# Network-Attached Commodity Storage for Scalable Bandwidth

A talk proposal for HOT INTERCONNECTS Symposium V-1997

Garth A. Gibson and David F. Nagle

Carnegie Mellon University, Pittsburgh, PA 15213-3891
{garth.gibson,david.nagle}@cmu.edu http://www.pdl.cs.cmu.edu/NASD

## Introduction

Storage systems represent a vital market that is growing faster than the personal computer market. No longer able to rely on the captive markets of single-vendor computing systems, the need to compete in the price-sensitive open market of personal computers has driven media areal growth rates from 25 percent per year to 60 percent per year. This increase in density growth rate has been accompanied by a 35-50 percent per year decrease in the cost per byte of storage. This trend is certainly not the last gasp of an obsolete technology. Storage hardware sales in 1995 topped $40 billion, including more than 60,000 terabytes of hard disk storage. In recent years, the amount of storage sold has been almost doubling each year; in the near future it is expected to sustain an annual growth of at least 60 percent. Secondary storage has a healthy place in future computer systems.

While many of these storage products are being directly attached to personal and home computers, 65 percent of the disk array products are in local area network file servers and this fraction is expected to rise to 75 percent over the next few years. This centralization of storage resources enables effective sharing, better administrative control and less redundancy. However, it increases the dependence on network and file server performance. With the emergence of high-performance cluster server systems based on commodity personal computers and scalable network switching, much higher demands on storage performance are anticipated. Specifically, storage performance must cost-effectively scale with customer investments in client processors, network links and storage capacity.

With today's distributed file system technology, all storage bytes are copied through file server machines between peripheral buses (typically SCSI) and client LANs. In essence these file server machines are acting as application-level inter-network routers, converting namespaces (disk block versus file range) and protocol layers (SCSI versus RPC/UDP/IP). This is a critical limitation for cost-effective scalable storage because it forces server resources to grow as rapidly as client processors and storage capacity to avoid serious bandwidth and latency bottlenecks.

Moreover, the sustained bandwidth of storage devices is rapidly outstripping installed interconnection technologies and rendering inexpensive store-and-forward servers impractical. Specifically, the rapid improvements in linear bit density and magnetic disk rotation rate is driving data rate up at 40 percent per year, insuring 25-40 MB/s sustained disk bandwidth by the end of the decade. With this much bandwidth possible from each commodity drive, the bandwidth possible from the number of drives it takes to offset the overhead cost of a low-cost workstation server is likely to be substantially more than the workstation can store-and-forward through its system bus and memory, forcing the use of a higher-cost workstation and even more drives.

Storage devices as small a disk drives, however, are already effective network data transfer engines. For example, Seagate's Fibre Channel Baracuda drives burst packetized SCSI at 1 GHz. Moreover, through careful hardware support for interlayer processing, the marginal cost of these network-attached disk drives is expected to be similar to that of high-end drive interfaces, such as differential SCSI [Anderson95].

It is our contention that cost-effective scalable storage performance depends on eliminating the file server's role as a inter-network router. Instead, we advocate exploiting the drive's ability to inject packets directly into the clients' network at high-bandwidth [VanMeter96]. With effective network-attached storage, striping of data over multiple devices effectively scales storage bandwidth [Patterson88, Hartman93].

## Managing Network-Attached Storage

The simplest network-attached storage architecture is the shared disk model in which the distributed file server is coded as a multithreaded application and every client runs one of the threads. While in some cluster systems there is

Point of contact: Assoc. Prof. Garth Gibson, School of Computer Science, Carnegie Mellon Univ., 5000 Forbes Ave., Pittsburgh PA 15213-3891, 412-268-5890, 412-268-3010 (FAX), garth.gibson@cs.cmu.edu.

This work was supported in part by DARPA contract N00174-96-0002 and by Data General, Symbios Logic, Hewlett-Packard, and Compaq. The US government has certain rights in this material. This work was performed in part according to the National Storage Industry Consortium (NSIC) NASD project agreement. The views contained in this document are those of the authors and should not be interpreted as representing the policies, either expressed or implied, of any supporting agency.

