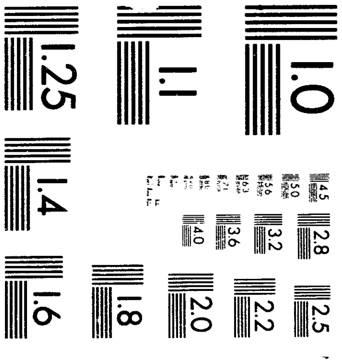
1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3

1 of 1

Conf-931115--13

UCRL-JC-115318
PREPRINT

The High Performance Storage System

Robert A. Coyne
Harry Hulen
Richard Watson

This paper was prepared for submittal to
Supercomputing 93
Portland, OR
November 15-19, 1993

September 1993

Lawrence
Livermore
National
Laboratory

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

The High Performance Storage System

Robert A. Coyne
Harry Hulen

IBM Federal Systems Company
3700 Bay Area Blvd., Houston, TX 77058

Richard Watson

Lawrence Livermore National Laboratory
P.O. Box 808, Livermore, CA 94550

Abstract

The National Storage Laboratory (NSL) was organized to develop, demonstrate and commercialize technology for the storage systems that will be the future repositories for our national information assets. Within the NSL four Department of Energy laboratories and IBM Federal Systems Company have pooled their resources to develop an entirely new High Performance Storage System (HPSS). The HPSS project concentrates on scalable parallel storage systems for highly parallel computers as well as traditional supercomputers and workstation clusters. Concentrating on meeting the high end of storage system and data management requirements, HPSS is designed using network-connected storage devices to transfer data at rates of 100 million bytes per second and beyond. The resulting products will be portable to many vendor's platforms. The three year project is targeted to be complete in 1995.

This paper provides an overview of the requirements, design issues, and architecture of HPSS, as well as a description of the distributed, multi-organization industry and national laboratory HPSS project.

1: Storage system requirements and challenges

The National Storage Laboratory (NSL) has initiated a project to develop a next generation High Performance Storage System (HPSS). Participants in the HPSS project are IBM Federal Systems Company, Lawrence Livermore National Laboratory, Los Alamos National Laboratory, Sandia National Laboratories, and Oak Ridge National Laboratory.

There are many requirements and challenges motivating the need for a new generation of storage system such as HPSS [4,5]. The requirements are driven both by the pull of applications and the push of technology. Today's storage systems can move one to ten million bytes of information per second, but current needs call for systems that can move data at 100 million bytes per second. Future needs will almost certainly reach 500 million to one billion bytes per second and beyond. On the technological front, the successful commercialization of parallel computers and disks has accelerated the already brisk rate of growth in the underlying hardware capabilities. The NSL was organized to develop, demonstrate, and commercialize high-performance storage system technologies to meet these challenges [9]. These new technologies will help meet the storage requirements of the Department of Energy and other government agencies as well as the private sector.

While the immediate objective of HPSS is to meet the individual and collective needs of the participating Department of Energy laboratories, the longer range objectives include participating in a national information infrastructure and transferring the technology to the commercial sector. The HPSS project leverages Department of Energy applications and expertise with that of industry to accelerate the development of U.S. technology for local and nationwide storage system architectures and to facilitate the availability of improved systems for the government and US industry.

Recognizing the importance to the nation's future of communicating, storing and manipulating large quantities of information, both the House of Representatives and the Senate have proposed legislation to create a national information infrastructure to facilitate broad-based access to information [25,26]. The Department of Energy, for example [27,28], has information on the global ecosystem, environmental remediation, petroleum reservoir modeling, the structure of novel materials, and plasma physics, to name only a few of its data resources that are of value to education, research, and industry. Many other government agencies, universities and commercial enterprises also have information assets whose potential economic value can only be fully realized if a national information infrastructure is created.

1.1: Requirement for storage systems to be part of an information infrastructure

To participate in a national information infrastructure, storage systems must become invisible components embedded in and supporting digital libraries of text and graphic information, scientific data, and multimedia data. Users will want to browse, access, and store data using a language and tools that are tied to the application domain, not the storage domain. The interface to the storage system must support appropriate primitives required by application domains such as education, science, and business. These requirements include the need for multiple levels of storage and mechanisms for migration and caching of data between storage levels, with explicit control by the application domain as well as implicit or automatic controls that are the rule in today's storage systems. They include mechanisms for efficient organization such as clustering, partitioning and explicit placement of data under the control of the application domain.

1.2: Requirements for scalability

A fundamental requirement of HPSS is that it must be scalable. There are several important dimensions to the scalability requirement:

- *Size*. The projected storage requirements are for billions of datasets, each potentially terabytes in size, for total storage capacities in petabytes. In addition the file naming environment must support millions of directory entries for thousands of users.

- *Data transfer rate*. The I/O architecture must scale, using parallel I/O mechanisms, to support individual application data transfers in the range of gigabytes per second and total system throughput of several gigabytes per second.

- *Topology*. Scalable parallel processors and parallel storage devices must be supported, as well as parallel data transfer at all levels. In the general case, the number of nodes at either end and the number of paths between can all be different.

- *Geography*. Multiple storage systems located in different geographical areas must be integratable into a single logical system accessible by client systems as wide ranging as personal computers, workstations, and supercomputers.

