Chapter 3

Decoupling the Storage Component: Putting Storage on the Network

In Chapter 2, we found the increasing size of online data to be a driving factor in pushing the limitations of the traditional client/server storage model. The processes that surround the design and development of business applications, and which fuel data growth, are not equipped to supplement the planning and design of storage infrastructures, no matter how critical storage becomes. Once applications reach production stage, the realities regarding actual users and the limitations of current storage environments are quickly quantified in application performance. The addition of support applications, generally addressed as an after thought, will further increase demands on new or current storage devices. As the likelihood of major renovations to existing storage infrastructures increases, post-implementation enhancements become very active, while the impact on business driven by the risk to successful application implementation becomes an even greater concern.

A solution to these problems is the design and implementation of more scalable and responsive storage infrastructures based upon storage networking concepts. As we discussed in Chapter 2, storage networking requires separating the traditional storage components from their server counterparts. This is a major shift from tightly coupled architectures of the traditional client/server era. However, once implemented, users can greatly benefit from increased scalability in both size and access and lower the risk of application implementation and maintenance failures.

This chapter will explore the concepts surrounding one of the current and increasingly popular models of storage networking: Network Attached Storage or NAS. NAS facilitates the decoupling of storage from the server by allowing storage to be directly connected to existing client/server networks. It provides a method of addressing the data access and storage challenges, while creating a scalable storage infrastructure that will support multiple applications. In addition, this model can compensate for the lack of planning that generally arises in the implementation frenzy of application development, not to mention the ad hoc need to provide ongoing storage capacity for business demands.

The NAS Idea

Prior to the days of Windows networking, Java downloads, and the commercialization of the ARPANET, early users of computer networks worked within the UNIX operating system environments. ARPANET, incidentally, was the global Internet's progenitor; it stood for Advanced Research Projects Agency Network, an agency of the U.S. Department of Defense. As their networks and applications grew, they found the problem of storage an increasing frustration. Providing storage on one's own workstation that was shared within the network could be hazardous to your work as well as to the users that shared that data. Accessing storage on someone else's workstation was just as frustrating, especially if they had left and turned off their computer for the night (see Figure 3-1).

Then an idea occurred to them. If users are going to share files, why not simply dedicate a computer to house large and shared files? For example, a computer connected to an existing network could have storage space shared with users throughout the

