7

Network Backup

Network backup systems can back up heterogeneous IT environments incorporating several thousands of computers largely automatically. In the classical form, network backup systems move the data to be backed up via the local area network (LAN); this is where the name 'network backup' comes from. This chapter explains the basic principles of network backup and shows typical performance bottlenecks for conventional server-centric IT architectures. Finally, it shows how storage networks and intelligent storage systems help to overcome these performance bottlenecks.

Before getting involved in technical details, we will first discuss a few general conditions that should be taken into account in backup (Section 7.1). Then the backup, archiving and hierarchical storage management (HSM) services will be discussed (Section 7.2) and we will show which components are necessary for their implementation (Sections 7.3 and 7.4). This is followed by a summary of the measures discussed up to this point that are available to network backup systems to increase performance (Section 7.5). Then, on the basis of network backup, further technical boundaries of server-centric IT architectures will be described (Section 7.6) that are beyond the scope of Section 1.1, and we will explain why these performance bottlenecks can only be overcome to a limited degree within the server-centric IT architecture (Section 7.7). Then we will show how data can be backed up significantly more efficiently with a storage-centric IT architecture (Section 7.8). Building upon this, the protection of file servers (Section 7.9) and databases (Section 7.10) using storage networks and network backup systems will be discussed. Finally, organisational aspects of data protection will be considered (Section 7.11).

Storage Networks Explained: Basics and Application of Fibre Channel SAN, NAS, iSCSI, InfiniBand and FCoE, *Second Edition*
U. Troppens R. Erkens W. Müller-Friedt N. Haustein R. Wolafka

7.1 GENERAL CONDITIONS FOR BACKUP

Backup is always a headache for system administrators. Increasing amounts of data have to be backed up in ever shorter periods of time. Although modern operating systems come with their own backup tools, these tools only represent isolated solutions, which are completely inadequate in the face of the increasing number and heterogeneity of systems to be backed up. For example, there may be no option for monitoring centrally whether all backups have been successfully completed overnight or there may be a lack of overall management of the backup media.

Changing preconditions represent an additional hindrance to data protection. There are three main reasons for this:

1. As discussed in Chapter 1, installed storage capacity doubles every 4–12 months depending upon the company in question. The data set is thus often growing more quickly than the infrastructure in general (personnel, network capacity). Nevertheless, the ever-increasing quantities of data still have to be backed up.
2. Nowadays, business processes have to be adapted to changing requirements all the time. As business processes change, so the IT systems that support them also have to be adapted. As a result, the daily backup routine must be continuously adapted to the ever-changing IT infrastructure.
3. As a result of globalisation, the Internet and e-business, more and more data has to be available around the clock: it is no longer feasible to block user access to applications and data for hours whilst data is backed up. The time window for backups is becoming ever smaller.

Network backup can help us to get to grips with these problems.

7.2 NETWORK BACKUP SERVICES

Network backup systems such as Arcserve (Computer Associates), NetBackup (Symantec/Veritas), Networker (EMC/Legato) and Tivoli Storage Manager (IBM) provide the following services:

- Backup
- Archive
- Hierarchical Storage Management (HSM)

The main task of network backup systems is to back data up regularly. To this end, at least one up-to-date copy must be kept of all data, so that it can be restored after a hardware or application error ('file accidentally deleted or destroyed by editing', 'error in the database programming').

The goal of archiving is to freeze a certain version of data so that precisely this version can be retrieved at a later date. For example, at the end of a project the data that was used can be archived on a backup server and then deleted from the local hard disk. This releases local disk space and accelerates the backup and restore processes, because only the data currently being worked on needs to be backed up or restored. Data archiving has become so important during the last few years that we treat it as a separate topic in the next chapter (Chapter 8).

HSM finally leads the end user to believe that any desired size of hard disk is present. HSM moves files that have not been accessed for a long time from the local disk to the backup server; only a directory entry remains in the local file server. The entry in the directory contains meta-information such as file name, owner, access rights, date of last modification and so on. The metadata takes up hardly any space in the file system compared to the actual file contents, so space is actually gained by moving the file content from the local disk to the backup server.

If a process accesses the content of a file that has been moved in this way, HSM blocks the accessing process, copies the file content back from the backup server to the local file system and only then gives clearance to the accessing process. Apart from the longer access time, this process remains completely hidden to the accessing processes and thus also to end users. Older files can thus be automatically moved to cheaper media (tapes) and, if necessary, fetched back again without the end user having to alter his behaviour.

Strictly speaking, HSM and backup and archive are separate concepts. However, HSM is a component of many network backup products, so the same components (media, software) can be used both for backup, archive and also for HSM. When HSM is used, the backup software used must at least be HSM-capable: it must back up the metadata of the moved files and the moved files themselves, without moving the file contents back to the client. HSM-capable backup software can speed up backup and restore processes because only the meta-information of the moved files has to be backed up and restored, not their file contents.

A network backup system realises the above-mentioned functions of backup, archive and HSM by the coordination of backup server and a range of backup clients (Figure 7.1). The server provides central components such as the management of backup media that are required by all backup clients. However, different backup clients are used for different operating systems and applications. These are specialised in the individual operating systems or applications in order to increase the efficiency of data protection or the efficiency of the movement of data.

The use of terminology regarding network backup systems is somewhat sloppy: the main task of network backup systems is the backup of data. Server and client instances of network backup systems are therefore often known as the backup server and backup client, regardless of what tasks they perform or what they are used for. A particular server

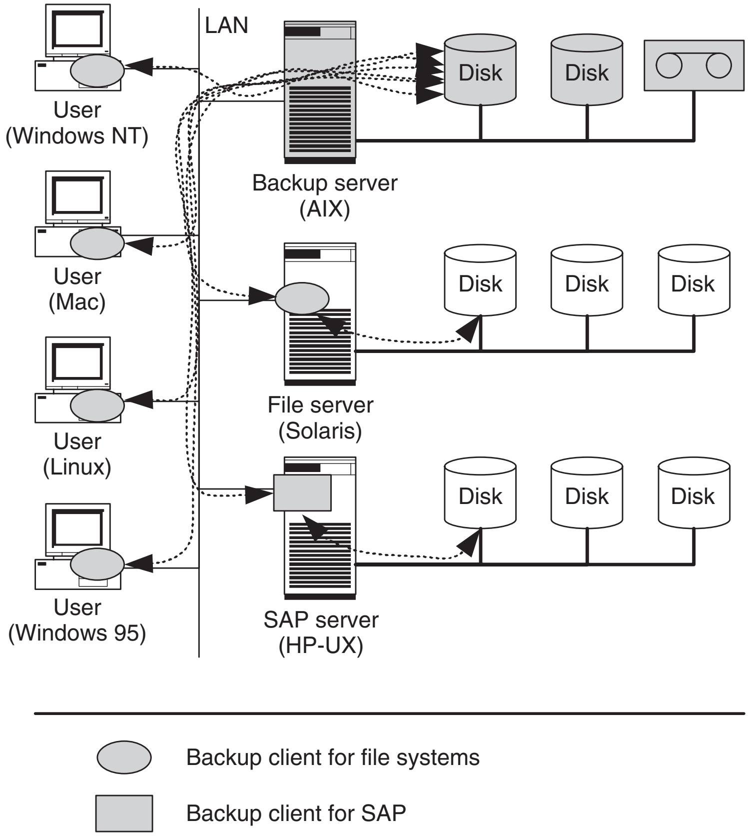

Figure 7.1 Network backup systems can automatically back up heterogeneous IT environments via the LAN. A platform-specific backup client must be installed on all clients to be backed up.

instance of a network backup system could, for example, be used exclusively for HSM, so that this instance should actually be called a HSM server – nevertheless this instance would generally be called a backup server. A client that provides the backup function usually also supports archive and the restore of backups and archives – nevertheless this client is generally just known as a backup client. In this book we follow the general, untidy conventions, because the phrase 'backup client' reads better than 'backup-archive-HSM and restore client'.

The two following sections discuss details of the backup server (Section 7.3) and the backup client (Section 7.4). We then turn our attention to the performance and the use of network backup systems.

7.3 COMPONENTS OF BACKUP SERVERS

Backup servers consist of a whole range of component parts. In the following we will discuss the main components: job scheduler (Section 7.3.1), error handler (Section 7.3.2), metadata database (Section 7.3.3) and media manager (Section 7.3.4).

7.3.1 Job scheduler

The job scheduler determines what data will be backed up when. It must be carefully configured; the actual backup then takes place automatically.

With the aid of job schedulers and tape libraries many computers can be backed up overnight without the need for a system administrator to change tapes on site. Small tape libraries have a tape drive, a magazine with space for around ten tapes and a media changer that can automatically move the various tapes back and forth between magazine and tape drive. Large tape libraries have several dozen tape drives, space for several thousands of tapes and a media changer or two to insert the tapes in the drives.

7.3.2 Error handler

If a regular automatic backup of several systems has to be performed, it becomes difficult to monitor whether all automated backups have run without errors. The error handler helps to prioritise and filter error messages and generate reports. This avoids the situation in which problems in the backup are not noticed until a backup needs to be restored.

7.3.3 Metadata database

The metadata database and the media manager represent two components that tend to be hidden to end users. The metadata database is the brain of a network backup system. It contains the following entries for every backup up object: name, computer of origin, date of last change, date of last backup, name of the backup medium, etc. For example, an entry is made in the metadata database for every file to be backed up.

The cost of the metadata database is worthwhile: in contrast to backup tools provided by operating systems, network backup systems permit the implementation of the incremental-forever strategy in which a file system is only fully backed up in the first backup. In subsequent backups, only those files that have changed since the previous backup are backed up. The current state of the file system can then be calculated on the backup server from database operations from the original full backup and from all subsequent incremental backups, so that no further full backups are necessary. The calculations in the metadata database are generally performed faster than a new full backup.

Even more is possible: if several versions of the files are kept on the backup server, a whole file system or a subdirectory dated three days ago, for example, can be restored (point-in-time restore) – the metadata database makes it possible.

7.3.4 Media manager

Use of the incremental-forever strategy can considerably reduce the time taken by the backup in comparison to the full backup. The disadvantage of this is that over time the backed up files can become distributed over numerous tapes. This is critical for the restoring of large file systems because tape mounts cost time. This is where the media manager comes into play. It can ensure that only files from a single computer are located on one tape. This reduces the number of tape mounts involved in a restore process, which means that the data can be restored more quickly.

A further important function of the media manager is so-called tape reclamation. As a result of the incremental-forever strategy, more and more data that is no longer needed is located on the backup tapes. If, for example, a file is deleted or changed very frequently over time, earlier versions of the file can be deleted from the backup medium. The gaps on the tapes that thus become free cannot be directly overwritten using current techniques. In tape reclamation, the media manager copies the remaining data that is still required from several tapes, of which only a certain percentage is used, onto a common new tape. The tapes that have thus become free are then added to the pool of unused tapes.

There is one further technical limitation in the handling of tapes: current tape drives can only write data to the tapes at a certain speed. If the data is transferred to the tape drive too slowly this interrupts the write process, the tape rewinds a little and restarts the write process. The repeated rewinding of the tapes costs performance and causes unnecessary wear to the tapes so they have to be discarded more quickly. It is therefore better to send the data to the tape drive quickly enough so that it can write the data onto the tape in one go (streaming).

The problem with this is that in network backup the backup clients send the data to be backed up via the LAN to the backup server, which forwards the data to the tape drive. On the way from backup client via the LAN to the backup server there are repeated fluctuations in the transmission rate, which means that the streaming of tape drives is repeatedly interrupted. Although it is possible for individual clients to achieve streaming by additional measures (such as the installation of a separate LAN between backup client and backup server) (Section 7.7), these measures are expensive and technically not scalable at will, so they cannot be realised economically for all clients.

The solution: the media manager manages a storage hierarchy within the backup server. To achieve this, the backup server must be equipped with hard disks and tape libraries. If a client cannot send the data fast enough for streaming, the media manager first of all stores the data to be backed up to hard disk. When writing to a hard disk it makes no difference what speed the data is supplied at. When enough of the data to be backed up has been temporarily saved to the hard disk of the backup server, the media manager automatically moves large quantities of data from the hard disk of the backup server to its tapes. This process only involves recopying the data within the backup server, so that streaming is guaranteed when writing the tapes.

