Fermilab's Multi-Petabyte Scalable Mass Storage System

Gene Oleynik, Bonnie Alcorn, Wayne Baisley, Jon Bakken, David Berg, Eileen Berman, Chih-Hao Huang, Terry Jones, Robert D. Kennedy, Alexander Kulyavtsev, Alexander Moibenko, Timur Perelmutov, Don Petravick, Vladimir Podstavkov, George Szmuksta, Michael Zalokar
Fermi National Accelerator Laboratory
{oleynik, alcorn, baisley, bakken, berg, berman, huangch, jonest, kennedy, aik, moibenko, timur, petravick, podstvkv, georges, zalokar}@fnal.gov

Abstract

Fermilab provides a multi-Petabyte scale mass storage system for High Energy Physics (HEP) Experiments and other scientific endeavors. We describe the scalability aspects of the hardware and software architecture that were designed into the Mass Storage System to permit us to scale to multiple petabytes of storage capacity, manage tens of terabytes per day in data transfers, support hundreds of users, and maintain data integrity. We discuss in detail how we scale the system over time to meet the ever-increasing needs of the scientific community, and relate our experiences with many of the technical and economic issues related to scaling the system. Since the 2003 MSST conference, the experiments at Fermilab have generated more than 1.9 PB of additional data. We present results on how this system has scaled and performed for the Fermilab CDF and D0 Run II experiments as well as other HEP experiments and scientific endeavors.

1. Introduction

Fermilab has developed a mass storage architecture and software suitable for the demands of high-capacity and high-throughput scientific endeavors, specifically for Run II of the CDF and D0 experiments that are currently in progress at the Fermilab Tevatron collider. These experiments record their data 24x7 to the Fermilab mass storage system, perform analysis of this data, and write the results back into the system all in real-time. In this paper, we will focus on the scalability aspects of the mass storage system and its performance.

.

The Fermilab mass storage architecture [1] consists of a data storage system, called "Enstore" that provides access to any number of automated tape libraries, and a disk caching front end to this, called "dCache", which permits transparent pre-staging of files from the libraries and efficient access to files by multiple users.

Enstore [2] is the mass storage system implemented at Fermilab as the primary data store for experiments' large data sets. Enstore provides distributed access to data on tape to both local and remote users. Enstore is designed to provide a high degree of fault tolerance and availability as well as easy administration and monitoring. It uses a client-server architecture that provides a generic interface for users and allows for hardware and software components that can be replaced and/or expanded dynamically.

File read/write requests to the mass storage system can also go through dCache [3], a disk caching system that uses Enstore as a permanent store. DCache decouples the (potentially slow) network transfer from the (fast) storage media I/O in order to keep the Enstore system from bogging down. Data exchanges between the dCache and Enstore are performed automatically and are transparent to the users.

The dCache project is a joint DESY[1]-Fermilab effort to overcome the accessibility limitations posed by the types of mass storage software and devices found at HEP labs. The dCache optimizes the location of staged copies and makes more efficient use of expensive tape drives and automated tape libraries. In addition, the dCache provides a uniform view of the storage repository, hiding the physical location of the file data (disk-cached or tape-only). The dCache provides several interfaces for off-site grid-based access to data in the Fermilab mass storage system: ftp, GridFTP, SRM and the dCache "dccp" interface for on-site access.

Mass storage is provided through a number of automated tape libraries and a disk cache. Currently, Fermilab has six Storage Tek 9310 libraries and one ADIC AML-2 library with a total potential capacity of around 8PB. These libraries are configured with over 100

[1] DESY – Deutsches Elektronen Synchrotron HEP research facility in Hamburg Germany

Proceedings of the 22nd IEEE/13th NASA Goddard Conference on Mass Storage Systems and Technologies (MSST 2005)
0-7695-2318-8/05 $20.00 © 2005 IEEE

tape drives consisting of 9940, 9940B, LTO, LTO2, and DLT devices. At the moment 180 TB of disk cache is provided, mainly for the CDF experiment. These libraries and drives are divided into three logical mass storage systems: CDF, D0, and CMS/General Use. The CDF system has two 9310 libraries, the D0 system two 9310 and the AML-2, and the General/CMS system two 9310 and one of the AML-2 quadratowers.