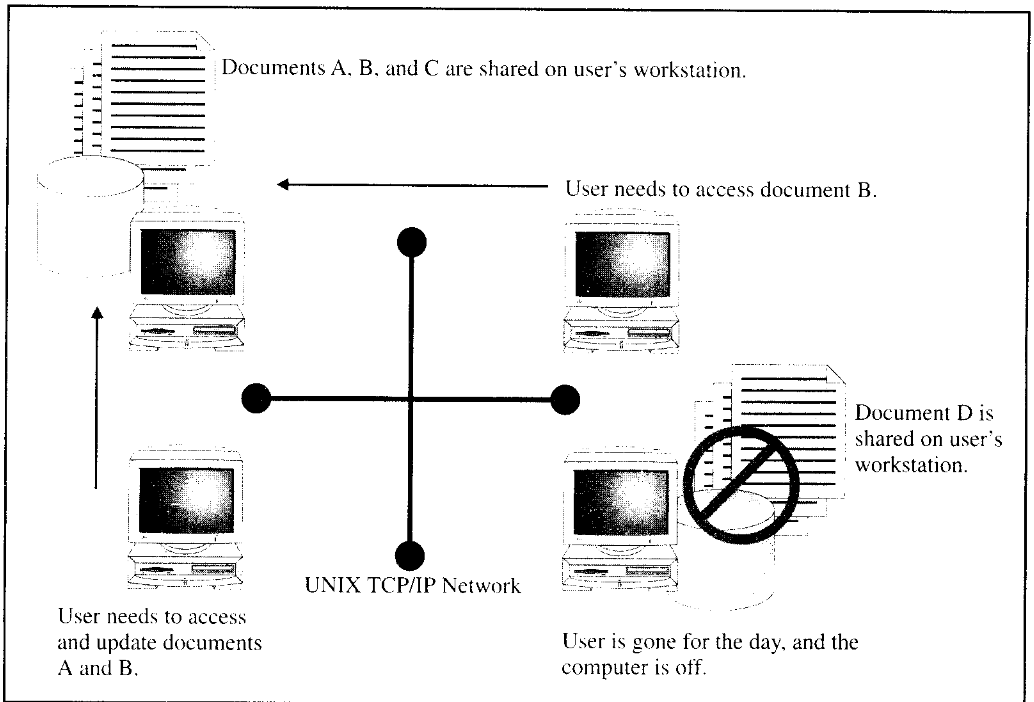


Figure 3-1. *Early UNIX networking storage issues*

network. That's exactly how early networked UNIX environments evolved to share data and depend on a file "Server," as depicted in Figure 3-2.

Given that working environment, the next logical step was the innovation of a server optimized for file I/O. The reasoning was that additional overhead for the operating system wasn't needed; the server should have the capacity for larger disk storage space, and be reliable most of the time. Any additional functionality that a user might require, like running other applications, or displaying and using graphic functions, wouldn't be needed.

Thus, it's necessary that a computer used in this fashion incorporate a large amount of disk storage to be integrated with a "thin" OS that only does I/O, and that it can be attached to existing networks. In addition, it should allow users to access their files on this special server as if the files were on their own workstation, essentially allowing the files to be shared with other users throughout the network. As UNIX environments became the platform for innovation—due to its unrestricted code base, use with academic communities, and virtually free price tag—the setting for the development of an optimized network file server was put in motion.

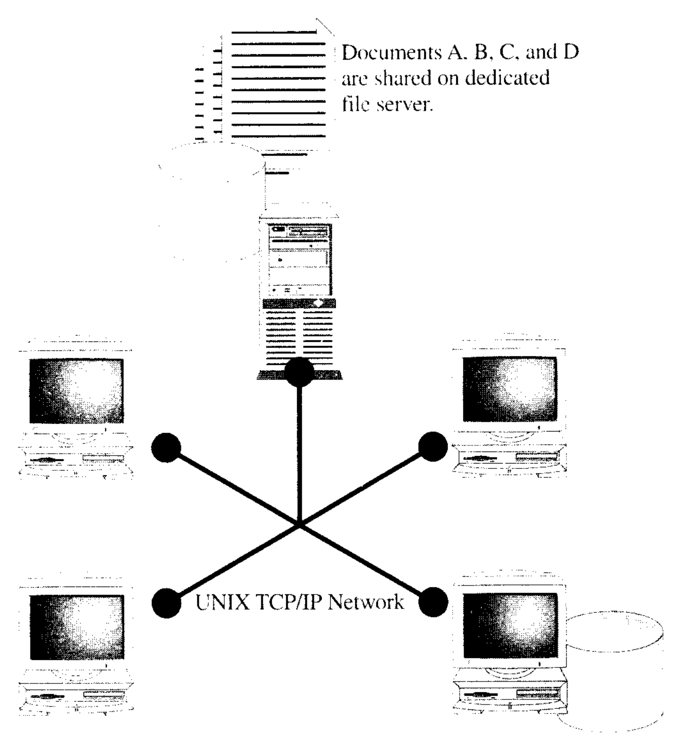


Figure 3-2. *Early UNIX file servers*

NAS Evolution and Development

The built-in networking functionality of UNIX prompted the development of a system that not only allowed users to share files across the network but which provided a method of organizing data that allowed the server to share its files with users within the network. Thus, the importance of the UNIX file system became apparent. The file system, as shown in Figure 3-3, forms the foundation for data sharing.

The file system is a method by which data is organized and accessed within a computer system. Generally a component of the operating system, it's employed by users and the OS itself. On behalf of application programs and interactive users, the system organizes data into logically defined instances, where the operating system can find information it needs to collect data on their behalf, and perform an I/O function to satisfy their request. The OS and the users form logical associations between the segments of data—which are referred to as files—through a naming convention as defined by the operating system.

