

Experiences From NASA/Langley's DMSS Project

A version of this paper was published in the proceedings for the 14th IEEE Symposium on Mass Storage Systems. This version is dynamically linked to some of current usage statistics from the DMSS system.

Abstract

There is a trend in institutions with high performance computing and data management requirements to explore mass storage systems with peripherals directly attached to a high speed network. The Distributed Mass Storage System (DMSS) Project at the NASA Langley Research Center (LaRC) has placed such a system into production use. This paper will present the experiences, both good and bad, we have had with this system since putting it into production usage. The system is comprised of:

- 1) National Storage Laboratory (NSL)/UniTree 2.1,
- 2) IBM 9570 HIPPI attached disk arrays (both RAID 3 and RAID 5),
- 3) IBM RS6000 server,
- 4) HIPPI/IPI3 third party transfers between the disk array systems and the supercomputer clients, a CRAY Y-MP and a CRAY 2,
- 5) a "warm spare" file server,
- 6) transition software to convert from CRAY's Data Migration Facility (DMF) based system to DMSS,
- 7) an NSC PS32 HIPPI switch, and
- 8) a STK 4490 robotic library accessed from the IBM RS6000 block mux interface.

This paper will cover:

- the performance of the DMSS in the following areas:
 - file transfer rates,
 - migration and recall, and
 - file manipulation (listing, deleting, etc.).
- the appropriateness of a workstation class of file server for NSL/UniTree with LaRC's present storage requirements in mind
- the role of the third party transfers between the supercomputers and the DMSS disk array systems in DMSS.
- a detailed comparison (both in performance and functionality) between the DMF and DMSS systems
- LaRC's enhancements to the NSL/UniTree system administration environment
- the mechanism for DMSS to provide file server redundancy
- the statistics on the availability of DMSS
- the design and experiences with the locally developed transparent transition software which allowed us to make over 1.5 million DMF files available to NSL/UniTree with minimal system outage

1. INTRODUCTION

The DMSS project has integrated emerging technologies from the areas of data storage hardware, high speed communications, and hierarchical storage management software into the latest production mass storage system to serve LaRC. This new system is characterized by peripherals directly attached to a high speed network, and a workstation acting as the file server. The file server no longer handles data in a high speed data transfers involving high speed clients like a CRAY Y-MP. Between 1991 and 1993 LaRC assembled and developed a prototype of this system as a proof of concept [1]. From the end of 1993 until the start of production (October 1994) LaRC concentrated on system stability issues.

DMSS is responsible for meeting the mass storage needs of the central scientific computing complex as well as the distributed computing systems spread over the local area network. It replaced the previous storage system which was run on the CRAY Y-MP using the Data Migration Facility (DMF) software.

2. DMSS DESIGN GOALS

The first design goal for DMSS was to provide a cost effective storage solution to replace the CRAY Y-MP implementation. Previous solutions to mass storage problems incorporated a supercomputer or mini-supercomputer as the file server in order to meet performance needs. Machines of these classes are expensive to buy and maintain. The separation of data and control as spelled out in the Mass Storage System Reference Model Version 4 [6] was a good way to achieve performance while reducing the throughput capacity needed on the server. Using network-attached peripherals has allowed the DMSS file server to become a workstation class machine.

The second goal was that the solution be an open systems solution so that it can provide an environment for evolvability when new technologies surface. Wherever possible, platforms which adhered to standards were chosen.

The third goal was high availability. Due to the limited disk space on the supercomputers, user jobs rely on the mass storage system for the retrieval of required files before running, and the storage of restart files and result files at the end of jobs. Thousands of workstations and PCs on the local area network also depend on the mass storage system for data sharing, system backup, and post-processing of supercomputer runs. System redundancy was provided in many system components in order to increase availability.

The fourth goal was to maximize the performance for supercomputer accesses. We achieved this via the Intelligent Peripheral Interface 3 (IPI3)[5] third party transfer protocol[8] over a High Performance Parallel Interface (HIPPI)[2,3] switch control[4] fabric.