2. Architecture and scalability

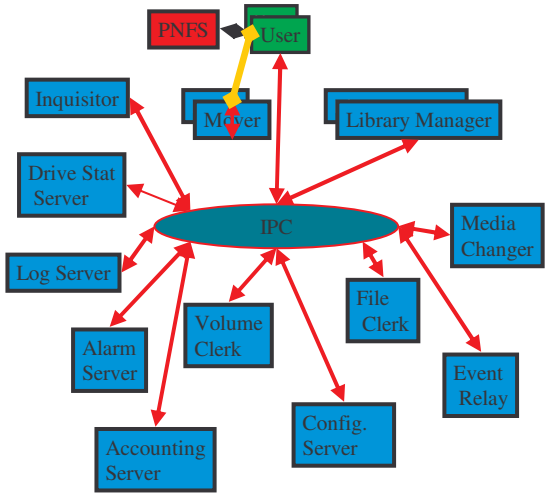

Figure 1. Enstore Architecture

The number of user computers is not restricted by Enstore or dCache, and enstore's components can be distributed over an unlimited number of computers, tape libraries and tape drives. Enstore and dCache are scaled by building the equipment out. Two things are required to make this possible: the decoupling of the user interface from the underlying technologies and a scalable design of the software architecture. Software has not been a factor in scaling the system to higher capacity, data rates, and users, with one exception in dCache which is discussed in the section titled 'DCache Scalability'.

Hardware consists of vendor supplied tape libraries and drives. The enstore and dCache systems run on commodity PCs using the Linux operating system. Critical disks (servers with databases or dcache nodes) are hardware RAID-5 or RAID-10. Enstore and dCache PCs are connected by Gigabit Ethernet to each other and typically to end-user machines. The systems span across three floors at Fermilab's Feynman Computing Center.

Enstore and dCache provide file based interfaces to end-users. Enstore users access their data through the encp interface which deals only with files and is independent of the underlying storage technologies. The underlying storage system can therefore be expanded and new storage technologies can be introduced without affecting the end user. Similarly, dCache users access their data through the dccp interface or the ftp, GridFTP, or SRM interfaces, all of which only expose the file interface and thus are independent of the storage system.

The Enstore software architecture is presented in Figure 1. Enstore is designed using a client-server architecture and provides a generic interface for users. Enstore components are:

- A configuration server keeps the system configuration information and provides it to the rest of the system. Configuration is described in an easily maintainable configuration file.
- A volume clerk maintains the volume database and is responsible for declaration of new volumes, assignments of volumes, user quotas, and volume bookkeeping.
- A file clerk maintains the file database, assigns unique bit file IDs and keeps all necessary information about files written into Enstore.
- Multiple distributed library managers provide queuing, optimization, and distribution of user requests to assigned movers.
- Movers write / read user data to tapes. A mover can be assigned to more than one library manager.
- A media changer mounts / dismounts tapes in the tape drives at the request of a mover.
- Alarm and log servers generate alarms and log messages from Enstore components correspondingly.
- An accounting server maintains an accounting database containing information about completed and failed transfers and mounts.
- A drivestat server maintains a database with information about tape drives and their usage.
- An inquisitor monitors the state of the Enstore components.
- PNFS namespace server implements a name space that externally looks like a set of Network File Systems.
- Events are used in the Enstore system to inform its components about changes in the configuration, completed and ongoing transfers, states of the servers, etc. An event relay transmits these events to its subscribers.

All Enstore components communicate using inter-process communications based on UDP. Great care has

been taken to provide reliable communications under extreme load conditions. The user command, encp, retries in case of an internal Enstore error.

