NAS and SAN Scaling Together (NASD Approach)

Luís Manuel Oliveira Soares

Departamento de Informática, Universidade do Minho
4710-057 Braga, Portugal
los@lsd.di.uminho.pt

Abstract. This communication exhibits the state of the art of distributed storage architectures based on a hybrid approach of NAS[1] (file oriented) and SAN[2] (block oriented) approaches, focusing an emergent solution, the NASD[3] architecture. It also brings out the benefits when combining a conceptual NASD device with smart devices such as Active Disks.

1. Introduction

The goal of network-shared storage is to enable efficient access to heterogeneous data contents, spread over a network. The actual Internet dependence is a fact as well as the growing necessity to reach information that is not close to the end user. This need compels storage manufacturers and software designers toward scientific research, in order to provide reliable, robust and scalable solutions.

Storage bandwidth becomes critical nowadays mostly due to the richer data types, such as video or data-intensive applications that travel through the Internet. A recent study [1] claims that explosive growth of information is only beginning: humankind will generate more original information over the next three years than was created in the previous 300,000 years combined. These throughput requirements for network-shared contents suggest efficient storage capability as well as easy retrieval. A question may arise when bringing up the network storage matter: why spread data over a network when it could be kept nearby. Networked storage reduces wasted capacity, the time to deploy new storage, and backup inconveniences. It also simplifies storage management, increases data availability, and enables the sharing of data among clients [2]. But there are still some details that must not be left out when handling network-shared storage. The system entropy is much higher, i.e., in order to make it all work, more machines have to function correctly, data is more vulnerable to privacy and integrity attacks and processing of network protocols is needed, which, by itself, is more expensive than a local hardware device access.

Two distinct approaches come forth when mentioning network storage issues. The NAS solution, which is mostly filesystem oriented and SAN, frequently declared as being a block-oriented solution. With these two distinct philosophies, potential problems may be solved, but still leave some situations uncovered. Along these facts, technology evolution and its side effects must be dealt with. It is not hard to perceive that the continuous development of computers has enabled and drastically increased the scale on which files are managed on behalf of multiple users and shared across networks.

This communication brings out a SAN/NAS hybrid solution that deals with server bottlenecks, providing a scalable solution for network storage problems. In Section 2, some

[1] Network Address Storage
[2] Storage Area Network
[3] Network Attached Secure Disks

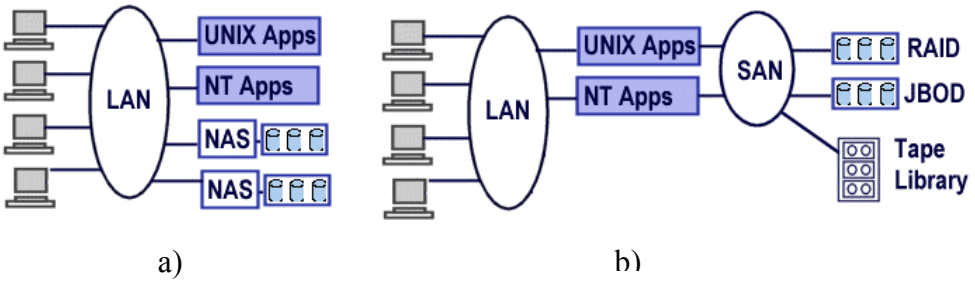

Fig. 1 : (a) Typical NAS Environment (b) Typical SAN Environment – *Courtesy of Zerowait Corp.*

background on NAS and SAN is presented. In Section 3, the two paradigms are drawn near to show the benefits when combining them. In Section 4, the NASD approach is presented. Section 5, presents a possible improvement on the NASD architecture, taking advantage of *Active Disks* [3] features. Finally, Section 6 concludes the paper.

2. Background

In 1979, the Shugart Associates, felt that people would be interested in a way to easily use and share disk devices. This feeling originated the definition of the SASI[4], which is the predecessor of SCSI[5]. This happening was shortly followed by NFS[6] creation when in the early eighties, a team of engineers at Sun Microsystems developed it. Right about the same time, IBM developed NetBIOS that became the foundation of the SMB[7] protocol which lead to the CIFS[8], the predominant method of sharing files in a Windows environment. Due to this historical overview, it is noticed that neither SAN nor NAS are revolutionary concepts; in fact, SANs and NAS may be considered to be the next step in the evolutionary path of SCSI and NFS/CIFS, respectively.

2.1 Storage Area Network

A SAN, is a set of two or more storage devices communicating through a serial SCSI protocol, such as Fibre Channel or iSCSI [4]. In a certain way, the concept *System Area Networks* refers to the storage device and the network hardware to which the storage is attached. Note that a LAN, which is used only for the purpose of handling storage traffic, is not a SAN. However, if that network deals with the storage traffic using a protocol such as iSCSI, then it may be considered a SAN.