The scalability requirements for HPSS are driven by applications and the distributed high performance computing environments necessary to run them. For example, large grand challenge applications such as three dimensional hydrodynamics, global climate modeling, lattice gas theory, materials processing, plasma modeling, and magnetic fusion reactor modeling currently generate datasets of the order of magnitude of tens to hundreds of gigabytes. When these models are scaled to take advantage of massively parallel computers, they will generate storage requirements of terabytes of data. Similarly data gathered by experimental devices and sensor based systems in the oil and gas industry, medical field, high energy physics, planetary space probes and earth observing satellites will create terabyte storage requirements. Digitized libraries, educational and other multimedia resources, and databases in commercial enterprises such as insurance companies also represent very large datasets.

1.3: Requirement for an industry standard infrastructure

Large storage systems live and evolve over decades. In order for such systems to have a long useful life, they need to be highly portable, adaptable to new applications, and accommodate new devices, storage system modules, policies and algorithms. A central requirement is that HPSS be implemented using industry standards wherever possible, including for its architecture, function, distributed environment, communications protocols, system management, and security. While standards in some of these areas are firmly established, many are emerging or are only now beginning to be discussed. Others, particularly in the parallel I/O area are at best in early research stages. The standards selected for HPSS are discussed in Section 2 of this paper.

1.4: Requirement for storage system management

Storage system management is the collection of functions concerned with control, coordination, monitoring, performance and utilization of the storage system. These functions are often interdependent, involve human decision making, and span multiple servers. The need for a separate, identifiable storage system management architecture arises from the requirement to exchange management information and to provide control in a consistent, predictable manner between components of the storage system. There is also a requirement for storage system management to be built around system management standards.

2: HPSS architecture, design and implementation considerations

HPSS is an evolving system designed to take advantage of current and future hardware and software technology. While many of the design features of HPSS are comfortably within the state-of-the-art for distributed, client-server, standards-based software projects, many features of HPSS, such as scalable parallel I/O, stake out positions at the leading edge of storage system technology. In this section we discuss the main architectural features and issues of HPSS. Figure 1 illustrates the system architecture that HPSS supports. Note that the architecture is network-centered, including a high speed network for data transfer and a separate network for control. In actual implementation, the control and data transfer networks may be physically separate as shown or may be combined. The data transfer network may also have parallel connections to storage subsystems such as disk arrays and tapes. The disk and tape servers are compatibility mechanisms that provide data network transfer interfaces for devices that cannot be directly network attached.

