# Network Storage

**Mladen Luksic, Ph.D.**

Fueled by the explosion of digital content, the pervasiveness of network applications, and the dramatic rise of Internet, data storage is rapidly moving to the center stage of computing technology. It is estimated that in the next few years, the demand for storage will grow at the rate of 100% per year. At the same time, customers and MIS management expect their data storage systems to deliver peak throughput performance, to safeguard data content from human and technical errors, to allow shared access in complex, multi-client, global, 24x7 environment, and to recover quickly after a disaster.

Such rigorous demands assume storage systems that offer un-interruptible upward scalability, reliability, and ease of management. Many individual enterprises cannot afford, or do not want, to maintain professional on-site storage system management resources. As a result, some are beginning to rely on simple-to-manage turnkey storage technology solutions, while others are resorting to arrangements even as complex as having third parties (often at the remote location) manage their storage needs. The latter is turning into a lucrative new industry--the "storage utility company". This polarization in storage management practice is resulting in the emergence of several new types of storage architectures, most notably the Network Attached Storage (NAS) Appliance, the Storage Area Network (SAN), and, although still behind the horizon, but most certainly coming, the Object Based Storage Device (OBSD).

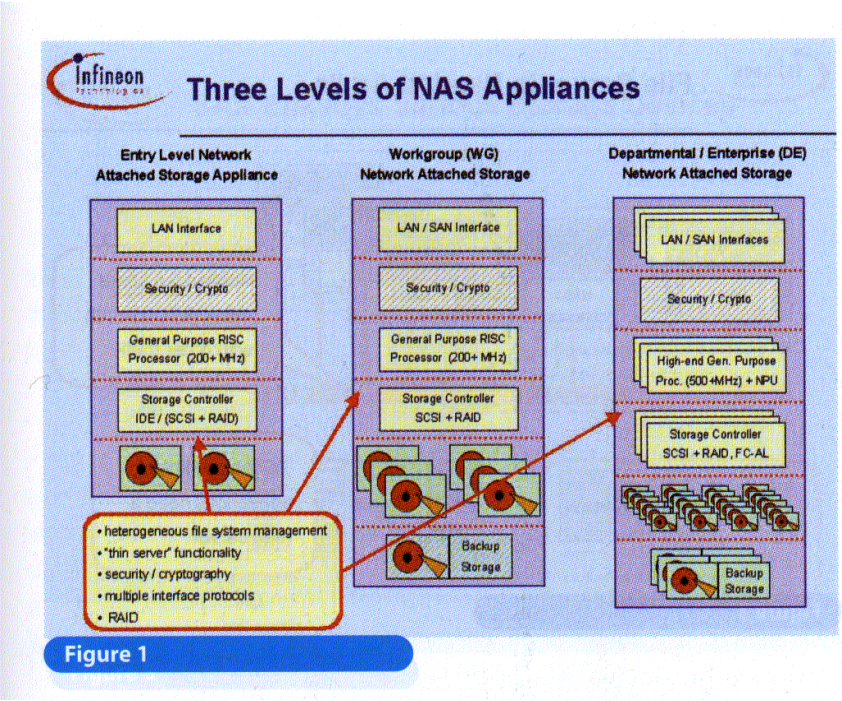
There is a lot written about NAS and SAN--what they are, how big a market they address, how they differ from each other and existing Server Attached Storage (SAS), etc. There are also a number of extensive discussions in the literature about general network storage concepts. So the purpose of this article is not to repeat the information readily available, but to try to clarify some of the most common confusing aspects of NAS, SAN, and OBSD. In particular, the article attempts to define the three storage architectures from the perspective of the role of the file system and the location of the file server, it describes how NAS and SAN differ and how they play together, and then it expands into the OBSD by explaining how the OBSD storage paradigm comes into being as a natural evolutionary step in the network storage technology. It briefly addresses the issue of security in OBSD, although the security and encryption, by the virtue of their paramount importance in network storage as well as in data communications in general, will be addressed in a separate follow-up article.

## Network Attached Storage (NAS) Appliance

The Network Attached Storage (NAS) Appliance is a storage system, which is attached to the network via network fabric (Ethernet); it has its own network identity (IP address), and contains a number of storage devices that are internally controlled by (proprietary) hardware and software. The hardware includes a general-purpose processor, an I/O engine, a network interface card (NIC), and the connection fabric. The processor runs (proprietary) software that "emulates" the file system of clients accessing the NAS appliance, i.e., the NAS appliance appears to an NT client as if it is running the NT file system, to a Unix client as if it is running a Unix file system, etc. This mechanism allows seamless file sharing in the heterogeneous network environment, and eliminates the need for subdividing and managing the network according to types of devices and applications running on the respective segments. In addition, NAS appliances can be configured and managed remotely, and easily added to an existing network without interrupting the work of other devices already connected to it. There are three levels of NAS appliances:

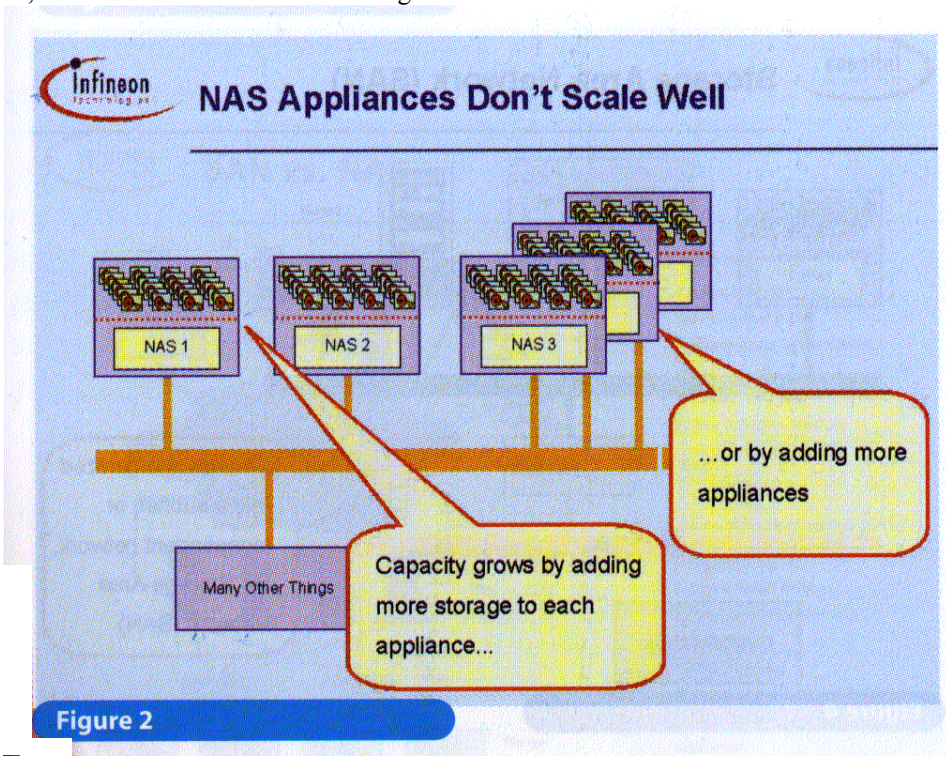
- Entry-level
- Workgroup
- Departmental/Enterprise (DE)

According to Dataquest, the Entry-level NAS appliance is an enclosed box, housing up to two (2) IDE disk drives supporting RAID 1 (mirroring), with no hot spares, which offers a simple "nontechie" GUI, limited bi-directional fail-over capabilities, and requires client-initiated back-up to an external tape. Thus, the Entry-level NAS appliance is not much more than a network attached disk drive. The Workgroup NAS appliance houses up to seven (most likely) SCSI drives supporting RAID levels 0,1, and possibly 5, has some hot spares, offers simpler GUI than traditional servers, can double up as a dedicated server for functions other than storage, and it allows network backup to an internal or external tape. The DE NAS appliance is a highly scalable high-end system which may have in access of 200 high performance SCSI or Fibre Channel drives supporting RAID levels 5 or 4, allows hot-swapping of drives, fans, and power supplies, offers clustering and central management of clusters, continuous data access with complex self-diagnostic and optimization algorithms, online backup, and disaster recovery, Figure 1.




## Adding More NAS Appliances

NAS appliances are very attractive from the cost of ownership and ease of management perspective s. For this reason they are expected to become a major player among network storage systems, and remain a complementary solution to other network storage architectures. However, NAS appliances have two significant shortcomings. First of all, they do not scale well in terms of storage capacity, i.e., if more NAS storage is needed, it can be done either by adding more storage devices to individual appliances, or by adding more appliances, Figure 2. Adding more storage devices to individual appliance is complicated, if not impossible, because a NAS appliance is a turnkey, in-the-box solution, not intended for hardware reconfiguration.




Adding more NAS appliances can saturate the network fabric, which already has to support the entire data traffic. However, a bigger problem is that the presence of more NAS appliances re-introduces the need for centralized file

system, bringing back the issue of file system centric storage. For example, imagine an enterprise application in which there are large, frequently updated relational databases of unpredictable dynamics and growth patterns (common in on-line trading, e-commerce, airline reservations, etc.). If a database has to be "smeared" across several NAS appliances, there has to be a mechanism that "tells" one NAS box what's in the others, and that correlates and synchronizes information that may be scattered in several boxes. And for this, a central file aggregation device (file server) is needed, Figure 3.

