CONF-9509237--17

THE HIGH PERFORMANCE STORAGE SYSTEM (HPSS)

K. L. KLIEWER

CENTER FOR COMPUTATIONAL SCIENCES
OAK RIDGE NATIONAL LABORATORY
P. O. BOX 2008
OAK RIDGE, TN 37831-6203

Ever more powerful computers and rapidly enlarging data sets require unprecedented levels of data storage and access capabilities. To help meet these requirements, the scalable, network-centered, parallel storage system HPSS was designed and is now being developed. The parallel I/O architecture, mechanisms, strategies and capabilities are described. The current development status and the broad applicability are illustrated through a discussion of the sites at which HPSS is now being implemented, representing a spectrum of computing environments. Planned capabilities and time scales will be provided. Some of the remarkable developments in storage media data density looming on the horizon will also be noted.

1. INTRODUCTION

As is distressingly clear to all of us in the world of high performance computing, I/O is a critical performance bottleneck. This reality has been lurking slightly beneath the surface for some time, but has now become a visible and nearly pervasive barrier to achieving otherwise achievable computational performance levels. While external data storage hardware capabilities, both tape and disk, continue to improve rapidly, far too little attention has been given to machine I/O interfaces and to the generation of parallel software systems for moving data sets rapidly and with ease into external storage environments. The latter is the focus of our attention here.

I might add, almost parenthetically, that the high energy physics community is more aware of the I/O dilemma than most. This is a consequence of the fact that the quantitative realities of the big machines and detectors - collision rates, trigger settings, memory and computational requirements per event, etc. - have compelled detailed analyses of I/O requirements. Accordingly, there are fewer surprises in the future for this community, but that in no way mitigates the burden of being able to deal properly with the coming data blasts!

At this time, data sets into the hundreds of gigabytes (GB) are not unusual, nor are total system storage requirements of 10 or more terabytes (TB). Such environments demand data path bandwidths of hundreds of megabytes (MB) per second. Looking into the near future, all of these numbers will expand by at least an order of magnitude, that is, data sets of TB size with total storage requirements of petabytes (PB) and data rates of more than a GB/sec. Further, the systems must provide superior user interfaces, excellent capabilities in data management, highly efficient use of storage space, and transparent distribution of data into network-connected storage sites with ready and rapid access.

To emphasize the scale of requirements noted above in the context of this meeting, we use data from the Fermi National Accelerator Laboratory (FNAL).[a] With the definition,

$$\text{Year} = \text{Snowmass Year} = 10^7 \text{ seconds},$$

[a]Thanks to Thomas Nash and Joel Butler of FNAL for providing these data.

the current collider run includes roughly 10 events/sec for each detector, D0 and CDF. For D0, characterizing each event in a computer requires about 600 KB and for CDF, 200 KB. The sum of these then requires a data rate of about 8 MB/sec and translates into total basic data of 80 TB/yr. Using the rule of thumb that processing inflates data sets by a factor of 3 or 4, the yearly accumulation is about 300 TB, which, to move, requires a data rate of about 30 MB/sec.

During the fixed target runs from 1996 to 1998, the eight experiments will generate data at a rate that is comparable to that of the total in the collider run.

However, 1999 brings Collider Run 2. The event rate will increase to 100 or even 1000 events/sec/detector. Being an optimist, let me use the latter for illustration. Allowing now 400 KB/event for D0 and 200 for CDF, we must prepare for processed data in the range of 20 PB/yr and continuous data rates of 2 GB/sec. We note also that these data, as well as the earlier data, will be kept for a number of years. This quick survey points up the formidable challenge ahead.

These numbers point unequivocally to the necessity for a high performance storage and access environment. To create a software system capable of providing performance at such levels, the HPSS development consortium was formed. Primary development responsibility resides with four U.S. Department of Energy National Laboratories - Lawrence Livermore National Laboratory (LLNL), Los Alamos National Laboratory (LANL), Oak Ridge National Laboratory (ORNL), and Sandia National Laboratories (SNL) - and IBM Government Systems. Important contributions have also been made by NASA-Langley Research Center, NASA-Lewis Research Center, and Cornell University.