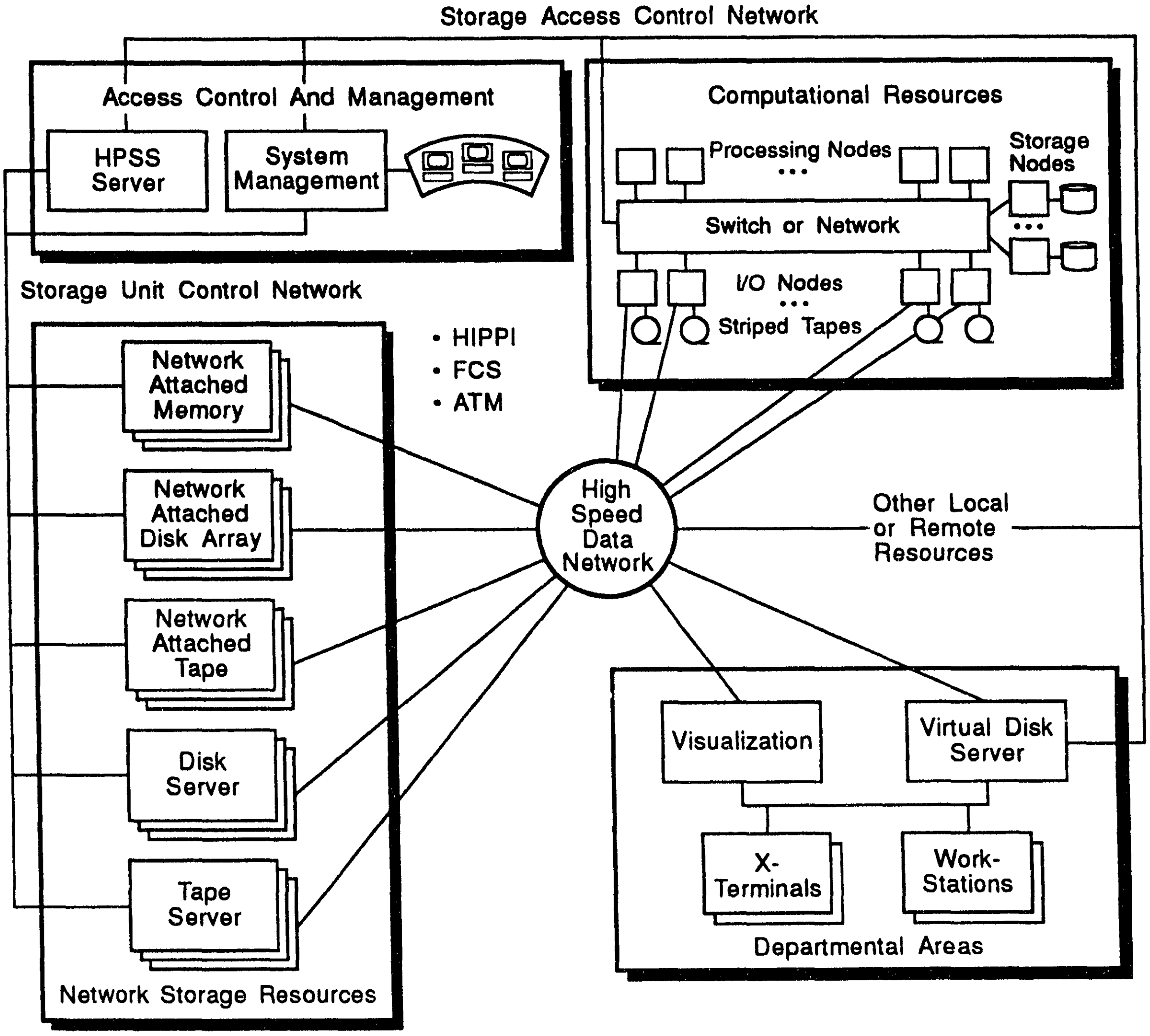

Figure 1. System architecture supported by HPSS

2.1: Modularity based on the Mass Storage Reference Model, Version 5

The basic architecture and modularity of HPSS have benefited from the nearly three years of collaborative work on the Version 5 of the Mass Storage Reference Model [8,16]. This new version of the Reference Model represents the work of over 50 experienced developers and users of storage systems from universities, industry, and government. The Reference Model details those components for which there are plans for developing standard interfaces and services and also recognizes other components and functionality needed in a complete system. The advantages of a design approach based on the Reference Model is four-fold. First, by basing the design on the Reference Model it was possible to complete the system design much quicker than would otherwise have been possible and to do so with the expectation that the design would be internally consistent. Second, by staying close to the Reference Model structure, it is hoped that it will be possible to provide and receive interchangeable software components to and from other vendors and development projects in the future. Third, since many members of the HPSS design team have participated in the Reference Model development, it provides a basis for communication between them and a basis on which to organize the project. Fourth, work on HPSS will provide an opportunity to obtain experience in working with the Reference Model in a new design and feedback our experience into the Model's ongoing evolution, as well as associated storage system standards work.

Recognizing that storage systems will increasingly be embedded in digital libraries and data management systems, the approach of the IEEE Storage System Standards Working Group, and therefore of HPSS, is to design storage systems in such a way as to provide layers of function with application interface points suitable for use by separate digital library, object storage and data management systems. Major components of the HPSS architecture and design, as shown in Figure 2, include the following:

- The Client is the application interface to the storage system. HPSS initially presents clients with a file system abstraction in which the files, and the application interface libraries that operate on the files appear similar syntactically and semantically to the POSIX file and I/O libraries. For historical reasons, the initial file system abstraction is supported by a server called a Bitfile Server. The client interface has been extended to support parallel data transfer interfaces and application level parallel I/O libraries. A POSIX Virtual File System interface is also planned. Initial file system daemons will be provided to support client interfaces to IBM's Vesta parallel file system [7], to Sun's Network File System (NFS) [23], and to the Open Software Foundation's Distributed Computing Environment (DCE) Distributed File System [20]. As HPSS becomes a foundation for digital libraries, large object stores and data management systems, additional application interfaces will be developed.

- The Name Server is recognized by the Reference Model as being a necessary component but one that is outside the boundaries of the model. The purpose of the Name Server is to map a name known to the client to a bitfile id known to the storage system. The initial HPSS Name Server provides a POSIX view of the name space. The strategic Name Server for HPSS is planned to be a name server under development at Cornell, that has the desired characteristics of scalability, performance efficiencies, and reliability. Additional Name Servers will be provided as new data management name spaces are introduced by users.

- The Location Server is planned for future releases to be scalable and keep track of the billions of distributed and replicated HPSS storage objects expected in mature usage.

- The Storage Server organizes physical storage volumes into virtual volumes. Virtual volumes can also span multiple physical volumes that have characteristics not found in physical volumes such as striping and replication groups. In HPSS, the storage server implements an additional storage abstraction called storage segments. Storage Segments are implemented over virtual volumes and provide a volume independent byte stream view of storage. Storage server clients using storage segments are relieved of the chores of allocating and deallocating virtual volume space and dealing with data migration and recovery issues. The Storage Server, based on type-of-service parameters, determines the parallel data transfer organization. It also determines the optimum use of parallel data transfer paths and parallel storage components such as disks and tapes.

- Movers are responsible for copying data to and from storage media as well as requesting transmission of data from a source to a sink. Of particular interest in HPSS is a type of Mover that implements third party data transfer, discussed in Section 2.6, that is key to the network-centered architecture of HPSS. Parallel I/O Movers also execute the parallel data transfer plan determined by the Storage System.