This storage hierarchy is used, for example, for the backup of user PCs (Figure 7.2). Many user PCs are switched off overnight, which means that backup cannot be guaranteed overnight. Therefore, network backup systems often use the midday period to back up user PCs. Use of the incremental-forever strategy means that the amount of data to be backed up every day is so low that such a backup strategy is generally feasible. All user PCs are first of all backed up to the hard disk of the backup server in the time window from 11:15 to 13:45. The media manager in the backup server then has a good

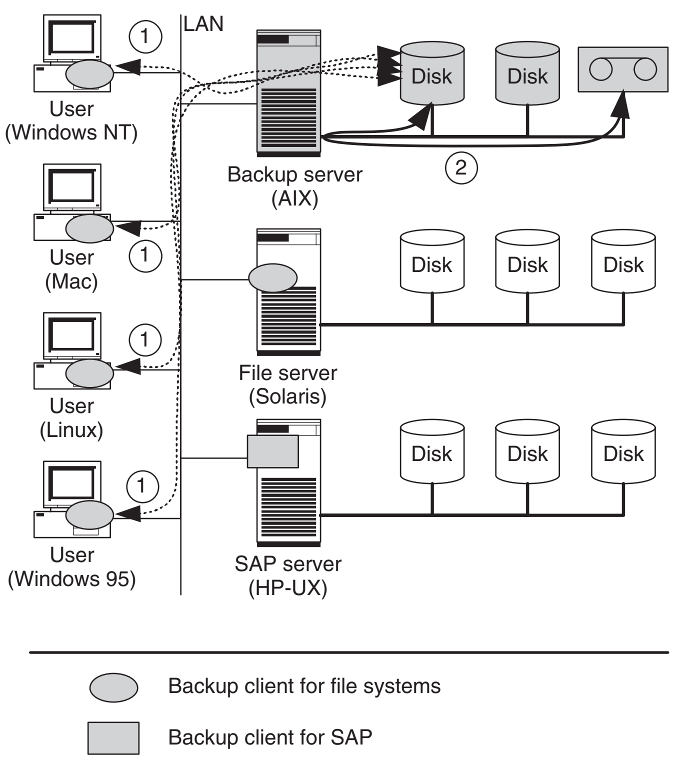

Figure 7.2 The storage hierarchy in the backup server helps to back up user PCs efficiently. First of all, all PCs are backed up to the hard disks of the backup server (1) during the midday period. Before the next midday break the media manager copies the data from the hard disks to tapes (2).

20 hours to move the data from the hard disks to tapes. Then the hard disks are once again free so that the user PCs can once again be backed up to hard disk in the next midday break.

In all operations described here the media manager checks whether the correct tape has been placed in the drive. To this end, the media manager writes an unambiguous signature to every tape, which it records in the metadata database. Every time a tape is inserted the media manager compares the signature on the tape with the signature in the metadata database. This ensures that no tapes are accidentally overwritten and that the correct data is written back during a restore operation.

Furthermore, the media manager monitors how often a tape has been used and how old it is, so that old tapes are discarded in good time. If necessary, it first copies data that is still required to a new tape. Older tape media formats also have to be wound back and forwards now and then so that they last longer; the media manager can also automate the winding of tapes that have not been used for a long time.

A further important function of the media manager is the management of data in a so-called off-site store. To this end, the media manager keeps two copies of all data to be backed up. The first copy is always stored on the backup server, so that data can be quickly restored if it is required. However, in the event of a large-scale disaster (fire in the data centre) the copies on the backup server could be destroyed. For such cases the media manager keeps a second copy in an off-site store that can be several kilometres away. The media manager supports the system administrator in moving the correct tapes back and forwards between backup server and off-site store. It even supports tape reclamation for tapes that are currently in the off-site store.

7.4 BACKUP CLIENTS

A platform-specific client (backup agent) is necessary for each platform to be backed up. The base client can back up and archive files and restores them if required. The term platform is used here to mean the various operating systems and the file systems that they support. Furthermore, some base clients offer HSM for selected file systems.

The backup of file systems takes place at file level as standard. This means that each changed file is completely re-transferred to the server and entered there in the metadata database. By using backup at volume level and at block level it is possible to change the granularity of the objects to be backed up.

When backup is performed at volume level, a whole volume is backed up as an individual object on the backup server. We can visualise this as the output of the Unix command ‘dd’ being sent to the backup server. Although this has the disadvantage that free areas, on which no data at all has been saved, are also backed up, only very few metadata database operations are necessary on the backup server and on the client side it is not necessary to spend a long time comparing which files have changed since the last backup. As a result, backup and restore operations can sometimes be performed more quickly at volume level than they can at file level. This is particularly true when restoring large file systems with a large number of small files.

Backup on block level optimises backup for members of the external sales force, who only connect up to the company network now and then by means of a laptop via a dial-up line or the Internet. In this situation the performance bottleneck is the low transmission capacity between the backup server and the backup client. If only one bit of a large file is changed, the whole file must once again be forced down the network. When backing up on block level the backup client additionally keeps a local copy of every file backed up. If a file has changed, it can establish which parts of the file have changed. The backup client sends only the changed data fragments (blocks) to the backup server. This can then reconstruct the complete file. As is the case for backup on file level, each file backed up is entered in the metadata database. Thus, when backing up on block level the quantity of data to be transmitted is reduced at the cost of storage space on the local hard disk.

In addition to the standard client for file systems, most network backup systems provide special clients for various applications. For example, there are special clients for Microsoft Exchange or IBM Lotus Domino that make it possible to back up and restore individual documents. We will discuss the backup of file systems and NAS servers (Section 7.9) and databases (Section 7.10) in more detail later on.

7.5 PERFORMANCE GAINS AS A RESULT OF NETWORK BACKUP

The underlying hardware components determine the maximum throughput of network backup systems. The software components determine how efficiently the available hardware is actually used. At various points of this chapter we have already discussed how network backup systems can help to better utilise the existing infrastructure:

- *Performance increase by the archiving of data*
  Deleting data that has already been archived from hard disks can accelerate the daily backup because there is less data to back up. For the same reason, file systems can be restored more quickly.
- *Performance increase by HSM*
  By moving file contents to the HSM server, file systems can be restored more quickly. The directory entries of files that have been moved can be restored comparatively quickly; the majority of the data, namely the file contents, do not need to be fetched back from the HSM server.
- *Performance increase by the incremental-forever strategy*
  After the first backup, only the data that has changed since the last backup is backed up. On the backup server the metadata database is used to calculate the latest state of the

data from the first backup and all subsequent incremental backups, so that no further full backups are necessary. The backup window can thus be significantly reduced.

- *Performance increase by reducing tape mounts*
  The media manager can ensure that data that belongs together is only distributed amongst a few tapes. The number of time-consuming tape mounts for the restoring of data can thus be reduced.
- *Performance increase by streaming*
  The efficient writing of tapes requires that the data is transferred quickly enough to the tape drive. If this is not guaranteed the backup server can first temporarily store the data on a hard drive and then send the data to the tape drive in one go.
- *Performance increase by backup on volume level or on block level*
  As default, file systems are backed up on file level. Large file systems with several hundreds of thousands of files can sometimes be backed up more quickly if they are backed up at volume level. Laptops can be backed up more quickly if only the blocks that have changed are transmitted over a slow connection to the backup server.

7.6 PERFORMANCE BOTTLENECKS OF NETWORK BACKUP

At some point, however, the technical boundaries for increasing the performance of backup are reached. When talking about technical boundaries, we should differentiate between application-specific boundaries (Section 7.6.1) and those that are determined by server-centric IT architecture (Section 7.6.2).

7.6.1 Application-specific performance bottlenecks

Application-specific performance bottlenecks are all those bottlenecks that can be traced back to the 'network backup' application. These performance bottlenecks play no role for other applications.

The main candidate for application-specific performance bottlenecks is the metadata database. A great deal is demanded of this. Almost every action in the network backup system is associated with one or more operations in the metadata database. If, for example, several versions of a file are backed up, an entry is made in the metadata database for each version. The backup of a file system with several hundreds of thousands of files can thus be associated with a whole range of database operations.

A further candidate for application-specific performance bottlenecks is the storage hierarchy: when copying the data from hard disk to tape the media manager has to load the data from the hard disk into the main memory via the I/O bus and the internal buses, only to forward it from there to the tape drive via the internal buses and I/O bus. This means that the buses can get clogged up during the copying of the data from hard disk to tape. The same applies to tape reclamation.

7.6.2 Performance bottlenecks due to server-centric IT architecture

In addition to these two application-specific performance bottlenecks, some problems crop up in network backup that are typical of a server-centric IT architecture. Let us mention once again as a reminder the fact that in a server-centric IT architecture storage devices only exist in relation to servers; access to storage devices always takes place via the computer to which the storage devices are connected. The performance bottlenecks described in the following apply for all applications that are operated in a server-centric IT architecture.

Let us assume that a backup client wants to back data up to the backup server (Figure 7.3). The backup client loads the data to be backed up from the hard disk into the main memory of the application server via the SCSI bus, the PCI bus and the system bus, only to forward it from there to the network card via the system bus and the PCI bus. On the backup server the data must once again be passed through the buses twice. In backup, large quantities of data are generally backed up in one go. During backup, therefore, the buses of the participating computers can become a bottleneck, particularly if the application server also has to bear the I/O load of the application or the backup server is supposed to support several simultaneous backup operations.

The network card transfers the data to the backup server via TCP/IP and Ethernet. Previously the data exchange via TCP/IP was associated with a high CPU load. However, the CPU load caused by TCP/IP data traffic can be reduced using TCP/IP offload engines (TOE) (Section 3.5.2).

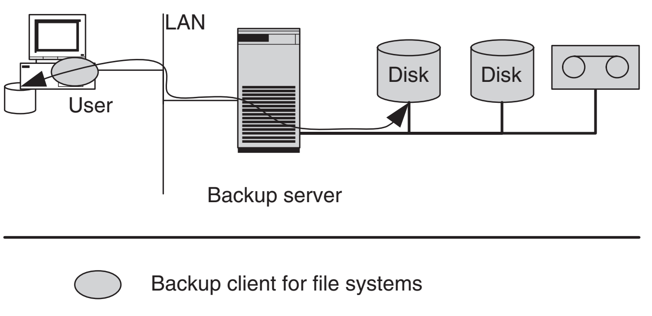

Figure 7.3 In network backup, all data to be backed up must be passed through both computers. Possible performance bottlenecks are internal buses, CPU and the LAN.

7.7 LIMITED OPPORTUNITIES FOR INCREASING PERFORMANCE

Backup is a resource-intensive application that places great demands upon storage devices, CPU, main memory, network capacity, internal buses and I/O buses. The enormous amount of resources required for backup is not always sufficiently taken into account during the planning of IT systems. A frequent comment is ‘the backup is responsible for the slow network’ or ‘the slow network is responsible for the restore operation taking so long’. The truth is that the network is inadequately dimensioned for end user data traffic and backup data traffic. Often, data protection is the application that requires the most network capacity. Therefore, it is often sensible to view backup as the primary application for which the IT infrastructure in general and the network in particular must be dimensioned.

In every IT environment, most computers can be adequately protected by a network backup system. In almost every IT environment, however, there are computers – usually only a few – for which additional measures are necessary in order to back them up quickly enough or, if necessary, to restore them. In the server-centric IT architecture there are three approaches to taming such data monsters: the installation of a separate LAN for the network backup between backup client and backup server (Section 7.7.1), the installation of several backup servers (Section 7.7.2) and the installation of backup client and backup server on the same physical computer (Section 7.7.3).

7.7.1 Separate LAN for network backup

The simplest measure to increase backup performance more of heavyweight backup clients is to install a further LAN between backup client and backup server in addition to the existing LAN and to use this exclusively for backup (Figure 7.4). An expensive, but powerful, transmission technology such as leading edge Gigabit Ethernet generations can also help here.