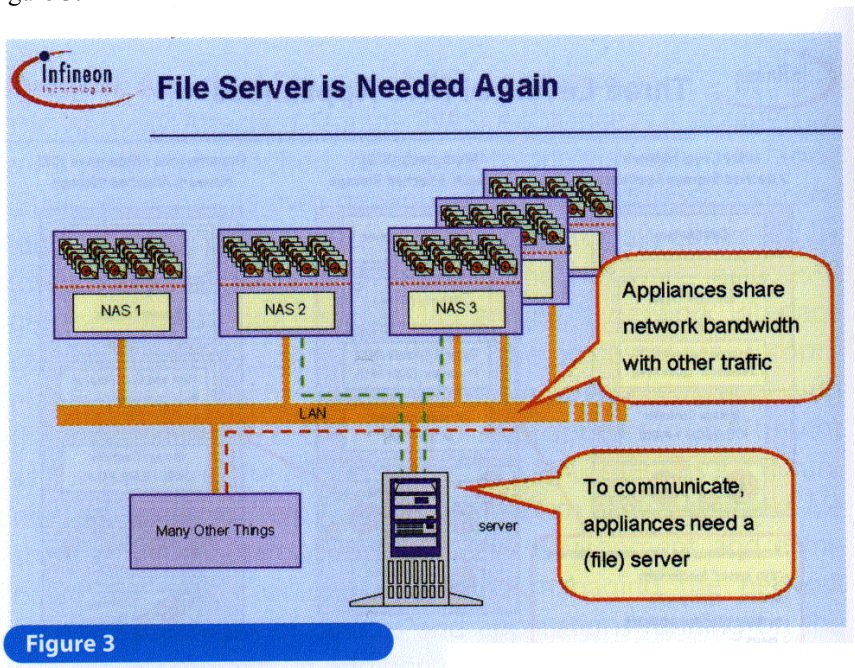


Figure 3

There are two basic approaches to resolve this. One is to separate completely the portion of the network structure that deals with data management and data movement into a dedicated (sub) network. Such a network is called the Storage Area Network (SAN), Figure 4. The other is to distribute the file system functions among storage devices themselves in a way that makes information that resides on these devices file (or object) oriented instead of block oriented. By doing this, the effort associated with aggregation of pieces of information (data blocks) residing on different storage devices is left at the level of devices, and the file server, now called the file manager, operates on file (or object) level. Storage systems based on this concept are often referred to as Object Based Storage Devices (OBSD), and are in very early stage. However, their commercial deployment is expected in the year 2002-2003.

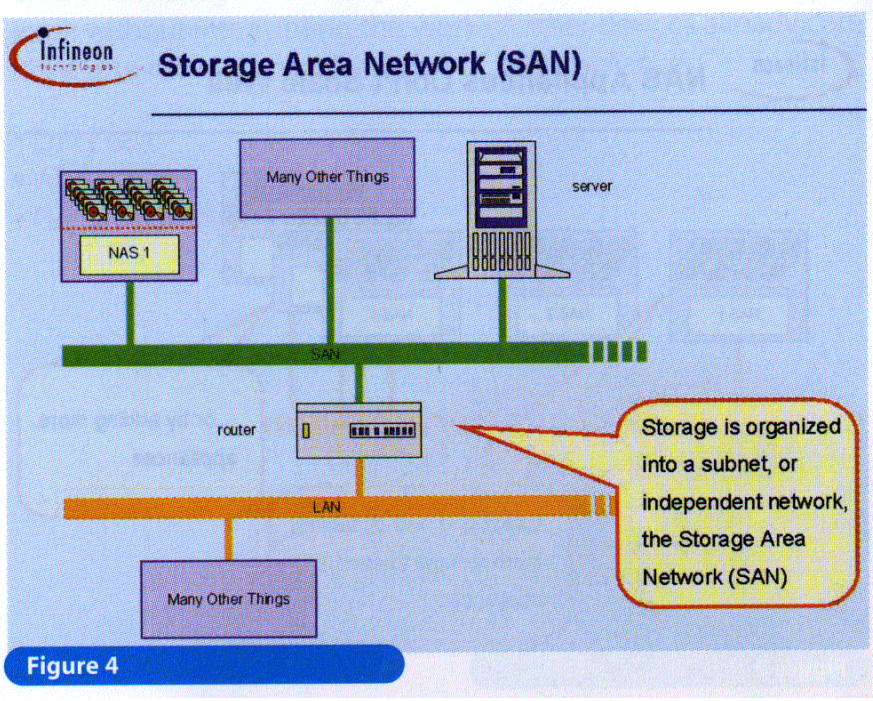


Figure 4

## Storage Area Network (SAN)

Storage Area Network is a dedicated network consisting of diverse storage devices, file servers, and fast net work fabric, which is connected to the rest of the network via dedicated (SAN) routers, switches, bridges, and hubs, much the same any two networks are connected. If the SAN is not built on the same fabric as the user network, it must also employ various interface protocol conversion devices. Today, the Fibre Channel (FC) appears to be the protocol of choice, and fiber, either copper or optical, is the underlying fabric of choice for storage area networks. However, there is a strong movement in industry to deploy storage area networks over TCP/IP protocol, particularly with the arrival of 10 Gbps Ethernet. But whatever the underlying physical fabric, it is important to keep in mind that SAN and NAS architectures are different things. The two should not be compared on the device level, or the performance level. A storage area network may incorporate many different types of storage devices, including NAS appliances, Figure 5.