A long list of attributes of HPSS is given in Figure 1; each of these has been given due attention in the development process. Let me emphasize that position on the list is not an indication of priority nor significance!

In addition to the list included in Figure 1, specific performance objectives include:

- Data transfer rates > 1 GB/sec; and
- File sizes > 1 TB.

Objectives further include versatility and scalability such that limits are imposed only through hardware availability rather than through any inherent HPSS software limits. The array of HPSS attributes in Figure 1 points up the comprehensive character of our HPSS effort.

Section 2 includes an illustration of a typical HPSS configuration, a description of the HPSS software environment, and an example of data transfer illustrating the roles of the software components. Some important perspective relating to security and reliability is provided in Section 3. Section 4 includes the chronology of the planned HPSS delivery and associated performance expectations. Section 5 focuses on a development that is, at this point, not an official part of HPSS, but rather a development at ORNL. The objective here is to provide a user-friendly interface for those individuals who wish to structure in some detail the incorporation of their files into the storage environment, while, at the same time, making it possible for those "wishing to leave all to the System Administrator" to do so. Section 6 provides examples of current and future deployment systems and strategies for some sites where early deployments are projected. We conclude with Section 7, in which we provide a brief summary of current storage media capabilities and a glimpse of the developing storage future.

Before beginning the HPSS description, let me emphasize a distinction and an opportunity. "Data Management" involves organizing, storing, and retrieving data based upon information content. "Storage Management" involves the containers - bitfiles,

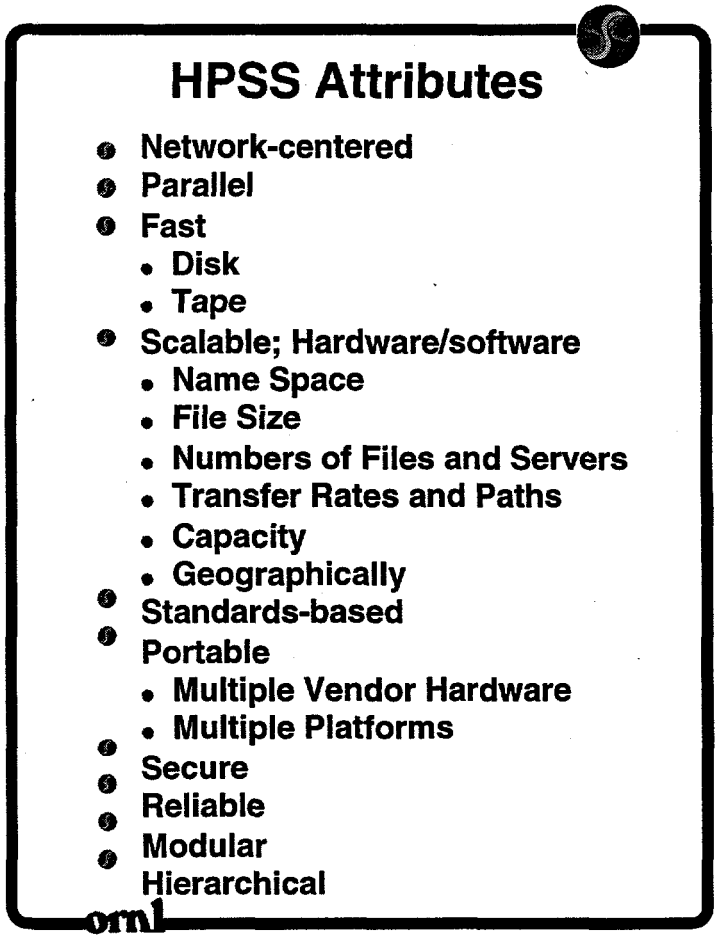

Fig. 1 HPSS features and characteristics, all specifically incorporated into the design and development process.

virtual volumes, physical volumes - into which data are organized. It is the latter that concerns us here. However, effective data management in the future demands a particularly smooth interface between the two, and therein lie problems and opportunities that need quick address.