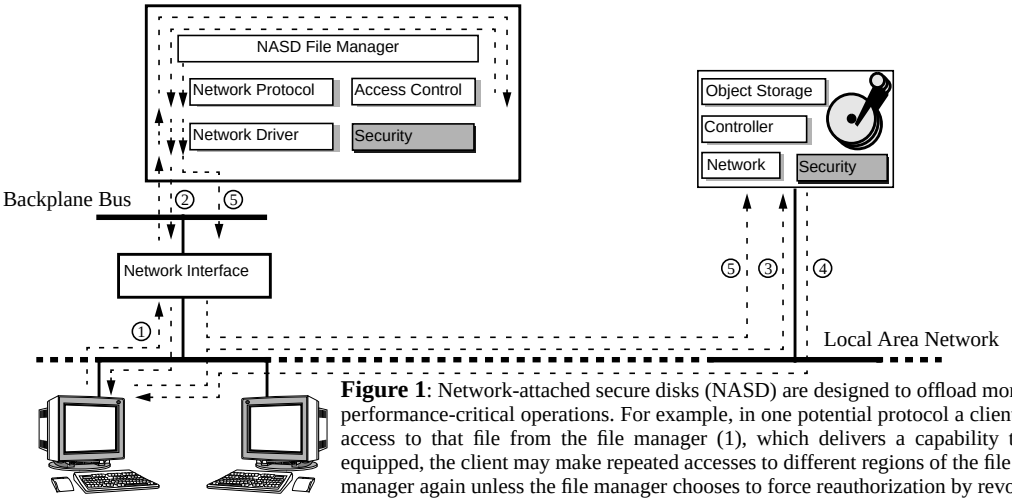


**Figure 1**: Network-attached secure disks (NASD) are designed to offload more of the file system's simple and performance-critical operations. For example, in one potential protocol a client, prior to reading a file, requests access to that file from the file manager (1), which delivers a capability to the authorized client (2). So equipped, the client may make repeated accesses to different regions of the file (3, 4) without contacting the file manager again unless the file manager chooses to force reauthorization by revoking the capability (5).

sufficient homogeneity in client operating systems to rely on the failure recovery and integrity of such distributed systems, we believe most environments will contain diverse clients too easily compromised by physical access.

Even if only for accident prevention in systems where the local network is assumed free of malicious entities, file protections and data/metadata boundaries should be checked by a small number of administrator-controlled file manager machines. Moreover, commodity storage must serve the semantics of many existing and yet-to-come file management systems (such as NFS, AFS, NTFS, Netware, etc.). Our goal is to show a network attached storage architecture that enables scalable client-storage performance while minimizing the vestigial file manager bottleneck.

We identify two basic architectures for direct network-attached storage [Gibson97]. The first, NetSCSI, makes minimal changes to the hardware and software of SCSI disks, while allowing NetSCSI disks to send data directly to clients, similar to the support for third-party transfers already supported by SCSI [Miller88, Drapeau94]. Drives' efficient data transfer engines ensure that the drive's sustained bandwidth is available to clients. Further, by eliminating file management from the data path, manager workload per active client decreases.

With storage directly participating in the delivery of data, integrity assurance must be supported by storage. Cryptographic hashes or digests, are essential for assuring integrity in NetSCSI without trusting all nodes attached to the network. For privacy in addition to integrity, encryption can be included.

The principal limitation of NetSCSI is that the file manager is still involved in each storage access; it is translating namespaces and setting up the 3rd party transfer on each request.

The second architecture, Network-Attached Secure Disks (NASD, see Figure 1), relaxes the constraint of minimal change from the existing SCSI interface and focuses on selecting a command interface that reduces the number of client-storage interactions that must be relayed through the file manager, avoiding the file manager's bottleneck without integrating file system policy into the disk. In NASD, data-intensive operations, such as reads and writes, go straight to the disk, while less-common policy making operations, including namespace and access control manipulations, go to the file manager.

Because clients directly request access to data in their files, a NASD drive must have sufficient metadata to map and authorize the request to disk sectors. Authorization, in the form of a time-limited capability applicable to the file's map and contents, is provided by the file manager to protect the manager's control over storage access policy. The storage mapping metadata is maintained by the drive, allowing smart drives to better exploit detailed knowledge of their own resources to optimize data layout, read-ahead, and cache management [Cao94, Patterson95, Golding95]. This is precisely the type of value-added opportunity that nimble storage vendors can exploit for market and customer advantage.

With mapping metadata at the drive controlling the layout of files, a NASD drive exports a namespace of file-like objects. Because control of naming is more appropriate to the higher-level file system, pathnames are not understood at the drive, and pathname resolution is split between the file manager and client. While a single drive object will suffice to represent a simple client file, multiple objects may be logically linked by the file system into one client file. Such an interface provides support for banks of striped files [Hartman93], Macintosh-style resource forks, or logically-contiguous chunks of complex files [deJong93].