The concept of installing a further network for backup in addition to the existing LAN is comparable to the basic idea of storage networks. In contrast to storage networks, however, in this case only computers are connected together; direct access to all storage devices is not possible. All data thus continues to be passed via TCP/IP and through application server and backup server which leads to a blockage of the internal bus and the I/O buses.

Individual backup clients can thus benefit from the installation of a separate LAN for network backup. This approach is, however, not scalable at will: due to the heavy load on the backup server this cannot back up any further computers in addition to the backup of one individual heavyweight client.

Despite its limitations, the installation of a separate backup LAN is sufficient in many environments. With Fast-Ethernet you can still achieve a throughput of over 10 MByte/s.

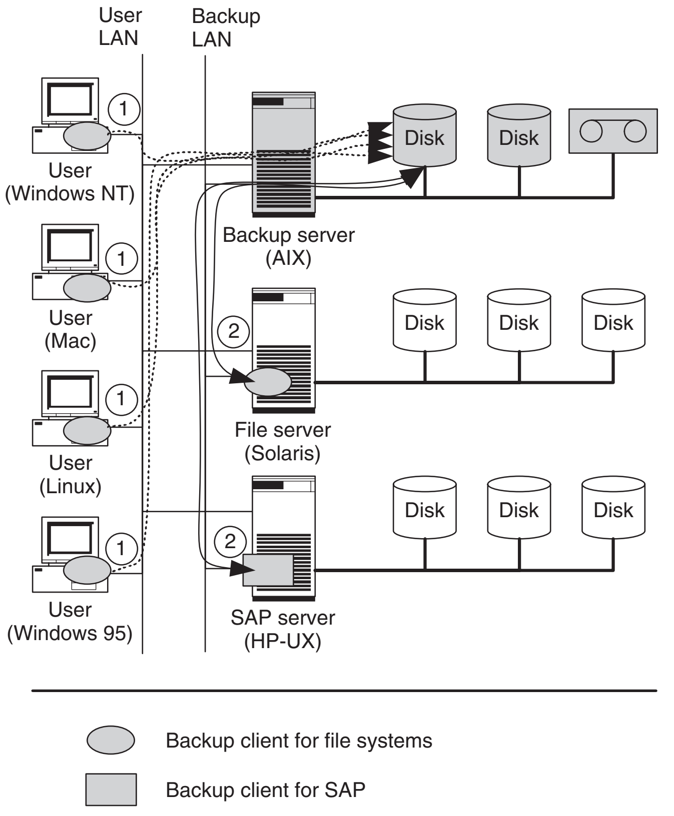

Figure 7.4 Approach 1: The throughput of network backup can be increased by the installation of a second LAN. Normal clients are still backed up via the User LAN (1). Only heavyweight clients are backed up via the second LAN (2).

The LAN technique is made even more attractive by Gigabit Ethernet, 10Gigabit Ethernet and the above-mentioned TCP/IP offload engines that free up the server CPU significantly with regard to the TCP/IP data traffic.

7.7.2 Multiple backup servers

Installing multiple backup servers distributes the load of the backup server over more hardware. For example, it would be possible to assign every heavyweight backup client a special backup server installed exclusively for the backup of this client (Figure 7.5). Furthermore, a further backup server is required for the backup of all other backup clients. This approach is worthwhile in the event of performance bottlenecks in the metadata

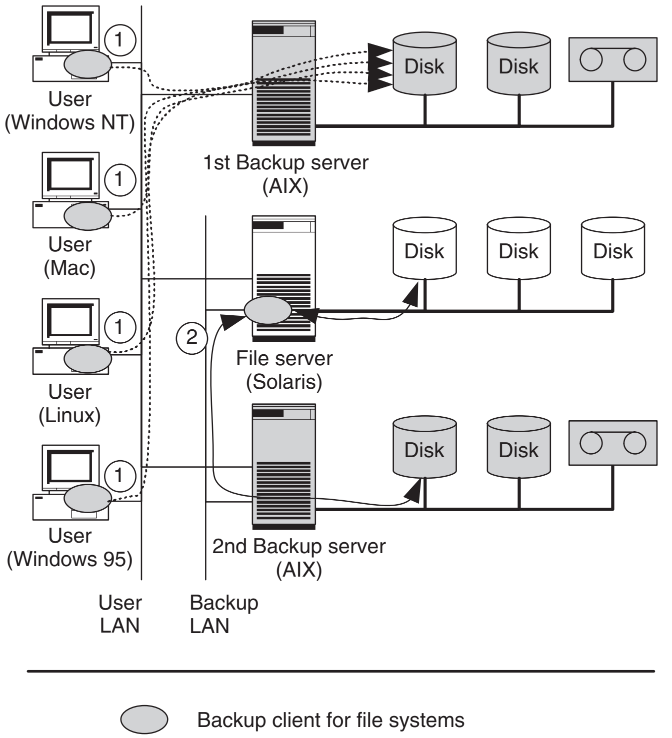

Figure 7.5 Approach 2: Dedicated backup servers can be installed for heavyweight backup clients. Normal clients continue to be backed up on the first backup server (1). Only the heavyweight client is backed up on its own backup server over the separate LAN (2).

database or in combination with the first measure, the installation of a separate LAN between the heavyweight backup client and backup server.

The performance of the backup server can be significantly increased by the installation of multiple backup servers and a separate LAN for backup. However, from the point of view of the heavyweight backup client the problem remains that all data to be backed up must be passed from the hard disk into the main memory via the buses and from there must again be passed through the buses to the network card. This means that backup still heavily loads the application server. The resource requirement for backup could be in conflict with the resource requirement for the actual application.

A further problem is the realisation of the storage hierarchy within the individual backup server since every backup server now requires its own tape library. Many small tape libraries are more expensive and less flexible than one large tape library. Therefore, it would actually be better to buy a large tape library that is used by all servers. In a server-centric IT architecture it is, however, only possible to connect multiple computers to the same tape library to a very limited degree.

7.7.3 Backup server and application server on the same physical computer

The third possible way of increasing performance is to install the backup server and application server on the same physical computer (Figure 7.6). This results in the backup client also having to run on this computer. Backup server and backup client communicate over Shared Memory (Unix), Named Pipe or TCP/IP Loopback (Windows) instead of via LAN. Shared Memory has an infinite bandwidth in comparison to the buses, which means that the communication between backup server and backup client is no longer the limiting factor.

However, the internal buses continue to get clogged up: the backup client now loads the data to be backed up from the hard disk into the main memory via the buses. The backup server takes the data from the main memory and writes it, again via the buses, to the backup medium. The data is thus once again driven through the internal bus twice.

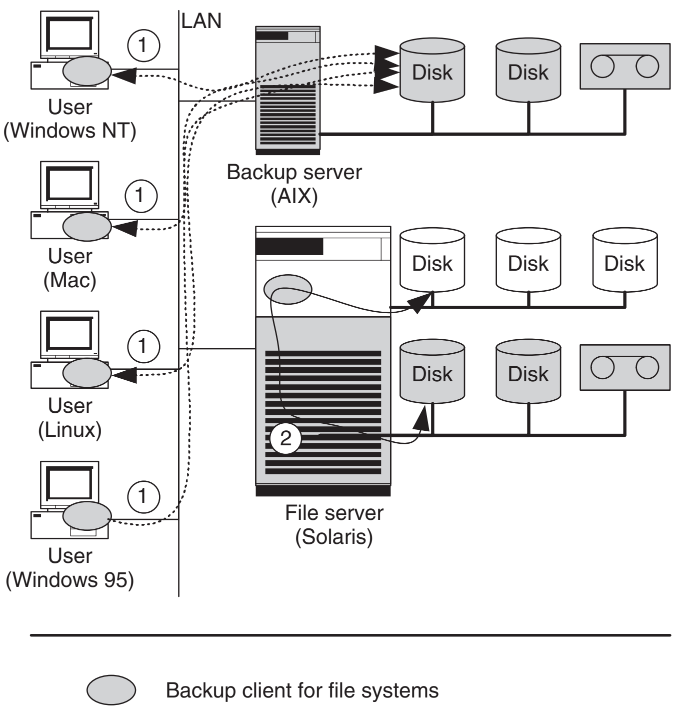

Figure 7.6 Approach 3: Application server, backup server and backup client are installed on one computer. Normal clients continue to be backed up on the first backup server (1). Only the heavyweight client is backed up within the same computer (2).

Tape reclamation and any copying operations within the storage hierarchy of the backup server could place an additional load on the buses.

Without further information we cannot more precisely determine the change to the CPU load. Shared Memory communication (or Named Pipe or TCP/IP Loopback) dispenses with the CPU-intensive operation of the network card. On the other hand, a single computer must now bear the load of the application, the backup server and the backup client. This computer must incidentally possess sufficient main memory for all three applications.

One problem with this approach is the proximity of production data and copies on the backup server. SCSI permits a maximum cable length of 25 m. Since application and backup server run on the same physical computer, the copies are a maximum of 50 m away from the production data. In the event of a fire or comparable damage, this is disastrous. Therefore, either a SCSI extender should be used or the tapes taken from the tape library every day and placed in an off-site store. The latter goes against the requirement of largely automating data protection.

7.8 NEXT GENERATION BACKUP

Storage networks open up new possibilities for getting around the performance bottlenecks of network backup described above. They connect servers and storage devices, so that during backup production data can be copied directly from the source hard disk to the backup media, without passing it through a server (server-free backup, Section 7.8.1). LAN-free backup (Section 7.8.2) and LAN-free backup with shared disk file systems (Section 7.8.3) are two further alternative methods of accelerating backup using storage networks. The use of instant copies (Section 7.8.4) and remote mirroring (Section 7.8.5) provide further possibilities for accelerating backup and restore operations. The introduction of storage networks also has the side effect that several backup servers can share a tape library (Section 7.8.6).

7.8.1 Server-free backup

The ultimate goal of backup over a storage network is so-called server-free backup (Figure 7.7). In backup, the backup client initially determines which data has to be backed up and then sends only the appropriate metadata (file name, access rights, etc.) over the LAN to the backup server. The file contents, which make up the majority of the data quantity to be transferred, are then written directly from the source hard disk to the backup medium (disk, tape, optical) over the storage network, without a server being connected in between. The network backup system coordinates the communication between source hard disk and backup medium. A shorter transport route for the backup of data is not yet in sight with current storage techniques.

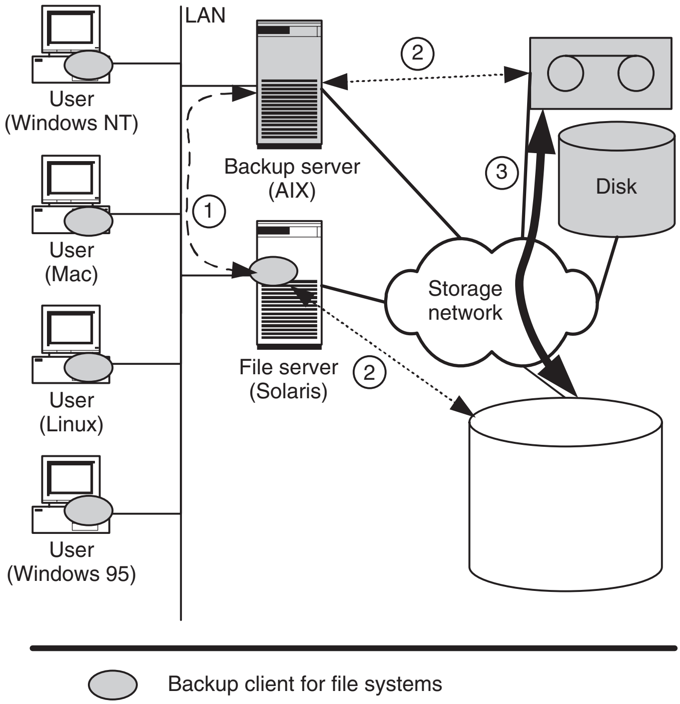

Figure 7.7 In server-free backup, backup server and backup client exchange lightweight metadata via the LAN (1). After it has been determined which data blocks have to be backed up, the network backup system can configure the storage devices for the data transfer via the storage network (2). The heavyweight file contents are then copied directly from the source hard disk to the backup medium via the storage network (3).