2. THE HPSS DESIGN AND COMPONENT RESPONSIBILITIES[1,2]

A typical HPSS configuration is shown in Figure 2. This configuration is based on the IEEE Mass Storage Performance Model, version 5[3]. Note the logical separation of control paths and high speed network-attached data flow paths. Data paths, which can be either parallel or sequential, are currently built around HIPPI (TCP/IP sockets and IPI-3), but Fiber Channel (FCS) and ATM will be available in the future (see also Section 6).

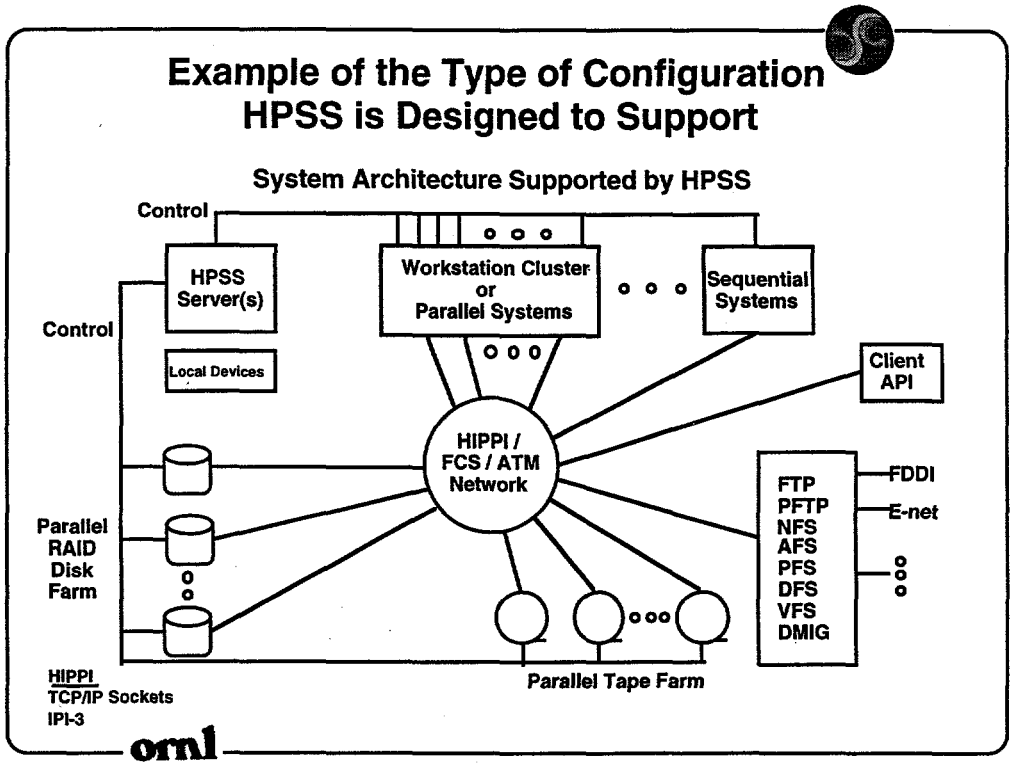

Fig. 2 An example of a configuration for which HPSS has been designed.

System architecture includes high-level access via client APIs, FTP, PFTP, NFS, and PFS. AFS, DFS, VFS, and DMIG interfaces are planned.

Control is handled with OSF DCE Remote Procedure Calls (RPCs). DCE multi-threading expedites HPSS serving large numbers of concurrent users with server multi-processing. The extensive use of transaction-based strategies is managed via Transarc Encina, used also for the important tasks of logging and ensuring the integrity of metadata.

The basic HPSS software architecture is shown in Figure 3. The primary storage entities with which we deal here are files (or bitfiles) that can be as large as 2^{64} bytes. A bitfile is the entity that a user manipulates. A user can read, write, and seek to any point in it. Further, the user can create holes in a bitfile by, for example, seeking beyond the current end-of-file and writing, thereby creating bitfile segments.

A bitfile's first encounter with HPSS will ordinarily be through the *Name Server*. The *Name Server* translates the "user name = human name" to a unique Bitfile ID. Note that the scale here can include billions of entries. The *Bitfile Server* structures and manages bitfile information into headers, including such descriptors as file size, access control information, and logical location. Further, the *Bitfile Server* is responsible for mapping the bitfile segments of a file onto real storage of some type. It does this by using the *Storage Server* to assign the file to one or more storage segments.