Enstore supports automated and manual storage libraries, which allows for a larger number of tapes than slots in the robotic storage. The user command interface program encp has a syntax similar to the Unix cp command with some additional options, allowing users to specify request processing parameters such as priority, and number of retries, whether to calculate a CRC, and to set a tape dismount time, etc.

Data stored in Enstore are grouped based on the storage group unique to each experiment, and file families inside of the storage group. The storage group is assigned by the storage system administrator while the file family is assigned by the user. Files in the same file family are written to the same set of tapes moderated by a file family width. The file family width controls the amount of simultaneous write transfers for a certain file family.

Enstore allows users to specify the priorities for data transfer. There are 2 kinds of priorities: regular and administrative or Data Acquisition (DAQ) Priority. The library manager will dispatch DAQ priority requests ahead of any regular priority requests. In this case the mover may dismount the currently mounted tape and mount the DAQ one. Priorities can be assigned to requests according to configuration parameters in the Enstore configuration file.

Another important feature of Enstore is its capability to specify the number of tape drives dedicated to an experiment (storage group). Experiments have separate budgets. Some of them may have their own tape drives installed in the general purpose robotic library. These drives are in the common pool but the experiment will preferentially be given access to the number of drives equivalent to its contribution. This amount is normally statically specified in the configuration file by the system administrators, and is used while processing the request queue.

DCache consists of

- Administrative Nodes which setup user requests to the read and write pool nodes
- Monitoring Node which monitors the other dCache nodes
- Namespace (PNFS) Server
- Read pool nodes from which files can be spooled to the user from Enstore
- Write pool nodes from which data can be written to Enstore

Of particular concern for scalability are the Movers, Library Managers, Media Changers, Namespace servers, and dCache Administrative and pool nodes. In general, all of these servers scale by replication. Their scalability is discussed below.

2.1. Movers and Media Changers

The knowledge of the underlying storage technologies is isolated to the media changers and the movers. The movers move data between the user and libraries and communicate with the media changers which instruct the libraries to fetch and store tapes as needed.

Movers transfer data between tape drives and the end-user over high speed network connections. Movers isolate the knowledge of the underlying tape technology and generally communicate with drives through a SCSI interface. In Enstore, 1 or 2 tape drives are managed by a single mover node which is a PC running Linux. System throughput is increased by adding tape drives and Mover nodes to perform the transfer function.

Media changers need to accommodate the number of user mount and dismount requests. Since automated tape libraries typically are limited to hundreds of exchanges per hour, this component of the system does not need to scale, but does provide the important function of isolating knowledge of the interfaces to the automated tape libraries.

2.2. Namespace and file/volume clerks

Enstore and dCache use a file system namespace called PNFS [4]. The users access files through PNFS names. A PNFS database maintains meta-data about these files names, and this information, in conjunction with file and volume databases, can be used to identify the location and size of a file in the mass storage system given its name. PNFS does not maintain the actual data, but rather the meta-data for Enstore and dCache to find the actual data.

The PNFS database is accessible using the NFS protocol. This means that users access PNFS via NFS file systems mounted on their nodes. A separate node is usually used to run a PNFS server. PostgreSQL [5] is used as the database management system for the name space. The size of the database can be up to 2^{64} bytes, therefore PNFS scalability is only limited by the number and the rate of the PNFS requests. To cope with the scaling of name space request load, we split the namespace and set

up additional PNFS server machines as needed. This scales just as an NFS file system would scale.

Scalability of the file and volume database is limited only by the scalability of the PostgreSQL DBMS in Enstore. The goal of the scalability in this case is to avoid overloading the system with requests. Scalability is achieved by adding databases and corresponding Enstore servers along with machines on which these servers run.

2.3. Library Managers

Library managers are responsible for processing user data transfer requests and therefore need to scale with the number of user requests for data. They can process thousands of queued requests and so far have not reached a saturation point, but new library managers can be added as needed.