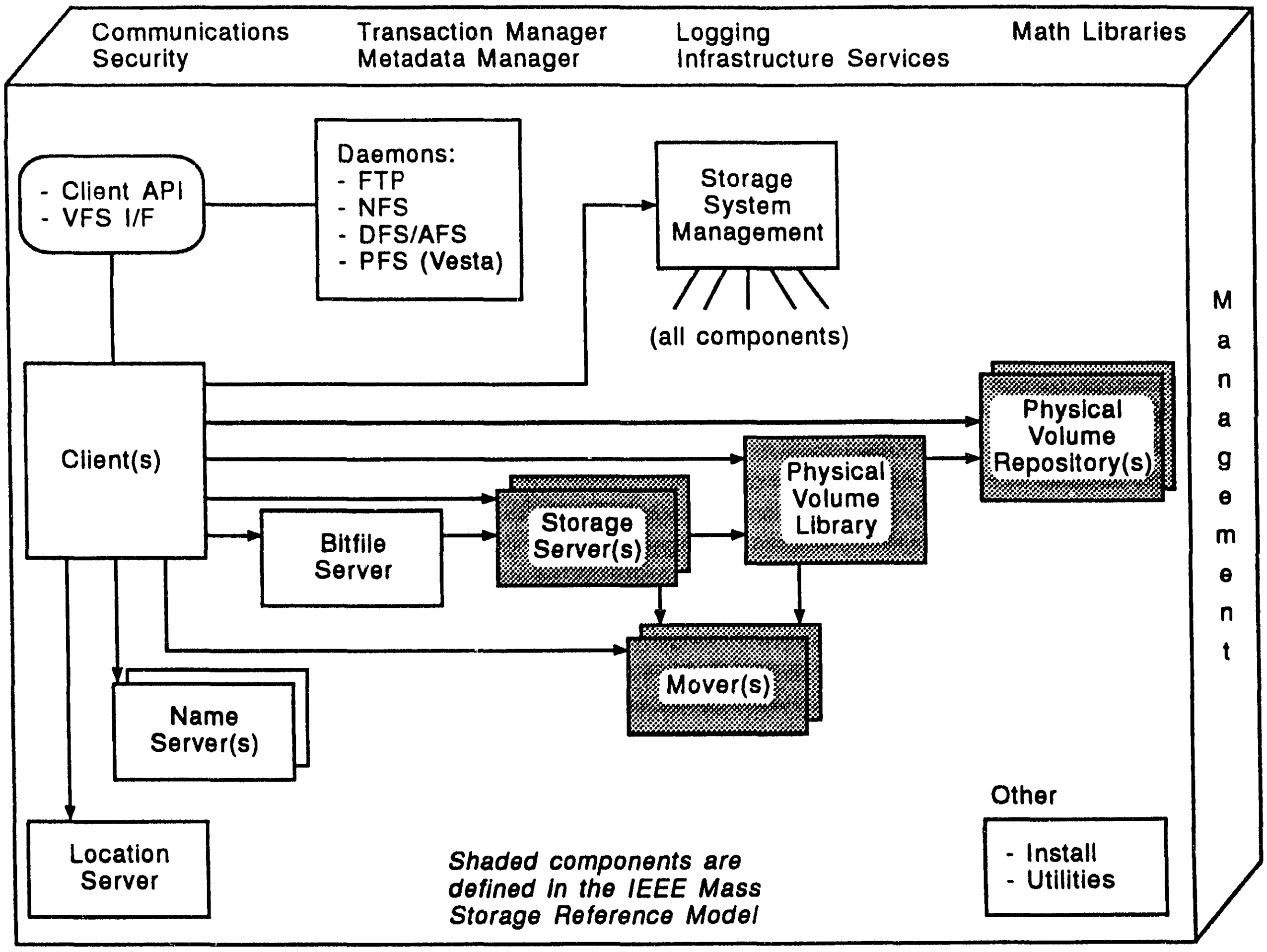

Figure 2. HPSS design is based on the IEEE Mass Storage Reference Model

- The Physical Volume Repository is responsible for storing removable physical volumes contained in cartridges and mounting these volumes onto physical drives, employing either robotic or human operators.
- The Physical Volume Library (PVL) integrates multiple Physical Volume Repositories into a distributed media library across an entire enterprise, similar to a distributed tape management system. The PVL in HPSS is responsible for atomic mounts of parallel tape cartridges.

In HPSS, each component has a well-defined application program interface. The APIs define the operations that may be performed on the objects owned by each component. Each API is therefore available directly to any application with proper authorization, setting the stage for the use of HPSS as an embedded storage system in data management, object store and digital library applications.

2.2: Distribution and multitasking based on DCE

One of the requirements of HPSS is that it be possible to distribute, replicate, and multiprocess its servers and clients. Further, it is required that HPSS provide concurrent access to hundreds of applications. The basic server architecture of HPSS is built on the Open Software Foundation's DCE. In particular, HPSS is based on DCE's remote procedure call (RPC) services for control messages and DCE threads for multitasking [20]. The latter is vital to both multiprocessing and serving large numbers of concurrent requests.

One of the requirements outlined earlier is the need for HPSS to integrate with distributed file services, particularly NFS and DFS. Progress has been made in developing approaches to integrating distributed file systems with mass stores [19]. While it will be relatively straight forward to integrate the HPSS and DFS naming and security environments because of the common DCE base, a greater challenge will be supporting network-connected storage devices, as described below, with a transparent client DFS environment. Currently DFS does not separate data and control messages as is required by HPSS. It performs all data and control transfers within the DCE RPC framework. Therefore modification to the DFS communication model will be required. The HPSS project plans to work closely with the DFS development community to solve this problem.