The performance of server-free backup is predominantly determined by the performance of the underlying storage systems and the connection in the storage network. Shifting the transport route for the majority of the data from the LAN to the storage network without a server being involved in the transfer itself means that the internal buses and the I/O buses are freed up on both the backup client and the backup server. The cost of coordinating the data traffic between source hard disk and backup medium is comparatively low.

A major problem in the implementation of server-free backup is that the SCSI commands have to be converted en route from the source hard disk to the backup medium. For example, different blocks are generally addressed on source medium and backup medium. Or, during the restore of a deleted file in a file system, this file has to be restored to a different area if the space that was freed up is now occupied by other files. In the backup from hard disk to tape, even the SCSI command sets are slightly different. Therefore, software called 3rd-Party SCSI Copy Command is necessary for the protocol conversion. It can be realised at various points: in a switch of the storage network in a box specially connected to the storage network that is exclusively responsible for the protocol conversion, or in one of the two participating storage systems themselves.

According to our knowledge, server-free backup at best made it to the demonstration centres of the suppliers' laboratories. Some suppliers were involved in aggressive marketing around 2002 and 2003, claiming that their network backup products support server-free backup. In our experience, server-free backup is basically not being used in any production environments today although it has been available for some time. In our opinion this confirms that server-free backup is still very difficult to implement, configure and operate with the current state of technology.

7.8.2 LAN-free backup

LAN-free backup dispenses with the necessity for the 3rd-Party SCSI Copy Command by realising comparable functions within the backup client (Figure 7.8). As in server-free backup, metadata is sent via the LAN. File contents, however, no longer go through the backup server: for backup the backup client loads the data from the hard disk into the main memory via the appropriate buses and from there writes it directly to the backup medium via the buses and the storage network. To this end, the backup client must be able to access the backup server's backup medium over the storage network. Furthermore, backup server and backup client must synchronise their access to common devices. This is easier to realise than server-free backup and thus well proven in production environments.

In LAN-free backup the load on the buses of the backup server is reduced but not the load on those of the backup client. This can impact upon other applications (databases, file and web servers) that run on the backup client at the same time as the backup.

LAN-free backup is already being used in production environments. However, the manufacturers of network backup systems only support LAN-free backup for certain applications (databases, file systems, e-mail systems), with not every application being supported on every operating system. Anyone wanting to use LAN-free backup at the moment must take note of the manufacturer's support matrix (see Section 3.4.6). It can be assumed that in the course of the next years the number of the applications and operating systems supported will increase further.

7.8.3 LAN-free backup with shared disk file systems

Anyone wishing to back up a file system now for which LAN-free backup is not supported can sometimes use shared disk file systems to rectify this situation (Figure 7.9). Shared disk file systems are installed upon several computers. Access to data is synchronised over the LAN; the individual file accesses, on the other hand, take place directly over the storage network (Section 4.3). For backup the shared disk file system is installed on the

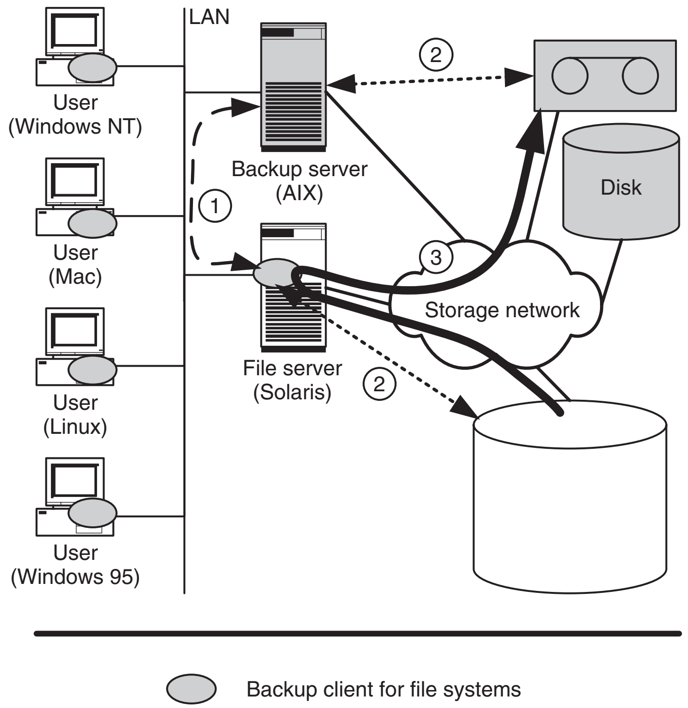

Figure 7.8 In LAN-free backup too, backup servers and backup clients exchange lightweight metadata over the LAN (1). The backup server prepares its storage devices for the data transfer over the storage network and then hands control of the storage devices over to the backup client (2). This then copies heavyweight file contents directly to the backup medium via the storage network (3).

file server and the backup server. The prerequisite for this is that a shared disk file system is available that supports the operating systems of backup client and backup server. The backup client is then started on the same computer on which the backup server runs, so that backup client and backup server can exchange the data via Shared Memory (Unix) or Named Pipe or TCP/IP Loopback (Windows).

In LAN-free backup using a shared disk file system, the performance of the backup server must be critically examined. All data still has to be passed through the buses of the backup server; in addition, the backup client and the shared disk file system run on this machine. LAN data traffic is no longer necessary within the network backup system; however, the shared disk file system now requires LAN data traffic for the synchronisation of simultaneous data accesses. The data traffic for the synchronisation of the shared disk file system is, however, comparatively light. At the end of the day, you have to measure whether backup with a shared disk file system increases performance for each individual case.

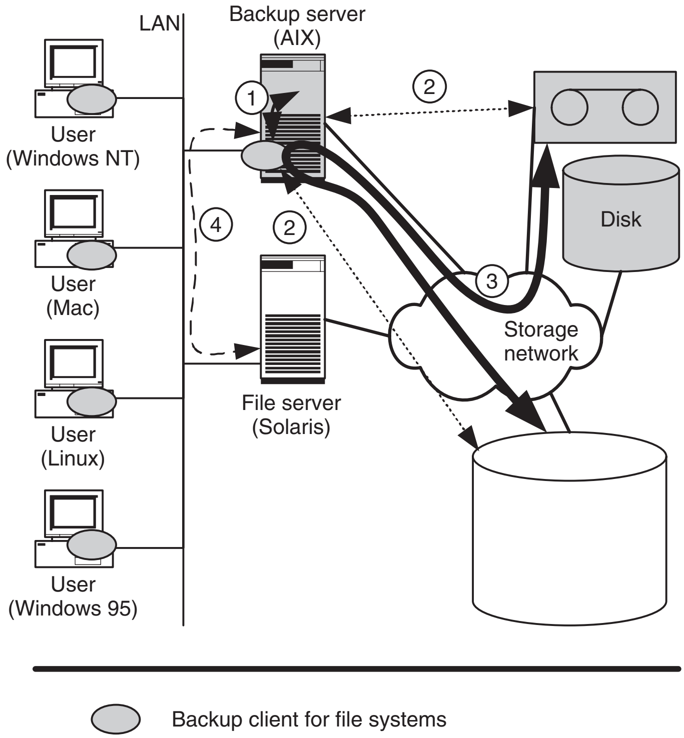

Figure 7.9 When backing up using shared disk file systems, backup server and backup client run on the same computer (1). Production data and backup media are accessed on the backup server (2), which means that the backup can take place over the storage network (3). The shared disk file system requires a LAN connection for the lightweight synchronisation of parallel data accesses (4).

The performance of LAN-free backup using a shared disk subsystem is not as good as the performance of straight LAN-free backup. However, it can be considerably better than backup over a LAN. Therefore, this approach has established itself in production environments, offering an attractive workaround until LAN-free (or even server-free) backup is available. It can also be assumed that this form of data backup will become important due to the increasing use of shared disk file systems.

7.8.4 Backup using instant copies

Instant copies can virtually copy even terabyte-sized data sets in a few seconds, and thus freeze the current state of the production data and make it available via a second access path. The production data can still be read and modified over the first access path, so that the operation of the actual application can be continued, whilst at the same time the frozen state of the data can be backed up via the second access path.

Instant copies can be realised on three different levels:

1. *Instant copy in the block layer (disk subsystem or block-based virtualisation)*
   Instant copy in the disk subsystem was discussed in detail in Section 2.7.1: intelligent disk subsystems can virtually copy all data of a virtual disk onto a second virtual disk within a few seconds. The frozen data state can be accessed and backed up via the second virtual disk.
2. *Instant copy in the file layer (local file system, NAS server or file-based virtualisation)*
   Many file systems also offer the possibility of creating instant copies. Instant copies on file system level are generally called snapshots (Section 4.1.3). In contrast to instant copies in the disk subsystem the snapshot can be accessed via a special directory path.
3. *Instant copy in the application*
   Finally, databases in particular offer the possibility of freezing the data set internally for backup, whilst the user continues to access it (hot backup, online backup).

Instant copies in the local file system and in the application have the advantage that they can be realised with any hardware. Instant copies in the application can utilise the internal data structure of the application and thus work more efficiently than file systems. On the other hand, applications do not require these functions if the underlying file system already provides them. Both approaches consume system resources on the application server that one would sometimes prefer to make available to the actual application. This is the advantage of instant copies in external devices (e.g., disk subsystem, NAS server, network-based virtualisation instance): although it requires special hardware, application server tasks are moved to the external device thus freeing up the application server.

Backup using instant copy must be synchronised with the applications to be backed up. Databases and file systems buffer write accesses in the main memory in order to increase their performance. As a result, the data on the disk is not always in a consistent state. Data consistency is the prerequisite for restarting the application with this data set and being able to continue operation. For backup it should therefore be ensured that an instant copy with consistent data is first generated. The procedure looks something like this:

1. Shut down the application.
2. Perform the instant copy.
3. Start up the application again.
4. Back up the data of the instant copy.

Despite the shutting down and restarting of the application the production system is back in operation very quickly.

Data protection with instant copies is even more attractive if the instant copy is controlled by the application itself: in this case the application must ensure that the data on disk is consistent and then initiate the copying operation. The application can then continue operation after a few seconds. It is no longer necessary to stop and restart the application.

Instant copies thus make it possible to backup business-critical applications every hour with only very slight interruptions. This also accelerates the restoring of data after application errors ('accidental deletion of a table space'). Instead of the time-consuming restore of data from tapes, the frozen copy that is present in the storage system can simply be put back.

With the aid of instant copies in the disk subsystem it is possible to realise so-called application server-free backup. In this, the application server is put at the side of a second server that serves exclusively for backup (Figure 7.10). Both servers are directly

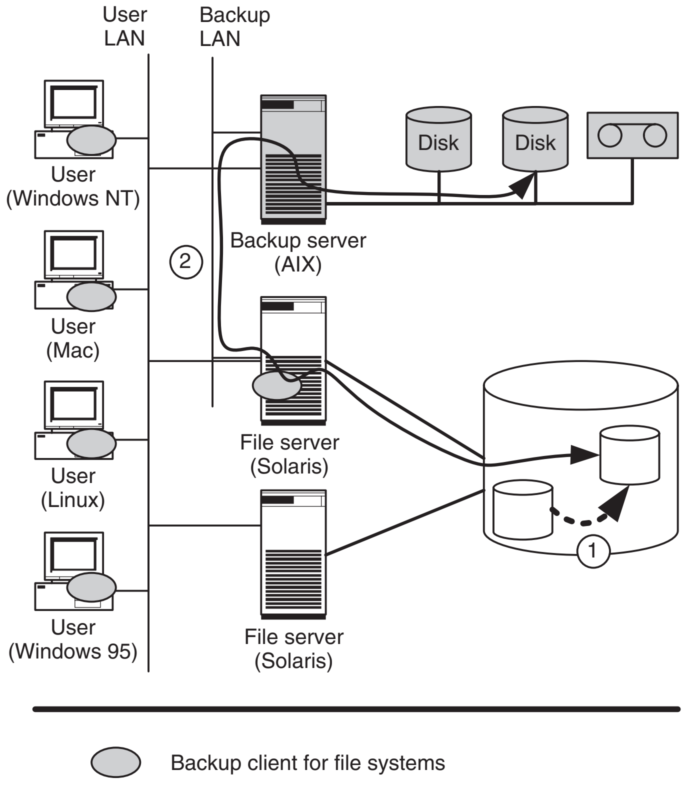

Figure 7.10 Application server-free backup utilises the functions of an intelligent disk subsystem. To perform a backup the application is operated for a short period in such a manner as to create a consistent data state on the disks, so that data can be copied by means of instant copy (1). The application can immediately switch back to normal operation; in parallel to this the data is backed up using the instant copy (2).

connected to the disk subsystem via SCSI; a storage network is not absolutely necessary. For backup the instant copy is first of all generated as described above: (1) shut down application; (2) generate instant copy; and (3) restart application. The instant copy can then be accessed from the second computer and the data is backed up from there without placing a load on the application server. If the instant copy is not deleted in the disk subsystem, the data can be restored using this copy in a few seconds in the event of an error.