The fifth goal was scalability. The solution must be easily expandable, should the requirements for mass storage change drastically within a short time. DMSS has removed the limitations of many of the current storage systems, such as limits on the types and amount of devices which can be attached and utilized.

These five goals mentioned above are by no means mutually exclusive. The cost of workstation class file servers enables the provision of file server redundancy in a cost-effective manner. Network-attached peripherals allow for expandability and high performance access from HIPPI-attached supercomputers. Openness, evolvability, expandability, and scalability all translate into savings on capital equipment and human resources.

3. IMPLEMENTATION

3.1 DMSS CONFIGURATION

Figure 1 shows the DMSS configuration. DMSS has two file servers, one IBM RS6000/990 and one IBM RS6000/970, that have 256 Megabytes (MB) of memory each. The 990 is the primary server and the 970 is the secondary server. The National Storage Laboratory (NSL)/UniTree, a distributed hierarchical storage management software package marketed by IBM Federal, runs on the servers. NSL/UniTree is a version of UniTree that has additional features of multiple storage hierarchies and support for IPI3 third party transfer drivers. NSL/UniTree manages two IBM 9570 disk array systems totalling 140 gigabytes (GB) and 14.4 Terabytes (TB) of magnetic tape storage in three silos of the StorageTek Automated Cartridge System (ACS 4490). The HIPPI-attached disk array systems and STK 4490 are accessible to both file servers. The STK 4490 access to the IBM RS 6000 is via block mux. When DMSS first went into production, the two file servers, two disk array systems and the CRAY Y-MP and CRAY 2 were all connected to a Network System Corporation (NSC) PS-32 HIPPI switch control which allows the simultaneous communication of 32 devices or computers, each at 100 MB/sec rate. The CRAY 2 has since been decommissioned and thus not included in the present DMSS.

In addition to the HIPPI-IPI3 disk array systems, each server also has 16 GB of SCSI disks, which store the NSL/UniTree databases. An external SCSI disk system, which can be attached to either the primary or secondary server, contains the primary copies of the databases. The purpose for this disk system is to make the switch over from one server to another easier.

3.2 CRAY HIPPI/IPI3 THIRD PARTY DRIVERS

LaRC has developed kernel drivers to perform HIPPI/IPI3 third party transfers on both of the CRAY Y-MP and CRAY 2. These drivers act as Movers as specified in the Mass Storage Reference Model Version 5 [7]. The drivers were first developed under the CRAY operating system UNICOS Release 7 and have been in production use. Currently, the CRAY Y-MP driver is being ported onto the UNICOS Release 8 platform. LaRC undertook the driver development because CRAY Research (CRI) indicated that they had no plans for such a product. CRI still does not support a third party driver. Our driver has both a kernel version and user space version. The user space version can easily be maintained without costly supercomputer system time, but its concurrent third party transfers are less efficient.

3.3 ACCESS TO DMSS

There are three different ways for the users to transfer files to/from DMSS: ftp, rcp, and the Explicit Archival and Retrieval System (EARS). The majority of accesses use either EARS or rcp. NFS is not used due to very poor performance.

Clients with HIPPI connections use third party transfers to transfer data directly to/from the HIPPI attached disk arrays. Clients without HIPPI get their data through the server.

FTP

NSL/UniTree supplies a special ftpd which communicates directly with NSL/UniTree when I/O is done and talks to clients using standard ftp protocol. Therefore, no modification is necessary at the ftp client side in order to transfer data to/from NSL/UniTree.

RCP

LaRC has modified rcp on the file server to act like the special ftpd. All I/O to the local machine has been replaced with the appropriate calls from the NSL/UniTree client API library (libnsl) to effect I/O directly with NSL/UniTree. A client's rcp still uses the standard rcp communications, so no client side modification is necessary.