These systems are often used for high-performance architectures that demand low latency and direct access to any small group of data units. Therefore, scalability for storage resources is required. On the other hand, the ability to supply file sharing for several hosts is not accomplished. This handicap originates in the incapacity of SAN systems to enable synchronized access to data. It is true that SANs allows multiple clients to access same

[4] Shugart Associates System Interface
[5] Small Computer System Interface
[6] Network File System
[7] Server Message Block
[8] Common Internet Filesystem

Table 1: Differences Between NAS and SAN

	SAN	**NAS**
Protocol	Serial SCSI-3	NFS/CIFS
Shares	Raw disk and tape drives	Filesystems

storage device directly, but its interface does not provide any mechanism to ensure synchronization in concurrent accesses.

2.2 Network Attached Storage

NAS originated in CIFS and NFS. These two file sharing protocols have some limitations that motivated some new thoughts about the network file sharing concept. In fact, these two protocols became quite popular and are generally accepted by the industry, despite their unfitness to communicate with each other. Another disadvantage is that both implementations behave as another application executing in the machine that leads to CPU and network resource consumption. To solve these problems, NAS was invented. The typical idea about Network Attached Storage is nothing more than a server in a stand alone box with hot-swappable RAID[9] arrays. This server continues to serve files using NFS and CIFS, but with customized versions, which make file system sharing easier, thus improving performance. Nevertheless, as it will be shown later, it is common for a NAS filer to be comprised of a filer head with SAN-attached storage behind it. Actually, many NAS systems internally interface with non-volatile magnetic media through a SAN-like interface. This situation is depicted in Figure 2. As it may be noticed from Table 1, these two approaches are quite different, but they may be linked together to produce an hybrid solution.

3. Combining NAS and SAN

The combination of NAS and SANs may overcome the flaws of each other. They both have several blemishes. On the NAS side, the server itself is the system bottleneck. These hinder strings up, because a single server collects data from the storage device/network, acting as a storage controller, and forwards it to the client host/network. In addition to this *store-and-forward* behaviour, the server also handles concurrency control and metadata consistency. Considering these facts, one notices that in high-performance distributed filesystems, there is an evident cost associated with the overhead produced by the filer, which serves files and handles all the filesystem semantics. This flaw becomes clearer when the server, apart from managing the tasks mentioned above, also bridges traffic between a storage network and a client network.

SANs present approximately the same problem, because a controller is needed to operate accesses to devices. To amortize the cost of the controller, it is usual to associate multiple disks per *storage-controller*, which degrades the controller performance. Another SANs incapacity is the inability to present file sharing support, due to the block oriented interface and the lack of explicit concurrency control mechanism. Deciding what network messages to trust is also a problem. Storage-system interconnects, including most of SANs

[9] Redundant Array of Inexpensive Disks

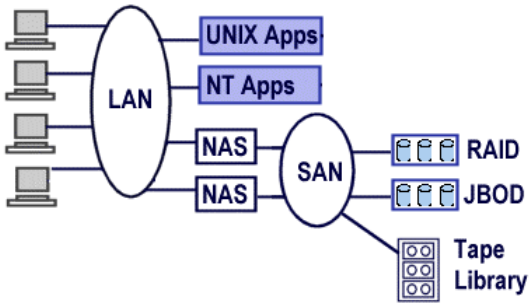

Fig. 2: NAS and SAN collaborating - *Courtesy of Zerowait Corp.*

technology, were originally designed as an extension of the internal buses of their hosts thence security provisions do not exist or are very restricted [2].

When combining both these technologies, one virtue may hopefully overcome the other's flaws, and vice versa. An easily data sharing on the NAS side combined with the scalability presented by the SANs is a very attractive solution. It is clear that not all the drawbacks may be solved combining these two approaches, but some are indeed. Actually, a strategy for avoiding controller bottlenecks is to separate control and datapaths, roughly like DMA. Another way is to maintain a load-balanced cluster of controllers, which is not a cost effective solution.

To solve some of the security problems mentioned above, NAS take advantage of Operating System access restriction techniques while SANs hold to VPNs[10] using IPSec[11] protocols, firewalls and Fibre Channel zoning, restricting trust only to participating systems.

4. Network Attached Secure Disks

NASD architecture enables cost-effective bandwidth scaling. Server bandwidth bottlenecks are eliminated because storage devices are modified granting them the ability to directly transfer data to clients. They are also able to share file server or database functionality with clients and servers.