7.8.5 Data protection using remote mirroring

Instant copies help to quickly restore data in the event of application or operating errors; however, they are ineffective in the event of a catastrophe: after a fire the fact that there are several copies of the data on a storage device does not help. Even a power failure can become a problem for a 24 $\times$ 7 operation.

The only thing that helps here is to mirror the data by means of remote mirroring on two disk subsystems, which are at least separated by a fire protection barrier. The protection of applications by means of remote mirroring has already been discussed in detail in Sections 6.3.3 and 6.3.5.

Nevertheless, the data still has to be backed up: in synchronous remote mirroring the source disk and copy are always identical. This means that if data is destroyed by an application or operating error then it is also immediately destroyed on the copy. The data can be backed up to disk by means of instant copy or by means of classical network backup to tapes. Since storage capacity on disk is more expensive than storage capacity on tape, only the most important data is backed up using instant copy and remote mirroring. For most data, backup to tape is still the most cost effective. In Section 9.5.5 we will take another but more detailed look at the combination of remote mirroring and backup.

7.8.6 Tape library sharing

It is sometimes necessary for more than one backup server to be used. In some data centres there is so much data to be backed up that, despite the new techniques for backup using storage networks presented here, several ten of backup servers are needed. In other cases, important applications are technically partitioned off from the network so that they have better protection from outside attacks. Ideally, everyone would like to have a separate backup server in every partitioned network segment with each backup server equipped with its own small tape library. However, having many small libraries is expensive in terms of procurement cost and is also more difficult to manage than large tape libraries. Therefore, a large tape library is frequently acquired and shared by all the backup servers over a storage network using tape library sharing (Section 6.2.2).

Figure 7.11 shows the use of tape library sharing for network backup: one backup server acts as library master, all others as library clients. If a backup client backs up data to a backup server that is configured as a library client, then this first of all requests a free tape from the library master. The library master selects the tape from its pool of free tapes and places it in a free drive. Then it notes in its metadata database that this tape is now being used by the library client and it informs the library client of the drive that the tape is in. Finally, the backup client can send the data to be backed up via the LAN to the backup server, which is configured as the library client. This then writes the data directly to tape via the storage network.

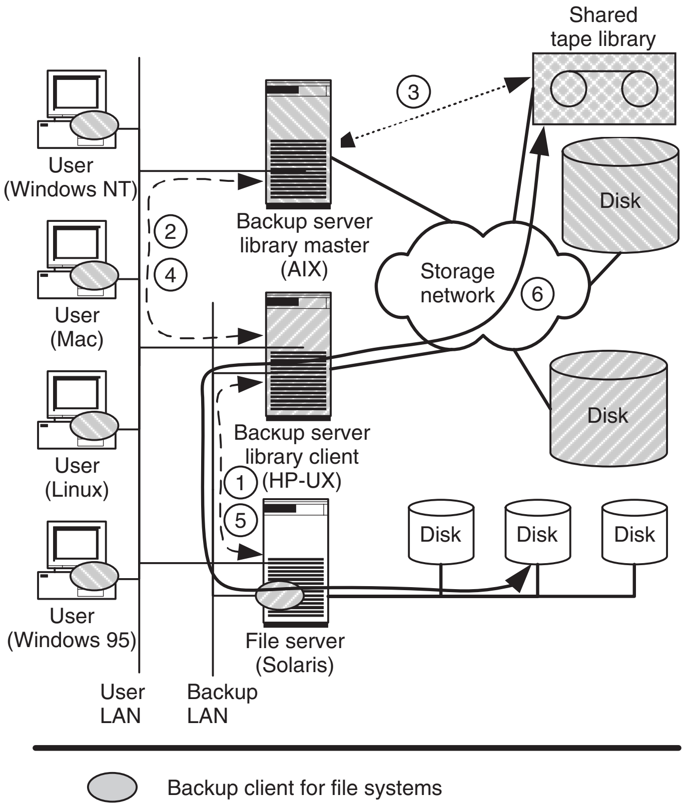

Figure 7.11 In tape library sharing two backup servers share a large tape library. If a client wants to back data up directly to tape with the second backup server (library client) (1) then this initially requests a tape and a drive from the library master (2). The library master places a free tape in a free drive (3) and returns the information in question to the library client (4). The library client now informs the backup client (5) that it can back the data up (6).

7.9 BACKUP OF FILE SYSTEMS

Almost all applications store their data in file systems or in databases. Therefore, in this section we will examine the backup of file servers (Section 7.9) and in the next section we will look more closely at that of databases (Section 7.10). The chapter concludes with organisational aspects of network backup (Section 7.11).

This section first of all discusses fundamental requirements and problems in the backup of file servers (Section 7.9.1). Then a few functions of modern file systems will be introduced that accelerate the incremental backup of file systems (Section 7.9.2). Limitations in the backup of NAS servers will then be discussed (Section 7.9.3). We will then introduce the Network Data Management Protocol (NDMP), a standard that helps to integrate the backup of NAS servers into an established network backup system (Section 7.9.4).

7.9.1 Backup of file servers

We use the term file server to include computers with a conventional operating system such as Windows or Unix that exports part of its local file systems via a network file system (NFS, CIFS) or makes it accessible as service (Novell, FTP, HTTP). The descriptions in this section can be transferred to all types of computers, from user PCs through classical file servers to the web server.

File servers store three types of information:

- Data in the form of files;
- Metadata on these files such as file name, creation date and access rights; and
- Metadata on the file servers such as any authorised users and their groups, size of the individual file systems, network configuration of the file server and names, components and rights of files or directories exported over the network.

Depending upon the error situation, different data and metadata must be restored. The restore of individual files or entire file systems is relatively simple: in this case only the file contents and the metadata of the files must be restored from the backup server to the file server. This function is performed by the backup clients introduced in Section 7.4.

Restoring an entire file server is more difficult. If, for example, the hardware of the file server is irreparable and has to be fully replaced, the following steps are necessary:

1. Purchasing and setting up of appropriate replacement hardware.
2. Basic installation of the operating system including any necessary patches.
3. Restoration of the basic configuration of the file server including LAN and storage network configuration of the file server.

4. If necessary, restoration of users and groups and their rights.
5. Creation and formatting of the local file systems taking into account the necessary file system sizes.
6. Installation and configuration of the backup client.
7. Restoration of the file systems with the aid of the network backup system.

This procedure is very labour-intensive and time-consuming. The methods of so-called Image Restore (also known as Bare Metal Restore) accelerate the restoration of a complete computer: tools such as 'mksysb' (AIX), 'Web Flash Archive' (Solaris) or various disk image tools for Windows systems create a complete copy of a computer (image). Only a boot diskette or boot CD and an appropriate image is needed to completely restore a computer without having to work through steps 2–7 described above. Particularly advantageous is the integration of image restore in a network backup system: to achieve this the network backup system must generate the appropriate boot disk. Furthermore, the boot diskette or boot CD must create a connection to the network backup system.

7.9.2 Backup of file systems

For the classical network backup of file systems, backup on different levels (block level, file level, file system image) has been discussed in addition to the incremental-forever strategy. The introduction of storage networks makes new methods available for the backup of file systems such as server-free backup, application server-free backup, LAN-free backup, shared disk file systems and instant copies.

The importance of the backup of file systems is demonstrated by the fact that manufacturers of file systems are providing new functions specifically targeted at the acceleration of backups. In the following we introduce two of these new functions – the so-called archive bit and block level incremental backup.

The archive bit supports incremental backups at file level such as, for example, the incremental-forever strategy. One difficulty associated with incremental backups is finding out quickly which files have changed since the previous backup. To accelerate this decision, the file system adds an archive bit to the metadata of each file: the network backup system sets this archive bit immediately after it has backed a file up on the backup server. Thus the archive bits of all files are set after a full backup. If a file is altered, the file system automatically clears its archive bit. Newly generated files are thus not given an archive bit. In the next incremental backup the network backup system knows that it only has to back up those files for which the archive bits have been cleared.

The principle of the archive bit can also be applied to the individual blocks of a file system in order to reduce the cost of backup on block level. In Section 7.4 a comparatively expensive procedure for backup on block level was introduced: the cost of the copying and comparing of files by the backup client is greatly reduced if the file system manages the quantity of altered blocks itself with the aid of the archive bit for blocks and the network backup system can call this up via an API.

Unfortunately, the principle of archive bits cannot simply be combined with the principle of instant copies: if the file system copies uses instant copy to copy within the disk subsystem for backup (Figure 7.10), the network backup system sets the archive bit only on the copy of the file system. In the original data the archive bit thus remains cleared even though the data has been backed up. Consequently, the network backup system backs this data up at the next incremental backup because the setting of the archive bit has not penetrated through to the original data.

7.9.3 Backup of NAS servers

NAS servers are preconfigured file servers; they consist of one or more internal servers, preconfigured disk capacity and usually a stripped-down or specific operating system (Section 4.2.2). NAS servers generally come with their own backup tools. However, just like the backup tools that come with operating systems, these tools represent an isolated solution (Section 7.1). Therefore, in the following we specifically consider the integration of the backup of NAS servers into an existing network backup system.

The optimal situation would be if there were a backup client for a NAS server that was adapted to suit both the peculiarities of the NAS server and also the peculiarities of the network backup system used. Unfortunately, it is difficult to develop such a backup client in practice.

If the NAS server is based upon a specific operating system the manufacturers of the network backup system sometimes lack the necessary interfaces and compilers to develop such a client. Even if the preconditions for the development of a specific backup client were in place, it is doubtful whether the manufacturer of the network backup system would develop a specific backup client for all NAS servers: the necessary development cost for a new backup client is still negligible in comparison to the testing cost that would have to be incurred for every new version of the network backup system and for every new version of the NAS server.

Likewise, it is difficult for the manufacturers of NAS servers to develop such a client. The manufacturers of network backup systems publish neither the source code nor the interfaces between backup client and backup server, which means that a client cannot be developed. Even if such a backup client already exists because the NAS server is based upon on a standard operating system such as Linux, Windows or Solaris, this does not mean that customers may use this client: in order to improve the Plug&Play-capability of NAS servers, customers may only use the software that has been tested and certified by the NAS manufacturer. If the customer installs non-certified software, then he can lose support for the NAS server. Due to the testing cost, manufacturers of NAS servers may be able to support some, but certainly not all network backup systems.

Without further measures being put in place, the only possibility that remains is to back the NAS server up from a client of the NAS server (Figure 7.12). However, this approach, too, is doubtful for two reasons.

First, this approach is only practicable for smaller quantities of data: for backup the files of the NAS server are transferred over the LAN to the NFS or CIFS client on which the backup client runs. Only the backup client can write the files to the backup medium using advanced methods such as LAN-free backup.