In the case of NFS access we are planning to use high speed NFS frontend servers integrated with appropriate automatic migration and caching mechanisms to and from the large hierarchical storage system. This will present several challenges, particularly when the same data is shared between multiple NFS frontends, given NFS limitations in providing full location transparency [23].

2.3: Built-in reliability through metadata and transaction management

HPSS is designed around a distributed metadata and transaction manager, an idea borrowed from database management systems. The metadata manager is a distributable entity that reliably stores the metadata each server requires for each of its abstract objects. This manager supports the concept of distributed atomic transactions involving multiple servers, that is, a set of actions that must be treated as a single operation and whose suboperations either all complete or no data is changed. Transaction management allows HPSS to provide facilities to commit or abort a transaction across multiple servers, thus providing assurance that metadata updates will not be left in a partially completed state. Transaction management guards against leaving data without pointers or pointers without data in the event a failure occurs in the midst of an update. For example, transaction management protects against having a name space entry without a corresponding file or vice versa, that has been a problem in previous storage systems. Transaction management is difficult to retrofit into an existing system. The requirement for transaction management is therefore one of the principal motivators for developing HPSS as a new system rather than as a derivative of an existing storage system. The metadata and transaction manager used in HPSS is Encina [12], that was developed by Transarc and is supported on several vendors' platforms.

In addition to use of transaction management, HPSS assures reliability and recoverability by replicating critical metadata and the naming database using services provided by Encina. Periodic backups are also provided of critical metadata. Another feature of the HPSS design simplifying recovery is that all levels of the system are implemented with both forward and backward pointers throughout critical metadata structures.

2.4: Built-in security through use of DCE

HPSS is designed to be a distributed storage system in which clients, intelligent storage devices, and control mechanisms communicate with each other over a network. Like all distributed architectures, this raises security issues. How, for example, can a server know that a request from a client or a storage server is an authorized, authentic request? In order to address this problem, HPSS is designed around DCE Security [20], that is based on Kerberos which originated at MIT's Project Athena. By using DCE, the mechanisms for providing security in a distributed environment are moved outside the domain of proprietary HPSS design and into the domain of an industry standard environment. The mechanism through which security is invoked is the DCE secure RPC mechanism. All of the control interfaces between HPSS modules are in the form of RPCs. The requirement for security and the decision to use a DCE infrastructure was another motivation that helped drive the decision to implement HPSS from the ground up, as it did not appear practical to retrofit such a capability into an existing design.

2.5: Multiple dynamic hierarchies

A concept pioneered at the NSL is multiple dynamic hierarchies. Current storage systems generally provide a single, simple, predefined hierarchy of storage media [4]. In the conventional hierarchy, frequently used data is kept on disk, less frequently used data is kept in an automated tape library, and infrequently used data is kept in tape vaults. With the availability of new media, such as solid state disk, disk arrays, and helical scan tape; and the wide range of devices offering different levels of cost, capacity, and performance, there is a need for more complex hierarchies that can be defined and dynamically redefined by a system administrator. The need for multiple hierarchies is based on such factors as location, reliability, data type, cost, performance and project affiliation. Each hierarchy must be adaptable to meet specific application and system configuration requirements and must be able to change over time under the control of a system administrator. The approach used in the initial storage system at NSL, which is based on UniTree [13] and adopted for HPSS was originated by NSL participants Buck and Coyne [3].

2.6: Network-connected storage concepts

The NSL has successfully demonstrated the concept of a network-connected storage system architecture in which network-attached storage devices communicate directly with supercomputers and other clients under the control of a storage system management and control entity [5,9]. Using a network-connected architecture, the NSL has shown, in its work with NSL UniTree, more than a tenfold increase in throughput over the former processor-centered architecture in use at the National Energy Research Supercomputer Center. NSL UniTree is an extension of the UniTree storage system from OpenVision [13]. The network-centered architecture is shown in Figure 1, which depicts processing nodes and storage devices connected by a high speed network such as HIPPI, FCS, or ATM.

Most operational storage systems at national laboratories and supercomputer centers use general purpose computers as storage servers. These storage servers connect to storage units such as disks and tapes and serve as intermediaries in passing data to compute-intensive nodes [4]. As data rates increase for storage devices and communications links, it becomes necessary to increase the size of the storage server to provide the required memory and data bandwidth. The alternative, that has been the subject of much of the work at NSL, is to attach storage devices directly to the network [6,9,15,17,24]. This network-centered alternative supports higher data rates than can be supported in the traditional processor-centered storage server configuration and does not require the storage server to be as large, thus reducing its cost. HPSS will extend the network-connected storage architecture to use Parallel I/O, discussed below.

2.7: Support for parallel clients and scalable parallel I/O

Most of the work in parallel computing has focused on processing and associated programming models. Very little work has gone into thinking about the equally important issues associated with parallel I/O, particularly parallel I/O involving a hierarchy of storage systems [2]. There have been important developments in RAID disk systems and striping I/O systems that have transfer rates in the range of 60 to 100 megabytes per second [10,11,14,22], but what is needed is another order of magnitude or more increase in performance, only possible through parallel transfers to parallel RAID disk and tape systems.

To meet its scalability requirements, the HPSS design is addressing many issues from simply providing adequate size fields in its data structures to complex issues such as fast recovery after a crash and efficient resource management algorithms for very large stores. One of the most challenging areas will be meeting the scalabilty requirements for I/O rates. This requires support for parallel I/O at all levels of the system architecture from application libraries and compilers down through the servers and the movers. The HPSS project plans to work closely with other research projects in the parallel I/O area [2].