4.1 NASD Based Storage Architecture

NASD based architecture essentially minimizes server-based data movement, embedding disk management functions into the device. Taking advantage of persistent capabilities granted by file managers, clients may access storage objects directly. If a client requests object access permission to the file manager, and it is granted, the next time this client needs to access the same object, it does not need to request it again, unless it was revoked. Therefore, all data and most control information, travels once across the network and directly between entities, avoiding expensive mechanism of store-and-forward.

Typically, file managers grant capabilities over object access, handle concurrency control, global naming and cache coherency. Note that a centralized controller is still required, but it is relieved from several management functions that can now be conducted at the device level. The storage device does no aim to replace the file server entirely, but it

[10] Virtual Private Network
[11] Short for IP Security, a set of protocols developed by the IETF to support secure exchange of packets at the IP layer

acts like a helper. Since high-level distributed filesystem functionality varies significantly, filesystem policies are defined at the file manager, leaving storage primitives to be efficiently operated at the NASD device itself. This architecture is instantiated in Figure 3.

Another point of interest is the NASD interface, which is an object-based one. This means that data layout management may be handled at the device, because, the device needs to know the storage metadata to define the data objects. Hence, the data units exported by the device are variable length objects instead of raw disk blocks. Consequently, the client does not access metadata but pre-defined data objects, thus it does not need to query the file manager on each access. The alternative, exporting storage metadata to clients and enabling client access to arbitrary disk blocks, ``is worse than insecure, it is accident-prone and puts the entire storage system at risk" [5].

4.2 Security

NASD security is based on cryptographic capabilities. Clients obtain the capabilities from the file manager through a private and secure protocol external to NASD. This process ensures the integrity of the requests, by filtering possible attacks and accidents. Cryptographic techniques are intuitively expensive and may be unacceptable in a disk. Software implementations operating at disk rates are not available with the computational resources we expect on a disk, but schemes based on multiple DES[12] functions blocks in hardware can be implemented in a few tens of thousand of gates and operate faster than disk rates [6].

4.3 NASD Properties

Generally, NASD exhibits four properties: *Direct Transfer*, *Asynchronous Oversight*, *Cryptographic Integrity*, *Object-based Interface* [6]. The first one asserts that data is transferred directly between drive and client without indirection or store-and-forward through a file server machine. The second one claims that client's have the ability to perform most operations without synchronous appeal to the file manager. Frequently consulted but infrequently changed policy decisions, such as authorization decisions, should be encoded into capabilities by the file manager and subsequently enforced by drives. Cryptographic Integrity affirms that drives attached to the network become vulnerable to direct attacks from adversaries, hence it is necessary to apply cryptographic techniques to defend against potential attacks. The last one allows drive entities to possess direct knowledge of the relationship between disk blocks and to minimize security overhead because, drives export variable length "objects" instead of fixed-size blocks. This also improves opportunities for storage self-management by extending into a disk an understanding of the relationships between blocks on the disk.

4. Active Disks and Network Attached Secure Disks

Current disk drives integrate all the components of a simple computer, i.e., a microprocessor, memory, and a communications subsystem with a control unit for signal processing control. Due to the improvements in the silicon circuit production, that enables production of integrated circuits using .25 or even .18 microns technology. The space available inside the drive circuit chip may now be used to incorporate a 200MHz embedded microprocessor.

[12] Data Encryption Standard Algorithm

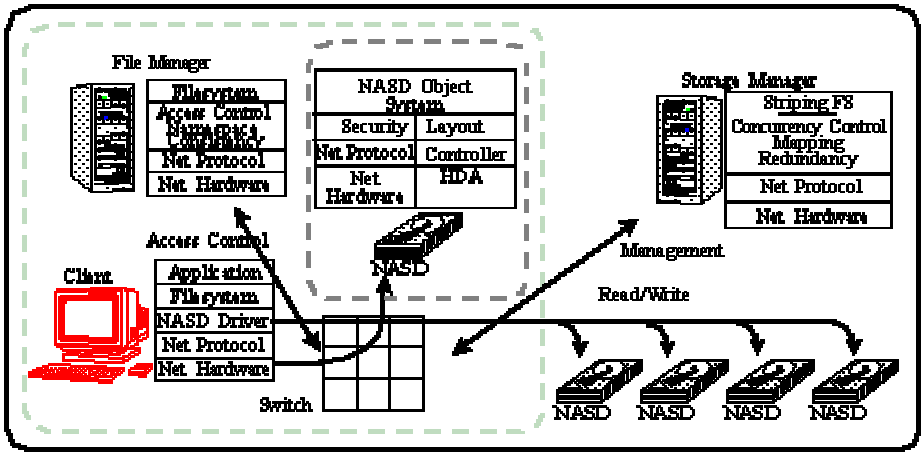

Fig. 3: Scalable bandwidth NASD system - *Courtesy of Carnegie Mellon University*

This is a powerful execution environment directly at the drive that enables new fields of research for distributed processing. Code can be executed near the data providing devices with the ability to examine data before it is released into the network. Not only the processing power is increased, but also the entity independence regarding the file manager may be strengthened.