Second, the backup of metadata is difficult. If an NAS server supports the export of the local file system both via CIFS and also via NFS then the backup client only accesses one of the two protocols on the files – the metadata of the other protocol is lost. NAS servers would thus have to store their metadata in special files so that the network backup system

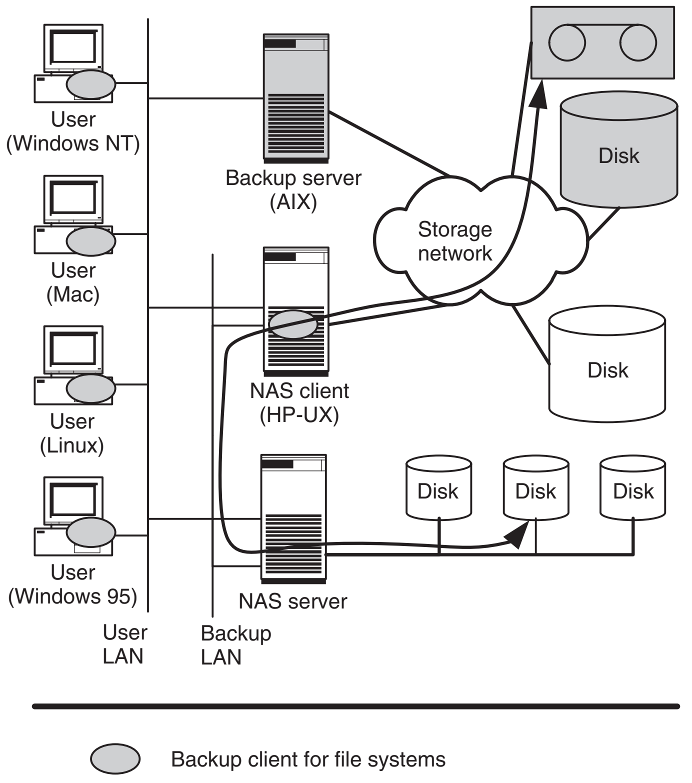

Figure 7.12 When backing up a NAS server over a network file system, the connection between the NAS server and backup client represents a potential performance bottleneck. Backup over a network file system makes it more difficult to back up and restore the metadata of the NAS server.

can back these up. There then remains the question of the cost for the restoring of a NAS server or a file system. The metadata of NAS servers and files has to be re-extracted from these files. It is dubious whether network backup systems can automatically initiate this process.

As a last resort for the integration of NAS servers and network backup systems, there remains only the standardisation of the interfaces between the NAS server and the network backup system. This would mean that manufacturers of NAS servers would only have to develop and test one backup client that supports precisely this interface. The backup systems of various manufacturers could then back up the NAS server via this interface. In such an approach the extensivity of this interface determines how well the backup of NAS servers can be linked into a network backup system. The next section introduces a standard for such an interface – the Network Data Management Protocol (NDMP).

7.9.4 The Network Data Management Protocol (NDMP)

The Network Data Management Protocol (NDMP) defines an interface between NAS servers and network backup systems that makes it possible to back up NAS servers without providing a specific backup client for them. More and more manufacturers – both of NAS servers and network backup systems – are supporting NDMP. The current version of NDMP is Version 4; Version 5 is in preparation.

More and more manufacturers of NAS servers and network backup systems support NDMP, making it a de facto standard. An Internet draft of NDMP Version 4 has been available for some time. Furthermore, a requirements catalogue exists for NDMP Version 5. NDMP Version 4 has some gaps, such as the backup of snapshots that some manufacturers deal with through proprietary extensions. These vendor-specific extensions relate to NDMP Version 4. However, NDMP has received widespread acceptance as a technology for the integration of NAS servers and network backup systems.

NDMP uses the term ‘data management operations’ to describe the backup and restoration of data. A so-called data management application (DMA) – generally a backup system – initiates and controls the data management operations, with the execution of a data management operation generally being called an NDMP session. The DMA cannot directly access the data; it requires the support of so-called NDMP services (Figure 7.13). NDMP services manage the current data storage, such as file systems, backup media and tape libraries. The DMA creates an NDMP control connection for the control of every participating NDMP service; for the actual data flow between source medium and backup medium a so-called NDMP data connection is established between the NDMP services in question. Ultimately, the NDMP describes a client-server architecture, with the DMA taking on the role of the NDMP client. An NDMP server is made up of one or more NDMP services. Finally, the NDMP host is the name for a computer that accommodates one or more NDMP servers.

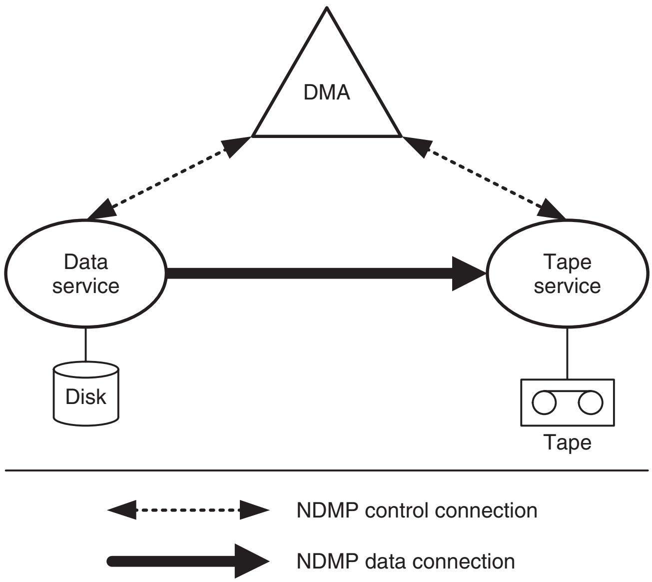

Figure 7.13 NDMP standardises the communication between the data management application (DMA) – generally a backup system – and the NDMP services (NDMP data service, NDMP tape service), which represent the storage devices. The communication between the NDMP services and the storage devices is not standardised.

NDMP defines different forms of NDMP services. All have in common that they only manage their local state. The state of other NDMP services remains hidden to an NDMP service. Individually, NDMP Version 4 defines the following NDMP services:

- *NDMP data service*
  The NDMP data service forms the interface to primary data such as a file system on a NAS server. It is the source of backup operations and the destination of restore operations. To backup a file system, the NDMP data service converts the content of the file system into a data stream and writes this in an NDMP data connection, which is generally created by means of a TCP/IP connection. To restore a file system it reads the data stream from an NDMP data connection and from this reconstructs the content of a file system. The data service only permits the backup of complete file systems; it is not possible to back up individual files. By contrast, individual files or directories can be restored in addition to complete file systems.

  The restore of individual files or directories is also called 'direct access recovery'. To achieve this, the data service provides a so-called file history interface, which it uses to forward the necessary metadata to the DMA during the backup. The file history stores the positions of the individual files within the entire data stream. The DMA cannot read this so-called file locator data, but it can forward it to the NDMP tape service in the event of a restore operation. The NDMP tape service then uses this information to wind the tape to the appropriate position and read the files in question.

- *NDMP tape service*
  The NDMP tape service forms the interface to the secondary storage. Secondary storage, in the sense of NDMP, means computers with connected tape drive, connected tape library or a CD burner. The tape service manages the destination of a backup or the source of a restore operation. For a backup, the tape service writes an incoming data stream to tape via the NDMP data connection; for a restore it reads the content of a tape and writes this as a data stream in a NDMP data connection. The tape service has only the information that it requires to read and write, such as tape size or block size. It has no knowledge of the format of the data stream. It requires the assistance of the DMA to mount tapes in a tape drive.
- *NDMP SCSI pass through service*
  The SCSI pass through service makes it possible for a DMA to send SCSI commands to a SCSI device that is connected to a NDMP server. The DMA requires this service, for example, for the mounting of tapes in a tape library.

The DMA holds the threads of an NDMP session together: it manages all state information of the participating NDMP services, takes on the management of the backup media and initiates appropriate recovery measures in the event of an error. To this end the DMA maintains an NDMP control connection to each of the participating NDMP services, which – like the NDMP data connections – are generally based upon TCP/IP. Both sides – DMA and NDMP services – can be active within an NDMP session. For example, the DMA sends commands for the control of the NDMP services, whilst the NDMP services for their part send messages if a control intervention by the DMA is required. If, for example, an NDMP tape service has filled a tape, it informs the DMA. This can then initiate a tape unmount by means of an NDMP SCSI pass through service.

The fact that both NDMP control connections and NDMP data connections are based upon TCP/IP means that flexible configuration options are available for the backup of data using NDMP. The NDMP architecture supports backup to a locally connected tape drive (Figure 7.14) and likewise to a tape drive connected to another computer, for example a second NAS server or a backup server (Figure 7.15). This so-called remote backup has the advantage that smaller NAS servers do not need to be equipped with a tape library. Further fields of application of remote backup are the replication of file systems (disk-to-disk remote backup) and of backup tapes (tape-to-tape remote backup).

In remote backup the administrator comes up against the same performance bottlenecks as in conventional network backup over the LAN (Section 7.6). Fortunately, NDMP local backup and LAN-free backup of network backup systems complement each other excellently: a NAS server can back up to a tape drive available in the storage network, with the network backup system coordinating access to the tape drive outside of NDMP by means of tape library sharing (Figure 7.16).

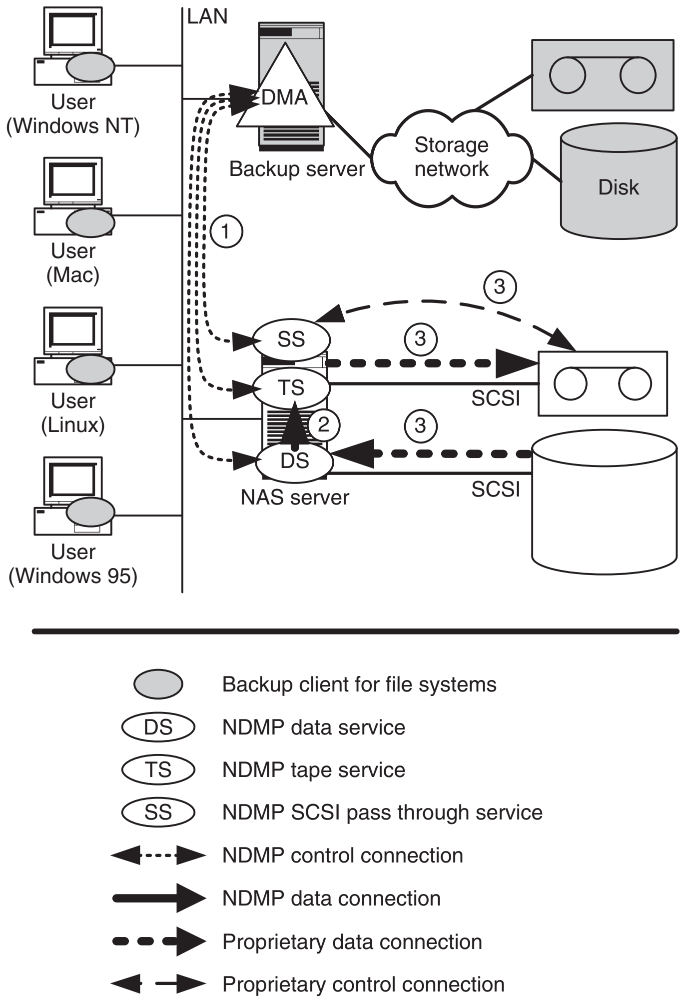

Figure 7.14 NDMP data service, NDMP tape service and NDMP SCSI pass through service all run on the same computer in a local backup using NDMP. NDMP describes the protocols for the NDMP control connection (1) and the NDMP data connection (2). The communication between the NDMP services and the storage devices is not standardised (3).

NDMP Version 4 offers the possibility of extending the functionality of the protocol through extensions. This option is being used by some manufacturers to provide some important functions that NDMP is lacking – for example, the backup and management of snapshots. Unfortunately, these extensions are vendor specific. This means, for example, that it is important in each individual case to check carefully that a specific network backup system can back up the snapshots of a certain NAS server.