The HPSS project is working to develop an application driven, scalable, parallel I/O model and architectures that provide the needed support. In parallel processing algorithms, a single large logical data object, such as a matrix, linked list, or other data structure, is broken into pieces that are distributed among the memories of various processors and storage devices to maximize overall algorithm performance. The pieces may form a non-overlapping partition of the object, or they may overlap slightly, where the redundancy of the overlapping may help reduce communication between processors. The programmer, the parallel libraries, and the compiler work to distribute the data in such a way as to increase parallel computation while at the same time decreasing the amount of costly interprocessor exchanges of data.

Parallel I/O services must minimize the time to write or read such distributed data objects to or from secondary and tertiary storage by making use of multiple storage devices and parallel data paths. The result is that a single logical object will have to be spread out (mapped) over many devices in such a way that the high-level structure of the data object, as seen at the application program level, can be reconstructed from the many pieces. This is complicated by the fact that the application may read or write the separate blocks of data in quite unpredictable order from any of the processors. Furthermore, a parallel data object may be stored with a distribution structure that is near optimal for the way the data set was produced but which may be very inefficient for subsequent patterns of access. The parallel I/O system needs to interact with whatever data partitioning methodologies are employed by client applications.

Figure 3 illustrates the problem of moving data between parallel nodes. The convention is that the data moves from the source nodes to the sink nodes. The source and sink nodes may be parallel processors or striped I/O devices. In Figure 3, four parallel source nodes are shown moving twelve blocks of data to three parallel sink nodes. The plan by which the data is moved is negotiated between the source and sink movers and the storage server. Then, the storage server manages the transfer.

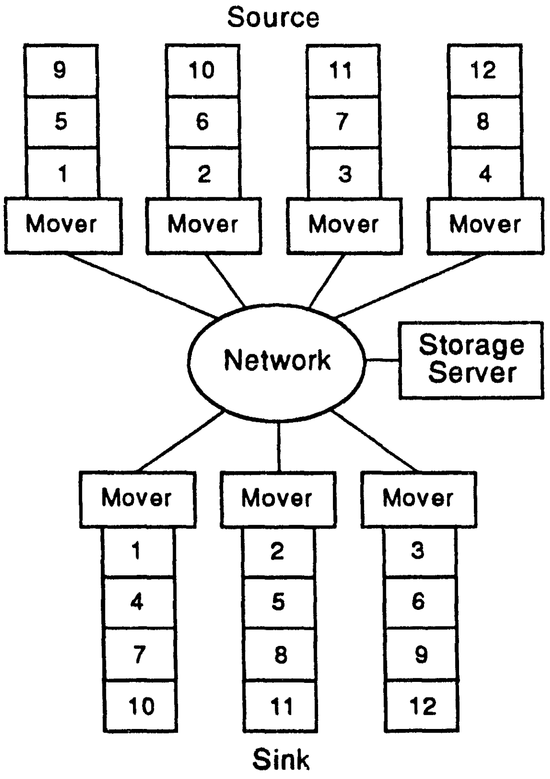

Figure 3. An example of moving data in parallel

Figure 3 could just as easily show interconnected parallel computer systems between which data transfer is desired. A project has been initiated under sponsorship of the NSL, DOE national laboratories, IEEE Storage System Standards Working Group in collaboration with the parallel computer vendor and research communities to define and prototype strategies for parallel data transfer interface with the intent of developing a standard for parallel data exchange [1]. HPSS will support clients that are either parallel or sequential computers. The initial implementation will interface to IBM Research Division's Vesta parallel file system [7]. Later versions will interface to any vendor's parallel computing system that supports the parallel data exchange protocols referenced above.

2.8: Storage system management

In a distributed storage system such as HPSS, the need for a storage system management architecture is much greater than in previous centralized systems. In the current prototype storage system at the National Storage Laboratory, some significant proof of concept work has been done using a storage management system with a graphical user interface [18]. For HPSS the decision was made to clearly identify during system design all system management functions and managed objects and to provide a common management system interface. The management system itself is still being designed, but the intent is to use, to the extent possible, an industry standard base such as the OSF Distributed Management Environment [21].

HPSS plans are to define the managed objects for storage systems in a manner consistent with the ISO/OSI Guidelines for the Definition of Managed Objects (GDMO) [29]. This structure will permit the HPSS storage system management framework to be standardized around the emerging technology from the OSF Distributed Management Environment.

3: Project methodology and status

The HPSS project began in 1992 as a series of requirements meetings. The initial goal was to determine if two or more organizations could converge on a single set of high level requirements called Level A requirements. A composite Level A requirements document was agreed to by representatives from the four DOE laboratories and IBM. In addition, Argonne National Laboratory, the National Center for Atmospheric Research, and Pacific Northwest Laboratory provided requirements and helpful input and review.

The next step was to expand the requirements into a more detailed Level B requirements specification for each of the modules and functional areas outlined in Section 2. This was accomplished, with the results cross-referenced to the Level A requirements, in March, 1993.

