Technologies for Storage Networks

Text Book: Storage Networks Explained

SERVER CENTRIC IT ARCHITECTURE

- In conventional IT architectures, storage devices are normally only connected to a single server (Figure 1.1). To increase fault-tolerance, storage devices are sometimes connected to *two servers*, the **storage device exists only in relation to the server to which it is connected.** Other servers cannot directly access the data; they always have to go through the server that is connected to the storage device.
- This conventional IT architecture is therefore called **server-centric IT architecture**. In this approach, servers and storage devices are generally connected together by SCSI cables.

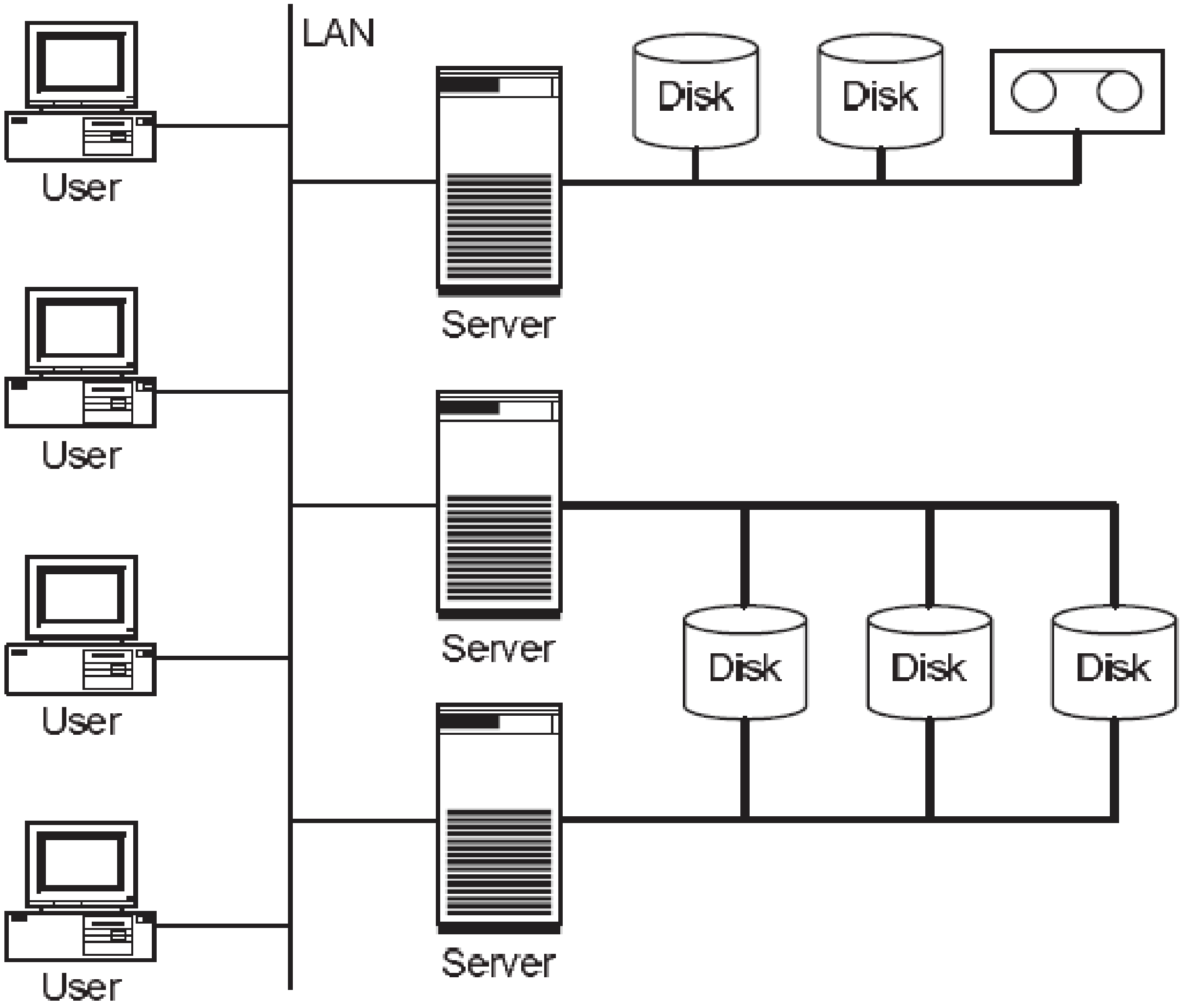

Figure 1.1 In a server-centric IT architecture storage devices exist only in relation to servers

LIMITATIONS

- In conventional server-centric IT architecture storage devices exist only in relation to the one or two servers to which they are connected. The failure of both of these computers would make it impossible to access this data.
- It is necessary to connect even more storage devices to a computer. This throws up the problem that **each computer can accommodate only a limited number of I/O cards** (for example, SCSI cards). Furthermore, the **length of SCSI cables is limited to a maximum of 25 m.** This means that the storage capacity that can be connected to a computer using conventional technologies is limited.

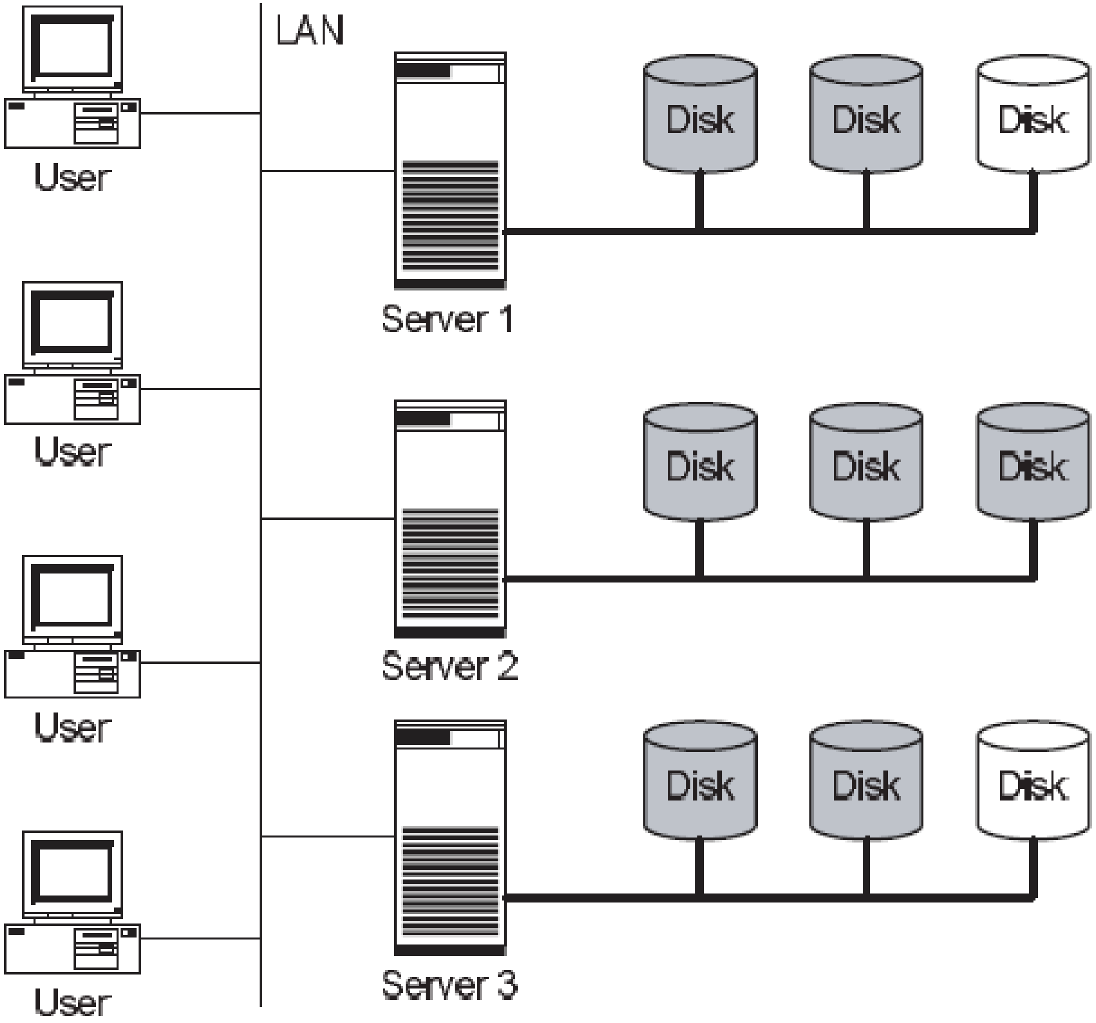

Figure 1.2 The storage capacity on server 2 is full. It cannot make use of the fact that there is still storage space free on server 1 and server 3

STORAGE CENTRIC IT ARCHITECTURE

- Storage networks can solve the problems of server-centric IT architecture that we have just discussed. Furthermore, storage networks open up new **possibilities for data management.**
- The idea behind storage networks is that the *SCSI cable* is replaced by a network that is installed *in addition to the existing LAN* and is primarily used for data exchange between computers and storage devices (Figure 1.3).
- In contrast to server-centric IT architecture, in storage networks storage devices exist completely independently of any computer.
- Several servers can access the same storage device directly over the storage network without another server having to be involved.
- IT architectures with storage networks are therefore known as **storage-centric IT architectures.**

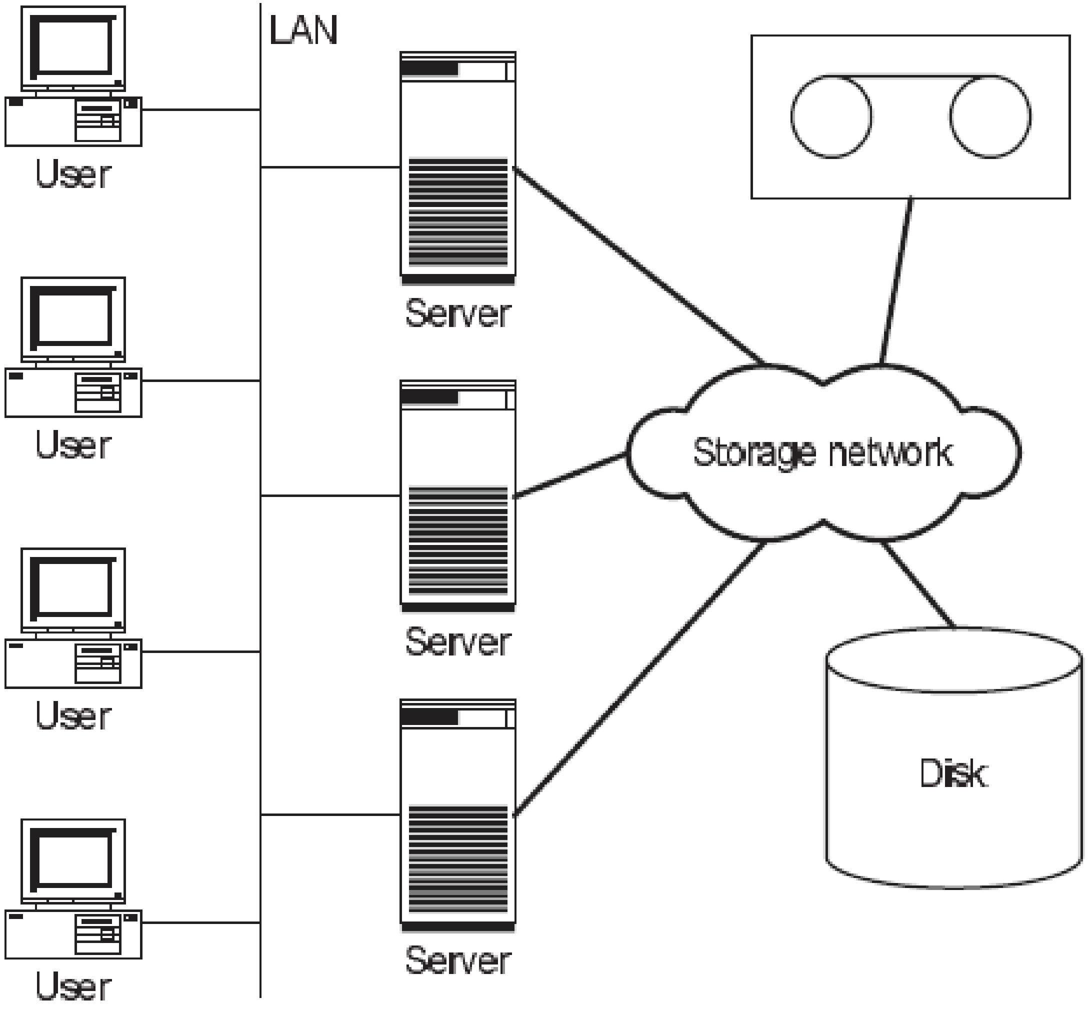

Figure 1.3 In storage-centric IT architecture the SCSI cables are replaced by a network. Storage devices now exist independently of a server

ADVANTAGES

- When a storage network is introduced, the storage devices are usually also consolidated. This involves replacing the many small hard disks attached to the computers with a large **disk subsystem**.
- The storage network permits all computers to access the disk subsystem and share it. Free storage capacity can thus be flexibly assigned to the computer that needs it at the time. In the same manner, many small tape libraries can be replaced by one big one.
- More and more companies are converting their IT systems to a storage-centric IT architecture. It has now become a permanent component of large data centers and the IT systems of large companies.