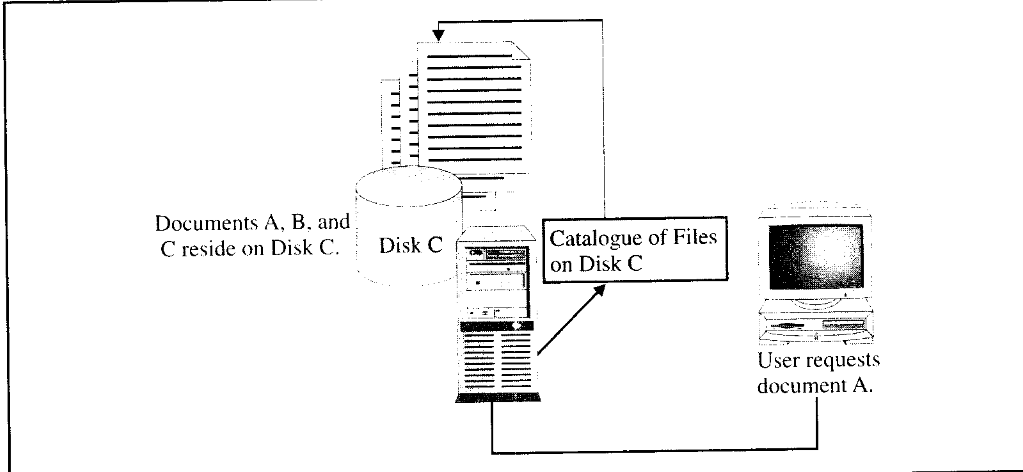


Figure 3-3. *The file system*

Emergence of the Network File System (NFS) drove the innovation behind the development of a special computer—whose only job was to do I/O. Two pioneering innovators, Auspex and Network Appliance, became the first to offer a downsized UNIX OS with larger amounts of disk storage and the capability to connect to a TCP/IP network. It leveraged an NFS file system organization that allowed users and their client computers to connect to their data, as shown in Figure 3-4. The result was the emergence of the first Network Attached Storage product.

The early NAS storage devices were UNIX-based and used largely in areas of academic research, scientific applications, and computer science development. The early deployment provided a method of storing large amounts of data used in research areas, such as simulating meteorological conditions, geological conditions, and nuclear testing—in other words, it aided any project that involved massive amounts of data which couldn't be easily stored on individual workstations or file servers. Early NAS products also proved to be quite productive in code development areas, where computer code had to be shared between developers and development teams. Figures 3-5a and 3-5b depict these types, respectively.

The emergence of networking required the file system to become shared throughout the computers on the network. Consequently, the network file system allows a computer to make available its file system to other users within the network. Other systems and users within the network can attach to the file system and access the data. The remote files from the shared computer are thus seen as local files by the local operating system, thereby providing a virtual set of files which don't actually exist on the local computer. Instead, their location is *Virtual*, or *Virtualized*.