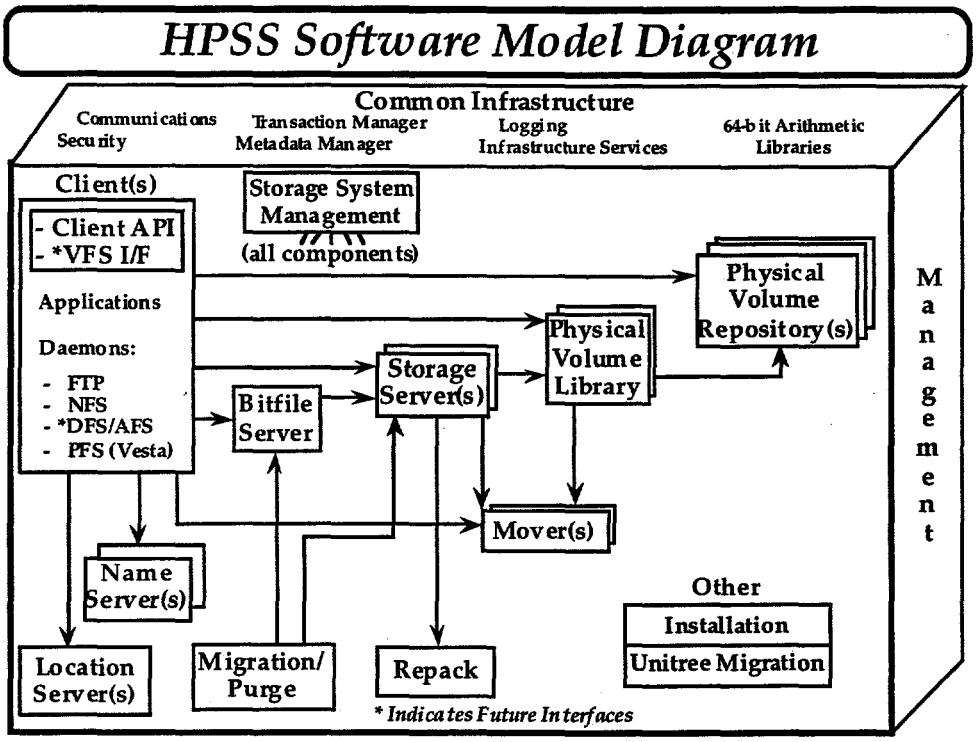

Fig. 3 The HPSS software components showing the overall plan and the modular character.

Here we need an important definition. The storage segment is the basic unit of space in the *Storage Server* which is allocated by the *Bitfile Server*. In other words, the storage segment in the smallest unit of space that can be allocated to satisfy any user request. Defining properly the storage segment size (or sizes) is a vital element in generating a well functioning storage system.

The *Storage Server* provides a hierarchy of storage objects - storage segments, virtual volumes, and physical volumes. The *Storage Server* translates references down this chain to the final peripheral addresses, and is responsible for all storage segment allocations in virtual and physical volumes.

Here we need some more definitions. A virtual volume is a collection of tape cartridges or disks equal in number to the stripe width for that virtual volume. For example, a tape stripe width of four would correspond to a virtual volume of four tapes. A physical volume is then the specific set of tape cartridges or disks, together with locations, where a bitfile resides. To emphasize the point, one virtual volume can spawn many physical volumes.

The *Storage Server* maintains a map that shows which storage segments are used. The *Storage Server* also schedules mounting and de-mounting of removable media

through the *Physical Volume Library*. Note that it is the *Storage Server* and the *Movers* that have primary responsibility for parallel I/O. *Movers*, as might be expected, transfer data from source to sink devices. It is sometimes useful to draw a contrast between devices with geometry (tape, disk) and devices without geometry (network, memory).

The *Physical Volume Library* manages physical volumes which include removable media. The *Physical Volume Library* maintains maps of physical volume to cartridge, cartridge to *Physical Volume Repository* and location in the *Physical Volume Repository*. The *Physical Volume Library* knows all the characteristics of the associated physical volumes. The *Physical Volume Repository* manages all robotic devices and their media.

