Storage System Software Solutions For High-End User Needs

Carole B. Hogan
Director of UniTree Technical Development

Distributed Computing Solutions, General Atomics
San Diego, California

Storage System Software Solutions For High-End User Needs

Abstract

Today's high-end storage user is one that requires rapid access to a reliable terabyte-capacity storage system running in a distributed environment. This paper discusses conventional storage system software and concludes that this software, designed for other purposes, cannot meet high-end storage requirements. The paper also reviews the philosophy and design of evolving storage system software. It concludes that this new software, designed with high-end requirements in mind, provides the potential for solving not only the storage needs of today but those of the foreseeable future as well.

Introduction

Since the days of computer infancy, people have used computers to generate long-term data, i.e., data that they desired to access over extended periods of time. To store this data and to permit access to it, computers included a system of storage hardware and storage software. Generally, these storage systems provided similar services to all users. The storage hardware consisted of a single storage media type. The storage software provided a convenient interface for users to access and manage data stored on the hardware media.

As the development and use of computers became more specialized and sophisticated, storage needs of users began to differ. Some user needs were simple, for example, large numbers of small files accessed frequently. Other user needs were more demanding, fewer numbers of large files that were accessed with varying frequencies. Even though user storage needs were diverging, storage systems generally continued to offer the same general services, failing to keep pace with differing user requirements.

Today, user storage needs are continuing to diverge and are doing so at a faster rate than in the past. Conventional storage systems, i.e., the systems that have grown up with the first generations of computer users, can no longer even pretend to meet the differing storage needs of current users. However, new storage systems are being provided that can meet the needs of a variety of storage users in today's computing environments and can also adapt to changing needs in the environments of tomorrow.

The remainder of this paper focuses on the particular storage software needs of high-end users, conventional storage systems software and the shortcomings that prevent it from meeting these needs, and the features of the newer storage system software that make it an ideal solution for this user community.

Storage Needs

In general, computer users with high-end needs are those that generate terabytes of long-term data from high-performance processors. Working in a network environment, these users require the data to be placed in a reliable storage system that can be rapidly and conveniently accessed from several points in the network. Because of the finite budgets, the storage system must be able to provide its services by efficient use of continually-advancing storage technology.

While high-end users have several needs from their storage system software, their primary needs are reliability, capacity, performance, hardware flexibility, software flexibility and reasonable cost. Each of these needs is discussed in turn.

Reliability

Reliability is the capability to guarantee that data accepted for storage will be returned in the same state in which it was received, and the capability to provide continuous storage system availability.

The data integrity component of the reliability need, i.e., returning data to the user in the same state in which it was stored, is the traditional hallmark of storage system software, particularly for storage systems designed as long-term archives. If users cannot trust the storage system software to provide data integrity then, regardless of its other attributes, the system is not useful to them.

The system availability aspect of the reliability need is particularly important for real-time applications such as seismic monitoring programs. Storage system software must provide continuous storage system service in order for these applications to successfully store the data received from the real-time source.

Capacity

Capacity is the capability to store increasing amounts of data upon demand.

The capacity requirement is primarily a software requirement to keep pace with developing hardware technology that permits the generation of larger and larger quantities of data. For example, the processor and main memory technology provided in computers from which users generate data is continually improving in speed and size. This allows users to generate more data in shorter periods of time.

Other types of sophistication in data generation techniques also require increasing storage capacity. for example, satellite-based data generation can result in petabyte capacity requirements. Storage system software must be able to meet these capacity requirements upon demand, in order to be useful in these types of computing environments.

The capacity requirement also prevents the placement of limitations on the numbers of files that can be stored or the size of the files that are stored in the system.

Performance

Performance is the capability to provide access to data as quickly as possible.

The need for fast response from storage system software, both on read and write accesses, is especially important when users generate large amounts of data in a network, or distributed, environment. In such an environment, the storage system software usually runs on a processor or processors dedicated to its use while users access the system from a variety of other computers. So in order for the software to provide optimum performance, it must be efficient enough to utilize high-speed networks and high-speed storage devices at their maximum performance capability.