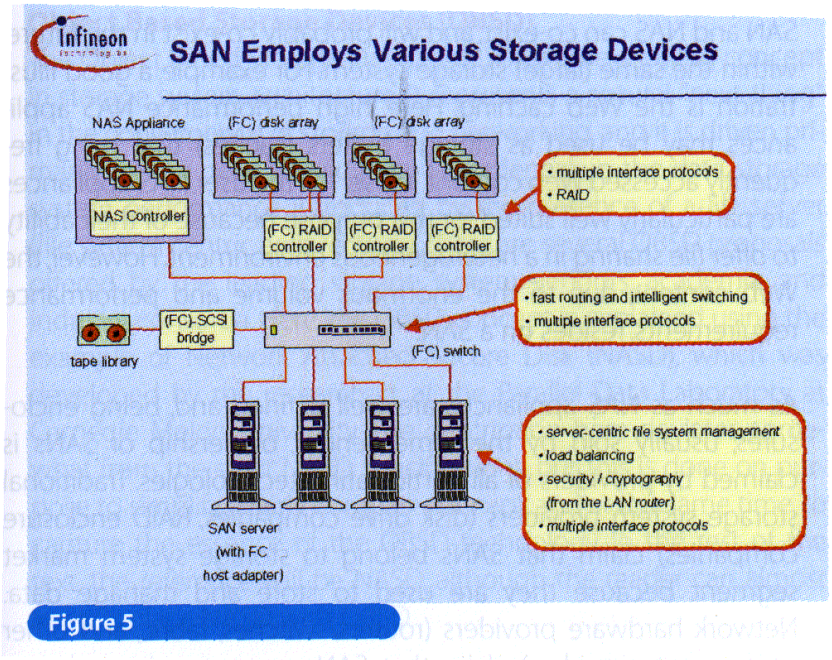


Figure 5

A NAS appliance may be a very high performance device, and inside the enclosure actually structured as SAN. But neither is NAS a SAN in the box, nor is SAN an opened NAS enclosure, as many sources tend to imply, Figure 6.

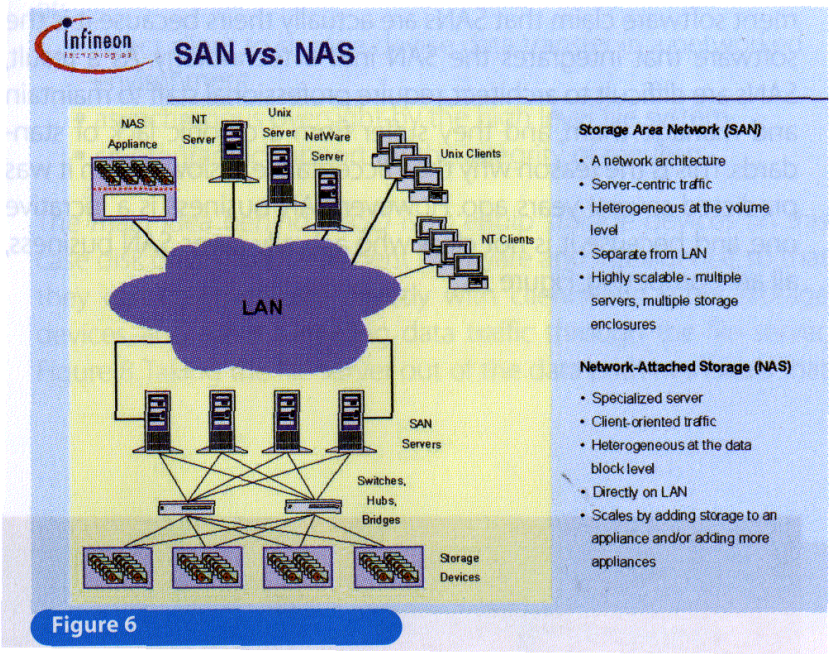


Figure 6

## SAN and NAS can Co-exist

SAN and NAS can co-exist, and will probably co-exist in the future, within the same (large) storage system. For example, a good illustration is the Web caching. Here, high performance NAS appliances may be used as storage buffers (caches) that bring frequently accessed Web content "closer" to the user. NAS appliances are particularly well suited for this purpose because of their ability to offer file sharing in a heterogeneous environment. However, the Web content, due to the enormous volume and performance requirements, resides on a SAN.