CASE STUDY: REPLACING A SERVER WITH STORAGE NETWORKS

- In the following we will illustrate some advantages of storage-centric IT architecture using a case study: in a production environment an application server is no longer powerful enough. The ageing computer must be replaced by a higher-performance device.
- It can be carried out very elegantly in a storage network.
- Steps for Replacing a server:
  - 1. Before the exchange, the old computer is connected to a storage device via the storage network, which it uses partially **(Figure shows stages 1, 2 and 3).**

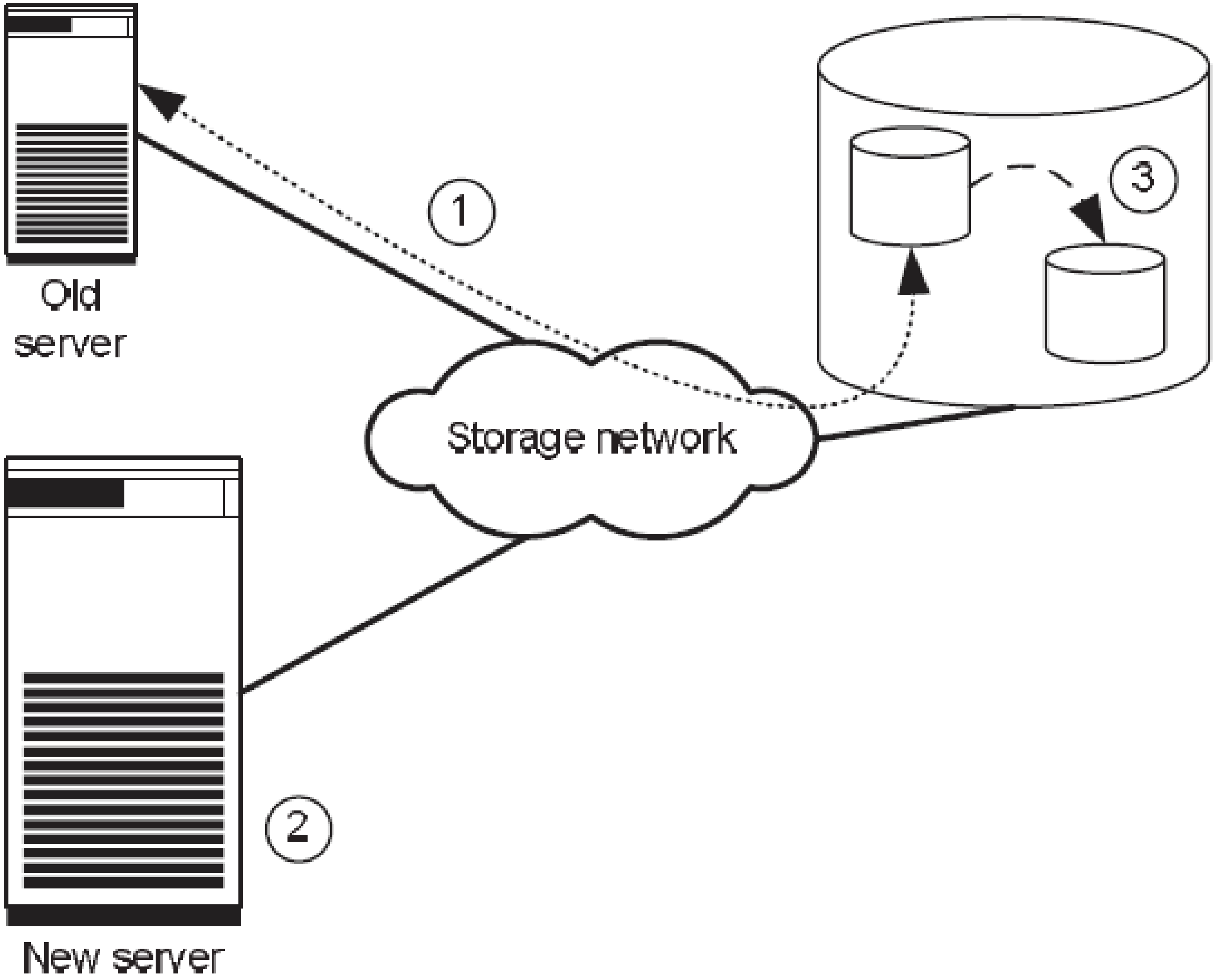

Figure 1.4 The old server is connected to a storage device via a storage network (1). The new server is assembled and connected to the storage network (2). To generate test data the production data is copied within the storage device (3)

- 2. First, the necessary application software is installed on the new computer. The new computer is then set up at the location at which it will ultimately stand.
- 3. Next, the production data for generating test data within the disk subsystem is copied. Modern storage systems can (practically) copy even terabyte-sized data files within seconds. **This function is called instant copy.**
- Then the copied data is assigned to the new computer and the new computer is tested intensively **(Figure 1.5).**
- 5. After successful testing, both computers are shut down and the production data assigned to the new server. The assignment of the production data to the new server also takes just a few seconds **(Figure 1.6 shows steps 5 and 6).**
- 6. Finally, the new server is restarted with the production data.

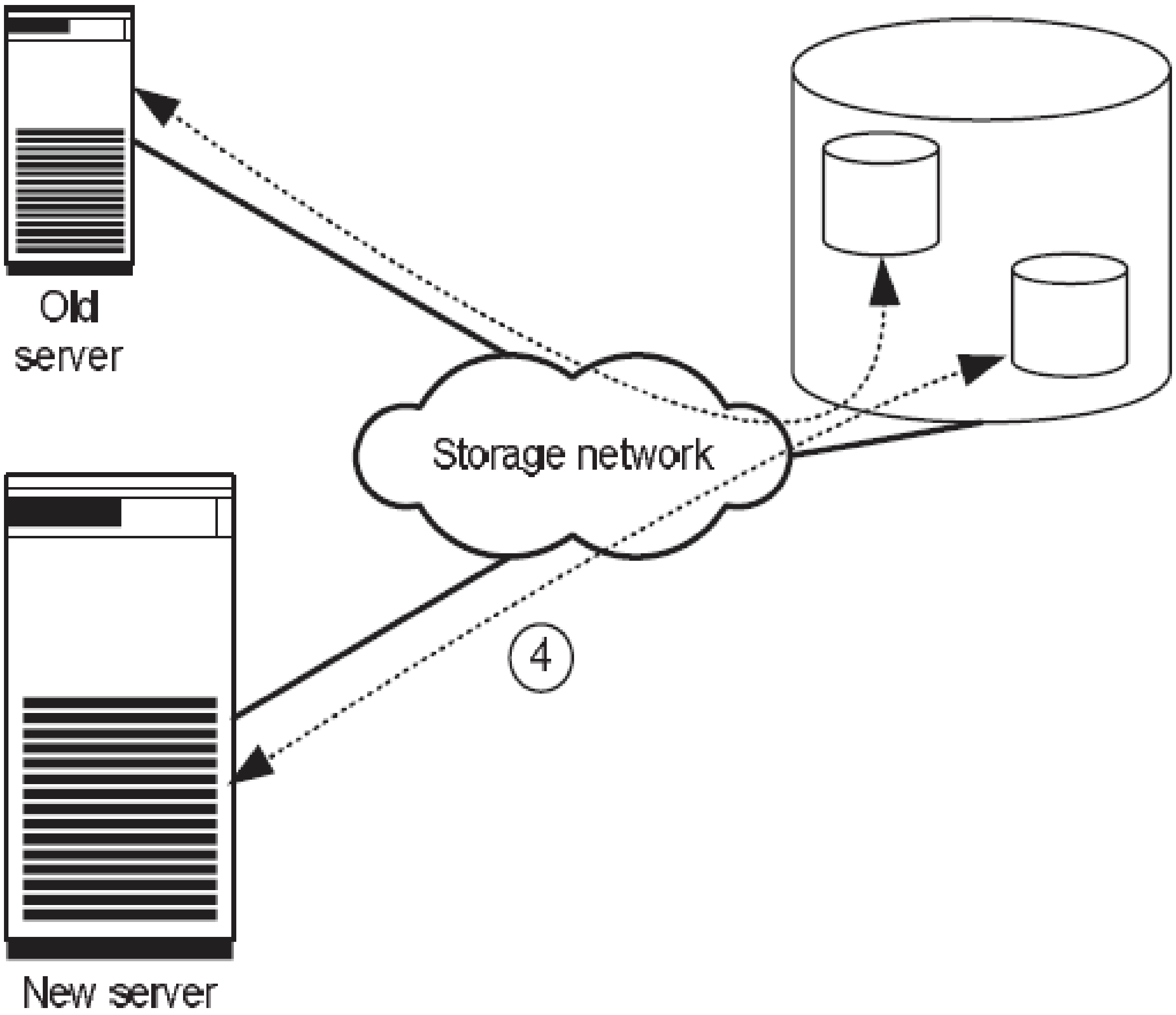

Figure 1.5 Old server and new server share the storage system. The new server is intensively tested using the copied production data (4)

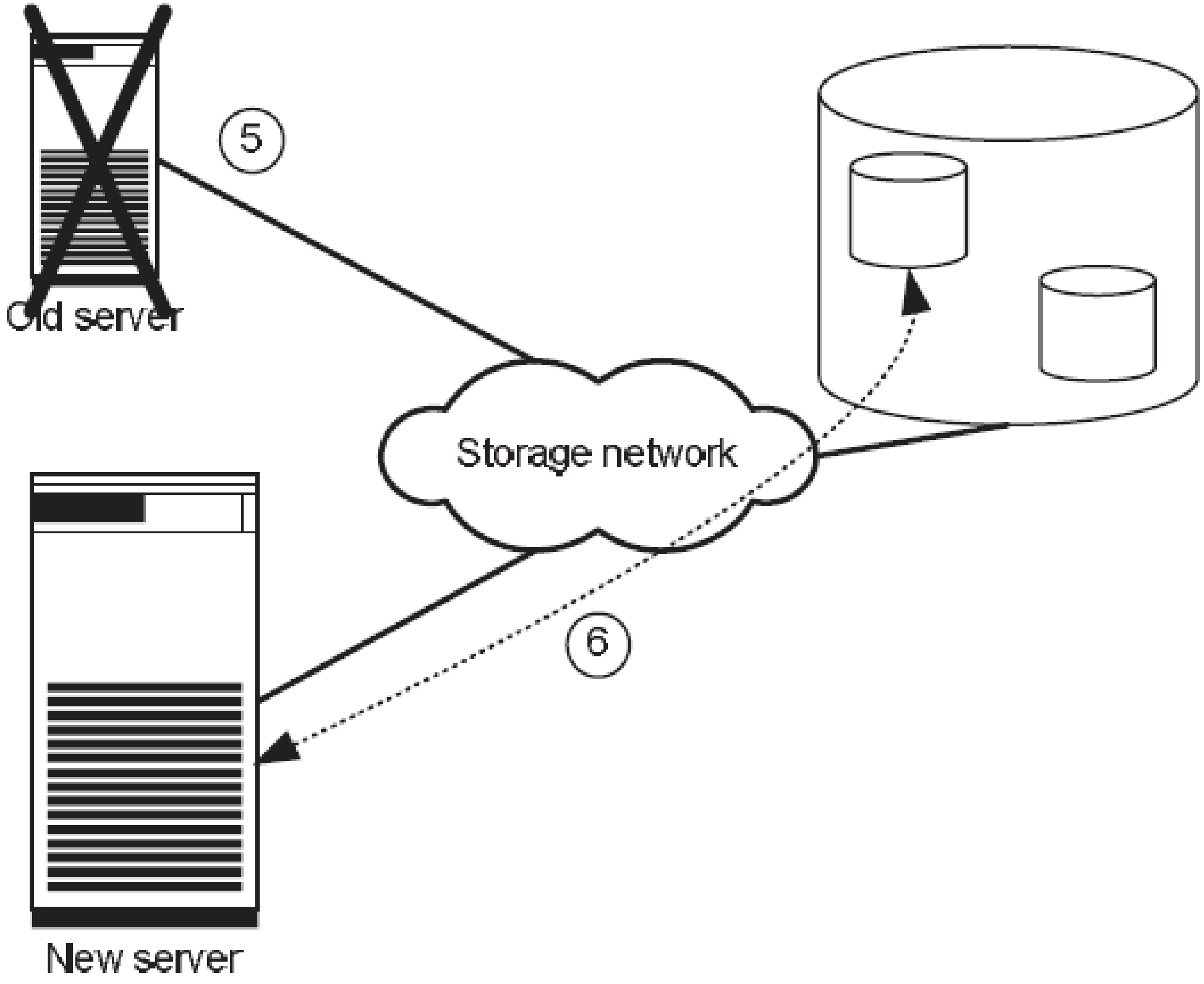

Figure 1.6 Finally, the old server is powered down (5) and the new server is started up with the production data (6)

Disk Subsystems

- When storage networks are introduced, the existing small storage devices are replaced by a few large storage systems (storage consolidation). For example, individual hard disks and small disk stacks are replaced by large disk subsystems
- That can store between a few hundred gigabytes and several ten terabytes of data, depending upon size.
- The administration of a few large storage systems is significantly simpler, and thus cheaper, than the administration of many small disk stacks.
- In contrast to a file server, a disk subsystem can be visualized as a hard disk server. Servers are connected to the connection port of the disk subsystem using standard I/O techniques. **(see next figure)**

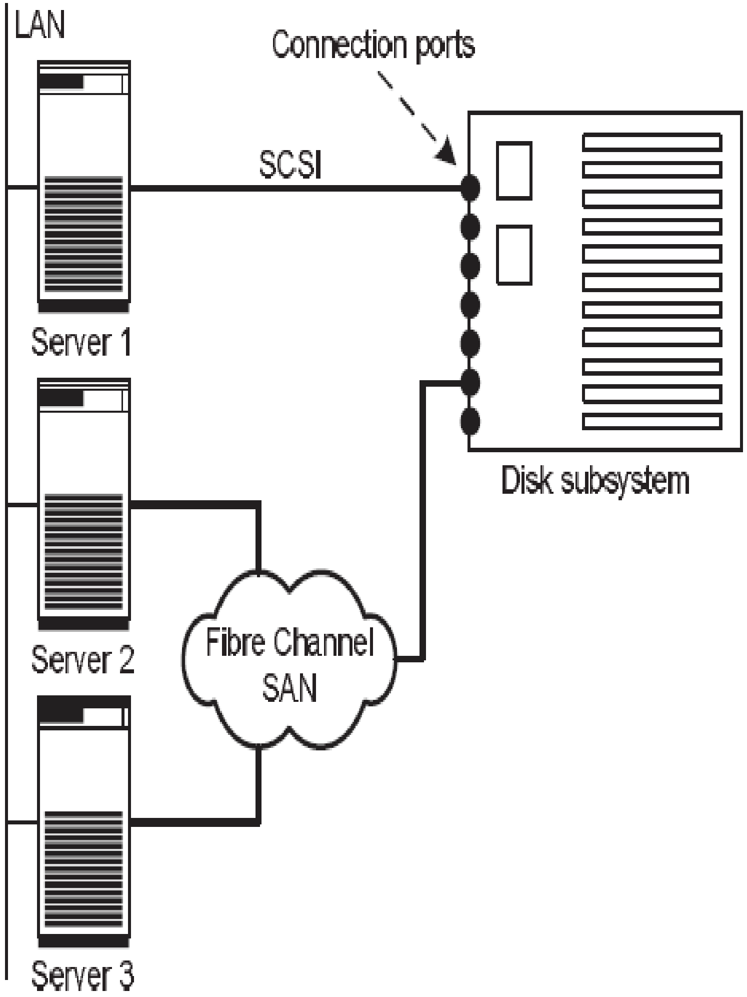

Figure 2.1 Servers are connected to a disk subsystem using standard I/O techniques. The figure shows a server that is connected by SCSI. Two others are connected by Fibre Channel SAN

- The internal structure of the disk subsystem is completely hidden from the server, which sees only the hard disks that the disk subsystem provides to the server.

- The connection ports are extended to the hard disks of the disk subsystem by means of internal I/O channels (Figure 2.2). In most disk subsystems there is a controller between the connection ports and the hard disks.
- All sizes of disk subsystems are available. Small disk subsystems have one to two connections for servers or storage networks, six to eight hard disks and – depending upon disk capacity – a storage capacity of around 500 gigabytes.
- Large disk subsystems have several ten ports for servers or storage networks, redundant controllers and several internal I/O channels
- Connection via a storage network means that a significantly greater number of servers can access the disk subsystem.