Hardware Flexibility

Hardware flexibility is the capability to adapt to new storage media, and the capability to manage multiple media in the same storage system.

Advances in hardware technology continue to provide faster, cheaper, greater-capacity storage media. In order for users to receive the benefit of these advances in technology, the storage system software must be flexible enough to store, retrieve and manage data on the new media with only minor modifications to the software.

Storage system software must also be able to manage different hardware media types in the same storage system to give users the option of utilizing media with different storage and cost/performance characteristics.

Software Flexibility

Software flexibility is the capability to operate on several software platforms, and the capability to adapt to new application requirements.

Generally, storage system software is designed to interface with the software operating system on a given processor. Because users are often best served by periodic upgrades to different storage processors with different operating systems, the storage system software must be flexible enough to run on a variety of software platforms.

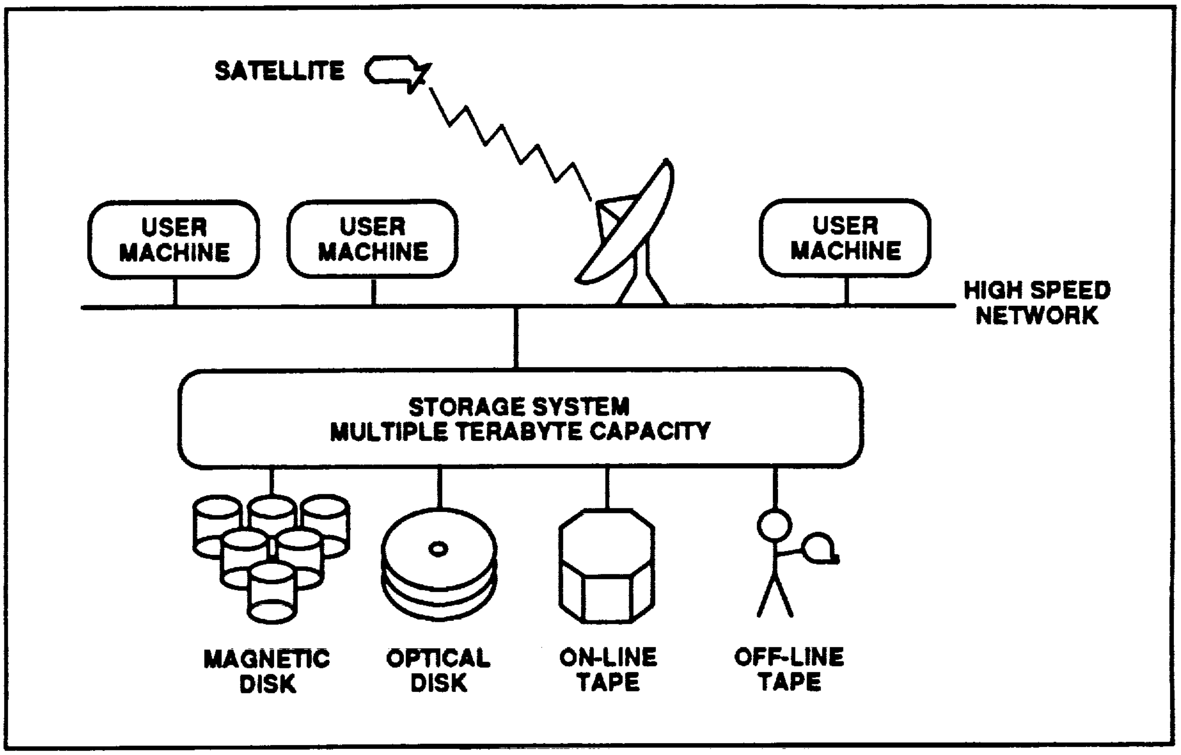
Storage system software must also be able to adapt to new application requirements such as providing the means to identify files by attribute rather than by more traditional means.