As much as NAS appliances are well defined and, being enclosures, usually sold by the same vendor, ownership of SANs is claimed by providers of all participating technologies. Traditional storage system providers (disk drive companies, RAID enclosure companies) claim that SANs belong to storage system market segment because they are used to store and manage data. Network hardware providers (routers/switches, fabric, and other component providers) claim that SANs are networks, and thus belong to networks. Providers of network and storage management software claim that SANs are actually theirs because it is the software that integrates the SAN into a technology. As a result, SANs are difficult to architect, require professional staff to maintain and manage them, and they suffer from a chronic lack of standards. This is the reason why their acceptance is slower than it was predicted several years ago. However, SAN business is a lucrative one, and because it is not clear who actually owns SAN business, all are competing, Figure 7.

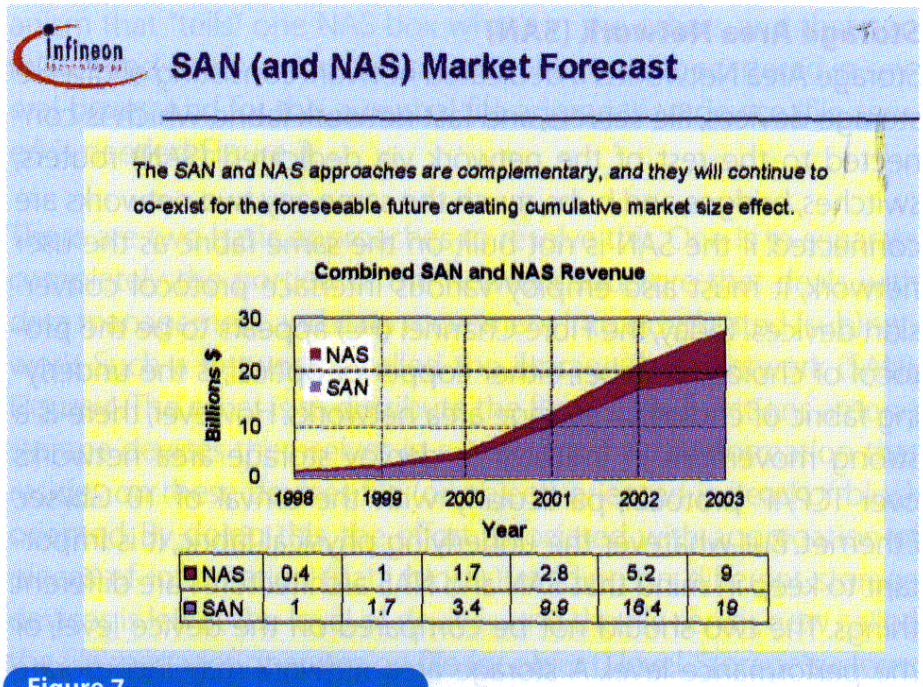



Figure 7

## Object Based Storage Devices (OBSD)

The Object Based Storage Device (OBSD) is a developing concept in storage system architecture. It represents a natural "next thing" in the convergence of storage and networking, and it is driven primarily by the need to eliminate a bottleneck in the data storage system performance introduced by the presence of a file server (file system centric architecture). There are several OBSD proposals offered by the leading storage system vendors, academia, and industry consortia. Here, the OBSD concept is explained using the example of Network Attached Secure Disk (NASD), which was developed by the researchers at the Parallel Data Laboratory at Carnegie Melon University. The description here deviates somewhat from the strict NASD specifications, but this is done on purpose to simplify a fairly complex concept, and at the same time, to capture the essence of the OBSD technology. In the rest of the text, the reference will be NASD, although the reader can almost freely replace NASD with OBSD. Loosely defined, a NASD (or OBSD) is any storage system capable of:

- Direct client to storage device data transfer in a networked environment
- Asynchronous oversight by the high-level file system
- Cryptographic support for the integrity of requests

The main idea behind NASD is to equip storage devices (in this case disk drives) with sufficient hardware and intelligence, so that they can communicate directly with clients and other storage devices, and avoid funneling data traffic through the file server, Figure 8. Taking the file server out of the data path mandates that the conventional concept of a file system executing on a dedicated file server workstation be abandoned in favor of a distributed file system, which retains some of its functions at the file server level, and exports others to disk drives. To understand how the partitioning of the file system functions is done in the NASD architecture, it helps to review the file system function in the conventional file server based architecture.