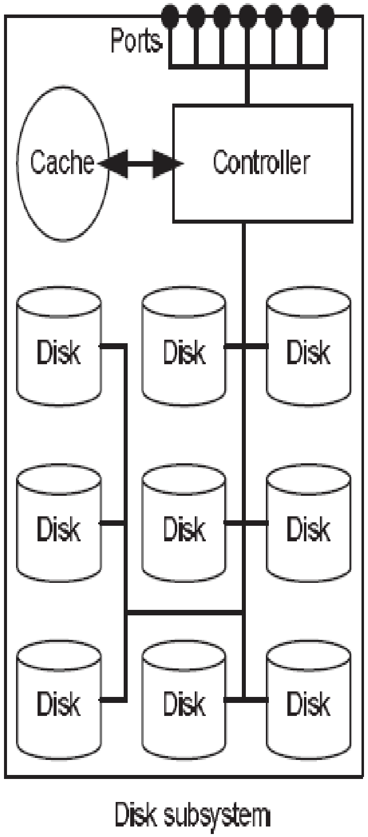

Figure 2.2 Servers are connected to the disk subsystems via the ports. Internally, the disk subsystem consists of hard disks, a controller, a cache and internal I/O channels

- Figure 2.2 shows a simplified schematic representation. The architecture of real disk subsystems is more complex and varies greatly.

- Regardless of storage networks, most disk subsystems have the advantage that free disk space can be flexibly assigned to each server connected to the disk subsystem
- In Figure 2.3 all servers are either directly connected to the disk subsystem or indirectly connected via a storage network. In this configuration each server can be assigned free storage.

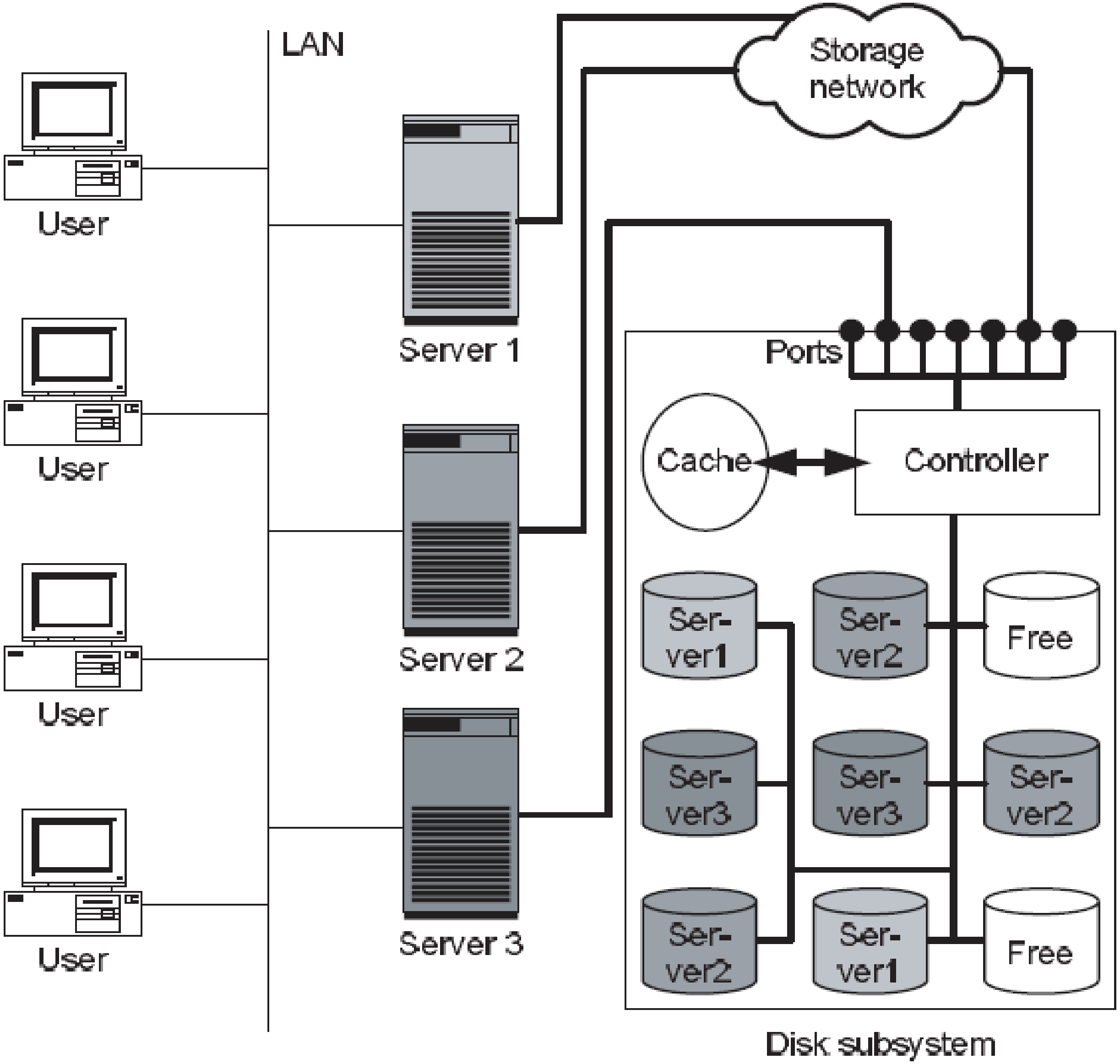

Figure 2.3 All servers share the storage capacity of a disk subsystem. Each server can be assigned free storage more flexibly as required

HARD DISKS AND INTERNAL I/O CHANNELS

- The controller of the disk subsystem must ultimately store all data on physical hard disks.
- Standard hard disks that range in size from 18 GB to 250 GB are currently (2003) used for this purpose.
- When selecting the size of the internal physical hard disks it is necessary to weigh the requirements of maximum performance against those of the maximum capacity of the overall system.
- With regard to performance it is often beneficial to use smaller hard disks at the expense of the maximum capacity: given the same capacity, if more hard disks are available in a disk subsystem, the data is distributed over several hard disks and thus the overall load is spread over more arms and read/write heads and usually over more I/O channels (Figure 2.4).

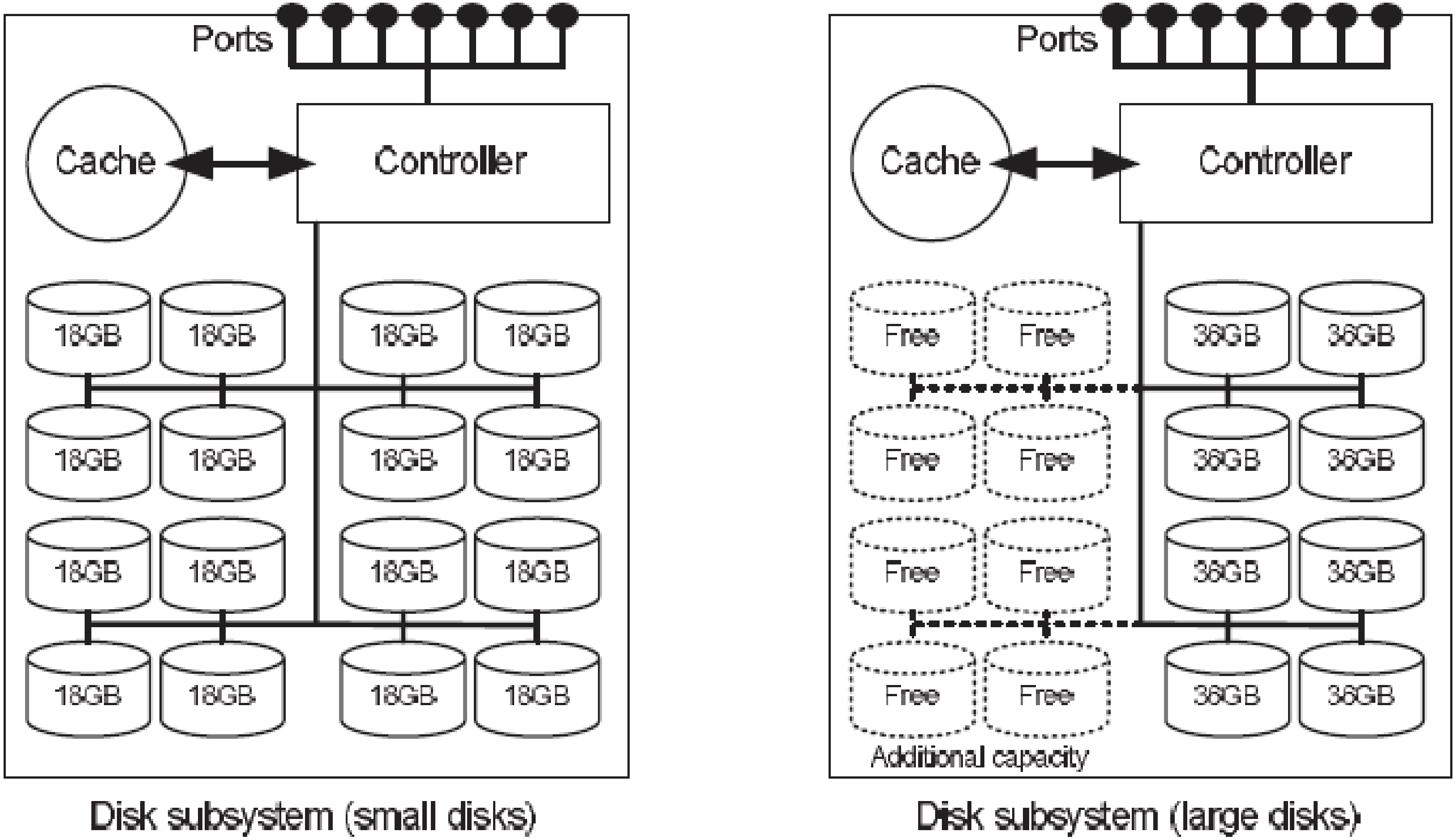

Figure 2.4 If small internal hard disks are used, the load is distributed over more hard disks and thus over more read and write heads. On the other hand, the maximum storage capacity is reduced, since in both disk subsystems only 16 hard disks can be fitted

- Standard I/O techniques such as SCSI and Fibre Channel, to an increasing degree SATA (Serial ATA) and sometimes also SSA (Serial Storage Architecture) are very often used for the internal I/O channels between connection ports and controller and between controller and internal hard disks.
- Sometimes, however, proprietary – i.e. Manufacturer specific – I/O technologies are used. Regardless of the I/O technology used, the I/O channels can be designed with built-in redundancy in order to increase the fault-tolerance of the disk subsystem.
- The following cases can be differentiated here:
  - **Active**
    - In active cabling the individual physical hard disks are only connected via one I/O channel (Figure 2.5, left). If this access path fails, then it is no longer possible to access the data.

- **Active/passive**
  - In active/passive cabling the individual hard disks are connected via two I/O channels (Figure 2.5, right). In normal operation the controller communicates with the hard disks via the first I/O channel and the second I/O channel is not used. In the event of the failure of the first I/O channel, the disk subsystem switches from the first to the second I/O channel.

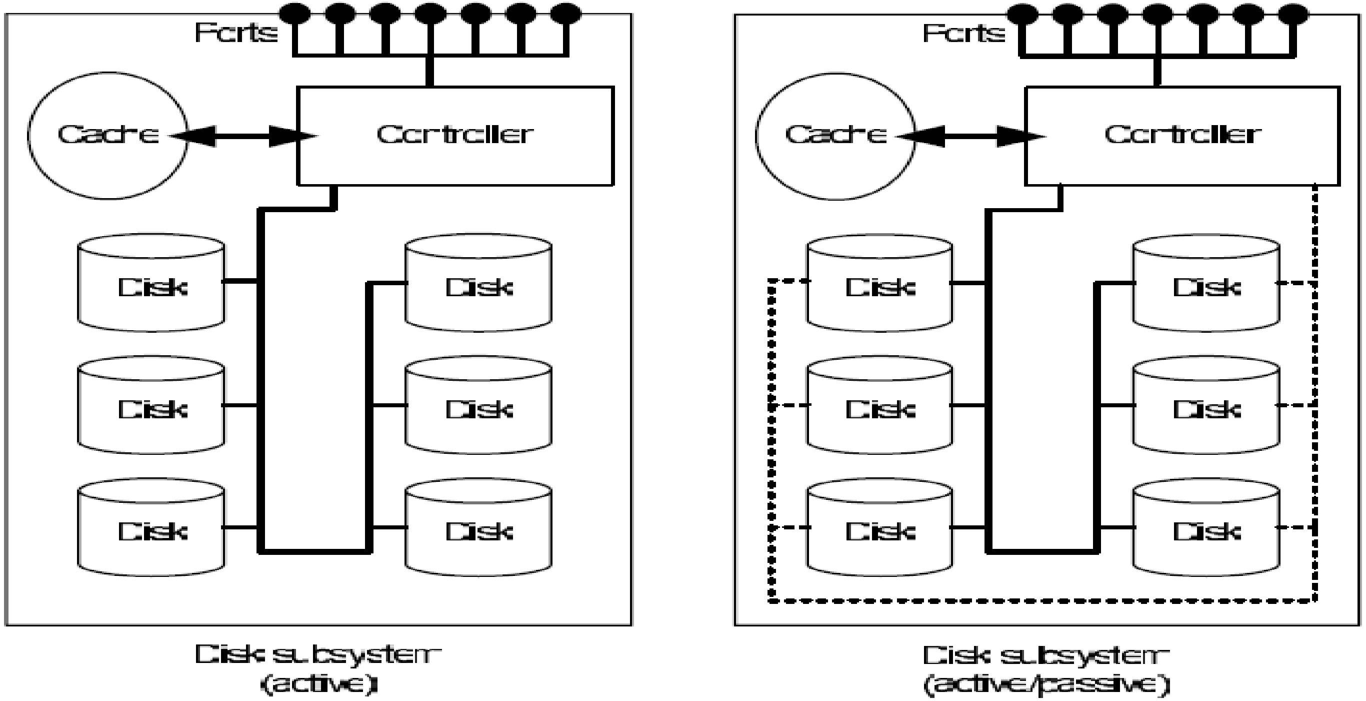

Figure 2.5 In active cabling all hard disks are connected by a just one I/O channel. In active/passive cabling all hard disks are additionally connected by a second I/O channel. If the primary I/O channel fails, the disk subsystem switches to the second I/O channel

- **Active/active (no load sharing)**
  - In this cabling method the controller uses both I/O channels in normal operation (Figure 2.6, left). The hard disks are divided into two groups: in normal operation the first group is addressed via the first I/O channel and the second via the second I/O channel. If one I/O channel fails, both groups are addressed via the other I/O channel.
- **Active/active (load sharing)**
  - In this approach all hard disks are addressed via both I/O channels in normal operation (Figure 2.6, right). The controller divides the load dynamically between the two I/O channels so that the available hardware can be optimally utilized. If one I/O channel fails, then the communication goes through the other channel only.
- Active cabling is the simplest and thus also the cheapest to realize but offers no protection against failure. Active/passive cabling is the minimum needed to protect against failure, whereas active/active cabling with load sharing best utilizes the underlying hardware.