EARS

EARS is an interface that LaRC has been using for years. Originally it was part of our MVS based mass storage system. EARS provides a UNIX-like set of commands for access to DMSS (ie. masls, maschmod, masrm, masput, masget, etc). On the server these commands are C programs which are linked with libnsl in order to communicate directly with NSL/UniTree.

On most clients the EARS commands are simple shell scripts which pass along the command they want to run to the server for execution. The exceptions are masput and masget which normally invoke rcp on the client (which then talks to the modified rcp on the server).

On clients with HIPPI connections masput and masget are the same C programs that exist on the server. This allows these clients to use the third party transfers to dramatically increase throughput.

3.4 DMF TO NSL/UNITREE TRANSITION

LaRC developed a mechanism for transparently transitioning over 1.7 million files from DMF to NSL/UniTree without having to move all of the data before the switch over to DMSS.

The design consists of 4 parts:

- DMF database extraction and UniTree population
- Modified tapesrvr/mvr code
- Mount-and-Read (MNR) task
- A background transition program (plowd)

3.4.1 DMF DATABASE EXTRACTION AND UNITREE POPULATION

Software was developed that traverses the DMF databases and the UNICOS file system and generates all of the information needed to find a DMF file. This data is then passed to a population program which makes the appropriate entries into the NSL/UniTree databases.

3.4.2 MODIFIED NSL/UNITREE SERVER CODE

Once the population is finished a second set of tape server processes (called dmfsrvr and dmfmovr) are started which deal with the new entries. The dmfsrvr is the same code as the normal NSL/UniTree tape server. A few things were changed such as disabling writes to the dmfsrvr. The main changes occur in the read logic. The new flow is as follows:

- Get file header
- while not done
 - Determine the next tape the data is on
 - Call MNR (see next section)

- Tell mover to move data
- Update MNR cache info
- go back to top of loop if not done

The dmfmovr does not actually get data off a tape. The MNR process (described in the next section) gets the data off tape and creates a disk image of the tape. The dmfmovr reads this disk image to get the data to send to the disk mover. The mover code has been changed to do this.

3.4.3 MOUNT AND READ (MNR) PROCESS

This program understands how to handle a DMF tape. When the dmfsrvr gets a request for a file it invokes MNR. MNR then mounts and reads the tape and creates a disk image of the tape. It then parses the image to create an index of DMF files and offsets. It then passes the offset and length back to the dmfsrvr, which is in turn passed to the dmfmovr.

MNR maintains a cache of these tape images. If a user makes a request for a file which is on a tape that is already on disk, then the response is almost immediate. MNR is responsible for managing the cache of images. We chose to cache the tape image to disk in order to minimize the number of tape mounts and tape positioning needed to transition all of the data into NSL/UniTree.

3.4.4 PLOWD

The dmfsrvr/dmfmovr and MNR will handle each individual users request for a DMF format file. The first time a DMF file is accessed via NSL/UniTree it is moved to the disk server. Once there it will be re-migrated to a new tape in the NSL/UniTree format. We want to have all old DMF files converted some day, however, the users will never request all of the old files. A program called plowd is responsible for eventually accessing all of the files. Plowd constantly looks at the tape image cache from MNR. When a new tape comes into the cache it then scans the database to determine what files are on that tape and then issues a stage request on those files (1 at a time). These requests go through the dmfsrvr just as if the user had requested the file. The dmfsrvr then calls MNR which finds the tape image already on disk and so it responds quickly.

If the system is idle plowd will start the process by itself, making requests for files, causing a tape to be cached to disk. Due to this activity NSL/UniTree has never been idle since it went into production.

As of 1/16/95 53 percent of the files and 70 percent of the data have be "plowed". The entire process will take approximately six months.

3.5 MISCELLANEOUS NSL/UNITREE ENHANCEMENTS

NSL/UniTree is lacking in the area of system management tools. Many functions we had taken for granted in other systems were not available in NSL/UniTree. For instance, we could not tell which files resided on which tapes, nor could we produce a report of how much data was stored in the system (other than a ls -lR of everything which takes 5 days). LaRC has written a number of tools to try and fill the gaps in the NSL/UniTree supplied utilities. These utilities allow LaRC to examine the NSL/UniTree databases either to investigate problems or produce usage reports.