Once the requirements were completed, a project organization was created. The basic project organization consists of an Executive Committee, a Technical Committee and Development Teams. The Executive Committee has a member from each participating organization and handles policy level issues. The Technical Committee is led by the HPSS project leader and has a member from each of the participating organizations, who are also members of the Development Teams. The Technical Committee meets at least once a week by teleconference to deal with resource allocation, scheduling and technical issues spanning the Development Teams. The Technical Committee is responsible for setting coding, documentation and other development standards for the project. The project is following high standards of software engineering. This is essential not only for system quality but to facilitate the distributed development nature of the project.

The Development Teams are organized around the modularity of the system and infrastructure areas. To the extent possible, the Teams are organized to take advantage of geographical closeness. The Teams interact frequently by teleconference, email and when necessary at a participant's site for design sessions and design reviews with the Technical Committee. Each Development Team submits a weekly progress report to the Technical Committee. A central on-line repository of project documentation, designs, schedules, and source code, is maintained. Computer systems are located at each site for development and there is shared access to certain systems as necessary to reduce software license fees. Subsystem integration is distributed, but system integration will be at one site. Our experience with the distributed development process is very positive.

By using the IEEE Reference Model as the starting point, a design for HPSS was achieved which consists of well-defined modules that could be developed relatively independently. This was an important factor considering that HPSS is being developed at five sites across the country. A design document was developed for each server along with interface definitions between servers. These designs were reviewed by the other server teams and the Technical Committee. By July, 1993, the design of the first release was formalized.

At this point, implementation of the first release is well under way. Staffing is up to 24 individuals, some part time, at five sites. Discussions are also in progress with other possible DOE laboratory and industrial participants. By the end of the first quarter of 1994, it is expected that the first release will be developed, integrated, and undergoing system test. The test plan then calls for the individual laboratories and IBM to each set up a pilot project to test the first release in as realistic an operational environment as possible. A second and third release are planned, that will follow the same design, development, and test methodology as the first release.

The expectation is that HPSS will, at the appropriate stage of development, be licensed by the developers for commercial use. This will include a source license so the system can be ported. It is anticipated that by the time of the second release, that should be toward the end of 1994 or early 1995, there will be sufficient functionality and maturity in HPSS to allow HPSS to be ported to other platforms and installed at other computer centers. The schedule for offering HPSS for use outside the sponsoring organizations has not been established, but will come shortly after HPSS has stabilized and proven itself in operation at the participating government laboratories.

4: The National Storage Laboratory

HPSS is a project of the National Storage Laboratory. The NSL collaboration, launched in 1992, has grown in just under two years into a major collaboration of governmental, industrial, and academic partners. The DOE laboratory partnership includes Lawrence Livermore, (LLNL) Los Alamos (LANL), Sandia (SNL), and Oak Ridge (ORNL) national laboratories, with planned participation of other DOE laboratories in the future. Industrial participants include IBM Federal Systems Company, IBM ADSTAR, Ampex Systems Corporation, Maximum Strategy Incorporated, Network Systems Corporation (NSC), OpenVision, Zitel Corporation, PsiTech, CHI Systems, Cray Research Inc., IGM, and Kinesix. University and supercomputing center participants include Cornell Office of Information Technologies and the San Diego Supercomputing Center. Discussions with other potential industrial, academic, and government agency participants are ongoing.

The approach of the NSL is based on the premise that no single company, government laboratory, or research organization has the ability to confront all the system-level issues that must be resolved before there is significant advancement in the storage systems technology needed for a national information infrastructure and for local high performance computing environments.

The NSL's primary prototype equipment site is located within the National Energy Research Supercomputer Center at LLNL. The NSL's collaborative software development projects are distributed across the DOE laboratories and other members sites. Work is performed under Cooperative Research and Development Agreements (CRADAs), with all participants funding their own participation. In particular, the DOE laboratories are supported by DOE Defense Programs Technology Transfer Initiative and DOE Energy Research funding.

Acknowledgments

We wish to acknowledge the many discussions and shared design, implementation, and operation experiences with our colleagues in the National Storage Laboratory collaboration, the IEEE Mass Storage Systems and Technology Technical Committee, the IEEE Storage System Standards Working Group, and in the storage community. Specifically we wish to acknowledge the people on the HPSS Technical Committee and Development Teams. At the risk of leaving out a key colleague in this ever-growing collaboration, the authors wish to acknowledge Dwight Barrus, Ron Christman, Danny Cook, and Tyce McLarty from LANL; Larry Berdahl, Jim Daveler, Dave Fisher, Mark Gary, Steve Louis, and Norm Samuelson from LLNL; Rena Haynes, Sue Kelly, Hilary Jones, and Bill Rahe from SNL; Randy Burris, Dan Million, and Vicky White from ORNL; and Paul Chang, Jeff Deutsch, Kurt Everson, Rich Ruef, Danny Teaff, and Terry Tyler from IBM Federal Systems Company and its contractors.

This work was, in part, performed by the Lawrence Livermore National Laboratory,* Los Alamos National Laboratory, Oak Ridge National Laboratory, and Sandia National Laboratories, under auspices of the U.S. Department of Energy, and by IBM Federal Systems Company under Independent Research and Development and other internal funding.