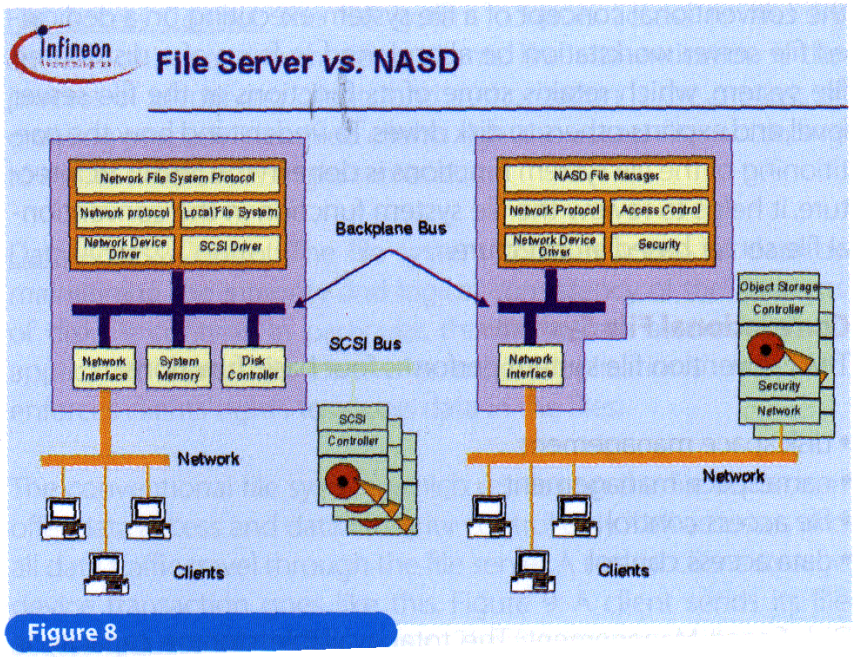


Figure 8

## Conventional File Systems

The convention file system performs four basic functions:

- Disk space management
- Namespace management
- File access control
- Data access control

***Disk Space Management*:** The total available storage capacity is partitioned into consecutively numbered logical blocks. The vector of all logical blocks is the Logical Block Address space, or the LBA space. The file system manages the LBA space by subdividing it dynamically in to file objects, or files. The file objects can vary in size, and can be created, extended, truncated, or completely deleted. The file system maintains a pool of unused logical blocks from which it allocates contiguous extents of space to file objects. Extents that become available because the file objects that occupied them have been truncated or deleted are returned to the pool. A file object may have more than one extent. The file system keeps track of what extents are associated with what file objects through tables of file extent locations. The tables are used to locate pieces of file objects and present each file object to the client as if it were a continuous byte stream, regardless of its actual location(s) in the LBA space.

***Namespace Management*:** File objects are assigned unique names. The set of all syntactically valid names is the namespace. The namespace is arranged into a tree structure of directories that allow organizing and locating file objects. The file system manages the namespace by maintaining consistent correlation between object file names and corresponding data extent locations.

***File Access Control*:** The file system controls access to files by authenticating and authorizing clients. The authentication is the process of identifying clients, and authorization is the process of Granting properly identified clients rights to access files according to established access right semantics.

***Data Access Control*:** The file system controls access to data by maintaining the integrity and logical consistency of the contents of disks it controls. In particular, the file system assures that an application does not overwrite other application's data, and enforces clients' rights to access data in the files.

The conventional file system, which resides on the file server, can offer data access and data integrity control only if all accesses and all data traffic travel through the file server. A typical client-storage device transaction goes like this, Figure 9: A client sends its file access request to the file system. The file system verifies client's access rights to the file in question, it establishes logical linkage between the client and the file, and, using its managed extent tables, converts the request for the file into a request for blocks of disk data. The storage device blindly honors any valid instruction that comes from the file system. The communication between the file server and the storage device(s) is invisible and inaccessible to the outside world.