The *Storage System Manager* monitors and controls activities that have occurred and are occurring in HPSS. In addition, the *Storage System Manager* reports to the system administrator and the system operators status as requested, and also, without request, system breakdowns in a variety of ways. *Migration/Purge* are key elements in the hierarchical storage management in that they maintain free space on the storage media by migrating bitfiles between devices as priority dictates. The principle purpose of disk migration is to free up disk space. *Migration* will select qualified bitfiles and copy them to the next level in the hierarchy following which *Purge* removes the original. *Repack* defragments physical volumes as appropriate.

To provide a somewhat less abstract picture of the roles of some of these components, let us examine the particular example shown in Figure 4, in which a parallel application, interfaced to HPSS through a client API, initiates a 4-node parallel read of a disk file. Associating the numbers here with those in the figure:

1) The parallel application opens the HPSS disk file, tells the *Name Server,* through the Client API code, the file in question, and the client API code receives in return a bitfile ID. Bitfile information then is returned to the application, and includes a complete description of the source (the file on the disks) to which the application then adds the appropriate buffer designations and details for the multi-node sink. All of this information is then relayed to the Client API code.
2) The Client API code then issues a *Bitfile Server* read command for the appropriate file.
3) The *Bitfile Server* translates the high-level metadata into a storage segment description and issues a read command to the *Storage Server*.
4) The *Storage Server* maps the storage segment description into virtual volumes and then physical volumes, followed by a multi-threaded command (one for each physical volume, four in this case) to the *Movers* to read.
5) These mover threads, in parallel and holding all relevant information, establish connection with the waiting (and expecting!) Client Movers and place the data into the Client Movers.
6) The client movers, knowing exactly where the data are to go, complete the process by placing the data into the Client Buffers.

If removable media, such as tape cartridges, had been involved, the *Physical Volume Library* and *Physical Volume Repository* would also have entered the picture.

3. SECURITY AND RELIABILITY/RECOVERY

SECURITY

Security in HPSS has been treated with particular care, as might be expected. Security is provided through DCE with Kerberos v5 authentication.

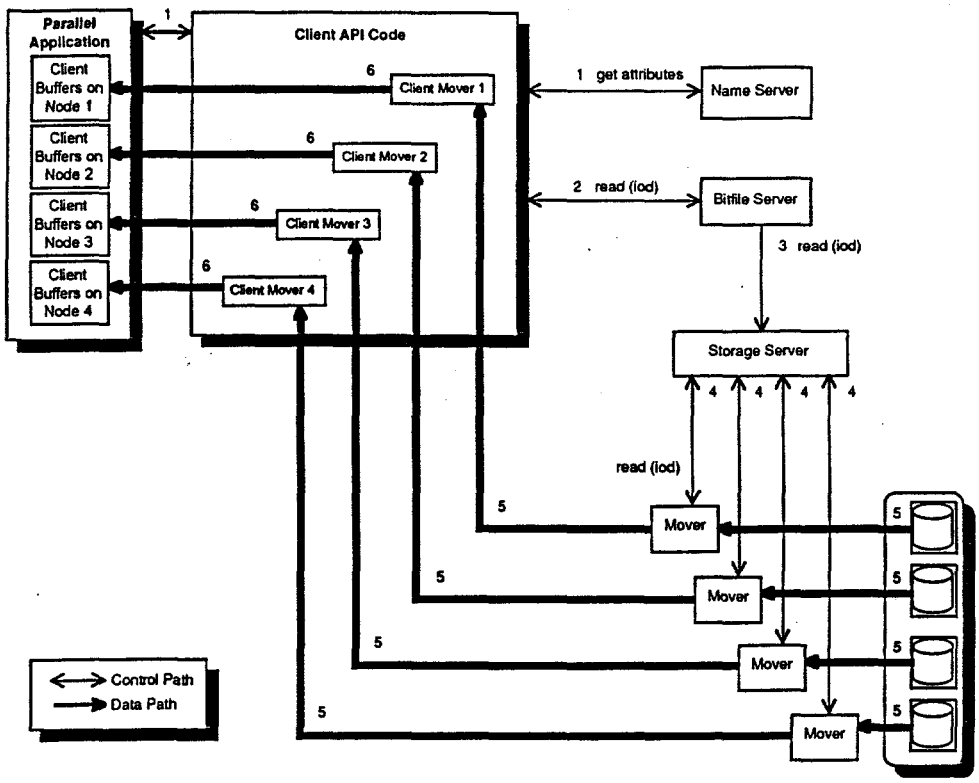

Fig. 4 An example parallel data transfer, effected by an application through an API Client code.