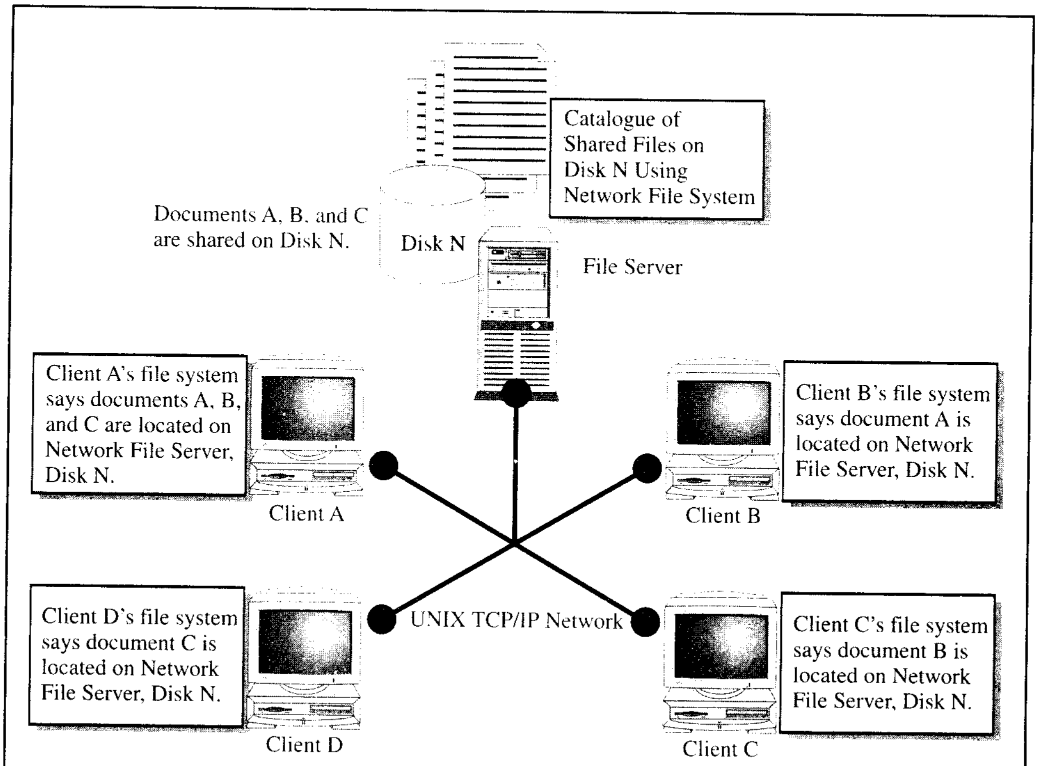


Figure 3-4. *The network file system*

The next evolution was the integration into the traditional client/server environments supporting Windows operating systems. This initially provided a much more hybrid approach, given the capacities of early windows networks, proprietary protocols, and usage of database servers running UNIX. The first stage of Windows networking was the formation of a type of network file server, still in use today, using proprietary networking systems. Figure 3-6 depicts these early configurations that functioned as "File Servers," not necessarily network attached storage, but which served in a similar capacity to the early UNIX servers mentioned previously.

However, Windows did have the capability to perform Remote Procedure Calls (RPCs) and through this function it was able to participate within a networked file system. The capability to provide and link to network drives became the Windows NFS compatible architecture, with the capability to participate within TCP/IP networks as well as proprietary networks based on Novell and Banyan proprietary packet-switched architectures. (More detailed information on file systems, network drives, and virtualization can be found in Part II of this book.)