Reasonable Cost

Reasonable Cost is the capability to provide the maximum amount of storage within defined budget guidelines.

To be cost effective, the storage systems software must be able to utilize cost-reducing advances in storage hardware technology such as managing cheaper storage media types as they become available. Also, storage system software that can utilize robotically-mounted storage media will permit reduction in the labor costs associated with manually mounted media.

Figure 1 presents an example of a network computing environment that depicts a storage solution for high-end storage needs.



(Figure 1)

Conventional Storage System Software

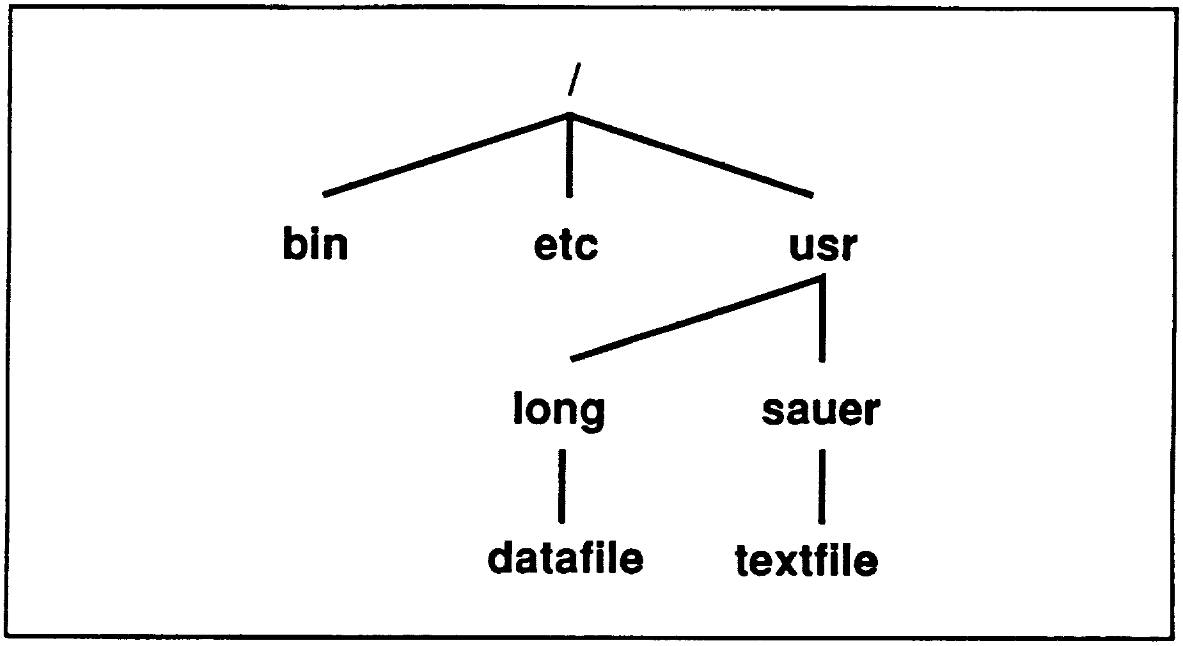
This section of the paper describe conventional storage system software and discusses the shortcomings inherent in this software that prevent it from meeting today's high-end storage needs.

Features

In general, conventional storage system software is designed to manage large amounts of data over long periods of time, and to provide convenient user access to the data. Convenient user access is provided through two abstractions, a file and a human name to identify a file.

The file abstraction allows users to manage their data in convenient units rather than as individual bits. A human name for a file permits users to access the file with that name rather than requiring them to supply the actual address on the storage media where the file is stored. The conventional human name for a file is a directory pathname.

