct the data of the defective disks using the three other data disks and the parity disk. In comparison to the examples in Figures 2.11 (RAID 0+1) and 2.12 (RAID 10), RAID 4 saves three physical hard disks. As in all other RAID levels, the server again sees only the virtual disk, as if it were a single physical hard disk.

From a mathematical point of view the parity block is calculated with the aid of the logical XOR operator (Exclusive OR). In the example from Figure 2.15, for example, the equation PABCD = A XOR B XOR C XOR D applies. The space saving offered by RAID 4 and RAID 5, which remains to be discussed, comes at a price in relation to RAID 10. Changing a data block changes the value of the associated parity block. This means that each write operation to the virtual hard disk requires (1) the physical writing of the data block, (2) the recalculation of the parity block and (3) the physical writing of the newly calculated parity block. This extra

cost for write operations in RAID 4 and RAID 5 is called the write penalty of RAID 4 or the write penalty of RAID 5.

The cost for the recalculation of the parity block is relatively low due to the mathematical properties of the XOR operator. If the block A is overwritten by block ˜A and _ is the difference between the old and new data block, then _ = A XOR ˜A. The new parity block ˜P can now simply be calculated from the old parity block P and _, i.e. ˜P = P XOR _. Proof of this property can be found in Appendix A. Therefore, if PABCD is the parity block for the data blocks A, B, C and D, then after the data block A has been changed, the new parity block can be calculated without knowing the remaining blocks B, C and D. However, the old block A must be read in before overwriting the physical hard disk in the controller, so that this can calculate the difference _. When processing write commands for RAID 4 and RAID 5 arrays, RAID controllers use the above-mentioned mathematical properties of the XOR operation for the recalculation of the parity block. Figure 2.16 shows a server that changes block D on the virtual hard disk. The RAID controller reads the data block and the associated parity block from the disk in question into its cache. Then it uses the XOR operation to calculate the difference

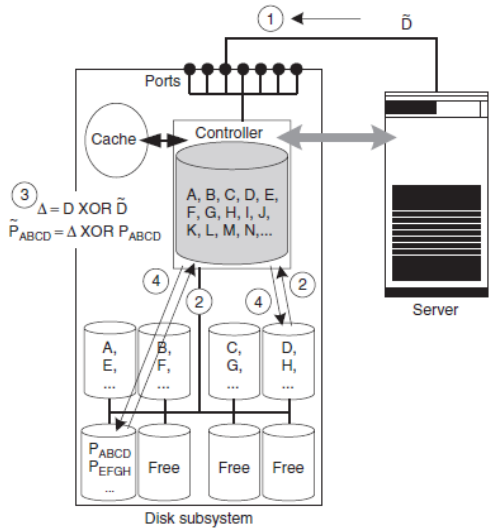

Figure 2.16 Write penalty of RAID 4 and RAID 5: The server writes a changed data block (1). The RAID controller reads in the old data block and the associated old parity block (2) and calculates the new parity block (3). Finally it writes the new data block and the new parity block onto the physical hard disk in question (4).

between the old and the new parity block, i.e. _ = D XOR D˜ , and from this the new parity block ˜PABCD by means of ˜PABCD = PABCD XOR _. Therefore it is not necessary to read in all four associated data blocks to recalculate the parity block. To conclude the write

operation to the virtual hard disk, the RAID controller writes the new data block and the recalculated parity block onto the physical hard disks in question.

Advanced RAID 4 and RAID 5 implementations are capable of reducing the write penalty even further for certain load profiles. For example, if large data quantities are written sequentially, then the RAID controller can calculate the parity blocks from the data flow without reading the old parity block from the disk. If, for example, the blocks E, F, G and H in Figure 2.15 are written in one go, then the controller can calculate the parity block PEFGH from them and overwrite this without having previously read in the old value. Likewise, a RAID controller with a suitably large cache can hold frequently changed parity blocks in the cache after writing to the disk, so that the next time one of the data blocks in question is changed there is no need to read in the parity block. In both cases the I/O load is now lower than in the case of RAID 10. In the example only five physical blocks now need to be written instead of eight as is the case with RAID 10. RAID 4 saves all parity blocks onto a single physical hard disk. For the example in Figure 2.15 this means that the write operations for the data blocks are distributed over four physical hard disks. However, the parity disk has to handle the same number of write operations all on its own. Therefore, the parity disk become the performance bottleneck of RAID 4 if there are a high number of write operations.

To get around this performance bottleneck, RAID 5 distributes the parity blocks over all hard disks. Figure 2.17 illustrates the procedure. As in RAID 4, the RAID controller writes the parity block PABCD for the blocks A, B, C and D onto the fifth physical hard disk. Unlike RAID 4, however, in RAID 5 the parity block PEFGH moves to the fourth physical hard disk for the next four blocks E, F, G, H.

RAID 4 and RAID 5 distribute the data blocks over many physical hard disks. Therefore, the read performance of RAID 4 and RAID 5 is as good as that of RAID 0 and almost as good as that of RAID 10. As discussed, the write performance of RAID 4 and RAID 5 suffers from the write penalty; in RAID 4 there is an additional bottleneck caused by the parity disk. Therefore, RAID 4 is seldom used in practice because RAID 5 accomplishes more than RAID 4 with the same amount of physical resources (see also Section 2.5.6). RAID 4 and RAID 5 can withstand the failure of a physical hard disk. Due to the use of parity blocks, the data on the defective

hard disk can be restored with the help of other hard disks. In contrast to RAID 10, the failure of an individual sector of the remaining physical hard disks always results in data loss. This is compensated for with RAID 6, whereby a second parity hard disk is kept so that data is protected twice (Section 2.5.5). In RAID 4 and RAID 5 the recovery of a defective physical hard disk is significantly more expensive than is the case for RAID 1 and RAID 10. In the latter two RAID levels only the mirror of the defective disk needs to be copied to the replaced disk. In RAID 4 and RAID 5, on the other hand, the RAID controller has to read the data from all disks, use this to recalculate the lost data blocks and parity blocks, and then write these blocks to the replacement disk. As in RAID 0+1 this high cost can be avoided by replacing a physical hard disk as a precaution as soon as the rate of read errors increases. If this is done, it is sufficient to copy the data from the hard disk to be replaced onto the new hard disk.

If the fifth physical hard disk has to be restored in the examples from Figure 2.15 (RAID 4) and Figure 2.17 (RAID 5), the RAID controller must first read the blocks A, B, C and D from the physical hard disks, recalculate the parity block PABCD and then write to the exchanged physical hard disk. If a data block has to be restored, only the calculation rule changes. If, in the example, the third physical hard disk is to be recreated, the controller would first have to read in the blocks A, B, D and PABCD, use these to reconstruct block C and write this to the replaced disk.

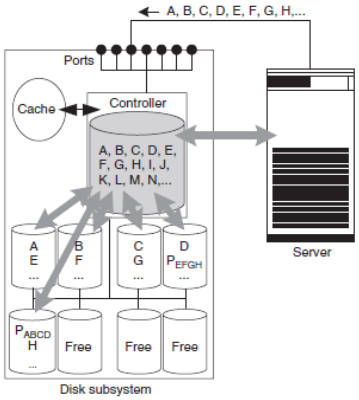

Figure 2.17 RAID 5 (striped parity): In RAID 4 each write access by the server is associated with a write operation to the parity disk for the update of parity information. RAID 5 distributes the load of the parity disk over all physical hard disks.