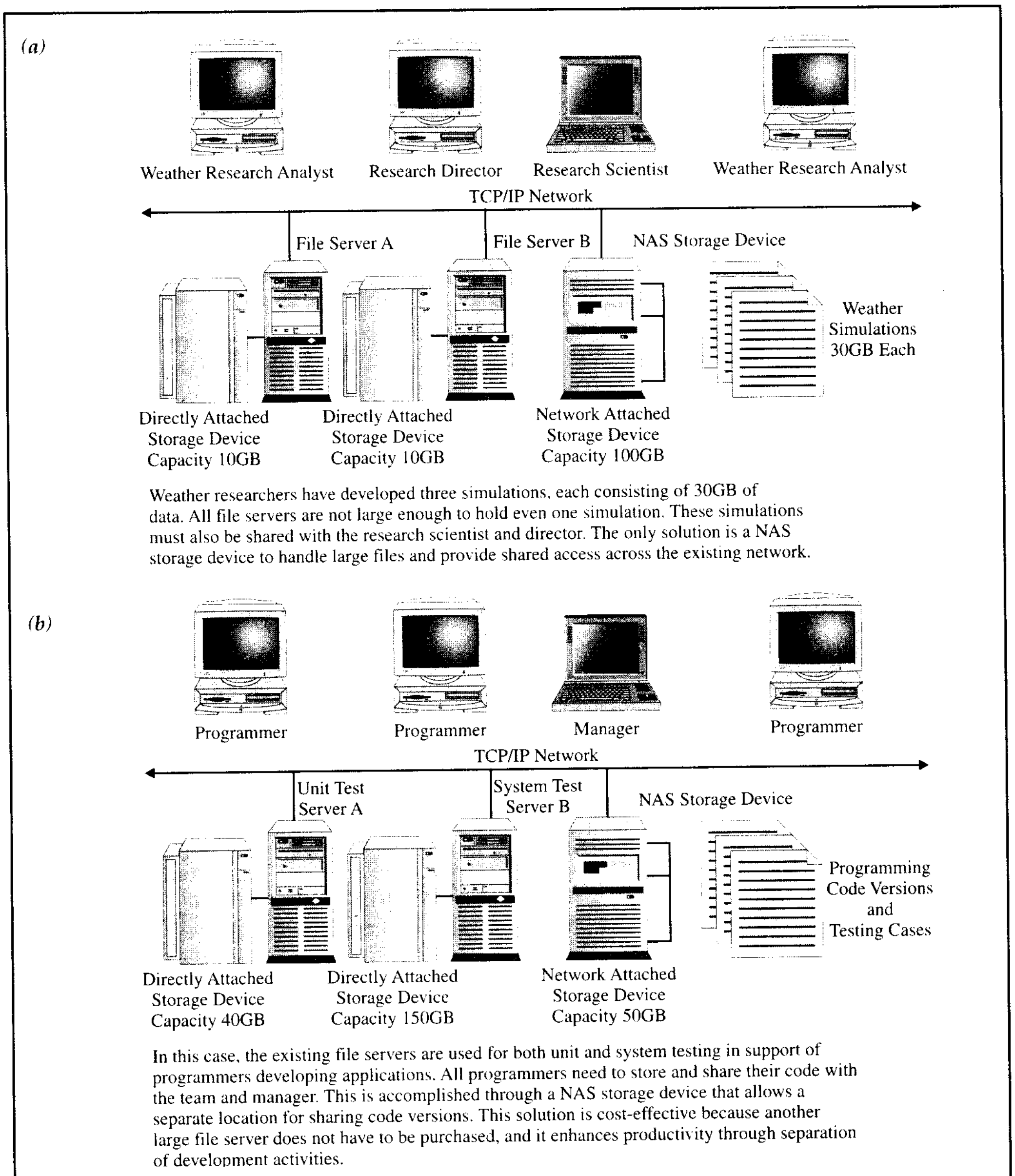


Figure 3-5. *(a) Early NAS scientific applications; (b) Early NAS code development applications*

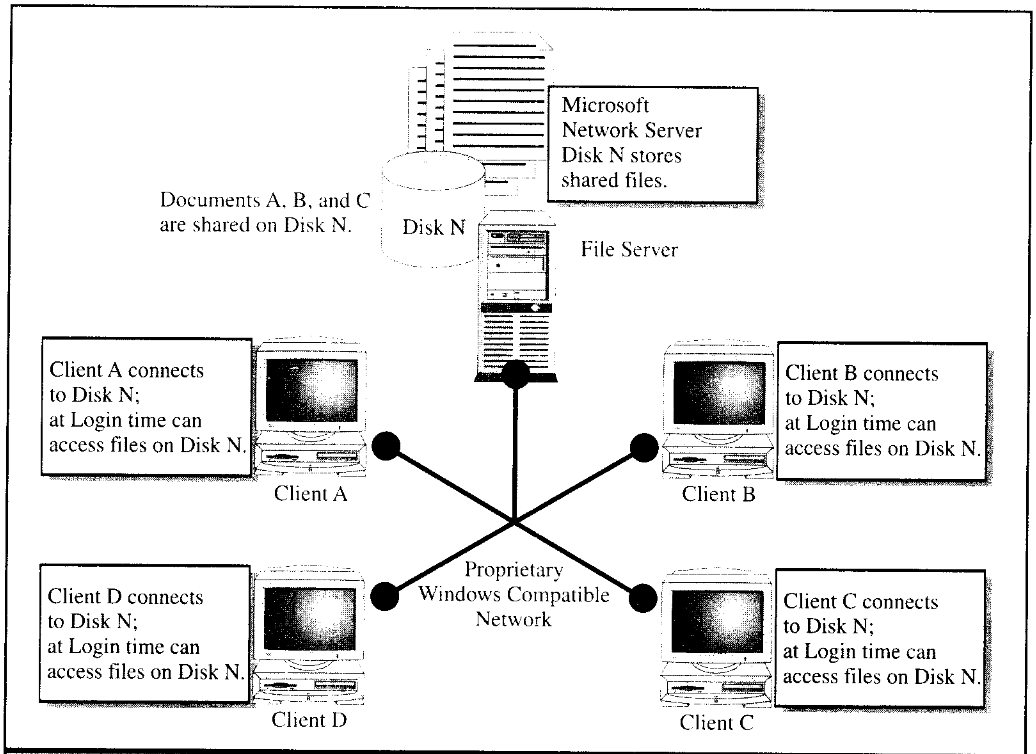


Figure 3-6. *Early Microsoft file servers*

These networking architectures gave way to the networking protocol standards of TCP/IP, based upon the standard OSI model. (Open Systems Interconnection is a standard description or "reference model" for how messages should be transmitted between any two points in a telecommunication network.) Although largely driven by everyone's attempt to integrate with the Internet and growing web application infrastructure, this was something that the early UNIX users had long been doing—for example, already connecting to ARPANET.