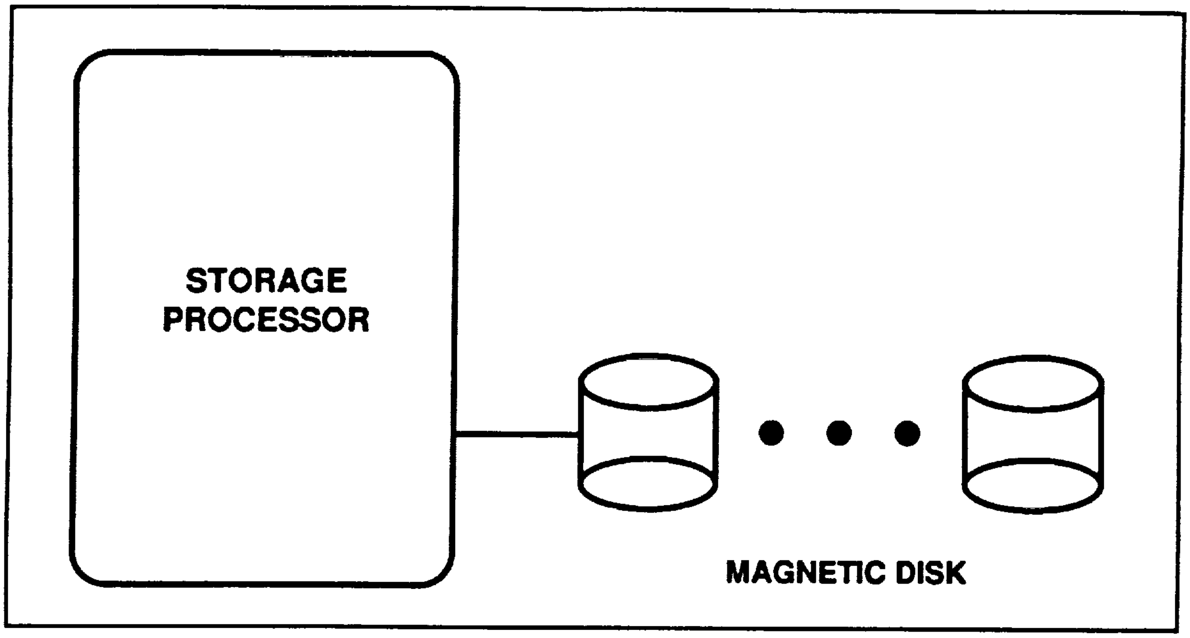
Figure 2 presents a sample directory structure of the UNIX file system, a conventional storage system familiar to many users. The two files in the directory structure shown have the human names "/usr/long/datafile" and "/usr/sauer/textfile".



(Figure 2)

Typically, conventional storage system software provide operations on files such as create, open, write, read, and close. It also manages files by storing them on a single storage media type. The conventional storage media is magnetic disk.

Figure 3 shows the hardware configuration for the conventional UNIX file system.



(Figure 3)

Meeting Storage Needs

Generally, conventional storage system software fails to fulfill each of today's high-end storage requirements because it is designed to meet simpler storage needs in simpler computing environments. Some of its specific shortcomings are discussed here.

Reliability

Storage system software provides data integrity by using algorithms that function correctly and by providing mechanisms that preserve data when hardware media fails. While conventional system algorithms function correctly, the software makes no provision for protecting data from media failure, such as providing redundant copies on different media or automatic backup procedures. Generally, assurance of data integrity in these systems is only as good as the reliability of the storage hardware media and the frequency of manual backup of the data to other media.

Further, providing continuous system availability is not a feature of conventional storage system software. For example, the flexibility to manage a single storage database from two copies of the same software running on two different processors is absent. Such a feature would permit continuous service across processor failure.

Capacity

The capacity to meet increasing storage capacity demands is also absent form conventional storage system software due to built-in limitations in the software. First, capacity is limited because the software manages only magnetic disk as the storage hardware media. There is a limit to the amount of disk that can be attached to any processor.