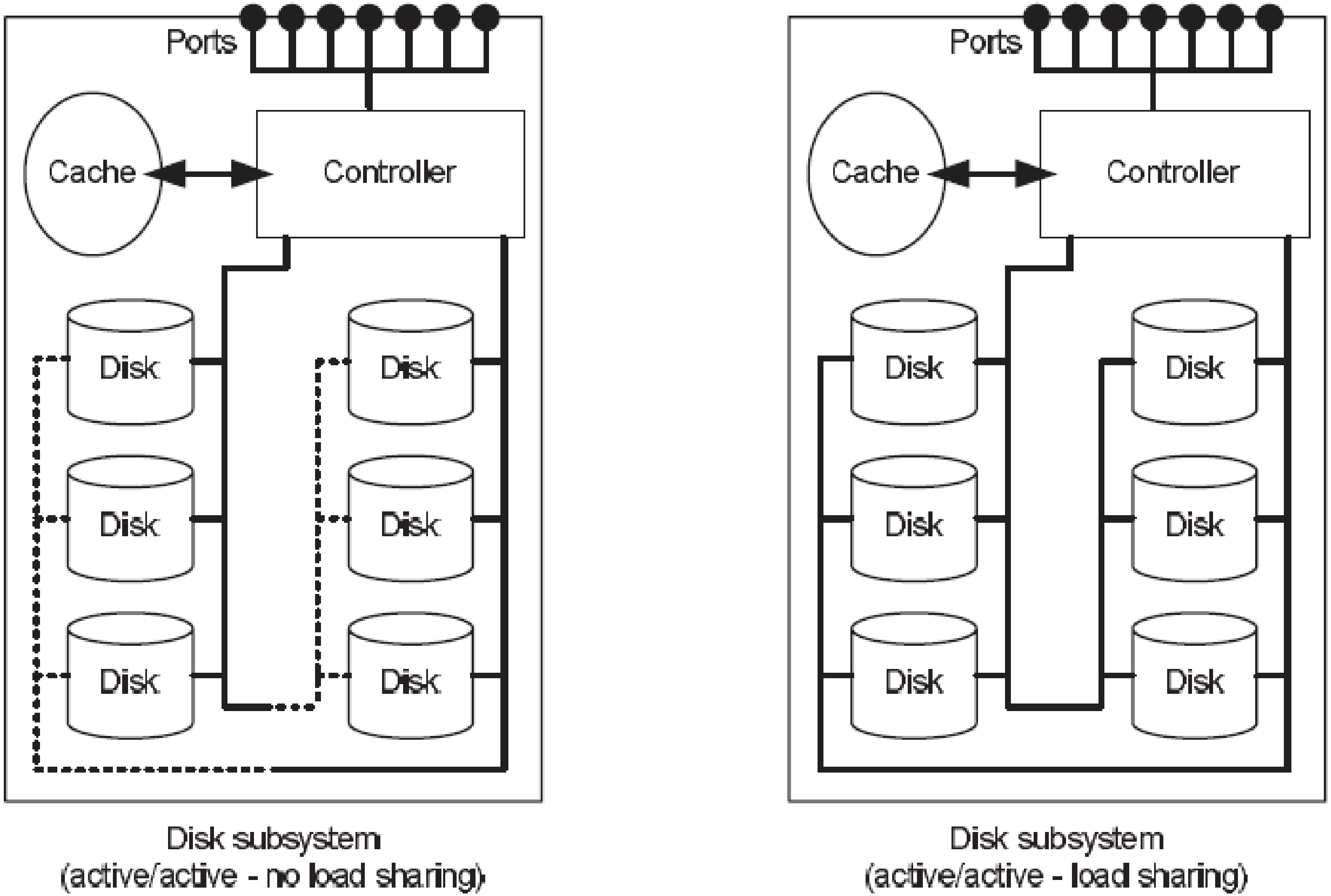

Figure 2.6 Active/active cabling (no load sharing) uses both I/O channels at the same time. However, each disk is addressed via one I/O channel only, switching to the other channel in the event of a fault. In active/active cabling (load sharing) hard disks are addressed via both I/O channels

JBOD: JUST A BUNCH OF DISKS

- If we compare disk subsystems with regard to their controllers we can differentiate between three levels of complexity:
  - **(1) No controller;**
  - **(2) RAID controller**
  - **(3) Intelligent controller** with additional services such as instant copy and remote mirroring
- If the disk subsystem has no internal controller, it is only an enclosure full of disks (Just a Bunch of Disks, JBOD).
- In this instance, the hard disks are permanently fitted into the enclosure and the connections for I/O channels and power supply are taken outwards at a single point. Therefore, a JBOD is simpler to manage than a few loose hard disks.
- Typical JBOD disk subsystems have space for 8 or 16 hard disks. A connected server recognizes all these hard disks as independent disks.

RAID

- A disk subsystem with a RAID controller offers greater functional scope than a JBOD disk subsystem.
- RAID was originally called **'Redundant Array of Inexpensive Disks'**. Today RAID stands for **'Redundant Array of Independent Disks'.**
- RAID has two main goals:
  - to increase performance by striping
  - to increase fault tolerance by redundancy.
- The bundle of physical hard disks brought together by the RAID controller are also known as **virtual hard disks**.
- A server that is connected to a RAID system sees only the virtual hard disk
- A RAID controller can distribute the data that a server writes to the virtual hard disk amongst the individual physical hard disks in various manners. These different procedures are known as **RAID levels**.

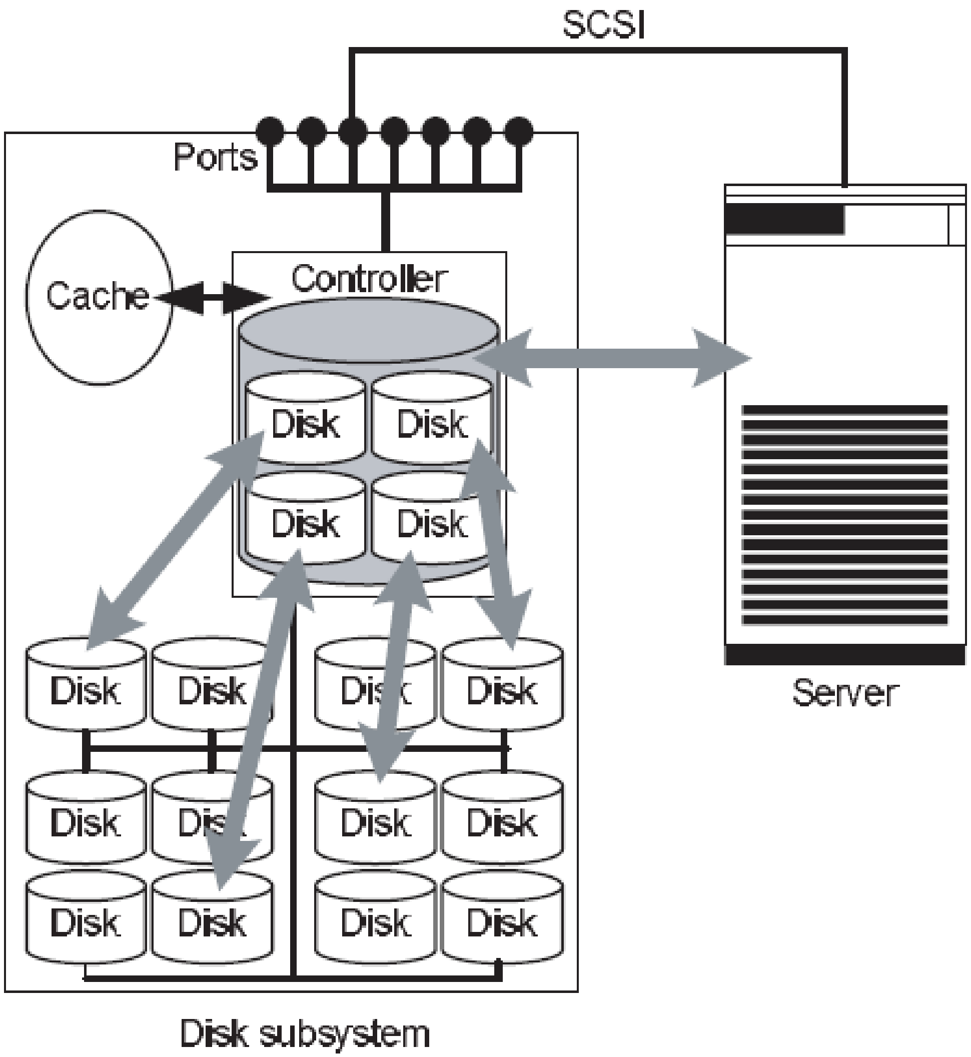

Figure 2.7 The RAID controller combines several physical hard disks to create a virtual hard disk. The server sees only a single virtual hard disk. The controller hides the assignment of the virtual hard disk to the individual physical hard disks

RAID LEVELS

- The various RAID levels are as follows:
  - **RAID 0: block-by-block striping**
  - **RAID 1: block-by-block mirroring**
  - **RAID 0+1/RAID 10: striping and mirroring combined**
  - **RAID 4 and RAID 5: parity instead of mirroring**
  - **RAID 2 and RAID 3**

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).
- 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.
- **Though 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. With 'RAID 0' standing instead for **'zero redundancy'**.

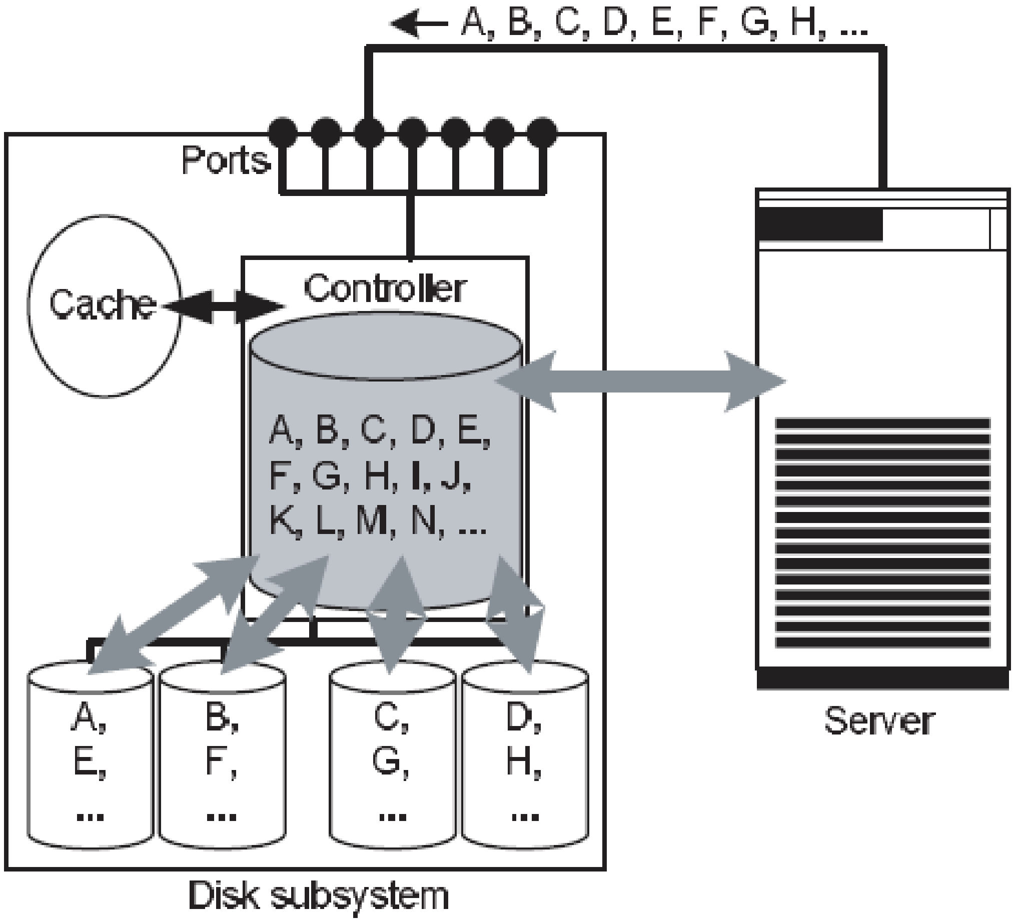

Figure 2.9 RAID 0 (striping): as in all RAID levels, the server sees only the virtual hard disk. The RAID controller distributes the write operations of the server amongst several physical hard disks. Parallel writing means that the performance of the virtual hard disk is higher than that of the individual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

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 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.

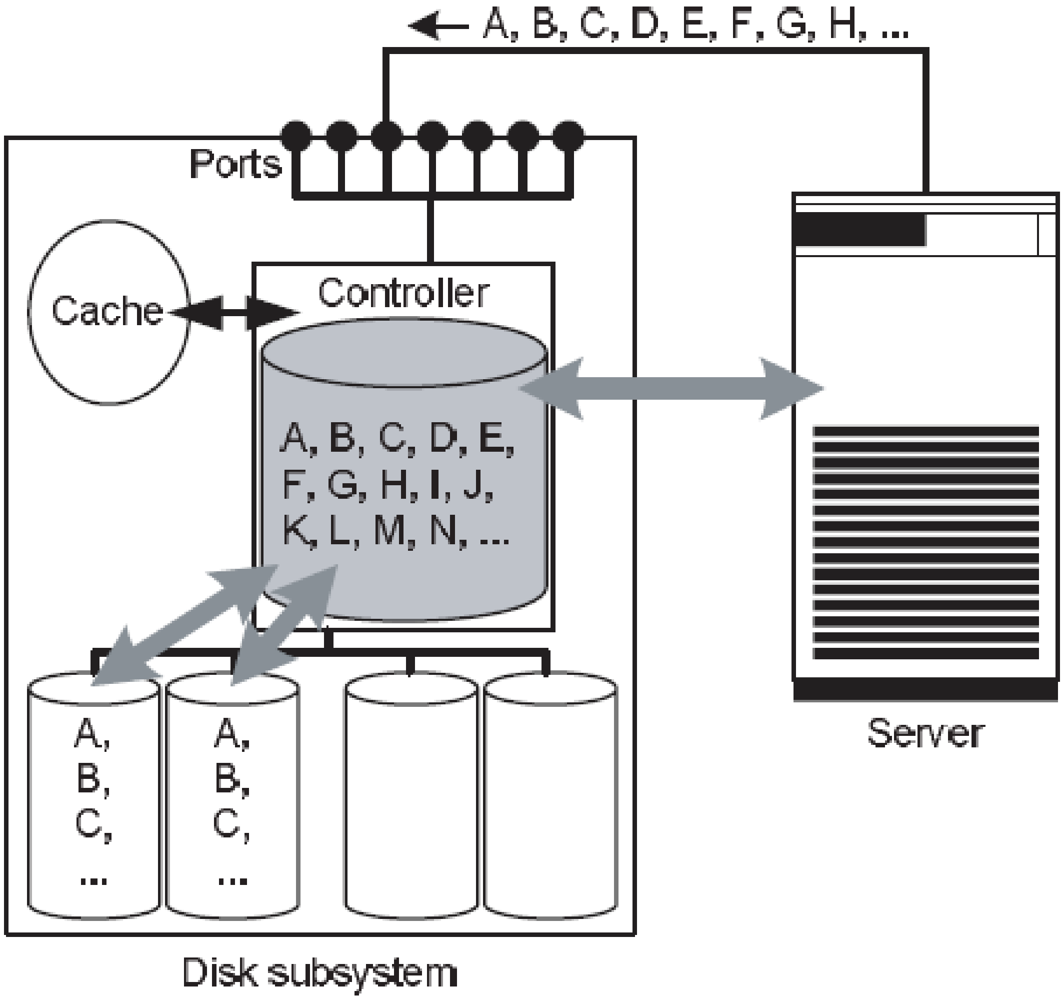

Figure 2.10 RAID 1 (mirroring): as in all RAID levels, the server sees only the virtual hard disk. The RAID controller duplicates each of the server's write operations onto two physical hard disks. After the failure of one physical hard disk the data can still be read from the other disk

RAID 0+1/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.
- 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.