These activities caused the development of a file system for use within the prolific web server software. Although Internet servers evolved and consisted largely of UNIX-based servers, the web application software phenomenon was founded on Windows systems. This formed the basis of the Common Internet File System (CIFS), a file system that was compatible with Windows web Internet environments and which could participate within a standard TCP/IP networked environment—and thus within UNIX-based environments.

Therefore those NAS vendors who already supported TCP/IP networks and network file systems, moved quickly to support multiple operating environments and file systems. Today's NAS devices come in a wide variety of flavors. However, most industry leaders support both UNIX-based environments based upon an NFS file system architecture with support for Windows through CIFS file systems. As indicated in Figure 3-7 these offerings often allow both systems access to a NAS device.

As we discussed previously, early NAS devices supported UNIX environments only. These file-oriented servers were used in academic, government, and computer science applications in which data requirements surpassed standard internal and external storage devices where no other device could be used. These environments furthered the "black box" characteristics of NAS as a storage device. The reason being that users could not configure NAS servers to run other applications or even access the OS itself. The OS in the NAS box is what is called a Real-Time Operating System (RTOS), a clever name for a specialized UNIX OS with a special set of functions for a specific assignment. In the NAS case, this was file I/O—for example, the capability to optimize the execution of file access to the connected storage, and leverage the NFS system as a shared catalogue for network users.

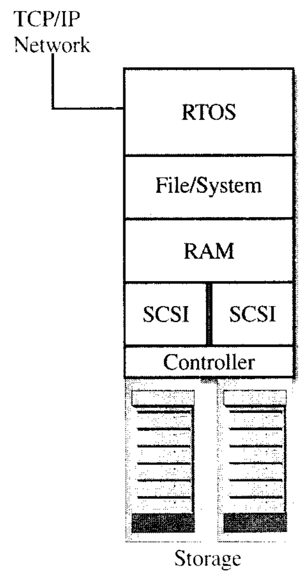


Figure 3-7. *Schematic of the NAS configuration*

An Operational Overview of NAS

NAS boxes remain thin servers with large storage capacities that function as dedicated I/O file servers. Requests from clients are routed to the NAS server through network file systems that are installed on subscribing application servers. As a result, NAS configurations become dedicated I/O extensions to multiple servers. Usage has given way to popular relational databases (for example, Microsoft SQL/Server and Oracle's relational database products), although these remain problematic given their file orientation (see the section titled "NAS Caveats" later in this chapter).

NAS servers work today in a number of different environments and settings, the most diverse being in storage networking. However, their value continues to focus on data access issues within high-growth environments, as well as how they address particular size challenges found in today's diversity of data types.

Leveraging NAS for Data Access

The inherent value of NAS continues to be its capability to provide storage quickly and cost effectively by using the resources that already exist in the data center. Today's solutions offer compatibility in both UNIX and Windows environments and connect easily into users' TCP/IP networks.

A typical solution for handling a large number of users who need access to data is depicted in Figure 3-8. Here, an end-user constituency of 1500 is represented. In the