Since all access methods to DMSS are under our control we have added many logging statements to ftp, rcp, and EARS. These logging statements give very good and up to the minute statistics on system

activity and file transfer traffic.

It is LaRC's intention to give all enhancements and bug fixes to IBM Federal to be incorporated into the NSL/UniTree product. The one modification to NSL/UniTree which LaRC has not passed back to IBM at this point is for lazy tape dismounting. This modification causes the tapesrvr to hold off dismounting a tape for 15 seconds in case another request comes for that tape. This helps to reduce the number of tape mounts needed in a day.

4. PERFORMANCE AND STATISTICS

4.1 CRAY Y-MP PERFORMANCE

This section contains performance comparisons of conventional file transfers via TCP/IP (using RCP) and the third party transfers of whole files between the DMSS's HIPPI/IPI3 disk array systems and the CRAY Y-MP's DD-41 and DD-42 disks (which are not striped).

Single file transfers between the DMSS and CRAY memory follow the performance curve shown in Figure 2. {Figure 2 will show the performance of the disk array in both 1st party and 3rd party modes. It will have performance on the Y-Axis and file size on the X-Axis. The file size will go from 0 to 64 MB}. The third party transfer rate approaches the first party rate of 50MB per second. Read operations from the disk array approach the maximum transfer rate faster than write operations (eg. a 4MB file can be read at 20 MB per second and be written at 18 MB per second).

Single file transfers going from DMSS to CRAY disk follow the performance curve shown in Figure 3. {Figure 3 will show the performance of file transfers using HIPPI and TCP/IP. The Y-Axis will have performance and the X-Axis will be file size.} Note that the performance of the CRAY disk and buffer cache drastically effect the performance. The files are transferred using 4 MB buffers. Increasing the buffer size beyond 4 MB has little effect on performance. For comparison Figure 3 also contains performance figures for CRAY to DMSS performance through RCP over FDDI. The RCP used is the same RCP referred to in section 3.3. This graph illustrates the performance benefits of HIPPI/IPI3 third party versus standard TCP/IP over FDDI. The aggregate performance of CRAY disk to DMSS transfers can approach the maximum of 50 MB per second (per disk array).

From these statistics, we conclude that, by employing a workstation file server and using third party transfer, our system has not suffered from poorer performance, but instead has out performed our previous system. It also validates that the third party kernel driver can be used to process many requests in parallel with a significant increase in performance over conventional FTP and RCP transfers. The better performance has provided an increase in CRAY Y-MP system efficiency and job turn around time.

4.2 MIGRATION/STAGING PERFORMANCE

The system currently is using 3490E tapes with a block mux channel. The implementation of the channel is poor. The best sustained performance observed has been 1.5 - 2.0 MB per second. The channel never performs more than a 2.0 MB per second aggregate rate. Also, when the channel is active it drains about 40 percent of the CPU. We plan to switch to SCSI STK access later this year.

NSL/UniTree only uses one tape drive when migrating data to tape. This limits the amount of data that can be migrated per day. Currently it can handle a maximum of about 50-75 GB per day. After the switch to SCSI this should increase to about 100 GB daily. The migration performance is currently acceptable for LaRC, but may not be for all sites.

At LaRC, NSL/UniTree has written an average of 510 MB per standard length 3490 tape and an average of 1086 MB per double length tape with the STK compression turned on. For uncompressible data NSL/UniTree gets an average of 437 MB per standard length tape.

Statistics for file stages from tape are difficult to interpret since there are many variables that affect performance (ie. file size, number of available tape drives, proximity of tape to drive, tape length, etc). Over a two day period where adequate tape drives were present for all requests the following statistics were collected: 291 files were staged from tape with a 134 second average wait time and a 113 second median wait time. The longest wait time was 482 seconds (500 MB file).