References

1. Berdahl, L., ed., "The Parallel Data Exchange", working document in progress, available from Lawrence Livermore National Laboratory, Sept. 1993.
2. Bershad, B., et al, "The Scalable I/O Initiative", White paper, available through the Concurrent Supercomputing Consortium, CalTech, Pasadena, CA, Feb. 1993.
3. Buck, A. L. and R. A. Coyne, Jr., "Dynamic Hierarchies and Optimization in Distributed Storage System," Digest of Papers, Eleventh IEEE Symposium on Mass Storage Systems, Oct. 7-10, 1991, IEEE Computer Society Press, pp. 85-91.
4. Coleman, S. and R. W. Watson, "The Emerging Paradigm Shift in Storage System Architectures," Proceedings of the IEEE, April 1993.
5. Coleman, S. S., R. W. Watson, R. A., Coyne, H. Hulen, "The Emerging Storage Management Paradigm", Proceedings of the Twelfth IEEE Symposium on Mass Storage Systems, Monterey, CA., April 1993.
6. Collins, B., et al., "Los Alamos HPDS: High-Speed Data Transfer" Proceedings of the Twelfth IEEE Symposium on Mass Storage, IEEE Computer Society Press, April, 1993.
7. Corbett, P. F., D. G. Feitelson, S. J. Baylor, J. Prost, "Parallel Access to Files in the Vesta File System," Proceeding of Supercomputing '93, IEEE Computer Society Press, November, 1993.
8. Coyne, R. A. and H. Hulen, "An Introduction to the Storage System Reference Model, Version 5," Proceedings of the Twelfth IEEE Symposium on Mass Storage, IEEE Computer Society Press, April, 1993.
9. Coyne, R. A., H. Hulen, and R. W. Watson, "Storage Systems for National Information Assets," Proceedings of Supercomputing 92, Minneapolis, IEEE Computer Society Press, Nov. 1992.
10. DeBenedictus, E., and S. Johnson, "Extending Unix for Scalable Computing", to appear in IEEE Computer, Nov. 1993.
11. Dibble, P. C., "A Parallel Interleaved File System", Ph.D. Thesis, Univ. of Rochester, 1989.
12. Dietzen, Scott, "Distributed Transaction Processing with Encina and the OSF/DCE", Transarc Corporation, September 1992.
13. DISCOS, "UniTree, the Virtual Disk System, An Overview," available from OpenVision, 1991.
14. Ghosh, J. and B. Agarwal, "Parallel I/O Subsystems for Distributed Memory Multicomputers", Proceeding of the Fifth International Parallel Processing Symposium, May 1991.
15. Hyer, R., R. Ruef, and R. W. Watson, "High Performance Direct Network Data transfers at the National Storage Laboratory" Proceedings of the Twelfth IEEE Symposium on Mass Storage, IEEE Computer Society Press, April, 1993.
16. IEEE Storage System Standards Working Group, "Mass Storage Reference Model Version 5," IEEE Computer Society Mass Storage Systems and Technology Committee, unapproved draft, August 5, 1993.
17. Katz, R. H., "High Performance Network and Channel-Based Storage," Proc. of IEEE, Aug. 1992.
18. Louis, S., and S. W. Hyer, "Applying IEEE Storage System Management Standards at the National Storage Laboratory" Proceedings of the Twelfth IEEE Mass Storage Symposium, IEEE Computer Society Press, April, 1993.
19. Nydict, D. et al., "An AFS-based Mass Storage System at the Pittsburgh Supercomputer Center" Digest of Papers Eleventh IEEE Symposium on Mass Storage Systems, Oct., 1991, pp. 117-122.
20. Open Software Foundation, Distributed Computing Environment Version 1.0 Documentation Set. Open Software Foundation, Cambridge, Mass. 1992.
21. Open Software Foundation Distributed Management Environment (DME) Architecture, Open Software Foundation, Cambridge, Mass., May 1992.
22. Peterson, D. G., and R. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)," Proc. SIGMOD Int. Conf. on Data Management, Chicago 1988, pp. 109-116.
23. Sandberg, R., et al., "Design and Implementation of the SUN Network Filesystem," Proc. USENIX Summer Conf., June 1989, pp. 119-130.
24. Sloan, J. L., B.T. O'Lear, D. L. Kitts, and E. E. Harano, "The MaSSIVE Project at NCAR" Proceedings of the Twelfth IEEE Symposium on Mass Storage, IEEE Computer Society Press, April, 1993.
25. U.S. Congress, "National Information Infrastructure Act of 1993, - H.R. 1757," Washington, DC, April 21, 1993.
26. U.S. Congress, "National Competitiveness Act of 1993 - S.4," Washington, DC, January 21, 1993.
27. U.S. Department of Energy, Office of Energy Research, Office of Scientific Computing, "Requirements for Supercomputing in Energy Research: The Transition to Massively Parallel Computing," National Technical Information Service, Publication DOE/ER-0587, February, 1993.
28. U.S. Department of Energy, Office of Energy Research, Office of Scientific Computing, "The DOE Program in High Performance Computing and Communications Energy Transition," National Technical Information Service, Publication DOE/ER-0536, March, 1993.
29. "Information Technology - Open Systems Interconnection - Structure of Management Information - Part 4: Guidelines for the Definition of Management Objects," ISO/IEC 10165-4, 1991.

*Under Contract No. W-7405-Eng-48.

DATE

FILMED

1 / 31 / 94

END