This way all the tasks regarding NASD procedures, like filesystem management, may be easily executed at the device. The extra cycles may then be used to execute application-level code. Facing these new drive capabilities, a distributed environment concept focusing distributed intelligent storage devices arises, since these devices cease to be thought of dummy storage drives that export data blocks to be understood as an entity capable of code execution and hence data processing.

Distributing application code throughout devices improves I/O performance and challenges the actual tendency that is to bring data to the application code. This new trend presents an innovate concept: sending application code to processing capable storage drives so that code executes near data, hoping that network traffic is reduced due to unnecessary data movement. Typically, this data movement was necessary because some part of the application code need it to evaluate some expression, without any output directed to the user, i.e., it was needed because of an in-between algorithm step. Note that this approach extends cluster based processing, because it does not take advantage of code processing only, it capitalizes data closeness so that code executes near data without generating extra network traffic.

Apart from incorporating a processor capable of filesystem semantics and data processing, there will still be available free area where it may fit on-chip DRAM functionalities or cryptographic support. This also favours the NASD approach, because it easies the cryptographic processing which is a mandatory NASD property.

6. Conclusions

Decisions regarding storage systems are becoming very important due to the role that modern storage devices play in the corporate data centres. Investment in such devices is often very expensive which makes these decisions to be thoughtful ones.

Two approaches to distributed storage systems are very popular today: NAS and SANs, that differ from each other mostly in the technology used. The first presents generic solutions that make use of *cheap* Ethernet exporting data as files while the second is based on *expensive* Fibre Channel and exports data as blocks instead of files. This second approach benefits from the SCSI-like interface that is usually used in SANs, which means that is very scalable. Nevertheless, these differences tend to blur and may soon not be recognizable. Moreover, NAS servers stand in an Ethernet network accepting client requests and fetch data through a SAN like private storage network. Although the combination of NAS and SANs brings considerable improvements to storage network, aspects such security, availability and data sharing still lack some efficiency.

To reduce these inefficiencies, a NASD architecture has been proposed. Network attached secure devices are defined by supporting: direct device-to-client transfers, secure interfaces (possibly through cryptographic techniques), asynchronous access to storage (accessing only once to the file manager) and object-based interface exporting variable length objects instead of fixed-size disk blocks.

Finally, NASD devices may take advantage of technology evolution due to the new generation disk drives with improved ASIC[13] chips. These new smart drives may embed in their integrated circuit all the actual functionalities plus an ARM or similar microprocessor, increasing the drive processing capabilities. Therefore, on-drive processing becomes available enabling the device-to-client communication to be handled as well as processing the cryptographic algorithms needed by NASD devices.

References

[1] Winter Corporation. Convergence of NAS and SAN With Highroad. Technical Report, Winter Corporation, (2001)

[2] Garth A. Gibson and Rodney Van Meter: Network Attached Storage Architecture. Communications of the ACM, (2000) 37-45

[3] Garth A. Gibson Erik Riedel, Christos Faloutsos and David Nagle: Active Disks for Large-Scale Data Processing. IEEE Computer, (2001) 68--74

[4] W. Curtis Preston. Using SANs and NAS O'Reilly, 1st edn, (2002)

[5] Garth A. Gibson, David F. Nagle, William Courtright II, Nat Lanza, Paul Mazaitis, Marc Unangst, and Jim Zelenka: NASD Scalable Storage Systems. Proc. USENIX, Linux Workshop, Monterey, CA, (1999)

[6] Garth A. Gibson, David F. Nagle, Khalil Amiri, Jeff Butler, Fay W. Chang, Howard Gobioff, Charles Hardin, Erik Riedel, David Rochberg, and Jim Zelenka: A Cost-Effective, High-Bandwidth Storage Architecture. Proc. 8th Int. Conf. on Architectural

[13] Application-Specific Integrated Circuit

Support for Programming Languages and Operating Systems, ACM Press (1998) 92-103.

[7] Garth A. Gibson, David F. Nagle, Khalil Amiri, Fay W. Chang, Eugene M. Feinberg, Howard Gobioff, Chen Lee, Berend Ozceri, Erik Riedel, David Rochberg, and Jim Zelenka: File Server Scaling With Network-Attached Secure Disks. Proc. ACM SIGMETRICS Int. Conf. on Measurement and Modelling of Computer Systems, ACM Press, (1997) 272--284

[8] Hyeran Lim Vikram Kapoor, Chirag Wighe and David H.-C. Du: Active Disk File System: A Distributed, Scalable File System. Proc. 8th IEEE Symposium on Mass Storage Systems, IEEE Press, (2001)