The lazy tape dismounting modification saved 3,793 tape mounts (19 percent of all mounts for read) for the first three months of operation.

Another measurement LaRC uses to gauge system performance is the hit rate on the disk cache. That is the percentage of file retrieval requests which come directly from disk without needing to be staged from tape. We plan to use this information to determine when we need to expand the disk cache. We currently have a 140 GB cache. This is big enough to hold one weeks worth of the most active files. Unfortunately, the DMF to UniTree transition is not complete. This activity severely reduces the cache hit rate. The current average hit rate is about 55 percent. When the transition is over we hope to see this rate rise to 90 percent.

4.3 FILE ACCESSING AND MANIPULATION

NSL/UniTree does a poor job when accessing large numbers of files. This has been the biggest complaint from the user community since switching over to NSL/UniTree. The best long listing performance (masls -l) achieved has been 18 entries listed per second. Short listings (without the -l) achieve almost UNIX speeds. Processes that change file attributes only handle around five to ten entries per second. File removal occurs at only about two per second.

4.4 AVAILABILITY

There are two separate conditions which effect DMSS availability. First, if the server is down then all access to DMSS is denied and jobs which require DMSS will abort. The second condition is when NSL/UniTree is down, but the server is up. In this case service is not denied but delayed. The EARS, FTP, and RCP programs have built-in logic to retry requests to NSL/UniTree when it is down. These requests will block until NSL/UniTree is reinitialized, at which time they will run to completion. This occurs for processes running when NSL/UniTree is brought down and processes started after the shutdown. This is desirable behavior. We can shut down NSL/UniTree for system backups, problem investigation, or software installation without impacting a running job (except for the delay).

The IBM RS 6000/9XXs have proven themselves to be very reliable machines. Since DMSS was in production in October 1994, we have had one AIX crash from which we immediately recovered and no occurrences of hardware problems.

Other DMSS components, which contribute to the DMSS's high availability, are the HIPPI-IPI3 disk array systems and the HIPPI switch. LaRC is pleased with the IBM 9570 disk array systems which have given the reliability anticipated. We have lost a total of four disks (out of 120 disks) in the previous year. When one disk is lost the array is still available. Reconstruction of the lost disk to a standby disk takes between eight and fifteen minutes. The only serious hardware problem occurred when two unexpected power outages occurred in the course of two hours. We lost two disks at the same time during the first outage which resulted in data loss (about 70 files which had not made it to tape). IBM has identified the faulty component which caused the failures and will be updating our drives in the spring. The HIPPI switch has not had a problem in the last year.

The 1994 Christmas and 1995 New Year holidays were the first time at LaRC that the whole central complex shutdown for 6 days without interrupting mass storage service for distributed users.

4.5 USAGE STATISTICS

Overall usage statistics are gathered once per week. The overall usage as of 1/16/95:

There were 1,880,082 files totalling 3.07 TB of space used. The average file size was 1.6 MB and the largest file was 1.9 GB.

Distribution of File Sizes

0KB - 1KB	338,667
1KB - 10KB	612,941
10KB - 100KB	411,110
100KB - 1MB	308,367
1MB - 10MB	105,684
10MB - 100MB	54,403
100MB - bigger	3,903

Weekly Traffic (10/94 - 1/95)

	Files Put	GB's Put	Files Retrieved	GB's Retrieved
High	33,618	81.62	17,831	119.17
Median	17,343	58.49	3,539	50.06
Low	9,885	37.20	1,739	22.59

Current system statistics are available for the following:

- Current Overall System Usage
- Last Weeks Data Traffic
- Daily Report

5. COMPARING DMF TO NSL/UNITREE

This section will make some comparisons between DMF and NSL/UniTree based on our experiences running both in production environments. This paper will not attempt to assign scores to the packages or pick a "winner". Both packages have different strengths and weaknesses.