Second, capacity is also limited because conventional software is not designed to continue writing a file to another disk with available space when the first disk becomes full.

This limitation is not only an inconvenience to users and system administrators but it also means that the largest file in the system is restricted to the size of the largest disk.

Third, total storage system capacity is also limited by a software restriction on the number of data blocks that can be addressed in the system. This restriction hold regardless of the number of disks managed by the software. Usually, this addressing restriction limits total storage capacity to hundreds of gigabytes on even the largest computers.

Performance

The greatest cause of performance degradation in conventional storage systems is the necessity for manual storage media management. When magnetic disks become full of data, a human operator must off-load the data to other media, such as tape, in order to create space on the disk for storing more files. Later, when a user desires to access a file that has been moved to tape, the operator must first locate the tape and read the file from the tape back to disk. Once the file is on disk again, the user can access it.

Any software system that requires manual media management will not perform satisfactorily in today's high-end computing environment.

Hardware Flexibility

The hardware flexibility requirement of adapting to new storage hardware media and of managing multiple media types in the same storage system software. Conventional software, as stated above, manages only magnetic disk as the storage hardware media.

Software Flexibility

Conventional storage system software is designed to run on only one type of operating system. Therefore, the same software cannot be used to manage user data if the storage processor is upgraded to one that provides a different operating system. In this case, another brand of storage system software must be used. This usually requires reformatting data from the old system format to the new system format.

Also, conventional storage system software lacks the flexibility to adapt to new application requirements such as new naming conventions for files. Because the conventional mechanism for naming files, i.e., the directory structure, is intertwined with file management, a new naming mechanism, such as a database management system, will require a different brand of storage system software.

Reasonable Cost

Conventional storage system software keeps storage costs relatively high by managing only magnetic disk and by requiring labor-intensive manual storage media management.

The remainder of the paper discusses the philosophy, features and architecture of today's evolving storage system software, and how this software can meet the storage needs of high-end users for the foreseeable future.

Evolving Storage System Software

Recognizing the need for storage system software that is capable of responding to user needs in the computing environments of today and tomorrow, the IEEE Technical Committee on Mass Storage Systems and Technology and the IEEE Storage System Standards Working Group are developing a model for mass storage systems. In its most recent published version, the Mass Storage System Reference Model[1] advocates concepts and an architecture that will allow storage system software to meet user needs for the foreseeable future.

Much of today's evolving storage system software takes its direction form the Mass Storage System Reference Model. Although the Model is not specifically described in this paper, its influence is visible in the philosophy, features and architecture of the evolving storage systems software discussed here.

Philosophy

The philosophy of evolving storage system software states that the software must be flexible enough to meet current application needs and to be easily adapted to changing application needs and hardware technology advances. This philosophy, seemingly unremarkable, deserves special notice because it is the first time storage software has been dedicated to meeting current and changing needs and technology. Certainly, conventional storage system software is designed without this motivation in mind and, as a result, fails to satisfy high-end user needs.

Key Features

There are several key features of evolving storage system software. These features are described here. Taken together, they provide a flexible storage system.

Redundancy

Redundancy is the ability to continue storage service in the event of hardware failure such as processor or storage media failure. If a storage processor fails, access to the storage system is threatened. If hardware media fails, access to data on that media is threatened.

Evolving storage system software is designed to allow continuance of service in the event of processor failure by providing the capability to access the same storage database from a standby processor. If the primary processor on which the storage system is operating fails, the standby processor can be started and the storage system run from that hardware platform.

Evolving storage system software also provide the capability to store redundant copies of data on different devices of the same media such as different tapes. If this feature is taken advantage of and a tape fails, data is not lost because copies exist on other tapes.

No Software Limits

A lack of software limits means that the capacity of the storage system is limited not by the software, but only by the amount of storage hardware media allocated to it.