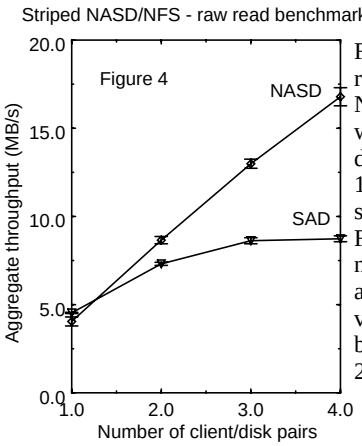


Figure 2: Per-client raw read bandwidth for striped NASD/NFS in comparison with NFSv3. Each NASD drive consists of two 1.0 GB HP C2247 drives striped at 64KB stripe unit. For SAD, the same total number of disks were attached to the file server via 4 independent SCSI busses and striped with a 256K stripe unit.

## NASD Implementation

To experiment with the performance and scalability of NASD, we designed and implemented a prototype NASD storage interface, ported two popular distributed file systems, AFS and NFS, to use this interface and then implemented a striping version of NFS on top of this interface [Gibson97b]. The NASD interface offers logical partitions containing a flat name space of variable length objects with size, time, security, clustering, cloning, and uninterpreted attributes. Access control is enforced by cryptographic capabilities authenticating the arguments of each request to a file manager/drive secret through the use of a digest.

In both NASD/AFS and NASD/NFS ports, the frequent data-moving operations and attribute read operations occur directly between client and NASD drive, while less-frequent requests are handled by the file manager. NFS's simple distributed filesystem model of a stateless server, weak cache consistency, and few mechanisms for filesystem management made it easy to port to a NASD environment; based on a client's RPC request opcode, RPC destination addresses are modified to deliver requests to the NASD drive. The AFS port was more interesting, specifically in maintaining the sequential consistency guarantees of AFS, and in implementing volume quotas. In both cases we exploit the ability of NASD capabilities to be revoked based on expired time or object attributes (size).

Using our implementations[1], we compared NASD/AFS and NASD/NFS performance against the standard Server-Attached Disk (SAD) implementations of AFS and NFS. Our perfectly load-balanced large-read benchmark (512K chunks) showed that NASD is able to scale linearly, up to the drive's aggregate transfer bandwidth, while SAD NFS and AFS systems are limited by the data throughput of the server to just three drives.

To demonstrate striping's ability to automatically load balance requests in a NASD environment, we implemented a striped NFS prototype. In this implementation, striping is transparent to both NASD/NFS file manager and NASD drives, encapsulating striping control in a separate striping manager that exports a NASD interface to the NASD/NFS file manager. Figure 2 shows the results for a synthetic benchmark consisting of 1 to 4 clients simultaneously reading a set of 5 different 8 MB files striped over all drives. Striped NASD/NFS scales linearly while SAD's throughput saturates quickly.

## Networking for Network-Attached Storage

The success of a NASD architecture for scalable storage systems depends critically on its networking environment. We can make a few observations from our experience to date.

**Thin protocol stack:** There is a high standard against which network-attached storage performance will be measured: the efficiency of existing SCSI peripheral access. As a link-level network layer, SCSI has credit-based flow control and reliable in-order delivery. Its corresponding network stack is notably "thin;" it is essentially RPC over the link-layer, so host processing is minimally reduced by sizable data transfer rates. Network-attached storage will be expected to provide the same level of efficiently when client and storage share a link-level medium. Specifically, unlike traditional networking whose efficiency goals seem to be to saturate the wire using up to all of the available CPU, high-bandwidth storage will be measured by the ability of clients to receive and process data concurrently.

**Small messages:** File access entails significant small message traffic: attribute manipulation, command and status, small file access, and metadata access. Network protocols that impose significant connection overhead and long codepaths will be a primary determinant in cached storage response time and (vestigial) file manager scalability. For example, in our experiments we observe that accessing storage over ATM/IP/RPC instead of SCSI induces significant new work for the file manager.

**Network media winner:** The most cost-conscious storage devices (disks, as opposed to array subsystems) will be carefully tailored to specific link-level protocols to enable highly integrated, cost-effective hardware implementations. Moreover, it is unlikely that drives will offer a wide variety of link-level and media alternatives. Instead, a small number (one or two) reasonable choices must lead to direct (switched) transfers between storage devices and high-performance client processors in most installations.

1 Our experimental testbed contains four NASD drives, each one a DEC Alpha 3000/400 (133MHz, 64 MB, Digital UNIX 3.2g-3) with a single 1.0 GB HP C2247 disk. We use four Alpha 3000/400's as clients. All are connected by a 155 Mb/s OC-3 ATM network (DEC Gigaswitch/ATM).