RELIABILITY/RECOVERY

Few things are as worthless as an unreliable storage system without recovery capabilities. Accordingly, reliability/recovery have been carefully woven into the HPSS structure. Client-server processes are structured as atomic transactions to assure continuing system consistency. Assisting in maintaining this consistency, the HPSS Metadata Manager is tightly linked with the Transaction Manager. If an HPSS component fails (e.g., the processor on which it resides fails) the information is relayed immediately to the Storage System Manager so that the system operator can relocate the failed component. Since component responsibilities are registered with the DCE Cell Directory Service, the "new" component will be "found" immediately by those components dependent upon it. To assist in recovery, metadata are mirrored. If bad media are encountered, HPSS can move storage segments from one virtual volume to another. This list, though incomplete, illustrates the care which we have accorded reliability/recovery.

4. HPSS CHRONOLOGY

So, where are we in HPSS development? Release 1, tape only, was available for initial testing late in 1994 and provided data rates of 10s of MB/sec. Release 3[b], currently in test, will be widely available in the second quarter of 1996. This release will provide data rates exceeding 1 GB/sec to both disk and tape. We are, at this time, projecting Release 4 for the second quarter of 1997 with a number of additional features, specifics of which are now being finalized. We anticipate enhanced performance in Release 4.

5. THE USER INTERFACE[c]

Some HPSS users will have no interest in specifying the form in which their files are stored, leaving all such matters to the System Administrator. However, other users will have specific storage expectations and requirements. It is our objective to create an interface that can satisfy, or nearly satisfy, both groups. We here describe a system that, perhaps in modified form, we intend to implement in the ORNL CCS.

In all HPSS environments, the System Administrator will define a Class of Service (COS) set. Basically, the COS set will match the Virtual Volumes in the system. Examples might be (Tape Robot 1, 4-way stripe), (RAID Array 2, 8-way stripe), (Tape Robot 2, 1-way stripe), etc. It is clear that there is a wide range of COS possibilities and they could include features well beyond those given in these examples. The COS system will clearly be site dependent and should be made available to all site HPSS users.

A user, requesting access to HPSS, will then be shown a screen titled COS Hints. A user may simply request the file be stored within a particular COS. The System Administrator priorities satisfied, this would then occur.

However, there will be alternatives. The COS Hints screen will include a number of features. Examples might be Medium Type, Medium Subtype, Block Size, Stripe Width, Stripe Size, Access Frequency, Transfer Rates, and the list could be extended greatly. For each entry on the list, the user can assign a value from 0 to 10; 0 represents "do not consider this," while 10 says "this is mandatory." (A10 that cannot be met means that the storage request would fail.) HPSS will then store the file according to the request, or provide a message indicating why the request cannot be met.

This system will in almost all regards be site dependent, and provides a striking degree of flexibility. Optimum structure of the COS Hints will emerge only through experience and a careful assessment of effective storage system usage from both users and system administrators.

6. EARLY DEPLOYMENT EXAMPLES

Operating experience with HPSS in a range of environments and with a variety of platforms is an essential requirement for bringing HPSS into reliable operation and meeting established expectations. We give here some examples of early deployments and plans.

---

[b]For reasons that we will not get into here, there is no Release 2.

[c]The material in this Section is not currently part of HPSS. Our view is that an interface much like that given here is essential. The particular one we now describe has emerged from discussions with Michael Gleicher of the ORNL CCS.