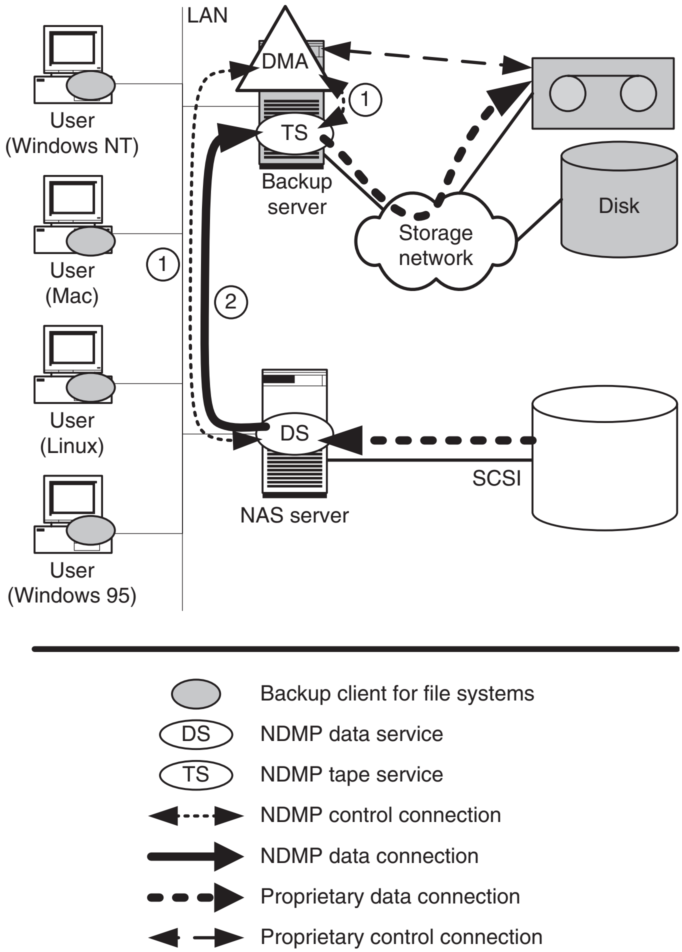

Figure 7.15 In a backup over the LAN (remote backup) the NDMP tape service runs on the computer to which the backup medium is connected. The communication between the remote services is guaranteed by the fact that NDMP control connections (1) and NDMP data connections (2) are based upon TCP/IP. The backup server addresses the tape library locally, which means that the NDMP SCSI pass through service is not required here.

In Version 5, NDMP will have further functions such as multiplexing, compressing and encryption. To achieve this, NDMP Version 5 expands the architecture to include the so-called translator service (Figure 7.17). Translator services process the data stream (data stream processor): they can read and change one or more data streams. The implementation of translator services is in accordance with that of previous NDMP services. This means that the control of the translator service lies with the DMA; other participating NDMP

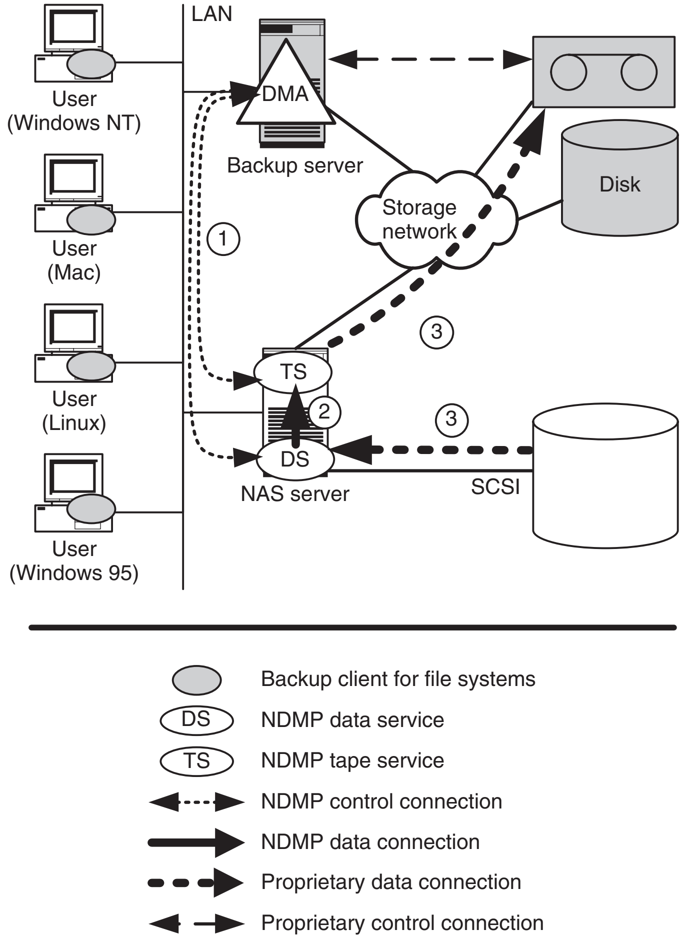

Figure 7.16 NDMP local backup can be excellently combined with the LAN-free backup of network backup systems.

services cannot tell whether an incoming data stream was generated by a translator service or a different NDMP service. NDMP Version 5 defines the following translator services:

- *Data stream multiplexing*
  The aim of data stream multiplexing is to bundle several data streams into one data stream (N:1-multiplexing) or to generate several data streams from one (1:M-multiplexing). Examples of this are the backup of several small, slower file systems onto a faster tape drive (N:1-multiplexing) or the parallel backup of a large file system onto several tape drives (1:M-multiplexing).

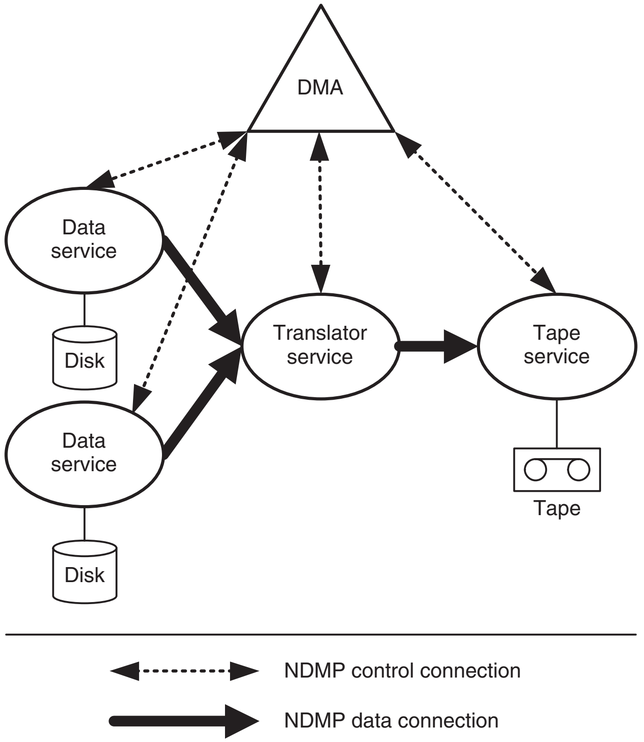

Figure 7.17 NDMP Version 5 expands the NDMP services to include translator services, which provide functions such as multiplexing, encryption and compression.

- *Data stream compression*
  In data stream compression the translator service reads a data stream, compresses it and sends it back out. Thus the data can be compressed straight from the hard disk, thus freeing up the network between it and the backup medium.
- *Data stream encryption*
  Data stream encryption works on the same principle as data stream compression, except that it encrypts data instead of compressing it. Encryption is a good idea, for example, for the backup of small NAS servers at branch offices to a backup server in a data centre via a public network.

NDMP offers many opportunities to connect NAS servers to a network backup system. The prerequisite for this is NDMP support on both sides. NDMP data services cover approximately the functions that backup clients of network backup systems provide. One weakness of NDMP is the backup of the NAS server metadata, which makes the restoration of a NAS server after the full replacement of hardware significantly more difficult (Section 7.9.1). Furthermore, there is a lack of support for the backup of file systems with the aid of snapshots or instant copies. Despite these missing functions NDMP has established itself as a standard and so we believe that it is merely a matter of time before NDMP is expanded to include these functions.

7.10 BACKUP OF DATABASES

Databases are the second most important organisational form of data after the file systems discussed in the previous section. Despite the measures introduced in Section 6.3.5, it is sometimes necessary to restore a database from a backup medium. The same questions are raised regarding the backup of the metadata of a database server as for the backup of file servers (Section 7.9.1). On the other hand, there are clear differences between the backup of file systems and databases. The backup of databases requires a fundamental understanding of the operating method of databases (Section 7.10.1). Knowledge of the operating method of databases helps us to perform both the conventional backup of databases without storage networks (Section 7.10.2) and also the backup of databases with storage networks and intelligent storage subsystems (Section 7.10.3) more efficiently.

7.10.1 Functioning of database systems

One requirement of database systems is the atomicity of transactions, with transactions bringing together several write and read accesses to the database to form logically coherent units. Atomicity of transactions means that a transaction involving write access should be performed fully or not at all.

Transactions can change the content of one or more blocks that can be distributed over several hard disks or several disk subsystems. Transactions that change several blocks are problematic for the atomicity. If the database system has already written a few of the blocks to be changed to disk and has not yet written others and then the database server goes down due to a power failure or a hardware fault, the transaction has only partially been performed. Without additional measures the transaction can neither be completed nor undone after a reboot of the database server because the information necessary for this is no longer available. The database would therefore be inconsistent.

The database system must therefore store additional information regarding transactions that have not yet been concluded on the hard disk in addition to the actual database. The database system manages this information in so-called log files. It first of all notes every pending change to the database in a log file before going on to perform the changes to the blocks in the database itself. If the database server fails during a transaction, the database system can either complete or undo incomplete transactions with the aid of the log file after the reboot of the server.

Figure 7.18 shows a greatly simplified version of the architecture of database systems. The database system fulfils the following two main tasks:

- *Database: storing the logical data structure to block-oriented storage*
  First, the database system organises the data into a structure suitable for the applications and stores this on the block-oriented disk storage. In modern database systems the relational data model, which stores information in interlinked tables, is the main model used for this. To be precise, the database system stores the logical data directly onto the disk, circumventing a file system, or it stores it to large files. The advantages and disadvantages of these two alternatives have already been discussed in Section 4.1.1.
- *Transaction machine: changing the database*
  Second, the database system realises methods for changing the stored information. To this end, it provides a database language and a transaction engine. In a relational database the users and applications initiate transactions via the database language SQL and thus call up or change the stored information. Transactions on the logical, application-near data structure thus bring about changes to the physical blocks on the disks. The transaction system ensures, amongst other things, that the changes to the data

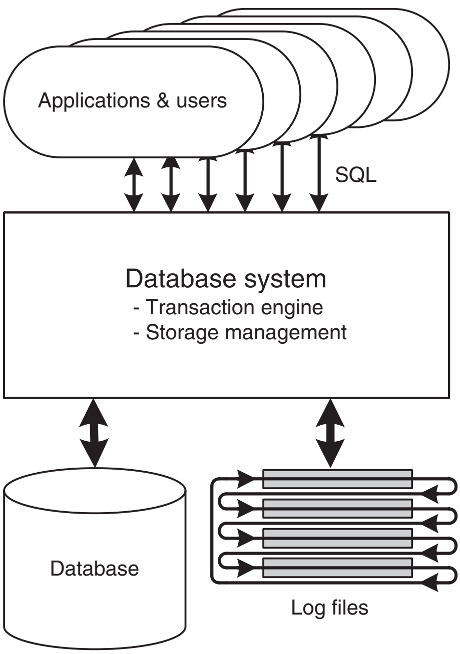

Figure 7.18 Users start transactions via the database language (SQL) in order to read or write data. The database system stores the application data in block-oriented data (database) and it uses log files to guarantee the atomicity of the transactions.

set caused by a transaction are either completed or not performed at all. As described above, this condition can be guaranteed with the aid of log files even in the event of computer or database system crashes.