2.4. DCache scalability

From the file access perspective, dCache is scaled by adding read or write pool nodes. DCache scales in simultaneous access by automatically duplicating frequently accessed files across pool nodes to spread the load.

The Administrative dCache nodes setup user requests to the pools. DCache can be scaled up to handle high rates of user requests by adding Administrative nodes.

The only software limitation to scalability encountered so far has been with PNFS on very large and highly utilized dCache systems due to the overhead of accessing the namespace server over the network for every file access. A new version of dCache removes this bottleneck by allowing local namespace resolution.

3. File and Metadata Integrity

At the current time, the Fermilab mass storage system contains on the order of 25000 tapes and monitors and maintains meta-data on these volumes and the files they contain. On this and ever growing scales, maintaining and monitoring the integrity of the files and meta-data is a great concern.

3.1. Maintaining File Integrity

The Enstore system takes the following steps to monitor and maintain the integrity of the files on the tape and in the data cache:

1. Extensive CRC checking at all stages of getting the files on and off tape and in and out of dCache.
2. A flexible policy for read after write CRC checking for tapes
3. Automated read checking of the CRC of randomly selected files on tapes
4. Automated CRC checking of dCache files
5. Automatically disabling access to tapes or tape drives that exhibit read or write problems
6. Cloning overused or problematic tapes

CRC checks are performed on the file transfers at all stages. A CRC is generated when the data is sent over the network to Enstore to be written on tape, and recalculated and compared when it is received by enstore. The original CRC is maintained in the metadata and is recalculated and compared at every stage of file movement, including in and out of dCache pool nodes. When a file is read from tape or a dCache pool, its CRC is calculated and checked against the one kept in the metadata, and the CRC is again checked when transferred to the user. If a dcache file with a bad CRC is alarmed, it is usually removed by an administrator. – future references will then get a fresh copy from tape.

The movers can be configured to read the first file written on a volume after it was mounted or not, and/or to check randomly selected files at a configurable frequency. For example, one experiment is using both approaches.

The Enstore system also automatically randomly selects files, reads them, calculates their CRC, and compares them to the files' metadata CRC. dCache also periodically checks the CRC of files that it has stored. This is the only part of data integrity checking that may be affected by the system scale. As the system grows, the frequency of these random checks can be increased.

CRC failures are alarmed and monitored by the administrative staff.

Enstore identifies two classes of errors: recoverable and unrecoverable. If enstore encounters an error executing some action, and if the error is identified as recoverable, it will retry the action until it succeeds or a configurable limit in retries is reached, in which case it reports the error as unrecoverable. Unrecoverable errors related to tape operations result in the tape being placed into the NOACCESS state. In addition, any time a write error occurs when writing to a tape, the tape is placed in the "READ-ONLY" state. If a configurable number of write errors or read errors occur in a row on the same tape

drive, the drive's mover process is automatically placed in the offline state and the drive cannot be used by end-users without administrative intervention. In addition, if a tape drive operation returns an unrecoverable error, or there is a mount or dismount failure, the drive's mover process is placed in the offline state and requires administrative intervention.

Tapes in the "read-only" or "noccess" state are alarmed, are checked by an administrator, and either are made available again, cloned, or sent to the vendor for recovery. A tape drive mover process placed offline because of errors is alarmed and an administrator runs tests on the drive. The tape drive is either put back online or a service call is placed to the drive vendor.

Finally, once a volume passes a mount count threshold, or there are problems reading files off the tape, the tape is cloned (copied) to a new tape and the old tape is discarded. Cloning is done manually by administrative staff and uses tape drive resources for long periods. Automated cloning is under consideration.