SANDIA NATIONAL LABORATORY - ALBUQUERQUE (SNL-A)

SNL-A will move aggressively to use HPSS as the primary storage software for its 1840-node Intel Paragon. Noteworthy is the SNL-A decision to use ATM connectivity to move files into both tape (IBM 3494 Tape Library with eight 3590 tape drives) and disks (IBM 7135 disk array). Additional ATM connectivity to FDDI networks and other computing platforms will also be provided.

LAWRENCE LIVERMORE NATIONAL LABORATORY (LLNL)

LLNL includes several facilities in which HPSS will have a major role.

Scalable I/O Facility (SIOF)

The SIOF provides an environment for testing I/O systems and strategies. The software architecture includes a parallel I/O API based on MPI-IO layered over HPSS. FCS provides parallel connectivity from a Meiko CS-2 to an array of tape and disk controllers.

Livermore Computing (LC)

LC handles the production computing for LLNL. HIPPI and FCS will provide parallel storage connectivity for the Meiko into network-attached arrays of both disks and tape. In addition, storage requirements of workstation clusters and other LANS will be served by HPSS.

National Energy Research Supercomputer Center (NERSC)

NERSC has a large user group distributed across the country. Projected storage requirements reach the PB range in 1997. Storage software currently includes the Common File System (CFS) and NSL-Unitree. Migration to HPSS is projected for 1997.

The current hardware configuration includes tape and disk arrays connected through HYPERchannel for CFS and HIPPI for NSL-Unitree. Cray computers, including a C-90, provide the computing power. The transition to HPSS will include connectivity provided through HIPPI or FCS. NERSC is also projecting the addition of a powerful MPP machine on the same time scale to complement the C-90.

CORNELL THEORY CENTER (CTC)

The CTC is moving aggressively to bring HPSS into production or near-production status. The principal file source will be the 512-node IBM SP2, with 80 GB of memory, 1.2 TB of disk, and rated peak computing capacity of 136 gigaFLOPS. Network connectivity will include all of HIPPI, ATM, HPS, and FIDDI. The AFS shared file system and PIOFS parallel file system are used.

CTC is projecting storage system migration from NSL-Unitree to HPSS in 1996, and will also pursue test-bed activities, an example being studies of AIMNET.

MAUI HIGH-PERFORMANCE COMPUTING CENTER (MHPCC)

Another major IBM Center, the MHPCC SP2 includes 400 nodes, 56 GB of memory, 784 GB of disk, and an additional 125 GB of NFS storage. A dedicated Essential HIPPI switch is available for HPSS testing and 10 HIPPI nodes are incorporated into the SP2. The hardware includes IBM 3490 E and NTP tape drives and Maximum Strategy GEN-5 disks.

NSL-Unitree is the current production software, with HPSS testing now under way.

CENTER FOR COMPUTATIONAL SCIENCES (CCS) AT ORNL

Our CCS computing environment includes a spectrum of machines (Intel Paragons: 66 GP-node XP/S 5, 512 GP-node XP/S 35, and 1024 MP3-node XP/S 150; a 16-node IBM SP2; and a KSR1-64) and a storage complement including a 100 TB IBM 3495 tape robot, soon to have eight NTP drives, 216 GB of Maximum Strategy disks, a 14.2 TB Storage Tek Silo (four Timberline and 2 Redwood drives with Powerhorn robotics), and a CREO optical tape system. While awaiting our fully configured ESCON-attached 3495 library, we are using a SCSI-attached 3494 library with four NTP drives for HPSS testing. The current production storage system is NSL-Unitree. At this point, HIPPI and ESCON switches provide network connectivity.

Storage systems, features, and development are a principal focus in the CCS, examples being the creation of the UTI interface for NSL-Unitree and the expansive storage system just described. Accordingly, we have established a test-bed environment, separate from our production environment, in which HPSS can be tested, even stressed. Tests of HPSS Release 1 utilized the Intel Paragon XP/S 5 and HIPPI/ESCON connections to both IBM and Storage Tek libraries, the latter configured with six parallel paths. For Release 3 tests, the system will be extended to include four parallel HIPPI paths into the Maximum Strategy RAID array and eight ESCON paths into the IBM 3495 library.