Evolving storage system software places no limitation on the length of files, the number of files, the lengths of human names for files or the number or size of directories, it that is the naming mechanism, that the storage system can contain. Unrestricted file length is particularly important since it means that a single file can span devices of the same storage media type such as disk or tape, and is no longer limited to the capacity of a single disk or tape.

Storage Media Hierarchy

Evolving storage system software has the capability to manage a storage media hierarchy. A hierarchy is different types of storage media configured in a layered or hierarchical fashion and managed as part of a single storage system. In general, the media in a storage hierarchy are layered according to speed of access, with the faster and generally more expensive media in the higher layers and the slower and generally cheaper media in the lower layers of the hierarchy.

For example, a storage hierarchy might consist of magnetic disk, optical disk, on-line magnetic tape and off-line magnetic tape. Data is moved between these layers according to site-configurable parameters and user access.

Automatic Migration and Caching

Evolving storage system software offers the features of migration and caching to automatically move data between the layers of the storage media hierarchy. Migration is the movement of data downward in the layers of the hierarchy. Caching is the movement of data upward in the layers of the hierarchy. The software moves the data automatically, without human operator intervention, according to site-configurable parameters and user access.

Migration parameters determine how often files are migrated to lower layers and which files are migrated. Caching upward in the hierarchy is triggered by user access of files that have migrated to lower levels. Because of migration and caching, active files, i.e., those currently being accessed, reside on the higher, faster access layers of the hierarchy, while inactive files reside on the lower, slower media.

Distributability

Evolving storage system software consists of modules that are capable of being distributed among several processor in a network environment. This means that the storage system software is not confined to a single processor but can run on several processors simultaneously as part of a single storage system. Distributability permits load balancing among components of the system and faster access to data.

Separation of Control and Data

User access to a storage system consists of commands transmitting data (e.g., write commands) and commands receiving data (e.g., read commands). Each command consists of two parts, control and data. The control component is the action to be taken, e.g., write or read. The data component is the actual data to be written or read.

Evolving storage system software separates the control and data components of storage access commands into two distinct entities that can be transmitted separately to and from the storage system. This separation permits control and data to flow over separate network paths.

For example, control can flow over slower network paths while data can flow over faster network paths. This type of separation permits faster access to the storage system. Since data transmissions are generally much larger than control transmissions, transmitting data over a separate path or paths can result in faster performance in general.

Extensibility

Evolving storage system software is extensible. Extensibility is the capability to easily integrate additional types of storage media into the storage media hierarchy. Easy integration of additional media permits sites to take advantage of new storage hardware technology as it develops.

For example, a site may begin with a storage media hierarchy consisting of magnetic disk, on-line tape and off-line tape. Because of the extensibility of the storage software, optical disk can be integrated into the hierarchy, for example, between the magnetic disk and on-line tape layers, when the site decides to take advantage of the features offered by optical disk.

Separate Naming Mechanism

Evolving storage system software is designed to easily adapt to new naming conventions so that users can access their data in the manner that is most efficient for their applications. For example, if applications at a given site generate many thousands of files, naming files by directory pathname can become inefficient. Instead, naming files by attribute such as time of creation may be preferred.

