REMOTE MIRRORING

Instant copies are excellently suited for the copying of data sets within disk subsystems. However, they can only be used to a limited degree for data protection. Although data copies generated using instant copy protect against application errors (accidental deletion of a file system) and logical errors (errors in the database program), they do not protect against the failure of a disk subsystem. Something as simple as a power failure can prevent access to production data and data copies for several hours. A fire in the disk subsystem would destroy original data and data copies. For data protection, therefore, the proximity of production data and data copies is fatal.

Remote mirroring offers protection against such catastrophes. Modern disk subsystems can now mirror their data, or part of their data, independently to a second disk subsystem, which is a long way away. The entire remote mirroring operation is handled by the two participating disk subsystems. Remote mirroring is invisible to application servers and does not consume their resources. However, remote mirroring requires resources in the two disk subsystems and in the I/O channel that connects the two disk subsystems together, which means that reductions in performance can sometimes make their way through to the application.

Figure 2.20 shows an application that is designed to achieve high availability using remote mirroring. The application server and the disk subsystem, plus the associated data, are installed in the primary data centre. The disk subsystem independently mirrors the application data onto the second disk subsystem that is installed 50 kilometres away in the backup data centre by means of remote mirroring. Remote mirroring ensures that the application data in the backup data centre is always kept up-to-date with the time interval for updating the second disk subsystem being configurable. If the disk subsystem in the primary data centre fails, the backup application server in the backup data centre can be started up using the data of the second disk subsystem and the operation of the application can be continued. The I/O techniques required for the connection of the two disk subsystems will be discussed in the next chapter.

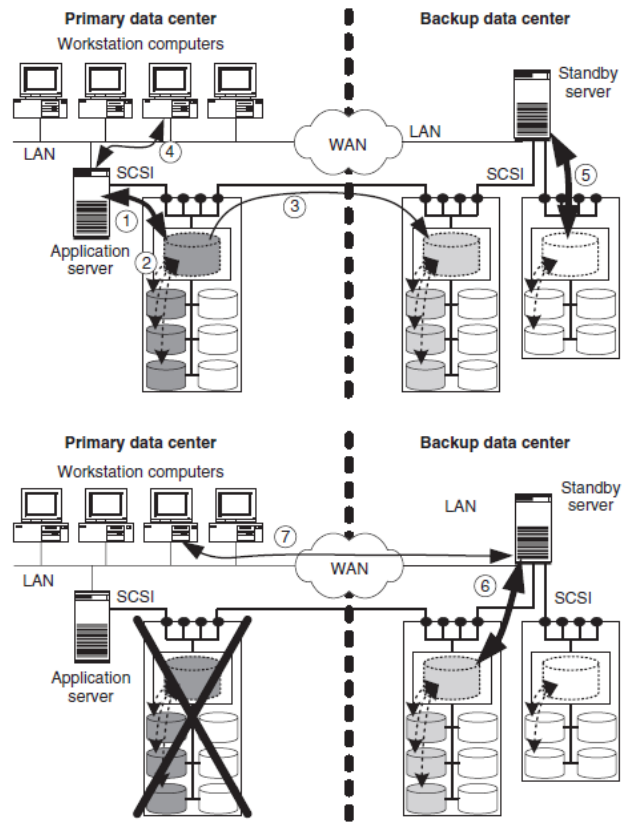

Figure 2.20 High availability with remote mirroring: (1) The application server stores its data on a local disk subsystem. (2) The disk subsystem saves the data to several physical drives by means of RAID. (3) The local disk subsystem uses remote mirroring to mirror the data onto a second disk subsystem located in the backup data centre. (4) Users use the application via the LAN. (5) The stand-by server in the backup data centre is used as a

test system. The test data is located on a further disk subsystem. (6) If the first disk subsystem fails, the application is started up on the stand-by server using the data of the second disk subsystem. (7) Users use the application via the WAN.

We can differentiate between synchronous and asynchronous remote mirroring. In synchronous remote mirroring the first disk subsystem sends the data to the second disk subsystem first before it acknowledges a server's write command. By contrast, asynchronous remote mirroring acknowledges a write command immediately; only then does it send the copy of the block to the second disk subsystem.