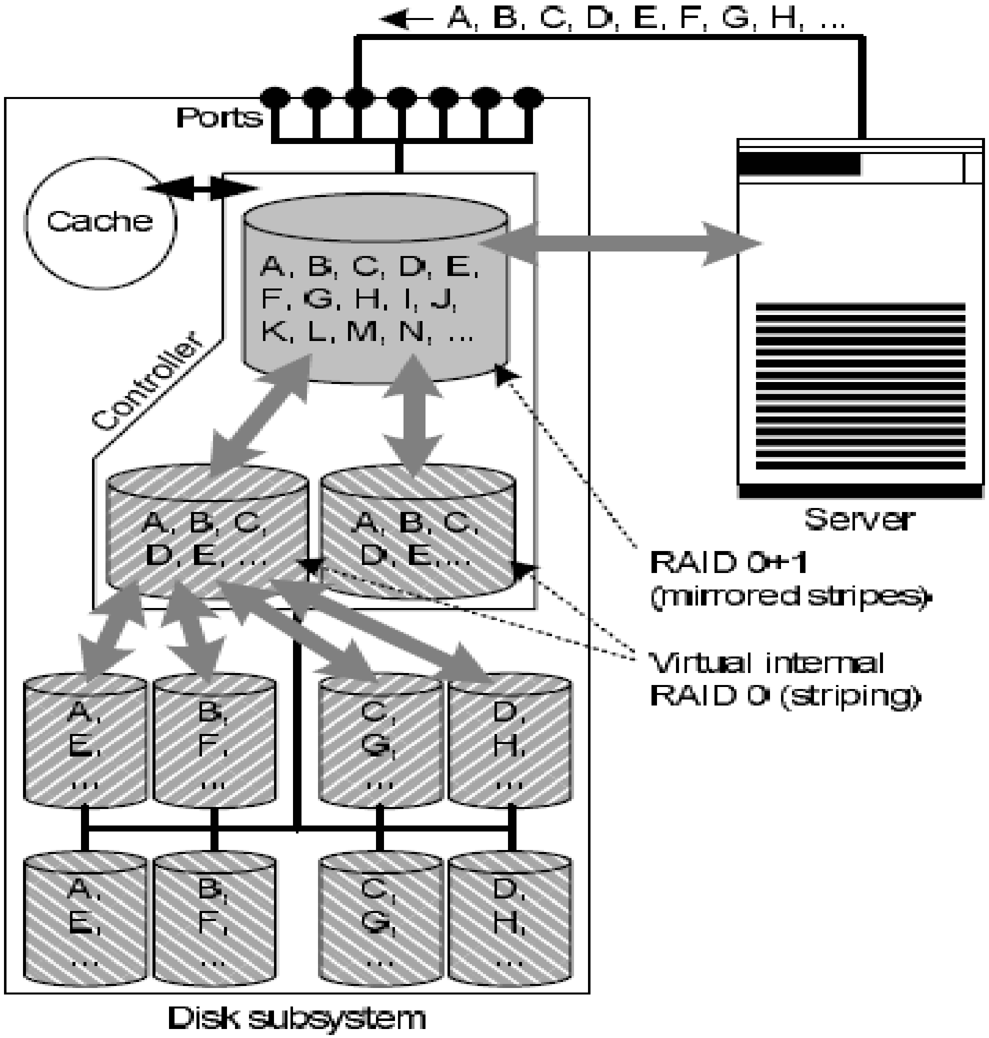

Figure 2.11 RAID 0+1 (mirrored stripes): as in all RAID levels, the server sees only the virtual hard disk. Internally, the RAID controller realizes the virtual disk in two stages: in the first stage it brings together every four physical hard disks into one virtual hard disk that is only visible within 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

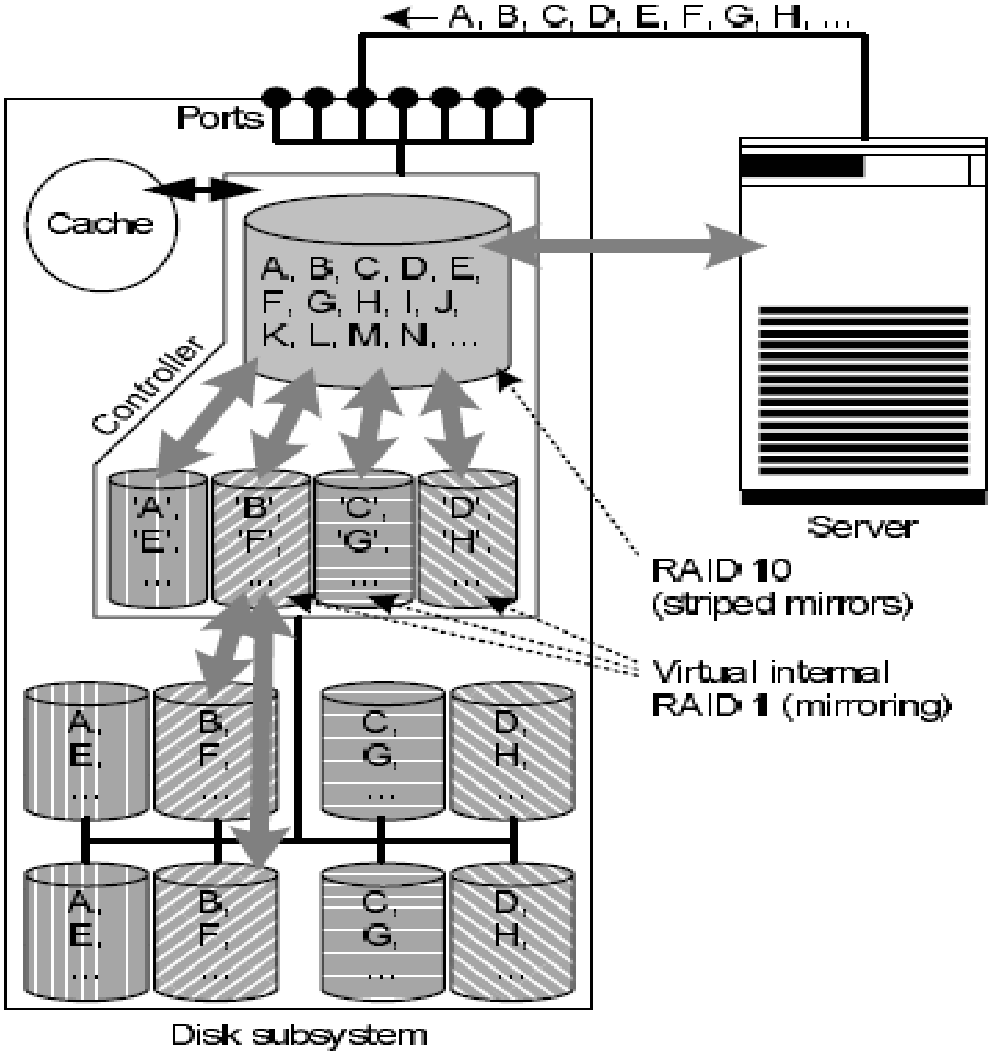

Figure 2.12 RAID 10 (striped mirrors): as in all RAID levels, the server sees only the virtual hard disk. Here too, we proceed in two stages. The sequence of striping and mirroring is reversed in relation to RAID 0+1. In the first stage the controller links every two physical hard disks by means of RAID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

WHICH OF THE TWO RAID LEVELS, RAID 0+1 OR RAID 10, IS PREFERABLE?

- 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)
- 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.

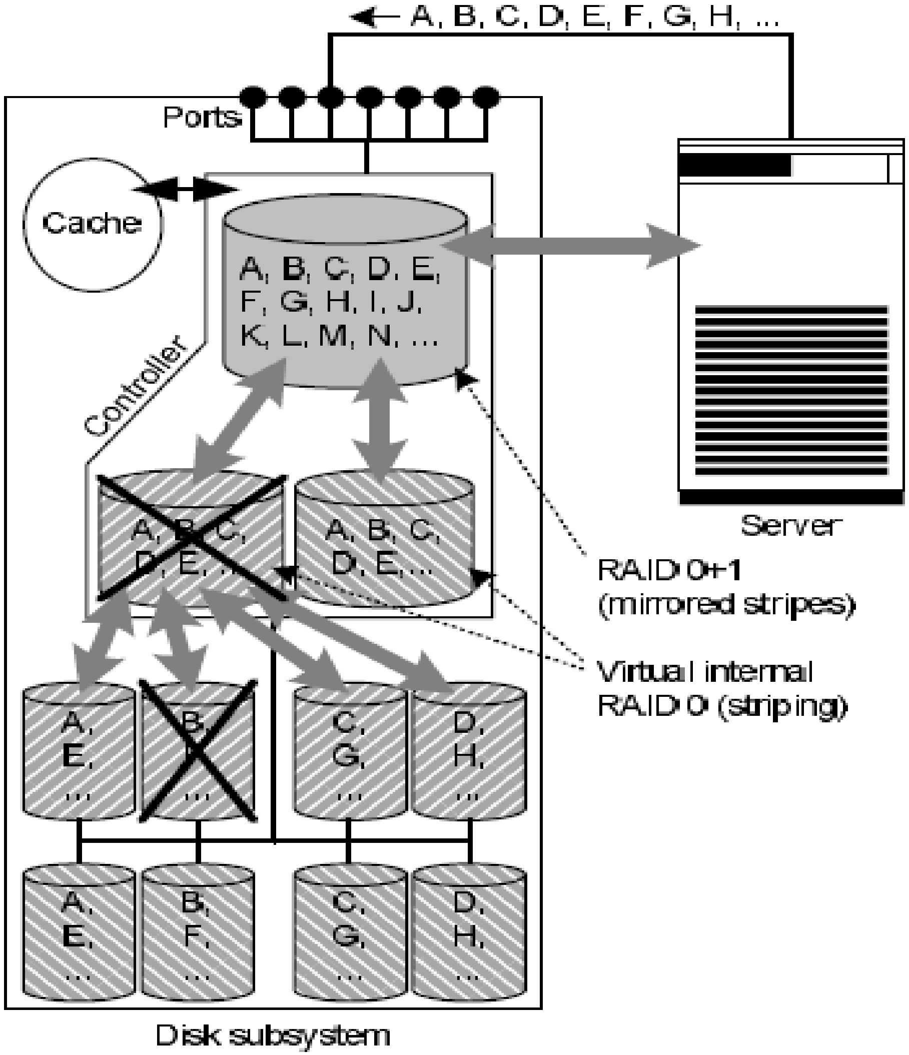

Figure 2.13 The consequences of the failure of a physical hard disk in RAID 0+1 (mirrored stripes) are relatively high in comparison to RAID 10 (striped mirrors). The failure of a physical hard disk brings about the failure of the corresponding internal RAID 0 disk, so that in effect half of the physical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

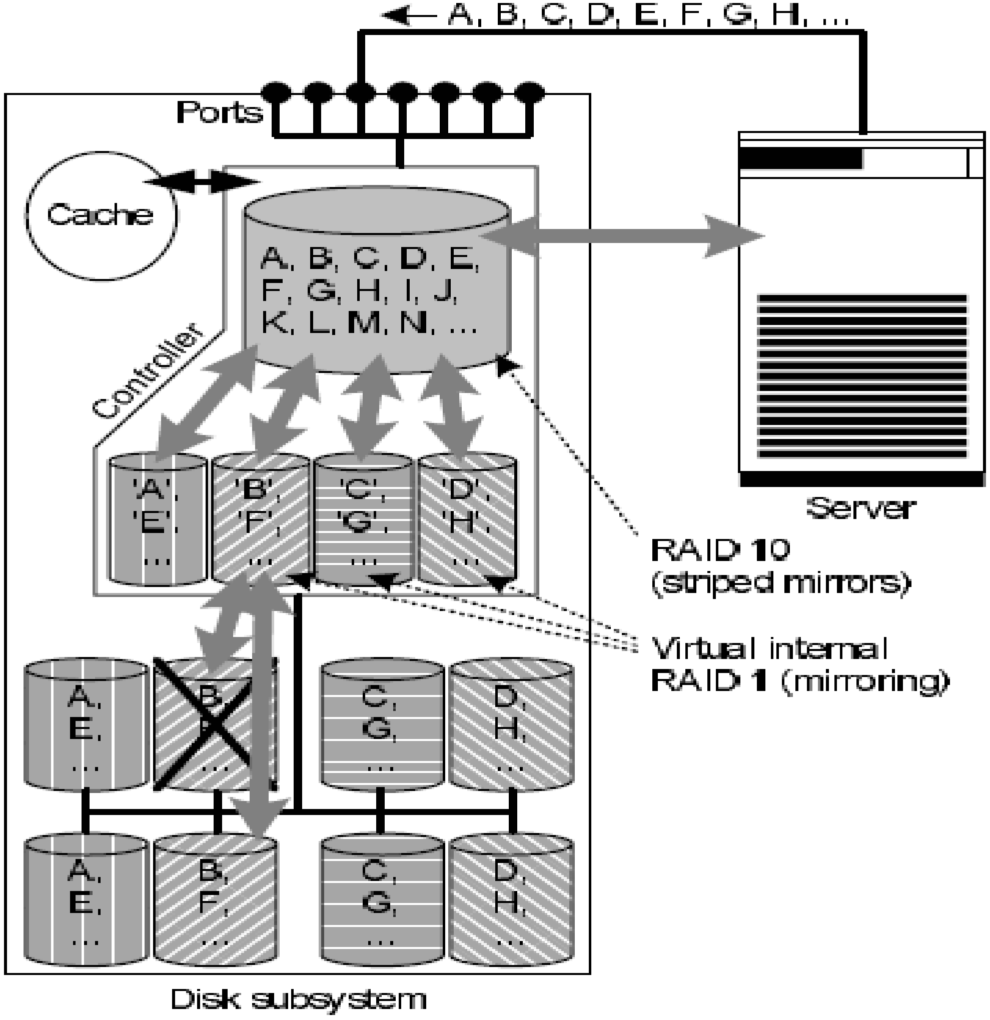

Figure 2.14 In RAID 10 (striped mirrors) the consequences of the failure of a physical hard disk are not as serious as in RAID 0+1 (mirrored stripes). All virtual hard disks remain intact. The restoration of the data from the failed hard disk is simple

RAID PARITY

- The term parity is used for the system memory error detection.
- The concept of RAID parity is similar to the concept to parity RAM.
- The principle of RAID parity is simple. For example, the RAID parity can be explained as consider the N pieces of data and compute extra piece of data for each of the following. Then take N+ 1 piece of data and try to store it in N+1 drives. You may have a chance to lose any of the N+ 1 of data, you can use remaining of the data regardless in which piece may be lost.
- The **parity information can be easily store on a separate or a dedicated drive or data across all the drives.**
- The term parity calculation is used when a logical operation such as **XOR or OR logical operator is true**. XOR operator is true if one of its operands is true.