5.1 MIGRATION/RECALL

A typical migration under DMF involved around four GB's of data and ten 3490 tapes (400MB each). It would finish in about thirty minutes and use all available tape drives. While using all of the drives increased migration speed it effectively interfered with any file recalls (generally for five to ten minutes). There was an average of three to five migrations per day. Using these numbers we can conclude that DMF could handle a maximum of 192 GB's per day. As described earlier NSL/UniTree will handle a maximum of 100 GB per day (with SCSI 3490E tapes). Both systems are capable of handling our current needs. However, if the average amount of incoming data approached or exceeded the maximum amount then the system would be unusable.

DMF wasted tape space because it did not make use of the compression on the STK drives. DMF was told that the 3490 tapes could hold 400 MB's. It never tried to put more than 400 MB's on a tape (even though the data was compressed and there was more room). For the same tapes NSL/UniTree averaged 510 MB's.

We did not collect recall statistics while using DMF.

DMF

- + Able to migrate more data per day.
- All drives used when migrating, thus causing recall delays.

NSL/UniTree

- Only uses one drive during migration.
- + Fully utilizes tapes.

5.2 OPERATIONS

DMF is a more mature product from an operations standpoint. The tools needed to analyze problems, analyze and fix database consistency, and produce usage and accounting statistics are present in DMF and not in NSL/UniTree. LaRC has written some utilities and has used some unsupported utilities from IBM to help fill the void. Both packages provide a tape repacking utility. As with the other operational tools the DMF version was more stable and mature than its NSL/UniTree counterpart.

NSL/UniTree logs are poorly suited for system monitoring and problem analysis. There are too many log files, each with differing formats. The information is cryptic, and in some cases there is too much useless information and in others there is not enough good information. It takes quite a bit of experience

to decode the logs and get an idea of what the system is or was doing.

NSL/UniTree can be brought down for extended periods of time for any reason (backups, problem analysis, etc.) without causing problems for applications needing NSL/UniTree services. Client programs will block and wait for NSL/UniTree to return to service. Thus batch jobs and cron processes will not be effected due to an NSL/UniTree outage. Applications needing mass storage files would abort instead of block if DMF was taken down.

NSL/UniTree lacks code to do an orderly shutdown. Any time it goes down there is a chance for database corruption. We have had one such instance since DMSS went into production.

While NSL/UniTree does not provide backup utilities for its databases (except the tape server) it has proven to be easier and faster to backup. We typically backup the databases once a night by writing them to the disk array. NSL/UniTree is down while this happens. It takes approximately 30 minutes to make the disk copies. Later in the day tape copies are made. Backups under DMF took three hours and during this no access could be made to the system. Backups were only made two times per week because of the system outage time.

NSL/UniTree's algorithm for purging files from the disk cache is not very flexible. Only one control is provided in the configuration files. Basically one can choose to favor size or time. If one chooses to favor time, most files over one day old become equally likely to be purged. If one favors size then a 500 MB file which is 30 minutes old will be purged before a 1 MB file which has not been accessed in five months. DMF allowed for a much more site tunable purging algorithm.

DMF

- + Mature product with better tools and utilities.
- System outages cause denial of service.

NSL/UniTree

- Poor logs.
- + Shutting down system causes applications to block not abort.
- + Faster backups allow for more frequent backups.
- Poor disk cache purging.
- No way to gracefully shutdown NSL/UniTree.

5.3 STABILITY

DMF proved to be a more stable product. It never crashed even when the system was very busy. Only on days of extreme activity (50,000 new files) did we have any problems. We would have to disable access to the system to allow DMF to catch up on migrations. Versions of NSL/UniTree prior to 2.1 were not stable enough for production use at LaRC. Version 2.1 has proven to be stable enough for our

production use. However, it is less stable than DMF. On busy days we see a variety of problems including NSL/UniTree processes crashing and restarting. These restarts generally cause only a delay in service, although they sometimes result in minor database consistency problems. Also when the system is very active (more than 40-50 simultaneous I/O requests) we can get write I/O errors. Users then have to re-transfer the data.