Figure 2.21 illustrates the data flow of synchronous remote mirroring. The server writes block A to the first disk subsystem. This stores the block in its write cache and immediately sends it to the second disk subsystem, which also initially stores the block in its write cache. The first disk subsystem waits until the second reports that it has written the block. The question of whether the block is still stored in the write cache of the second disk subsystem or has already been written to the hard disk is irrelevant to the first disk subsystem. It does not acknowledge to the server that the block has been written until it has received confirmation from the second disk subsystem that this has written the block.

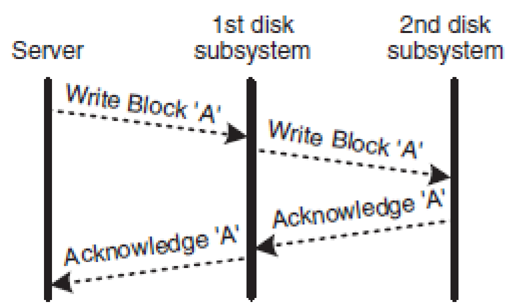

Figure: In asynchronous remote mirroring one disk subsystem acknowledges a write operation as soon as it has saved the block itself. The price of the rapid response time achieved using asynchronous remote mirroring is obvious. In contrast to synchronous remote mirroring, in asynchronous remote mirroring there is no guarantee that the data on the second disk subsystem is up-to-date. This is precisely the case if the first disk subsystem has sent the write acknowledgement to the server but the block has not yet been saved to the second disk subsystem.

If we wish to mirror data over long distances but do not want to use only asynchronous remote mirroring it is necessary to use three disk subsystems (Figure 2.23). The first two may be located just a few kilometres apart, so that synchronous remote mirroring can be used between the two. In addition, the data of the second disk subsystem is mirrored onto a third by means of asynchronous remote mirroring. However, this solution comes at a price: for most applications the cost of data protection would exceed the costs that would be incurred after data loss in the event of a catastrophe. This approach would therefore only be considered for very important applications.

An important aspect of remote mirroring is the duration of the initial copying of the data. With large quantities of data it can take several hours until all data is copied from the first disk subsystem to the second one. This is completely acceptable the first time remote mirroring is established. However, sometimes the connection between both disk subsystems is interrupted later during operations – for example, due to a fault in the network between both systems or during maintenance work on the second disk subsystem.

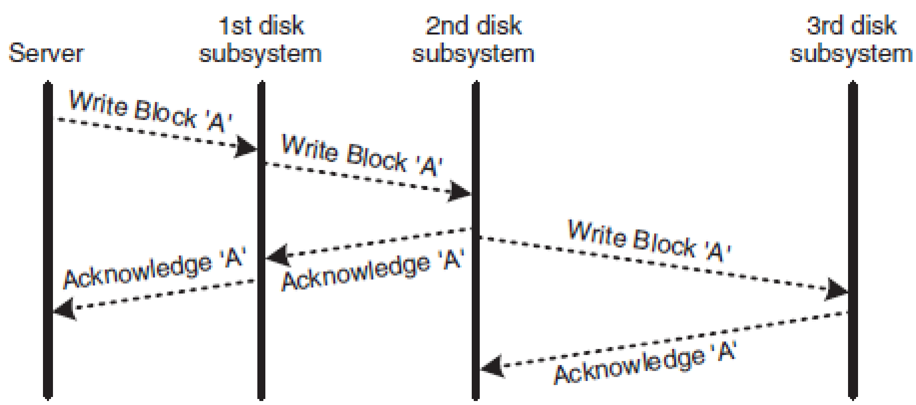

Figure The combination of synchronous and asynchronous remote mirroring means that rapid response times can be achieved in combination with mirroring over long distances.

After the appropriate configuration the application continues operation on the first disk subsystem without the changes having been transferred to the second disk subsystem. Small quantities of data can be transmitted in their entirety again after a fault has been resolved. However, with large quantities of data a mechanism should exist that allows only those blocks that were changed during the fault to be transmitted. This is also referred to as suspending (or freezing) remote mirroring and resuming it later on. Sometimes there is a deliberate reason for suspending a remote mirroring relationship. Later in the book we will present a business continuity solution that suspends remote mirroring relationships at certain points in time for the purposes of creating consistent copies in backup data centres (Section 9.5.6).

For these same reasons there is a need for a reversal of remote mirroring. In this case, if the first disk subsystem fails, the entire operation is completely switched over to the second disk subsystem and afterwards the data is only changed on that second system. The second disk subsystem logs all changed blocks so that only those blocks that were changed during the failure are transmitted to the first disk subsystem once it is operational again. This ensures that the data on both disk subsystems is synchronised once again.