The database system changes blocks in the data area, in no specific order, depending on how the transactions occur. The log files, on the other hand, are always written sequentially, with each log file being able to store a certain number of changes. Database systems are generally configured with several log files written one after the other. When all log files have been fully written, the database system first overwrites the log file that was written first, then the next, and so on.

A further important function for the backup of databases is the backup of the log files. To this end, the database system copies full log files into a file system as files and numbers these sequentially: logfile 1, logfile 2, logfile 3, etc. These copies of the log files are also called archive log files. The database system must be configured with enough log files that there is sufficient time to copy the content of a log file that has just been fully written into an archive log file before it is once again overwritten.

7.10.2 Classical backup of databases

As in all applications, the consistency of backed up data also has to be ensured in databases. In databases, consistency means that the property of atomicity of the transactions is maintained. After the restore of a database it must therefore be ensured that only the results of completed transactions are present in the data set. In this section we discuss various backup methods that guarantee precisely this. In the next section we explain how storage networks and intelligent storage systems help to accelerate the backup of databases (Section 7.10.3).

The simplest method for the backup of databases is the so-called cold backup. For cold backup, the database is shut down so that all transactions are concluded, and then the files or volumes in question are backed up. In this method, databases are backed up in exactly the same way as file systems. In this case it is a simple matter to guarantee the consistency of the backed up data because no transactions are taking place during the backup.

Cold backup is a simple to realise method for the backup of databases. However, it has two disadvantages. First, in a 24 × 7 environment you cannot afford to shut down databases for backup, particularly as the backup of large databases using conventional methods can take several hours. Second, without further measures all changes since the last backup would be lost in the event of the failure of a disk subsystem. For example, if a database is backed up overnight and the disk subsystem fails on the following evening all changes from the last working day are lost.

With the aid of the archive log file the second problem, at least, can be solved. The latest state of the database can be recreated from the last backup of the database, all archive log files backed up since and the active log files. To achieve this, the last backup of the database must first of all be restored from the backup medium – in the example above the backup from the previous night. Then all archive log files that have been created since the last backup are applied to the data set, as are all active log files. This procedure, which is also called forward recovery of databases, makes it possible to restore the latest state even a long time after the last backup of the database. However, depending upon the size of the archive log files this can take some time.

The availability of the archive log files is thus an important prerequisite for the successful forward recovery of a database. The file system for the archive log files should, therefore, be stored on a different hard disk to the database itself (Figure 7.19) and additionally protected by RAID. Furthermore, the archive log files should be backed up regularly.

Log files and archive log files form the basis of two further backup methods for databases: hot backup and fuzzy backup. In hot backup, the database system writes pending changes to the database to the log files only. The actual database remains unchanged

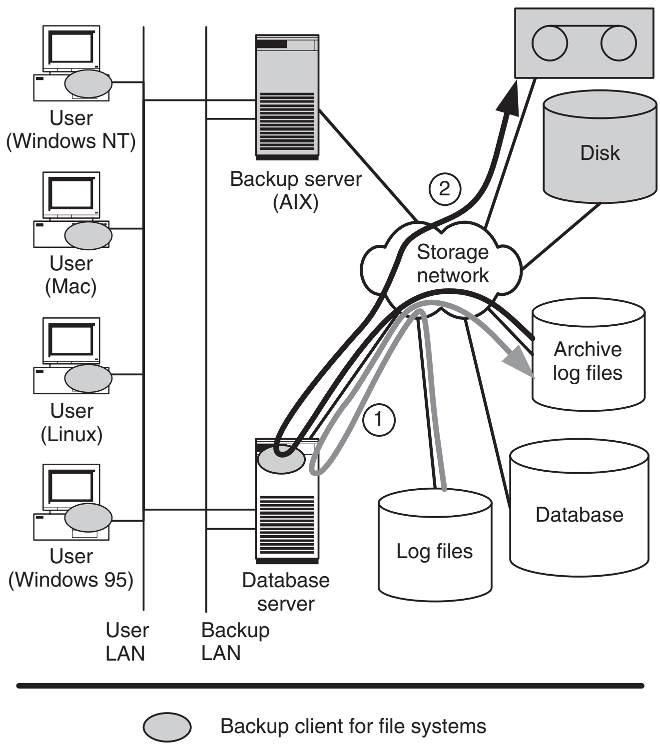

Figure 7.19 The database system copies the archive log files into a file system (1) located on a different storage system to the database and its log files. From there, the archive log files can be backed up using advanced techniques such as LAN-free backup.

at this time, so that the consistency of the backup is guaranteed. After the end of the backup, the database system is switched back into the normal state. The database system can then incorporate the changes listed in the log files into the database.

Hot backup is suitable for situations in which access to the data is required around the clock. However, hot backup should only be used in phases in which a relatively low number of write accesses are taking place. If, for example, it takes two hours to back up the database and the database is operating at full load, the log files must be dimensioned so that they are large enough to be able to save all changes made during the backup. Furthermore, the system must be able to complete the postponed transactions after the backup in addition to the currently pending transactions. Both together can lead to performance bottlenecks.

Finally, fuzzy backup allows changes to be made to the database during its backup so that an inconsistent state of the database is backed up. The database system is nevertheless capable of cleaning the inconsistent state with the aid of archive log files that have been written during the backup.

With cold backup, hot backup and fuzzy backup, three different methods are available for the backup of databases. Network backup systems provide backup clients for databases, which means that all three backup methods can be automated with a network backup system. According to the principle of keeping systems as simple as possible, cold backup or hot backup should be used whenever possible.

7.10.3 Next generation backup of databases

The methods introduced in the previous section for the backup of databases (cold backup, hot backup and fuzzy backup) are excellently suited for use in combination with storage networks and intelligent storage subsystems. In the following we show how the backup of databases can be performed more efficiently with the aid of storage networks and intelligent storage subsystems.

The integration of hot backup with instant copies is an almost perfect tool for the backup of databases. Individually, the following steps should be performed:

1. Switch the database over into hot backup mode so that there is a consistent data set in the storage system.
2. Create the instant copy.
3. Switch the database back to normal mode.
4. Back up the database from the instant copy.

This procedure has two advantages: first, access to the database is possible throughout the process. Second, steps 1–3 only take a few seconds, so that the database system only has to catch up comparatively few transactions after switching back to normal mode.

Application server-free backup expands the backup by instant copies in order to additionally free up the database server from the load of the backup (Section 7.8.5). The concept shown in Figure 7.10 is also very suitable for databases. Due to the large quantity of data involved in the backup of databases, LAN-free backup is often used – unlike in the figure – in order to back up the data generated using instant copy.

In the previous section (Section 7.10.2) we explained that the time of the last backup is decisive for the time that will be needed to restore a database to the last data state. If the last backup was a long time ago, a lot of archive log files have to be reapplied. In order to reduce the restore time for a database it is therefore necessary to increase the frequency of database backups.

The problem with this approach is that large volumes of data are moved during a complete backup of databases. This is very time-consuming and uses a lot of resources, which means that the frequency of backups can only be increased to a limited degree. Likewise, the delayed copying of the log files to a second system (Section 6.3.5) and the holding of several copies of the data set on the disk subsystem by means of instant copy can only seldom be economically justified due to the high hardware requirement and the associated costs.

In order to nevertheless increase the backup frequency of a database, the data volume to be transferred must therefore be reduced. This is possible by means of an incremental backup of the database on block level. The most important database systems offer backup tools for this by means of which such database increments can be generated. Many network backup systems provide special adapters (backup agents) that are tailored to the backup tools of the database system in question. However, the format of the increments is unknown to the backup software, so that the incremental-forever strategy cannot be realised in this manner. This would require manufacturers of database systems to publish the format of the increments.

The backup of databases using the incremental-forever strategy therefore requires that the backup software knows the format of the incremental backups, so that it can calculate the full backups from them. To this end, the storage space of the database must be provided via a file system that can be incrementally backed up on block level using the appropriate backup client. The backup software knows the format of the increments so the incremental-forever strategy can be realised for databases via the circuitous route of file systems.

7.11 ORGANISATIONAL ASPECTS OF BACKUP

In addition to the necessary technical resources, the personnel cost of backing data up is also often underestimated. We have already discussed (1) how the backup of data has to be continuously adapted to the ever-changing IT landscape; and (2) that it is necessary to continuously monitor whether the backup of data is actually performed according to plan. Both together quite simply take time, with the time cost for these activities often being underestimated.

As is the case for any activity, human errors cannot be avoided in backup, particularly if time is always short due to staff shortages. However, in the field of data protection these human errors always represent a potential data loss. The costs of data loss can be enormous: for example, Marc Farley (Building Storage Networks, 2000) cites a figure of US$ 1000 per employee as the cost for lost e-mail databases. Therefore, the personnel requirement for the backup of data should be evaluated at least once a year. As part of this process, personnel costs must always be compared to the cost of lost data.

The restore of data sometimes fails due to the fact that data has not been fully backed up, tapes have accidentally been overwritten with current data or tapes that were already worn and too old have been used for backups. The media manager can prevent most of these problems.

However, this is ineffective if the backup software is not correctly configured. One of the three authors can well remember a situation in the early 1990ies in which he was not able to restore the data after a planned repartitioning of a disk drive. The script for the backup of the data contained a single typing error. This error resulted in an empty partition being backed up instead of the partition containing the data.

The restore of data should be practised regularly so that errors in the backup are detected before an emergency occurs, in order to practise the performance of such tasks and in order to measure the time taken. The time taken to restore data is an important cost variable: for example, a multi-hour failure of a central application such as SAP can involve significant costs.

Therefore, staff should be trained in the following scenarios, for example:

- Restoring an important server including all applications and data to equivalent hardware;
- Restoring an important server including all applications and data to new hardware;
- Restoring a subdirectory into a different area of the file system;
- Restoring an important file system or an important database;
- Restoring several computers using the tapes from the off-site store;
- Restoring old archives (are tape drives still available for the old media?).

The cost in terms of time for such exercises should be taken into account when calculating the personnel requirement for the backup of data.

7.12 SUMMARY

Storage networks and intelligent storage subsystems open up new possibilities for solving the performance problems of network backup. However, these new techniques are significantly more expensive than classical network backup over the LAN. Therefore, it is first necessary to consider at what speed data really needs to be backed up or restored. Only then is it possible to consider which alternative is the most economical: the new techniques will be used primarily for heavyweight clients and for 24 × 7 applications. Simple clients will continue to be backed up using classical methods of network backup and for medium-sized clients there remains the option of installing a separate LAN for the backup of data. All three techniques are therefore often found in real IT systems nowadays.

Data protection is a difficult and resource-intensive business. Network backup systems allow the backup of data to be largely automated even in heterogeneous environments. This automation takes the pressure off the system administrator and helps to prevent errors such as the accidental overwriting of tapes. The use of network backup systems is indispensable in large environments. However, it is also worthwhile in smaller environments. Nevertheless, the personnel cost of backup must not be underestimated.

This chapter started out by describing the general conditions for backup: strong growth in the quantity of data to be backed up, continuous adaptation of backup to ever-changing IT systems and the reduction of the backup window due to globalisation. The transition to network backup was made by the description of the backup, archiving and hierarchical storage management (HSM). We then discussed the server components necessary for the implementation of these services (job scheduler, error handler, media manager and metadata database) plus the backup client. At the centre was the incremental-forever strategy and the storage hierarchy within the backup server. Network backup was also considered from the point of view of performance: we first showed how network backup systems can contribute to using the existing infrastructure more efficiently. CPU load, the clogging of the internal buses and the inefficiency of the TCP/IP/Ethernet medium were highlighted as performance bottlenecks. Then, proposed solutions for increasing performance that are possible within a server-centric IT architecture were discussed, including their limitations. This was followed by proposed solutions to overcome the performance bottlenecks in a storage-centric IT architecture. Finally, the backup of large file systems and databases was described and organisational questions regarding network backup were outlined.

The next chapter (Chapter 8) on archiving considers another important field of application for storage networks. In contrast to backup, archiving is about 'freezing' data and keeping it for years or even decades. In the subsequent chapter (Chapter 9) we will discuss backup as an important building block in business continuity.