- The implementation of the RAID Parity has a high cost, because the drives has a duplicate data and improve performance for many applications.
- The RAID levels such as RAID 3, RAID5 and RAID 7 uses a concept which is exactly similar to parity but which is not exactly the same as the RAID parity.
- **The main aim is to use the RAID parity is to better the performance in a RAID.** RAID improves performance in different ways and buy RAID parity the performance is tremendously increased to very large extend.
- As a result the RAID parity is used for higher levels of RAID in which accuracy of the application performance matters to a very large extend.

RAID 4 AND RAID 5: PARITY INSTEAD OF MIRRORING

- As we have seen that RAID 10 provides excellent performance at a high level of fault-tolerance, but 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 and then 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.**

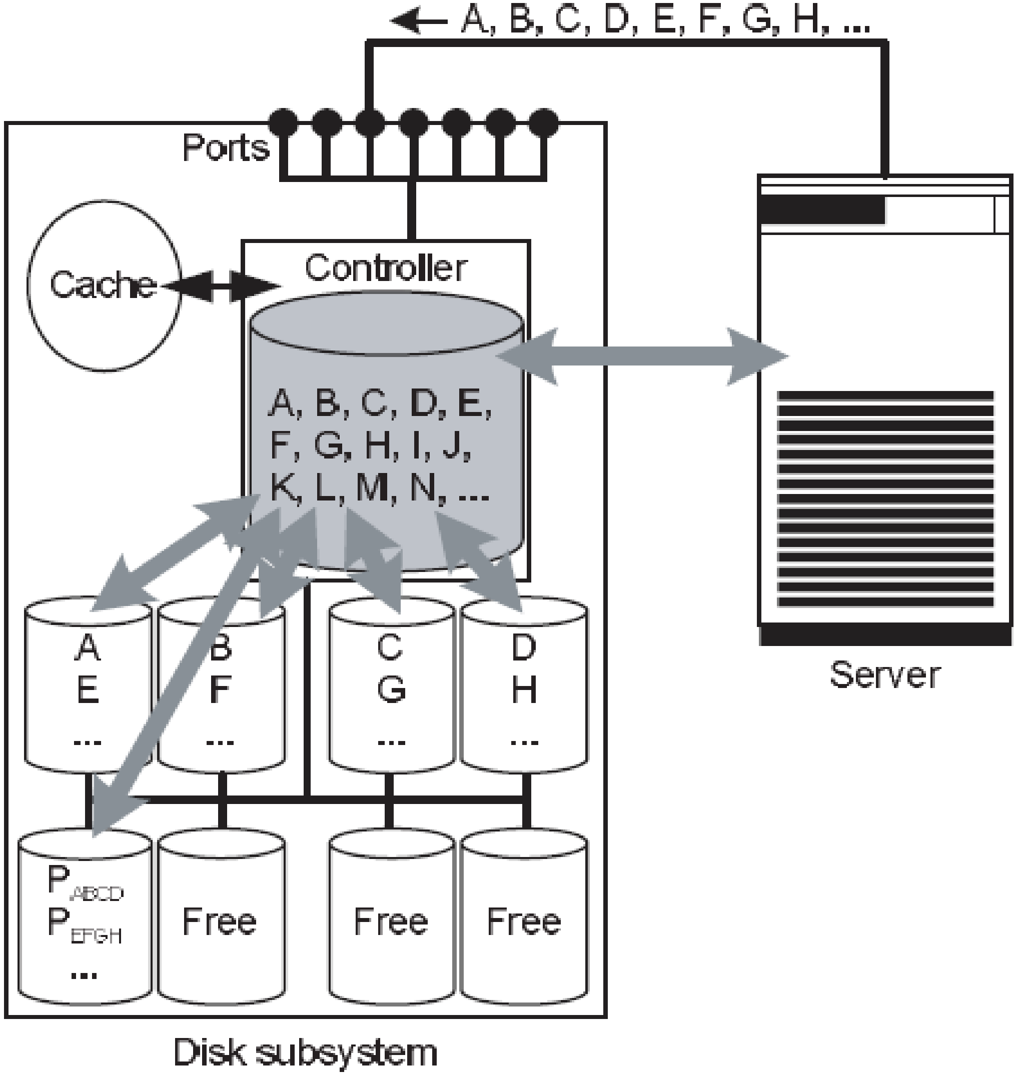

Figure 2.15 RAID 4 (parity disk) is designed to reduce the storage requirement of RAID 0+1 and RAID 10. In the example, the data blocks are distributed over four physical hard disks by means of RAID 0 (striping). Instead of mirroring all data once again, only a parity block is stored for each four blocks

- For example, the RAID controller calculates the parity block P_{ABCD} 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.
- If we compare it with examples in Figures 2.11 **(RAID 0+1) and 2.12 (RAID 10) then** RAID 4 saves three physical hard disks. This situation is shown in next slide (Figure 2.15).
- From a mathematical point of view the parity block is calculated with the aid of the **logical XOR operator (Exclusive OR)**.

$$P_{ABCD} = A \text{ XOR } B \text{ XOR } C \text{ XOR } D$$

- 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.

- RAID 4 saves all parity blocks onto a single physical hard disk. 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.
- Unlike RAID 4, however, in RAID 5 the parity block P_{EFGH} moves to the fourth physical hard disk for the next four blocks E, F, G, H.

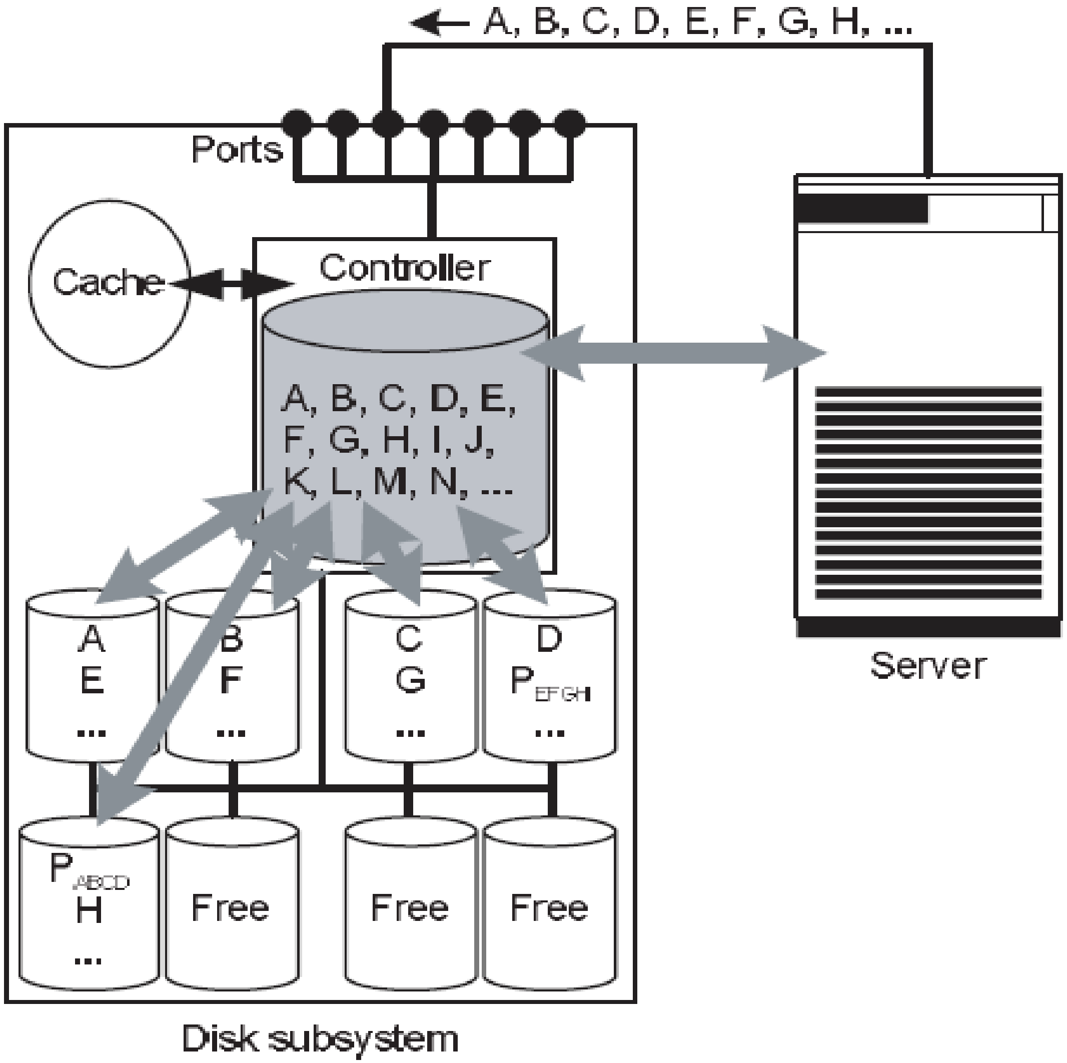

Figure 2.17 RAID 5 (striped parity): in RAID 4 each write access by the server is associated with a write operation to the parity disk for the comparison of parity information. RAID 5 distributes the load of the parity disk over all physical hard disks

A comparison of the RAID levels

- The various RAID levels raise the question of **which RAID level should be used when.**
- Table 2.1 compares the criteria of **fault-tolerance, write performance, read performance and space requirement** for the individual RAID levels.
- *CAUTION PLEASE: The comparison of the various RAID levels discussed in this section is only applicable to the theoretical basic forms of the RAID level in question.*
- *In practice, manufacturers of disk subsystems have design options in:*
  - *the selection of the internal physical hard disks;*
  - *the I/O technique used for the communication within the disk subsystem;*
  - *the use of several I/O channels;*
  - *the realization of the RAID controller;*
  - *the size of the cache; and*
  - *the cache algorithms themselves.*

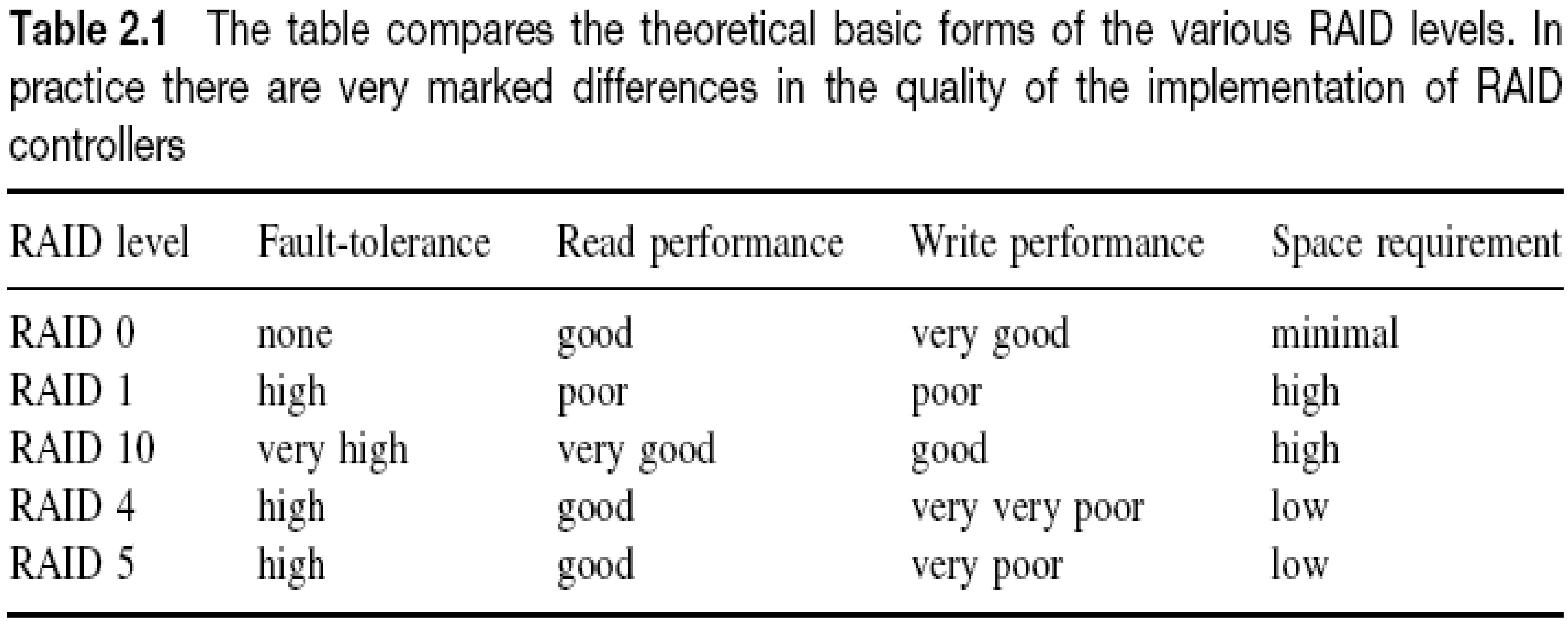

Table 2.1 The table compares the theoretical basic forms of the various RAID levels. In practice there are very marked differences in the quality of the implementation of RAID controllers

RAID level	Fault-tolerance	Read performance	Write performance	Space requirement
RAID 0	none	good	very good	minimal
RAID 1	high	poor	poor	high
RAID 10	very high	very good	good	high
RAID 4	high	good	very very poor	low
RAID 5	high	good	very poor	low

RAID 4 and RAID 5 save disk space at the expense of a poorer write performance. For a long time the rule of thumb was to use RAID 5 where the ratio of read operations to write operations is 70 : 30. At this point we wish to repeat that there are now storage systems on the market with excellent write performance that store the data internally using RAID 4 or RAID 5.

HOT SPARING

- As we have seen that one factor common to almost all RAID levels is that **they store redundant information.**
- If a physical hard disk fails, its data can be reconstructed from the hard disks that remain intact.
- The defective hard disk can even be replaced by a new one during operation if a disk subsystem has the appropriate hardware. Then the RAID controller reconstructs the data of the exchanged hard disk.
- Modern RAID controllers initiate this process automatically. This requires **the definition of so-called hot spare disks (Figure 2.8).**
- The hot spare disks are not used in normal operation. If a disk fails, the RAID controller immediately begins to copy the data of the remaining intact disk onto a hot spare disk.