3.2. Maintaining Metadata Integrity

Enstore metadata integrity is very critical and much detail has been paid to assuring its integrity and reproducibility in case of a catastrophic failure. Steps include:

1. redundancy
2. automated checks
3. backups
4. replicated (hardware) databases

One copy of the Enstore system metadata is kept in the PNFS database and the other in the Enstore file database. The data in one can be recreated from the other. In addition, Enstore compares the metadata in both of these databases on each file transfer, and reports discrepancies to Enstore administration and to the user. Enstore also periodically scans the metadata in the databases to ensure their integrity.

Enstore uses features of PostgreSQL to perform routine database backups. Full backups are normally performed once per hour, and each backup is kept online for a two day period. One backup is copied to tape daily. In addition, the backups are monitored to ensure that these critical jobs are running successfully.

Inventories are performed once per hour on the live databases to provide statistical information. Internal database corruption, if any, is likely to be found in this process. If corruption is detected, the database is fixed using backups and the journals.

Enstore also keeps a transaction journal (independent of PostgreSQL). In a catastrophic database failure, this journal, along with the most recent good backup, can be used to reconstruct the database to its last consistent state.

Database backup time is a concern as the amount of data in the Fermilab mass storage system scales up. Some of the experiment databases are getting large enough that the backups are taking many hours to complete. If a catastrophic database failure occurs before a backup completes, it would take a significant amount of time to restore the database from the previous backup and journal. For this reason, we have started implementing replicated database servers, using PostgreSQL features, so that we have a hot spare database to switch to in the event of a database failure.

4. Scaling through Migration

Another way Fermilab scales capacity is by migrating to newer technologies with denser media. The capacity of a tape library is thereby increased by using fewer slots for the same amount of data.

The CDF experiment completed a program of migrating 4557 tapes written with 60GB 9940A drives to 200GB 9940B tapes at the end of 2004. This migration increased the capacity of their tape library by a factor of 3, and the media, in this case, was reusable. This migration took place over one year during several periods when the CDF tape library was made available for this purpose, with 3 staff working on it part time, and using 1 9940B and several 9940A drives. Out of the 4557 9940A tapes that were migrated, 42 had problems reading files. Scripts specific to CDF performed the migration and validation. Meta-data for the files had to be swapped manually as the last step of the migration. Since the CDF migration, validation, pnfs duplication, and meta-data swapping are now incorporated in enstore (see below).

The migration of 1240 Eagle tapes to 9940B tapes was just completed on the general use mass storage system, freeing up over 1000 needed slots. This effort used the recently developed auto-migration feature of enstore, and took around two months to complete. On the best day, 46 volumes were migrated. Less than 20 problem tapes were encountered, most of which had read errors. All but one was copied successfully on a second pass. One bad file was recovered by reading several times with dd. The legacy Eagle tapes are sold on the used market.

The core of the auto-migration feature is a process that migrates files. To migrate a file, a copy is made on a different volume, then its meta-data is swapped so that the same /pnfs entry points to the new copy. The file is read back in the same way as a user would do. This completes a migration cycle. The file is always available to the end user during migration.

This migration process can be 99% automatic. A script recommends the volumes to migrate, a person initiates the migration process, and then when migration completes, all files on the volumes are properly migrated and verified. The reason it is partly manual is to place a person in the loop to catch errors early. We are currently investigating further automation of such migrations and using the migration tools to clone problematic or overused tapes.

5. Administration and Maintenance

Because of its size and due to the diversity of technology, the distributed nature of the Fermilab mass storage system, and the fact it must operate efficiently 24x7, monitoring must be diligent. The system is maintained by a staff of 4 administrators, 3 Enstore developers, and 3.5 dCache developers. This staff currently has the following responsibilities:

- Monitors and maintains 7 tape libraries with approximately 100 tape drives, 150 movers, file servers, and server PC nodes
- Recycles volumes
- Monitors capacity vs. use
- Clones overused tapes
- Troubleshoots problems
- Installs new hardware and software

In order to administratively deal with the volume of problems and information, the mass storage system has the following administrative facilities:

1. 24x7 vendor support on tape libraries and tape drives.
2. 24x7 on-call support by primary and secondary on-call administrative Fermilab staff.
3. Automatic generation of alarms by Enstore and dCache software integrated in with the Fermilab helpdesk system to generate tickets and page administrators.
4. In-house on-hour helpdesk and off hour call centers to respond to end users and generate pages and tickets.
5. Extensive web based logs, plots, status, statistics and metrics, alarms, and system information.
6. An animated graphical view of Enstore resources and data flows.