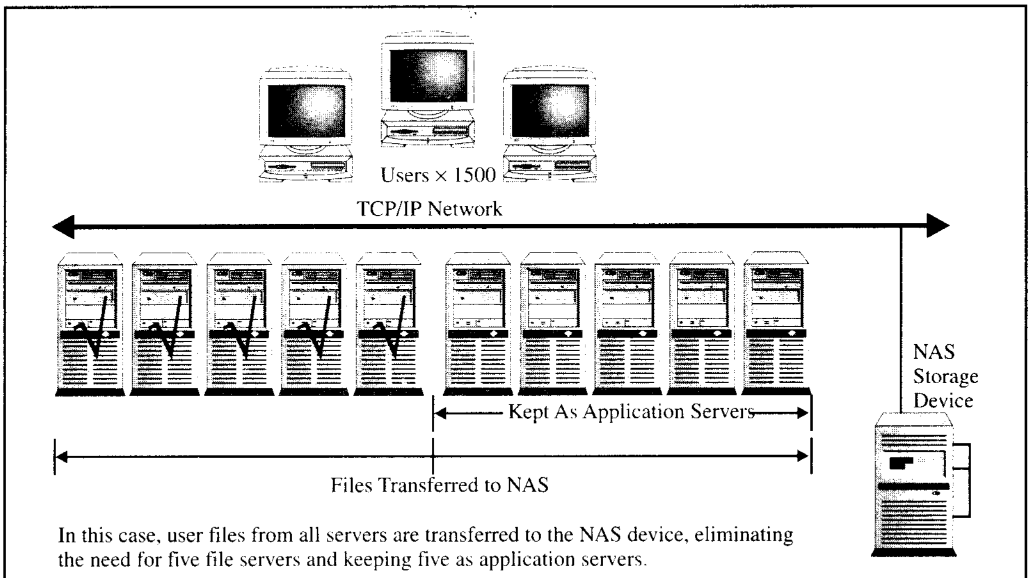


Figure 3-8. *A NAS for data access*

scenario, the users access data files under 100KB in size for read-only purposes that are stored across a number of servers. If the user base increases by another thousand over the next year, the workload could be handled by upgrading current servers as well as adding new servers with larger storage sizes to support this increase in data and traffic. This solution, however, would be costly and difficult to manage.

However, as Figure 3-8 shows, the NAS solution is used to combine the data spread across the current ten servers, placing it on the NAS storage device. Users' requests for data now come to one of the five application servers attached to the network and are then redirected to the NAS box where their request for data from a particular file is executed. Upon completion of the requested I/O operation within the NAS box, the data request is redirected to the requesting user. Because the NAS devices are optimized to handle I/O, their capacity for I/O operations exceeds typical general-purpose servers. The increased I/O capacity accounts for the collapsing of servers from ten to five. Moreover, they can scale higher in storage capacity. As the users and their respective data grows, increasing the NAS devices provides a more scalable solution for handling increased data access requirements within the scenario.

This solution does two major things for the data center. First, it provides a much larger window in which the applications can operate before they reach the non-linear performance curves. Given that both users and storage resources can increase on a linear basis, this adds to the value and stability of the enterprise storage infrastructure and can support multiple applications of this type through these configurations.

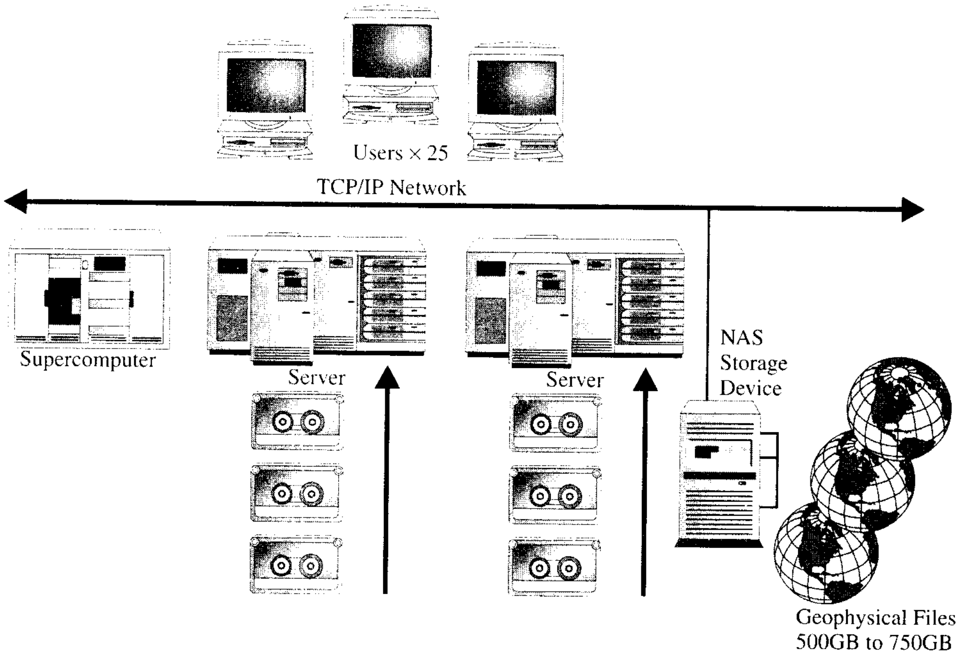
Secondly, it provides a cost-effective solution by applying optimized resources to the problem. By utilizing resources that existed within the data center—for example, existing server file systems and the TCP/IP network—it eliminated the need for additional general-purpose servers and associated network hardware and software. More importantly, it reduced the need for additional personnel to manage these systems.