Over the years we have had data loss due to events such as disk crashes and bad tapes, but have had no occurrences of data loss or corruption due to software bugs in either system in production use.

DMF

+ Very stable.

NSL/UniTree

- Less stable.

5.4 PERFORMANCE

DMSS has provided better transfer rates than the CRAY Y-MP based solution. This improvement is seen by all clients from HIPPI attached supercomputers to ethernet attached workstations. The transfer rates have been between two and ten times better depending on the situation.

NSL/UniTree suffers from poor file access speeds (as mentioned in section 4.3). Any applications which access (list, remove, etc.) even a moderate number of files see a performance penalty. DMF does not suffer the same problem. It gives typical UNIX file system response.

NSL/UniTree

+ Better file transfer rates.

- Poor file access performance.

5.5 FUNCTIONALITY

NSL/UniTree provides a number of useful features which DMF does not including dynamic storage hierarchies, file families, and a file by file dual copy mechanism. Dynamic storage hierarchies allow the definition of any number of hierarchies consisting of many different devices. The hierarchy definitions tell NSL/UniTree how to migrate and cache files using the available devices. Each file can be assigned to any defined hierarchy by the user or the application. This feature allows a site and its users to better meet performance and cost requirements.

NSL/UniTree has the concept of file families which allow files to be grouped together on tape. While this feature is nice it only allows 32 families. This limits its usefulness to special requests and projects.

NSL/UniTree, like DMF, allows for multiple tape copies of a file to be written. NSL/UniTree is more flexible by allowing up to 16 copies of a file and allowing the user to specify the number of copies they want for each file.

NSL/UniTree

+ Extra features (especially dynamic storage hierarchies).

6. CONCLUSIONS

LaRC has placed a new cost-effective mass storage solution into production which provides high availability and high performance. The system is also flexible and scalable, allowing LaRC to better match user requirements with the appropriate level of service.

Comparing NSL/UniTree to DMF we find that NSL/UniTree provides greater flexibility and functionality, but has some deficiencies (especially under heavy loads or when accessing large numbers of files). DMF is the more mature product but lacks a number of features that effect availability and scalability.

7. ACKNOWLEDGEMENTS

We would like to thank those people and organizations which helped to make this project a success. In particular we would like to recognize Travis Priest and Tim Starrin of the Computer Sciences Corporation (CSC) for their excellent development work and Ken Evans, Doyle Franklin, and John Kaiser of CSC for their efforts to keep this and previous systems up and running smoothly. Also we wish to thank IBM for providing assistance when needed.

REFERENCES:

1. Pao, J and D. Humes, "NASA Langley Research Center's Distributed Mass Storage System", Proc. of Third NASA Goddard Conference on Mass Storage Systems and Technologies, College Park, October 1993
2. ANSI, "High Performance Parallel Interface - Mechanical, Electrical and Signaling Protocol Specification (HIPPI-PH)", American Standards Institute, X3.183-1991.
3. ANSI, "High Performance Parallel Interface - Framing Protocol(HIPPI-FP) Preliminary Draft", American Standards Institute, X3.210-199x.
4. ANSI, "High Performance Parallel Interface - Physical Switch Control (HIPPI-SC)", American Standards Institute, X3.91-023-1991.
5. ISO/IEC, "Information Technology - Intelligent Peripheral Interface Part 3: Device Generic Command Set for Magnetic and Optical Disk Drives", ISO/IEC 9318-3, September 1990.
6. Coleman, S. and S. Miller, eds., "Mass Storage System Reference Model Version 4", IEEE Technical Committee on Mass Storage Systems and Technology, May 1990.
7. Buck, L., S. Coleman, R. Garrison, D. Issac, eds., "Mass Storage System Reference Model Version 5", IEEE Storage Systems Standards Working Group, September 1994.

8. Hyer, R., R. Ruef, R. Watson, "High-Performance Data Transfer Using Network-Attached Peripherals at the National Storage Laboratory", Proc. Twelfth IEEE Symposium on Mass Storage Systems, Monterey, April 1993.