Our test objectives include scalable parallel file transfers and comprehensive feature tests to ensure that the anticipated Release 3 capabilities exist and to assist in establishing features and capabilities for Release 4.

This description of emerging HPSS deployments is certainly not complete; the objective here has been to demonstrate the variety of platforms into which HPSS is moving. HPSS is also being deployed in distributed environments at LANL, the University of Washington, and, of particular interest here, Fermilab. We encourage those interested in HPSS to explore configurations and performance in detail with staff members at all early deployment sites.

7. THE STORAGE FUTURE

We see HPSS as ensuring the availability of a storage software system capable of meeting the rapidly expanding high-end storage needs for an extended period. As we noted earlier, coordinated efforts to assure seamless merger of innovative developing file and database structures with the storage environment are essential.

But what about the hardware? Current magnetic tape performance includes data densities up to 50 megabits per square inch (Mb/in^2) and read/write speeds as rapid as 400 Mb/sec. For disks, data densities, both optical and magnetic, reach the 600 Mb/in^2 regime. Magnetic disks provide data rates up to 100 Mb/sec while current optical disk rates are smaller by a factor of 30 or so.

The near future should see these density numbers move into the ten Gb/in^2 territory without a major change in the basic technology. The exploitation of magnetoresistance[4] and giant magnetoresistance[5] systems points in this direction.

However, the drama in storage will probably come from distinctly different directions. Scanning tunneling and scanning force microscopes, magneto-optical systems, and near-field optical microscopes give promise of density increases of several or even many orders of magnitude.

I will not elaborate on these systems here, because Bruce Terris does so in these Proceedings. However, there is one system which he has not mentioned that I will note briefly.

Stutz and Lamartine[6] of LANL have developed a desktop-size system which uses a gadolinium ion beam to write on small cylinders. Cylinders of both stainless steel and iridium have been used. Data densities exceeding 100 Gb/in^2 on "nail-size" cylinders have been attained with write speeds of 0.7 Mb/sec and read speeds of 1 Mb/sec.

It is clear that the storage systems of the future will be very different from those of today!

8. MORE INFORMATION

Extensive information about HPSS is available over the World Wide Web - http://www.ccs.ornl.gov/hpss.

ACKNOWLEDGMENTS

Many thanks to Michael Gleicher of the CCS, Randall Burris of Lockheed Martin Energy Systems, and Richard Watson of LLNL for discussions that I, at least, found both interesting and informative. Thanks also to R. Michael Cahoon of SNL, Douglas Carlson of Cornell, John Sobolewski of the MHPCC and Richard Watson of LLNL for information and transparencies relating to HPSS deployments at their sites.

Principal support for the development of HPSS has come from the Department of Energy, Defense Programs, through Cooperative Research and Development Agreements (CRADAS) between the DOE Laboratories and IBM Government Systems. Additional support for HPSS has been provided at ORNL through the DOE Office of Computational and Technology Research (OCTR), under contract DE-AC05-84OR21400.

REFERENCES

1. D. Teaff, R. Watson, and R. Coyne, "The Architecture of the High Performance Storage System (HPSS)," Proceedings of the Goddard Conference on Mass Storage and Technologies, College Park, MD, March 1995.
2. R. W. Watson and R. A. Coyne, "The Parallel I/O Architecture of the High Performance Storage System (HPSS),"Proc. Fourteenth IEEE Symposium on Mass Storage Systems, Monterey, CA, September 1995, pp. 27-44.
3. IEEE Storage System Standards Working Group (SSSWG) (Project 1244),"Reference model for Open Storage Systems Interconnection, Mass Storage Reference Model Version 5," September 1994. Available from the IEEE SSSWG Technical Editor Richard Garrison, Martin Marietta [(215) 532-6746].
4. C. Tsang, M.-M. Chen, and T. Yogi, "Gigabit-Density Magnetic Recording," Proc. IEEE **81** 1344 (1993).
5. W. H. Butler, X.-G. Zhang, and D. M. C. Nicholson, "Spin Dependent Scattering and Giant Magnetoresistance," J. Mag. Magn. Mat (to be published).
6. R. Stutz and B. Lamartine, private communication. Further information is available from Bigbear321@aol.com.

DISCLAIMER