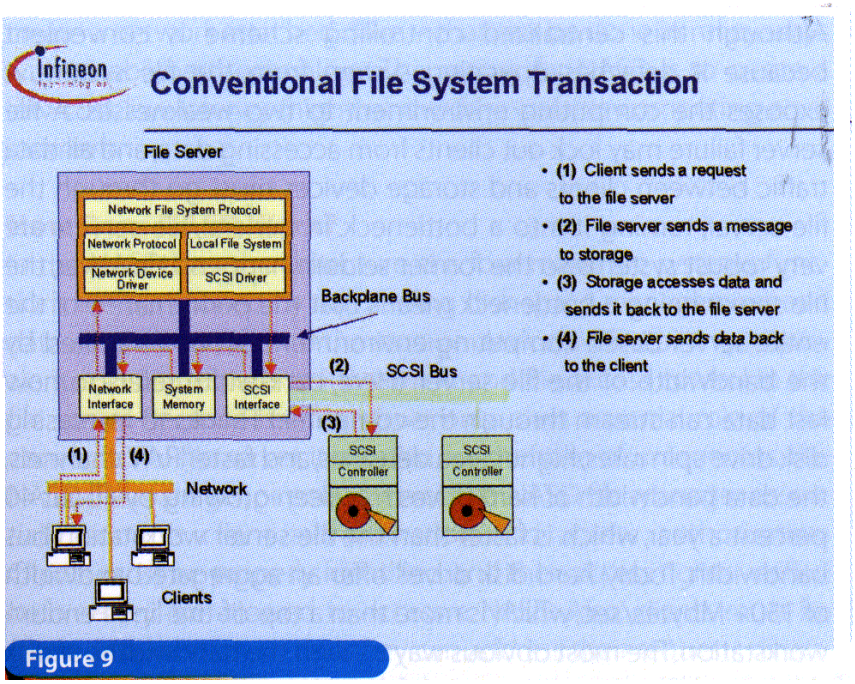


Figure 9

Although this centralized controlling scheme is convenient because it detaches operating system from the file system, it exposes the computing environment to two weaknesses: A file server failure may lock out clients from accessing data, and all data traffic between clients and storage devices must go through the file server, turning it into a bottleneck. In general, file servers are very robust systems so the former seldom happens. However, the file server being a bottleneck means that the performance of the entire server based computing environment becomes limited by the bandwidth of the file server. (Here, bandwidth refers to how fast data can stream through the computer.) Thanks to increasing disk drive spin rates, higher area densities, and faster R/W channels, the data bandwidth of hard drives has been growing by about 40 percent a year, which is faster than the file server workstation bus bandwidth. Today, hard disk drives offer an aggregate bandwidth of 150+ Mbytes/sec, which is more than a top-of-the-line Pentium workstation. The most obvious way to align the bandwidths of disk drives and file servers is to get faster, specialized buses on the servers. This approach is expensive, and as the number of disks in the environment grows, it ultimately reaches the point of diminishing return. The other approach is to remove (partially) the file server from the data path. This approach is the main premise behind NASD technology. So the reason for NASD is not just to create a network attached drive—that can be done in many ways with established technology. The reason for NASD is to create a low-cost, low-latency (high-bandwidth), scalable computing environment.

## How Does NASD Work?

In the NASD based architecture, the four major file systems functions are partitioned by transferring disk space management and data access control onto the storage device, and retaining (equivalents of) namespace management and file access control in the file server. Because the file server only plays a partial role of the conventional file server, it is called, for the sake of uniqueness, the NASD file manager, or policy server. The NASD file manager essentially controls client authentication and access authorization. Clients must be properly identified, and their access rights verified before they are allowed to access a NASD. Also, clients must be given data location, so that

they can address their requests directly to the appropriate NASD. At the same time, the clients' rights must be made known to NASD, so that they can honor authorized requests.

***Disk Space Management***: A NASD manages its own LBA space. Instead of a single LBA space, a NASD reads and writes blocks of data in extents of storage. A NASD client cannot address logical blocks directly. Extents are not permanent. The NASD creates them, extends them, truncates them, and deletes them in response to client commands.

***Data Access Control***: A NASD controls access to data. Unlike an ordinary disk drive, which indiscriminately executes any valid R/W request, a NASD only permits reading and writing within extent boundaries. For example, a NASD will reject an application request to write block 103 of a 100-block extent. Therefore, NASD clients cannot overwrite data in other clients' extents. Moreover, a NASD only performs properly authorized operations for authenticated clients.

The actual process of client-NASD device transaction goes like this, Figure 10: A client sends its first file access request to the NASD file manager, which returns an encrypted ticket, or token, called a capability.