Pursuing this last point further, with network-attached disks there are three identifiable network infrastructures: multiprocessor interconnection network (message or memory semantics), storage network (NASD), and internet access (TCP/IP). Cost-effectiveness argues that office and machine room wiring should only be done once and the number of interface cards minimized. We see two compelling and apparently incompatible configurations: cluster SANs and workgroup LANs.

**Cluster SAN:** High-performance commodity cluster servers will be based on commodity interconnection networks, system area networks (SAN), and will employ protocols optimized for high-bandwidth and low-latency. Such a SAN is a natural fit for the needs of scalable storage. Internet traffic, however, is considerably less bandwidth intensive and will be forwarded out of the SAN by one or more gateway nodes. Storage offering only a SAN-optimized protocol can be made available over the internet by bringing back the protocol converting router (file server) for remote access only as a function in the gateway(s).

**Workgroup LAN:** Collections of client workstations sharing a distributed file system is the other commodity environment that network-attached storage will target. Traditionally, all workstations in such a workgroup have an internet interface to a local area network (LAN) and do all interprocessor communication using it. In this case, network-attached storage must be LAN-attached, but the internet protocol suite is a less effective match for storage. For example, a recent measurement of client CPU overhead in a COTS workstation showed large internet suite transfers over ATM consuming as much as 10 times as much CPU as comparable bandwidths over SCSI.

However, a workgroup LAN that employs a cluster SAN as its LAN overcomes the above concerns. Specifically, storage for clusters and workgroups uses the same media and link-layers, increasing its commodity advantages and, for storage access to processors local to the workgroup, providing appropriately thin protocols and support for small messages. Similarly, remote access to workgroup storage can use the same server or gateway (in this case, router) solution as suggested for clusters. Finally, the use of a SAN for a workgroup enables the workgroup to be employed as a cluster for more effective closely coupled distributed applications.

The actual choice of SAN media remains unclear. Storage implementers certainly favor Fibre Channel since it is already being implemented in storage subsystems and drives. However, another widely understood, more cost-effective SAN might displace Fibre Channel as the obvious network for network-attached storage.

## Acknowledgments

Although in no way to blame for the speculations in this proposal, our ideas were developed in numerous conversations with members of CMU's Parallel Data Lab, notably Jim Zelenka, Eugene Feinberg, and Berend Ozceri, members of the NSIC NASD projects, notably David Anderson, Jim Hughes and John Wilkes, and CMU networking researchers, notably Hui Zhang, Peter Steenkiste, and David Johnson.

## Bibliography

[Anderson95] Anderson, D., Seagate Technology Inc., Personal communication, 1995.

[Cao94] Cao, P., et al.,, "The TickerTAIP parallel RAID Architecture," ACM Transactions on Computer Systems 12(3). August 1994, 236-269.

[Drapeau94] Drapeau, A.L. et al., "RAID-II: A High-Bandwidth Network File Server", 21st ISCA, 1994, pp.234-244.

[deJonge93] de Jonge, W., Kaashoek, M. F., Hsieh, W.C., "The Logical Disk: A New Approach to Improving File Systems," 14th SOSP, Dec. 1993.

[Gibson97] Gibson, G.A., et al., "File Server Scaling with Network-Attached Secure Disks," SIGMETRICS 1997, June 1997.

[Gibson97b] Gibson, G.A. et al,. "Filesystems for Network-Attached Secure Disks," in preparation, CMU-CS-97-118.

[Golding95] Golding, R., Shriver, E., Sullivan, T., Wilkes, J., "Attribute-managed storage," Workshop on Modeling and Specification of I/O, San Antonio, TX, October 1995.

[Hartman93] Hartman, J.H., Ousterhout, J.K., "The Zebra Striped Network File System", 14th SOSP, Dec. 1993, pp. 29-43.

[Miller88] Miller, S.W., "A Reference Model for Mass Storage Systems", Advances in Computers 27, 1988, pp. 157-210.

[Patterson88] Patterson, D.A., Gibson, G., Katz, R.H., "A Case for Redundant Arrays of Inexpensive Disks (RAID)", 1988 SIGMOD, June 1988, pp. 109-116.

[Patterson95] Patterson, R.H. et al., "Informed Prefetching and Caching", 15th SOSP, Dec. 1995.

[VanMeter96a] Van Meter, R., Holtz, S., Finn G., "Derived Virtual Devices: A Secure Distributed File System Mechanism", 5th NASA Goddard Conf. on Mass Storage Systems and Technologies, College Park, MD., Sept. 1996.