CONT'D

- After the replacement of the defective disk, this is included in the pool of hot spare disks. Modern RAID controllers can manage a common pool of hot spare disks for several virtual RAID disks.
- Hot spare disks can be defined for all RAID levels that offer redundancy.
- The recreation of the data from a defective hard disk takes place at the same time as write and read operations of the server to the virtual hard disk, so that from the point of view of the server, performance reductions at least can be observed.

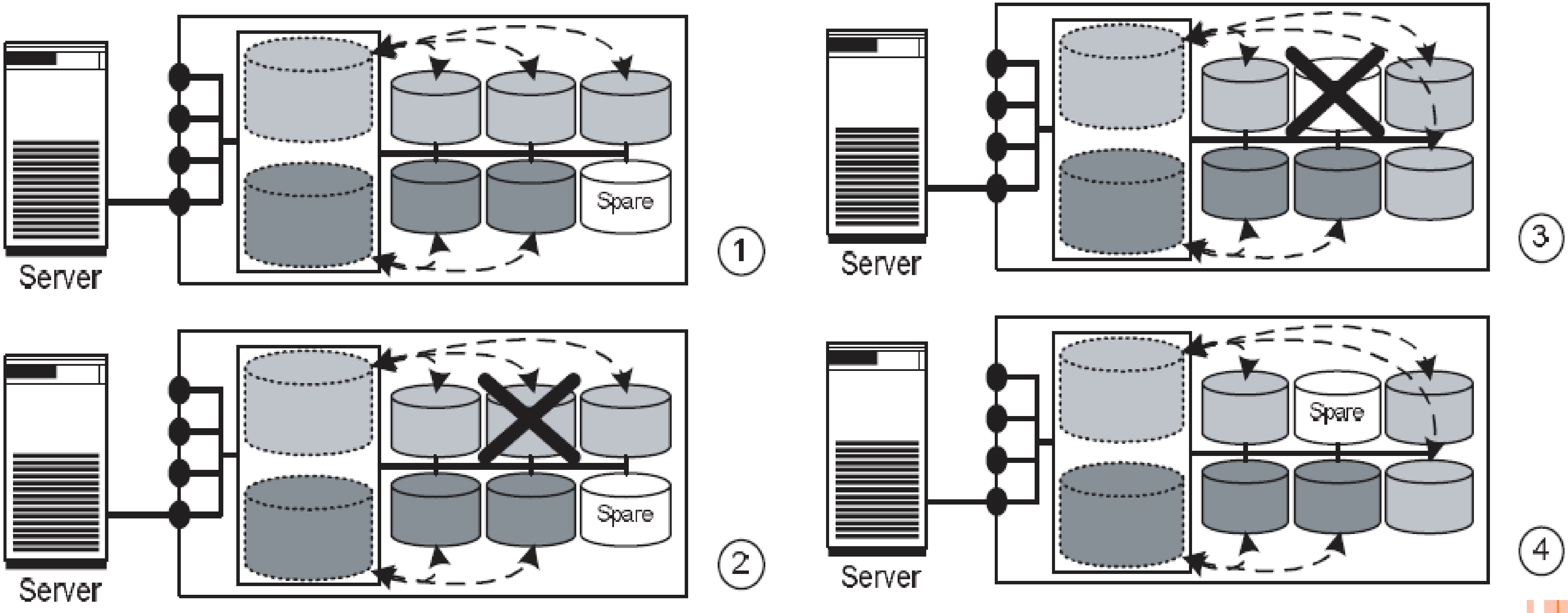

Figure 2.8 Hot spare disk: the disk subsystem provides the server with two virtual disks for which a common hot spare disk is available (1). Due to the redundant data storage the server can continue to process data even though a physical disk has failed, at the expense of a reduction in performance (2). The RAID controller recreates the data from the defective disk on the hot spare disk (3). After the defective disk has been replaced a hot spare disk is once again available (4)

HOT SWAPPING IN RAID

- The most of the available types of the RAID; support the concept of the hot swapping. The hot swapping can be even implemented in the IDE RAID, SCSI RAID, and also the SATA RAID.
- The term hot swapping means that changing the hard disk drive with out even switching off the power. Even if the disk drive fails then also the system is not halted for the purpose of even changing the hard disk.
- This ensures that the server runs well with out any interruptions.
- More over there are different types of swapping techniques that are available:
  - Hot Swap,
  - Warm Swap
  - Cold Swap.

- The warm swapping is the one where in the system is not switched off and the power supply is not discontinued but the **user processes needs to be stopped.**
- The warm swapping halts the work that is being done.
- In cold swapping the normal process of the swapping takes place where in the hard disk power supply and the power to the **system must be switched off** in order to replace the hard disk drive.
- This is nothing but the normal replacement that is carried out for the replacement of the hard disk drive.

CACHING: ACCELERATION OF HARD DISK ACCESS

- In all fields of computer systems, caches are used to speed up slow operations by operating them from the cache.
- Specifically in the field of disk subsystems, caches are designed to accelerate **write and read accesses to physical hard disks.**
- In this connection we can differentiate between two types of cache:
  - (1) cache on the hard disk
  - (2) cache in the RAID controller.

The **cache in the RAID controller is subdivided into write cache and read cache.**

1) Cache on the hard disk

- Each individual hard disk comes with a very small cache. If a server or a RAID controller writes a block to a physical hard disk, the disk controller stores this in its cache.
- The disk controller can thus write the block to the physical hard disk in its own time whilst the I/O channel can be used for data traffic to the other hard disk.
- Read access is accelerated in a similar manner. If a server or RAID controller wishes to read a block, it sends the address of the requested block to the hard disk controller.
- The hard disk controller transfers the block from its cache to the RAID controller or to the server at the higher data rate of the I/O channel.

2a) Write cache in the controller of the disk subsystem

- In addition to the cache of the individual hard drives **many disk subsystems come with their own cache**, which **in some models is gigabytes in size**.
- The write cache should have a battery back-up and ideally be mirrored. The battery back-up is necessary to allow the data in the write cache to survive a power cut.
- If a server sends several data blocks to the disk subsystem, the **controller initially buffers all blocks into a write cache** and immediately reports back to the server that all data has been securely written to the drive.
- The disk subsystem then copies the data from the write cache to the slower physical hard disk in order to make space for the next write peak.

2B) Read cache in the RAID controller

- The **acceleration of read operations is difficult** in comparison to the acceleration of write operations using cache.
- To speed up read access by the server, the disk subsystem's controller must copy the relevant data blocks from the slower physical hard disk to the fast cache before the server requests the data in question.
- **The problem with this is that it is very difficult for the disk subsystem's controller to work out in advance what data the server will ask for next.**
- Consequently, the controller can only analyze past data access and use this to extrapolate which data blocks the server will access next. In sequential read processes this prediction is comparatively simple, in the case of random access it is almost impossible.

INTELLIGENT DISK SUBSYSTEMS

- Intelligent disk subsystems represent the third level of complexity for controllers after JBODs and RAID arrays.
- The controllers of intelligent disk subsystems offer additional functions over and above those offered by RAID.
- In the Intelligent disk subsystems that are currently available on the market these functions are usually
  - **Instant copies**
  - **Remote mirroring**
  - **LUN masking.**

1) Instant copies

- Instant copies can practically copy data sets of several terabytes within a disk subsystem in a few seconds.
- Instant copies are used, for example, for the **generation of test data, for the backup of data and for the generation of data copies for data mining.**
- All realizations of instant copies require controller computing time and cache and place a load on internal I/O channels and hard disks.

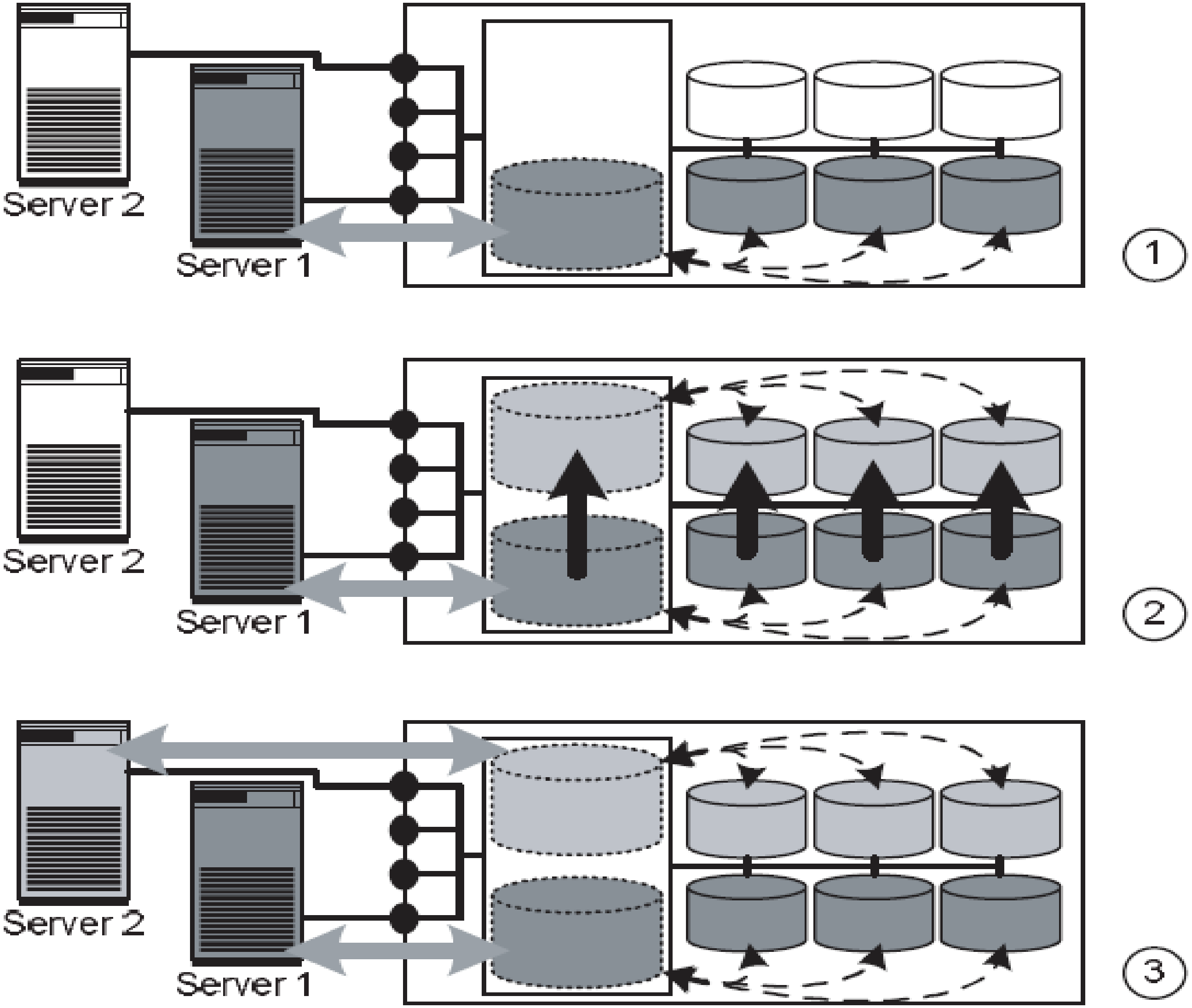

Figure 2.18 Instant copies can practically copy several terabytes of data within a disk subsystem in a few seconds: server 1 works on the original data (1). The original data is practically copied in a few seconds (2). Then server 2 can work with the data copy, whilst server 1 continues to operate with the original data (3)

2) 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.
- A fire in the disk subsystem would destroy original data and data copies.
- **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.**

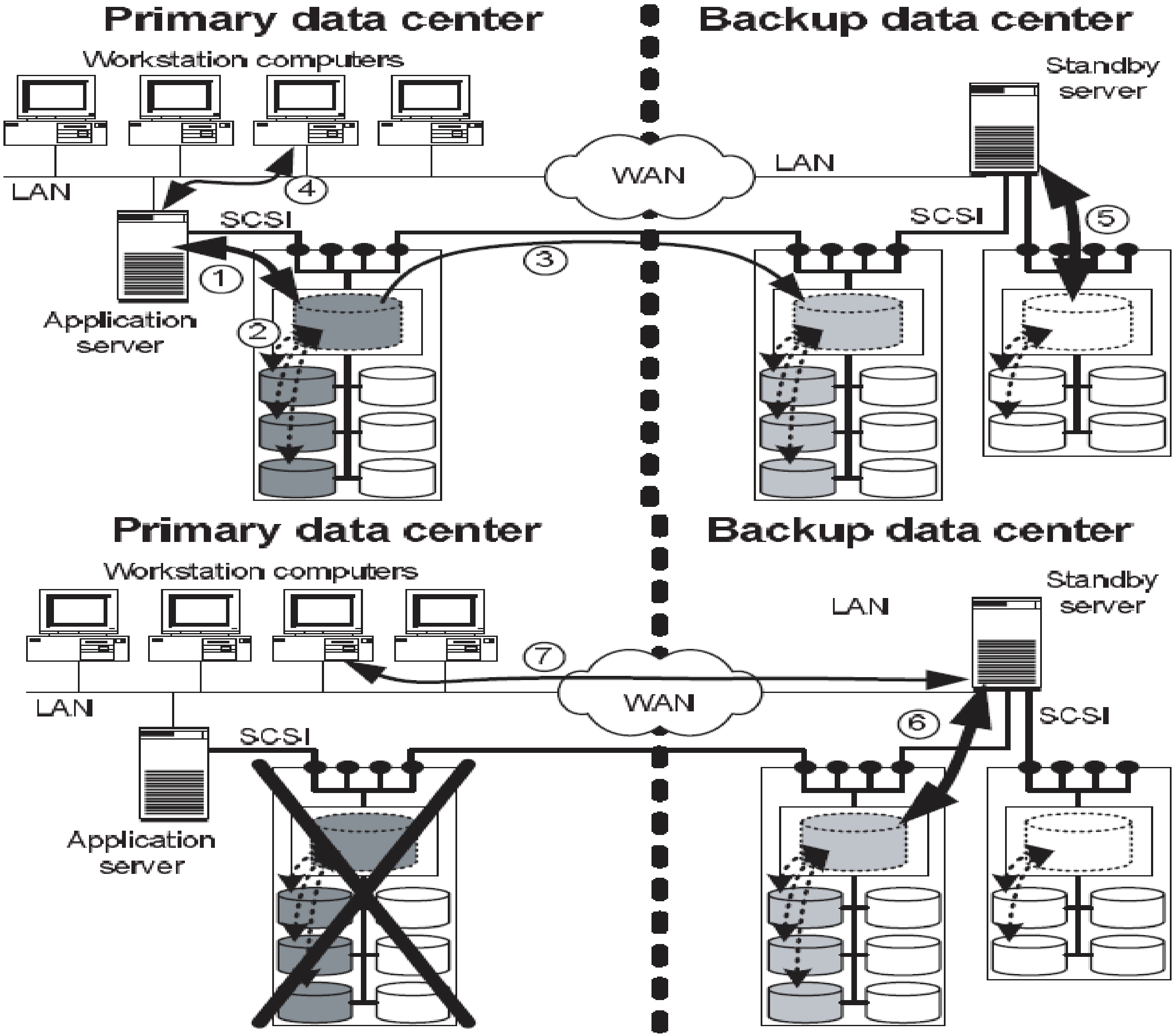

Figure 2.19 High availability with remote mirroring: (1) The application server stores its data on a local disk subsystem. (2) The disk subsystem saves the data to several physical drives by means of RAID. (3) The local disk subsystem uses remote mirroring to mirror the data onto a second disk subsystem located in the back-up data centre. (4) Users use the application via the LAN. (5) The stand-by server in the back-up 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 2.20 illustrates the data flow of synchronous remote mirroring.