Enstore and dCache monitors a large set of hardware and software components:

- States of the Enstore servers
- Amount of Enstore resources such as tape quotas, number of working movers
- User request queues
- Plots of data movement, throughput, and tape mounts
- Volume information
- Generated alarms
- Completed transfers
- Computer uptime and accessibility
- Resource usage (memory, CPU, disk space)
- Status and space utilization on dCache systems

The monitored information is available on the web in a set of static and dynamic web pages published on the Enstore web site at:

http://www-isd.fnal.gov/enstore/

Most of the plots in this paper were automatically produced by Enstore and displayed on this web interface. Enstore monitoring is described in more detail in [3].

The metrics in the table below depict a weeks administration work (chosen at random, slower than typical)

Table 1. Weekly Administration metrics

Item	Occurrences
9940B drives replaced	1
9940A drives replaced	1
LTO1&2 drives replaced	1
Installs	1
Replaced server nodes	0
Replaced mover nodes	0
Replaced file servers	0
Tape library maintenance	0
Server/Mover Maintenance	0
Mover interventions	3
Server Interventions	4
Tape Drive Interventions	2
Fileserver Interventions	2
Tape interventions	3
File interventions	4
Tapes clobbered/recycled	0
Tapes labeled/entered	40
Tapes cloned	2
Enstore service requests	3
Raid disk replacements	0
Off hour calls	0
Data Integrity Issues	1

The volume of information can be quite overwhelming and is continually being streamlined in order to scale with the system.

Many maintenance operations, including maintenance requiring the library to be shut down for brief periods, can be performed without interrupting the user's work flow (besides delaying processing). In these cases, an administrator pauses the enstore queues prior to the maintenance, and then resumes them after it is completed. Additional user's requests queue up during the maintenance period.

Our experience with PCs and Linux has been very good. We have typically incurred a several hardware failures each year on the PCs. For server nodes this has resulted in some brief system downtimes and for mover nodes short unavailability of the tape drives. Raid array disk failures are fairly infrequent, and replacement usually does not affect system availability

6. Performance and Cost

Since reported on in the 2003 MSST conference, the Fermilab mass storage systems have been built out with an additional

- 2 Storage Tek 9310 tape libraries (~10000 slots) to a total of 6 9310 libraries (~ 30000 slots)
- Activation of second ADIC/AML-2 arm and a second ADIC/AML-2 quadratower (~ 4300 slots) for a total of about 9000 slots.
- 37 9940B tape drives for a total of 49
- 14 LTO2 tape drives for a total of 14
- 8 DLT
- About 50 PC nodes for a total of > 150
- 80 TB of disk cache

In these 2 years the data stored in Fermilab's mass store has grown from 600 TB to 2.5 PB, and is continuously growing (see Figures 2-4). 10-20 TB of data are transferred daily. The aggregate mount rate in the libraries is more than 3000 mounts/day. The system now maintains about 25000 volumes of data.

To achieve this level of performance, the system is configured with more than 100 9940, 9940B, LTO1, LT02, and DLT tape drives. The total number of dCache pool nodes is more than 75.

Enstore is implemented with more than 125 PCs running and dCache more than 80 PCs. Costs of the system are mitigated by using commodity PC systems running the Linux operating system.