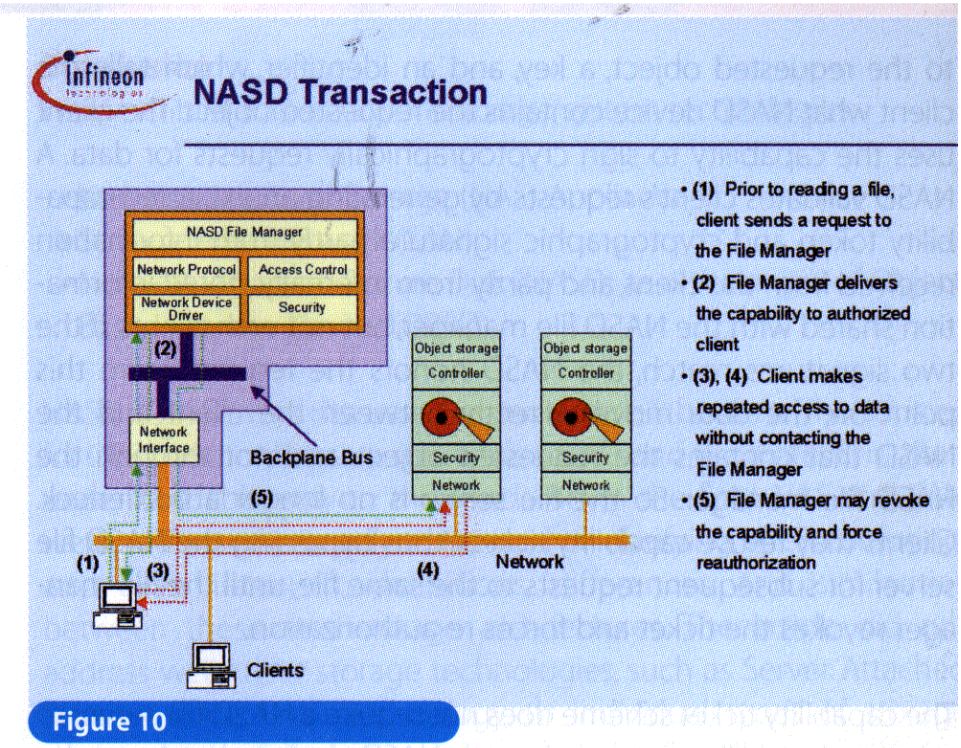


Figure 10

The capability contains the client's access rights to the requested object, a key, and an identifier, which tells the client what NASD device, contains the requested object. The client uses the capability to sign cryptographically requests for data. A NASD validates client's requests by generating an expected capability token and cryptographic signature partly from information received from the client, and partly from internally stored information shared with the NASD file manager, but not with clients. If the two signatures match, the NASD honors the request. From this point on, the data moves directly between the client and the NASD that contains the requested object, and not through the NASD file manager. So the file server is no longer a bottleneck. Clients may re-use capability tickets, thus bypassing the NASD file server for subsequent requests to the same file, until the file manager revokes the ticket and forces reauthorization.

The capability ticket scheme does not require a NASD file server to communicate client permissions to NASD devices, but instead, a NASD device honors all requests that present a valid capability ticket, without having to know client's identity. This protocol minimizes the network traffic required for a client to gain access to a file, and the amount of information that a NASD device must maintain. The NASD file management does not require a lot of computational work, so the NASD file manager computer can be an inexpensive PC. Also, as there is no file server in the data path once a client's right to access a file has been established, the NASD file manager does not require high-performance internal data bus and extensive device connectivity hardware that is common in today's file

servers and causes cost to increase faster than capacity. As the NASD system grows, the network infrastructure supporting it can grow as well, using standard network techniques such as segment isolation, routing, etc., to manage network's bandwidth for optimal performance. The NASD approach provides scalable architecture for high-per-forming I/O subsystems capable of supporting many clients and storage devices. The (upward) scalability is possible because the resource growth is inherent in the NASD concept, i.e., as more storage capacity is added to a NASD system, more processing power is added with it. Although NASD is primarily targeted to Enterprise computing, and its embodiment (best) suited for LAN environment, the concept works with any file system that can be partitioned into space management and data access control tasks, The piece that is "local", and thus given to NASD, and authentication and authorization tasks, the piece that is "distributed", and thus left at the NASD file manager level. Therefore, nothing prevents a NASD based implementation in a single RAID, where disk drives are replaced with NASD drives, SCSI interface with fibre channel, or some other fast interface, and file server with the NASD file manager, or extending the NASD to WAN environment.

## Security in a NASD System

If the NASD architecture is implemented in the network environment, all NASD devices must support networking, which is fairly standard in terms of additional hardware and additional performance bandwidth required to support specialized software. However, having storage devices attached directly on the network, exposes interfaces previously hidden behind the file server, so NASD devices must be responsible for their own security. In secure communications, security means two things: integrity and privacy.

A system provides integrity if it protects message exchange from tampering by an adversary. In most secure communication systems, two general methods are used to provide integrity: message digest (MD) with public key signature, and message authentication c ode (MAC). In both cases, binding the data to a cryptographic key protects integrity.

A system provides privacy if an observer of message traffic cannot learn the contents of the messages. Privacy is guaranteed through data encryption. Most common methods in data encryption are public key cryptography, which is computationally very intensive, and symmetric key cryptography, which is less intensive, but requires more complicated key management. Frequently, the synergy of the two methods is used with key distribution done via public key cryptography, and bulk data encryption done with symmetric key cryptography. Unfortunately, cryptography, if done exclusively by software, is computationally extremely intensive, and it can seriously impact the overall performance of the computing environment, erasing all the benefits introduced by NASD architecture. Hardware sup-port can dramatically improve performance, especially in cryptographic primitive s, but dedicated cryptographic hardware is expensive.

The NASD approach integrates security, both integrity and privacy, into the NASD device through a combination of three techniques: hierarchical message authentication code that enables pre-computation of integrity information, incremental verification of data integrity over large transfers that minimizes buffer requirements, and integration of the cryptographic hardware with a data path of a device. Together, the three techniques offer near non-secure communication performance at the cost increment that is accept-able to price-concerned disk drive manufacturers. The details of security in the NASD device are too complex for a short article, and will be discusses separately.

## Conclusion

We described three network storage system technologies:
• Network Attached Storage (NAS) Appliance
• Storage Area Network (SAN)
• Network Attached Secure Disk (NASD) as an example of Object

Based Storage Device (OBSD) NASD (OBSD) was discussed in more detail, because the sources for OBSD technology are still rather scarce, and the material is often understandable only to the specialists in the field. NAS and SAN were addressed primarily to focused on clarifying often confusing, and even incorrect, definitions, differences, and similarities between these two storage architectures. The article did not address well know storage technologies, such as Server Attached Storage (SAS), RAID technology, or individual types of storage devices. In the follow-up articles, more details will be provided in terms of the current status of NAS and SAN industries (key players, markets, market size and trends, revenue projections, etc.)