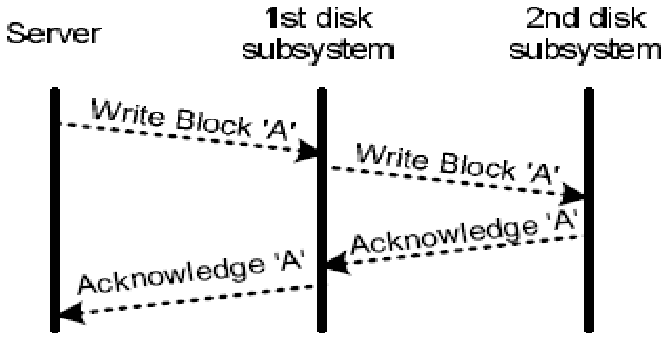

Figure 2.20 In synchronous remote mirroring a disk subsystem does not acknowledge write operations until it has saved a block itself and received write confirmation from the second disk subsystem

- Synchronous remote mirroring has the advantage that the copy of the data held by the second disk subsystem is always up-to-date.
- The disadvantage is that copying the data from the first disk subsystem to the second and sending the write acknowledgement back from the second to the first increases the response time of the first disk subsystem to the server.
- Figure 2.21 illustrates the data flow in asynchronous remote mirroring.

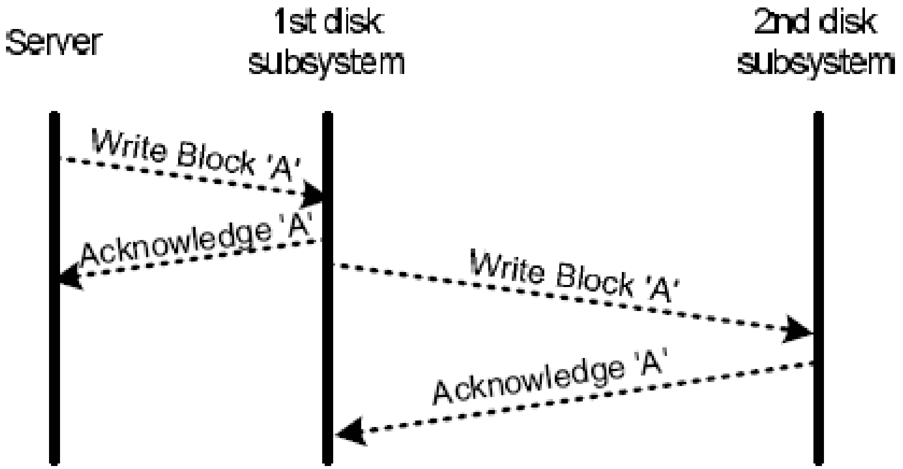

Figure 2.21 In asynchronous remote mirroring one disk subsystem acknowledges a write operation as soon as it has saved the block itself

3) LUN MASKING

- LUN (Logical Unit Number) masking brings us to the third important function – after instant copy and remote mirroring – that intelligent disk subsystems offer over and above that offered by RAID.
- **LUN masking limits the access to the hard disks** that the disk subsystem exports to the connected server.
- Based upon the **SCSI protocol**, all hard disks – physical and virtual – that are visible outside the disk subsystem are also known as LUN (Logical Unit Number).
- Without LUN masking every server would see all hard disks that the disk subsystem provides. Shown in **(Figure 2.23)**

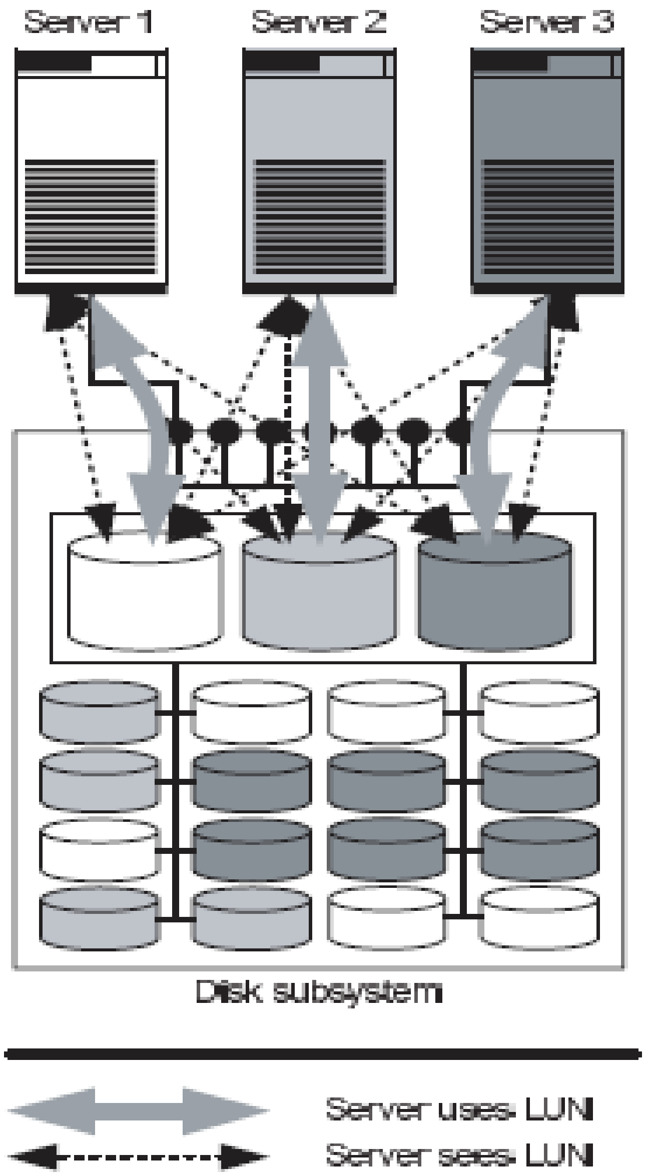

Figure 2.23 Chaos: each server works to its own virtual hard disk. Without LUN masking each server sees all hard disks. A configuration error on server 1 can destroy the data on the other two servers. The data is thus poorly protected

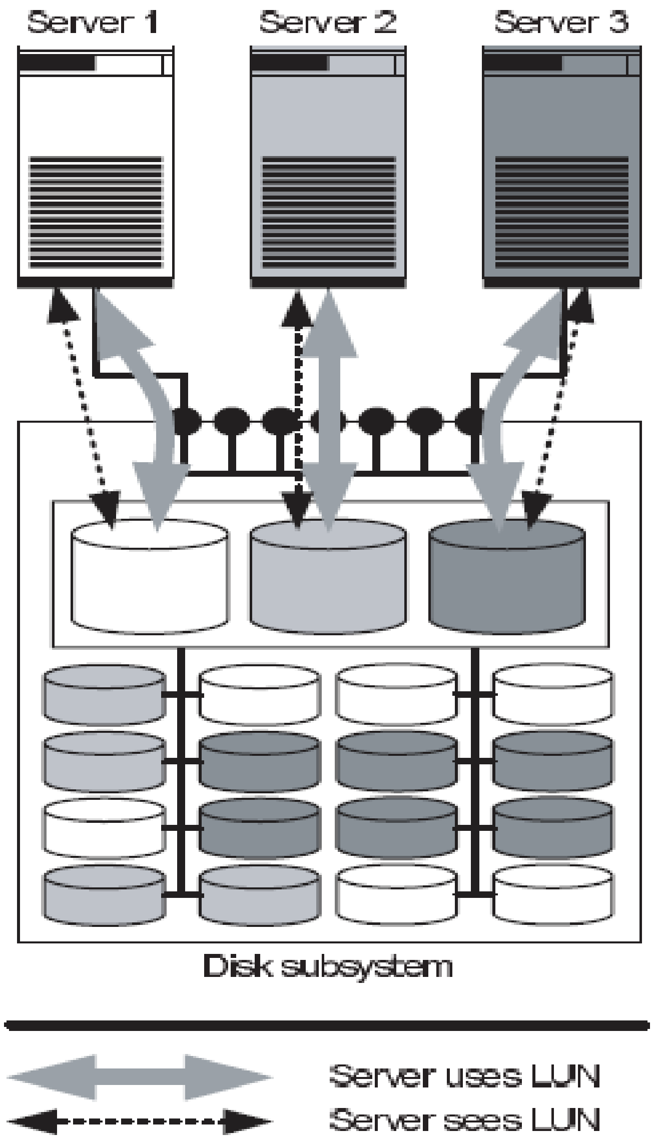

Figure 2.24 Order: each server works to its own virtual hard disk. With LUN masking, each server sees only its own hard disks. A configuration error on server 1 can no longer destroy the data of the two other servers. The data is now protected

- We differentiate between **port-based LUN masking and server-based LUN masking.**
- **Port-based LUN masking** is the 'poor man's LUN masking', it is found primarily in low-end disk subsystems. In port-based LUN masking the filter only works using the granularity of a port. This means that all servers connected to the disk subsystem via the same port see the same disks.
- **Server-based LUN masking** offers more flexibility. In this approach every server sees only the hard disks assigned to it, regardless of which port it is connected via or which other servers are connected via the same port.

Tape Drives

- Tape drives are used universally for backup and restore operations.
- The **primary difference** **between tape and disk storage** is that
  - **Tape media is removable**, which means it can be transported for safekeeping from disasters .
  - One of the less-obvious differences is their **read/write ratios.** Disk drives are most often used to read data, **tape drives are usually used to write data.**
- This section discusses following topics concerning with tapes and tape drives:
  - Tape media
  - Caring for tape
  - Caring for tape heads
  - Tape drive performance
  - The tale of two technologies

TAPE MEDIA

- The media where data is stored on a tape drive is - **magnetic tape.**
- The result of a great deal of chemicals, materials, and manufacturing technology. **Magnetic tape is constructed in four basic layers:**

1. **Backing**
   - The backing of a tape is the foundation material that gives the **tape its inherent flexibility and strength.**
2. **Binder:**
   - Tape binder is the **flexible glue-like material that** adheres to the backing as well as the **magnetic material**.
3. **Magnetic material:**
   - The magnetic materials in tape are where the action is, of course, and **where data is written and read.** The **magnetic properties are provided by fine metal oxides, which are smooth to the human eye but somewhat jagged and rough at a microscopic level.**
4. **Coating:**
   - The **coating layer levels the surface of the tape and** provides a smoother surface for running over the tape heads.

CARING FOR TAPE

- In general, tapes deteriorate slowly over time.
- They **develop cracks in the surface**, they tear along the edges, and the metal oxides corrode.
- It is important to use and store data tapes under conditions of **moderate temperature and low humidity**. This includes tapes that have not been used yet but are being stored for future use.

CARING FOR TAPE HEADS

- Unlike **disk drive heads that float at microscopic levels** above the platter, **tape heads are designed to be in contact with the tape** when reading or writing data.
- **As a result,** tape heads eventually wear out over time due to the constant friction of regular operations.
- Tape heads should be cleaned after **every 30 hours** of use.
- Unlike disk drives, which have limited exposure to airborne particles, tape drives are exposed when tapes are removed and inserted. This makes it practically impossible to keep particulate matter away from the tape heads, and **that's why it's important to operate tape drives in a clean environment**.

TAPE DRIVE PERFORMANCE

- **Tape drives have wide performance ranges based on two variables:**
  - A sufficient amount of data being transferred
  - The compressibility of data
- **Streaming and Start-Stop Operations**
  - Unlike disk drives, tape drives run at different speeds. A drive's streaming transfer rate represents the speed at which data is written from buffers onto tape.
  - Start-stop operations occur when there is insufficient data to maintain streaming mode operations. Start-stop speeds are typically far less than streaming speeds because the tape has to be stopped, rewound slightly, and started up again.
- **Compression**
  - Tape drives **typically incorporate compression technology** as a feature to boost data transfer rates. Compression can increase performance several times beyond native (uncompressed) data transfer rates, but that depends on how much the data can be compressed.

THE TALE OF TWO TECHNOLOGIES

- Tape drives used in storage networks **can be divided into two broad technology areas**, with two contestants in each area all of them being incompatible with the others.
- **Linear Tape Technology**
  - Linear tape reads and writes data just as it sounds by placing **"lines"** of data that **run lengthwise on the tape media**.
  - Linear tape drives use multiple heads operating in parallel, reading and writing data simultaneously.
  - Tape used in linear tape drives **is .5 inches wide**.
  - Linear tape cartridges have only one spool to hold tape.
  - There are two primary, competing linear tape technologies:
    1. **Super digital linear tape (SDLT)**
    2. **Linear tape open (LTO)**

Helical Scan Tape Technology

- Helical scan tape technology **was originally developed for video recording applications.**
- Most helical scan tape drives used for data storage applications **use 8 mm tape**, which is approximately **.25 inches wide.**
- Helical scan drives **write data in diagonal strips** along the tape.
- In general, helical scan tape cartridges are much smaller than linear tape cartridges, even though they have two reels, unlike linear tape.
- Two primary helical scan technologies are used in storage networking environments:
  1. **Mammoth-2**
  2. **AIT-3.**

COMPARING TAPE TECHNOLOGIES

The following table compares the leading tape technologies used in storage networks.

Table 4-3. Comparison of Tape Technologies Used in Storage Networks

Technology	Linear or Helical	Capacity (Native/Compressed)	Maximum Transfer Rate (Native/Compressed)
Mammoth-2	Helical	60 GB/150 GB	12 Mbps/30 Mbps
AIT-3	Helical	100 GB/250 GB	12 Mbps/30 Mbps
SuperDLT	Linear	110 GB/220 GB	10 Mbps/20 Mbps
LTO Ultrium	Linear	340 GB/680 GB	20 Mbps/40 Mbps

QUESTIONNAIRE-1

- Text book and Reference book ?
- Why Storage network required?
- What are Five Pillars of IT?
- What is data proliferation?
- Various problems caused by data proliferation?
- What are the proposed solutions to overcome from these problems?
- What are the various stages of ILM?
- What is storage hierarchy?
- What are the various constituent parts of any disk storage subsystems?
- Differentiate between port based and server based LUN masking.
- Name the four basic layers of magnetic tape.