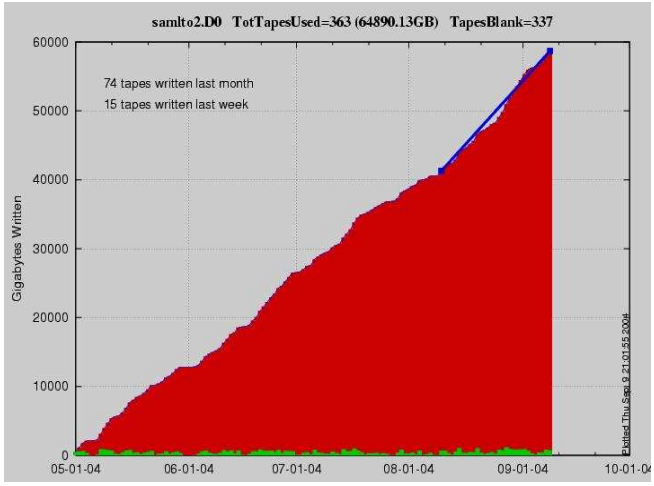

Figure 2. D0 LTO2 tape consumption rate (top is 60TB)

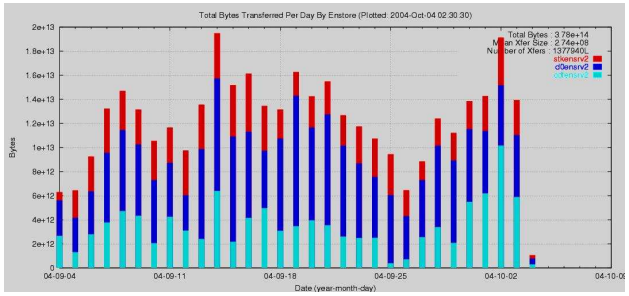

Figure 3. Aggregate System Transfer Rate (Top is 20 TB)

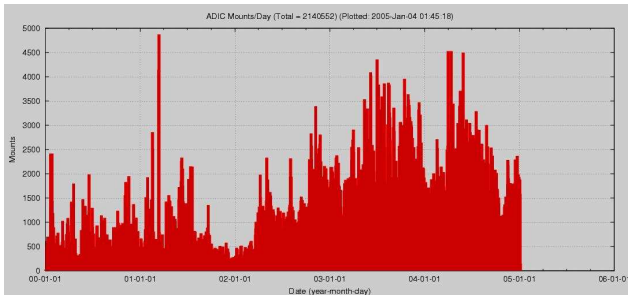

Figure 4. D0 Aggregate mounts/day (Top is 5000)

7. Conclusions

Fermilab has constructed a mass storage system that is scalable beyond the PB level. Over the past two years, the system has been scaled up to handle over 1.9 PB of additional data from its primary users, the run II experiments CDF and D0.

The system can support different underlying storage technologies and scales by building these technologies out or adding/migrating to new technologies. The costs of build-out are mitigated by using commodity PCs and public domain operating systems.

Currently the system supports 2.5 PB with 10-20TB/day in transfers and more than 3000 tape exchanges/day. Over the next year, we expect to see over 1 PB of additional data written to the mass storage system, and the CDF experiment is expected to double their rate of RAW data to tape. The current system is expected to be able to absorb this with no difficulty. For the longer term, we are considering newer commodity tape technologies and newer automated libraries as may be required by the Fermilab program.

8. References

[1] A. Moibenko, "The Fermilab Data Storage Infrastructure"Proceedings of the 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies, Fermilab, April 2003, p. 101-104.

[2] J. Bakken et al. Enstore Technical Design Document, http://www-isd.fnal.gov/Enstore/design.html

[3] M. Ernst et al. DCache, a distributed storage data caching system, http://www.dCache.org/manuals/talk-4-005.pdf

[4] P. Fuhrmann, A Perfectly Normal File System, http://www-pnfs.desy.de/info.html

[5] PostgreSQL Organization home page http://www.postgresql.org/

[6] J. Bakken et al. Monitoring a Petabyte Scale Storage System. Proceedings of CHEP-2004 Conference, Interlaken, September-October 2004.