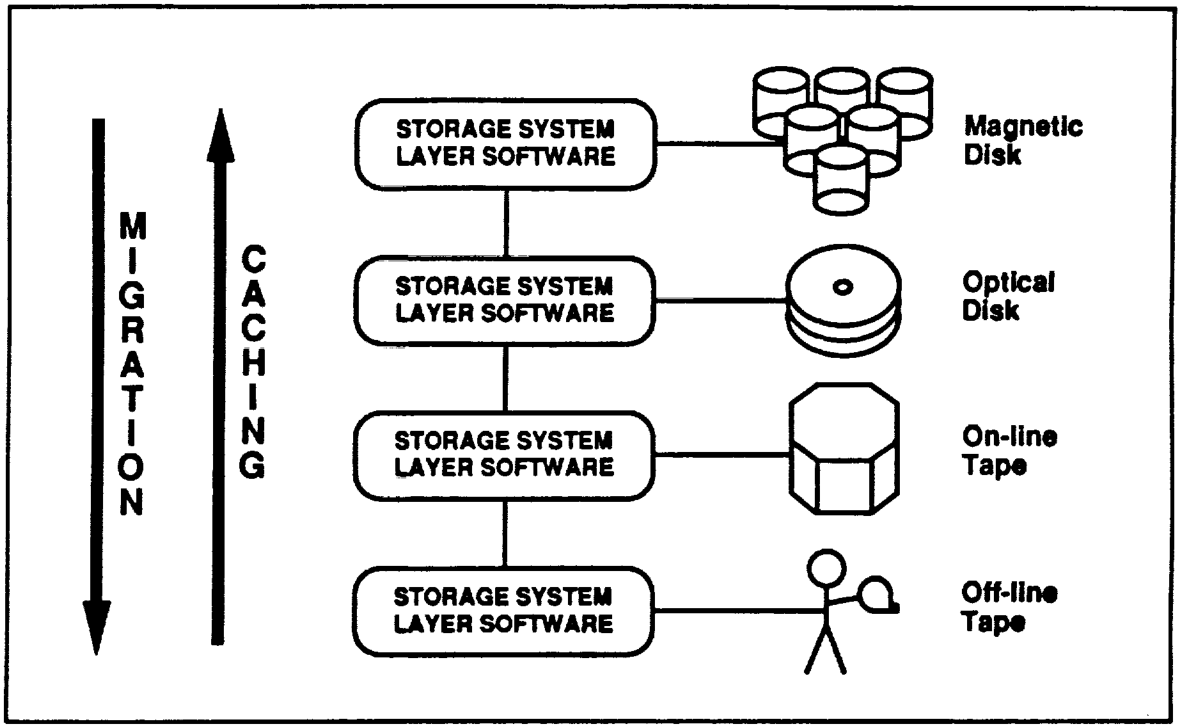
Evolving storage system software will permit this transition in naming mechanism from directory pathname to file attribute because the software modules that map names to files are separate from , and peripheral to, the modules that manage the files. Therefore, to transition to a new naming mechanism means only to change a few peripheral modules rather than the entire storage system software.

Portability

Evolving storage system software is portable. It is designed to run on a variety of software operating systems. This means that if the storage processor is upgraded to a different manufacturer with a different operating system. It also means that the same storage software can manage a single distributed storage system by running on a variety of different processors with different operating systems.