Leveraging NAS for Data Size

The ability of a single server to handle large amounts of data is becoming a common problem. As Figure 3-5 illustrates, a configuration supporting storage of geophysical data can be tremendously large but with relatively few users and less-sensitive availability requirements. However, storing gigabyte-sized files is a problem for general-purpose servers. Not only is it difficult to access the data, but it's also very costly to copy the source data consisting of external storage devices such as tape to the processing servers.

By adding a NAS solution to this scenario (see Figure 3-9), the data, as in our first example, is consolidated on the NAS devices and optimized for I/O access through the network. Although the sizes of the files accessed are much larger, the number of users are low and the resulting user requests for data can be handled by the existing network. Therefore, the current resources are effectively utilized within the required service levels of the sophisticated applications.

In this case, the user requests for data are similarly received by the server and redirected to the NAS devices on the existing network. The NAS completes the requested I/O, albeit one that is much larger in this case, achieving levels of 300 to 500MB per file, and is then redirected by the server back to the requesting user. However, an additional



In this case, geophysical files are stored on the NAS device because of their size. They are updated and archived through tape devices that can be multi-streamed to and from the NAS devices. This configuration enhances the storage and movement of data while separating the processing of simulations on the Supercomputer.

Figure 3-9. *A NAS for data size*

anomaly to this example is the sourcing of the data through tapes that are sent to the data center. The tapes are read from the server and copied directly to the NAS devices. Conversely, data from the NAS is archived back to tape as new simulations are developed. Given that multiple servers can reach the NAS devices, multiple input streams can be processing simultaneously in order to perform I/O to and from the offline media, which in this case are the tape units.

This solution, although different in perspective from our first example, provides the same benefits. The larger window of operation for the application itself demonstrates that appropriate storage resources balanced with data size and the number of users will stabilize the linear performance of the application. It also shows that NAS provides a more cost-effective solution by consolidating the data within an optimized set of devices (for instance, the NAS devices) and utilizing the existing resources of the data center which in this case were the server and associated file systems and network resources.

NAS Caveats

The caveats for NAS are twofold. The architecture employed by NAS products does not lend itself to applications that are dependent on sophisticated manipulation of the storage media—meaning any application that either has its own embedded file system to map and store data to the online storage directly, or that works with data management products, such as relational databases, which do the same. Secondly, applications that deal with highly volatile data do not perform well—in other words, data that is subject to constant and continuous updating by a large user base.

Writing to disk directly is a characteristic of many applications first developed within the UNIX operating environment. Relational databases commonly have this architecture. The problem manifests itself in performance as these systems have developed their own ways of manipulating data and in a sense have their own internal file system. Having these run under an additional file system penalizes the application by forcing it to do certain things twice.

However, any application that contains a database designed in a monolithic fashion will have performance problems when operating within NAS storage environments. For example, e-mail applications have embedded databases that are not scalable. The monolithic nature of their databases can therefore force all traffic into a single path or device creating a performance bottleneck and recovery problem if the storage device or path is encumbered in any way.

On the other hand, NAS provides the most cost-effective solution for applications that are file-oriented, contain read-only files, or which have large user bases. The architecture behind NAS within these application environments provides a scalable way for large numbers of users to obtain information quickly without encumbering the servers. This also provides an effective way to upgrade storage quickly through existing networking resources. This has proven itself within the ISP data centers where NAS storage devices are the solution of choice given their capability to handle large amounts of data through the storage of web pages, and users who require random read-only access.

We have discussed both the NAS architecture and issues in summary fashion. More detailed discussions of the components that make up the NAS solution, how they operate, and how to apply the NAS solution in data center settings will be discussed in Part V of this book.