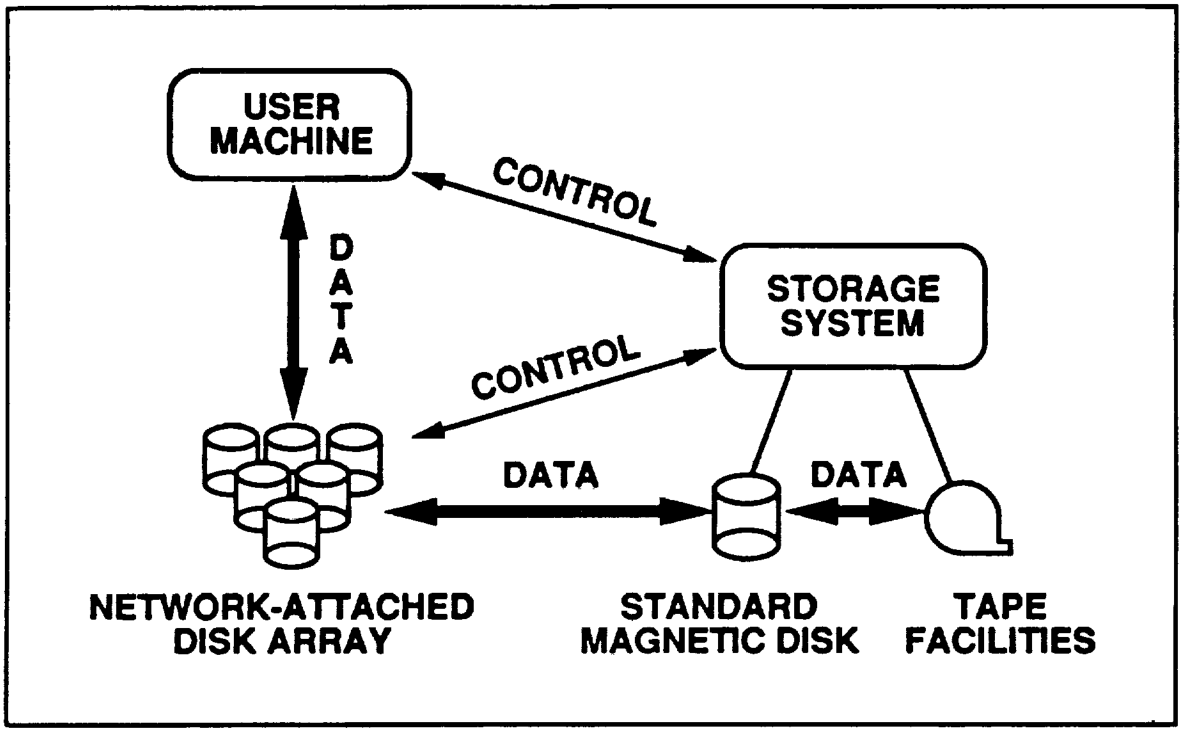
Architecture

The architecture of evolving storage system software is presented here graphically. Figure 4 shows the software and hardware layers in a sample storage media hierarchy. Each layer of media is managed by a different set of software modules.


MIGRATION
CACHING
STORAGE SYSTEM LAYER SOFTWARE
STORAGE SYSTEM LAYER SOFTWARE
STORAGE SYSTEM LAYER SOFTWARE
STORAGE SYSTEM LAYER SOFTWARE
Magnetic Disk
Optical Disk
On-line Tape
Off-line Tape

(Figure 4)

Figure 5 shows the separate network paths that control and data can take between the user and the storage system. In this Figure, data is stored in a three-layer hierarchy, consisting of a network-attached disk array, standard magnetic disk and tape. The last two layers in the hierarchy are attached to the storage processor. Storage system software is running on the disk array controller as well as the storage processor itself. In this way, data can flow directly between the user machine and the disk array without passing through the storage machine.



(Figure 5)

Meeting Storage Needs

The features and architecture of evolving storage system software permit it to meet the storage needs of high-end computer users. The manner in which the software meets these needs is discussed here.

Reliability

System availability and data integrity are provided by the redundancy features of evolving storage system software. Redundancy provided the capability to continue service from a standby processor. It also preserves data in the event of damage or loss of a storage media device such as a tape by providing the capability of storing multiple copies of the same data on different tapes.

Capacity

Evolving storage system software provides the potential for unlimited storage capacity by managing a hierarchy of storage media and by a lack of limits in the number of files that can be stored, the size of files that can be stored, and so forth. Multi-media management in a hierarchical fashion allows tape to be included in the storage system. Tape as a storage medium, coupled with a lack of limitation in file size and numbers, means that the system capacity can grow to as many tapes as a site purchases.

Performance

Evolving storage system software provides performance through the features of automatic migration and caching, separation of control and data and distributability. Automatic migration and caching eliminate the performance degradation caused by conventional manual media management. Separation of control and data permit data to flow to and from the storage system over faster network paths, improving performance in general. Distributability allows components or layers of the storage software to run on different processors, permitting load balancing and faster access to the system as a whole.

Hardware Flexibility

The extensibility feature allows evolving storage system software to provide hardware flexibility. With this flexibility, new storage media can be integrated into the storage hierarchy. The ability to integrate new media permits users to take advantage of advancing hardware technology.

Software Flexibility

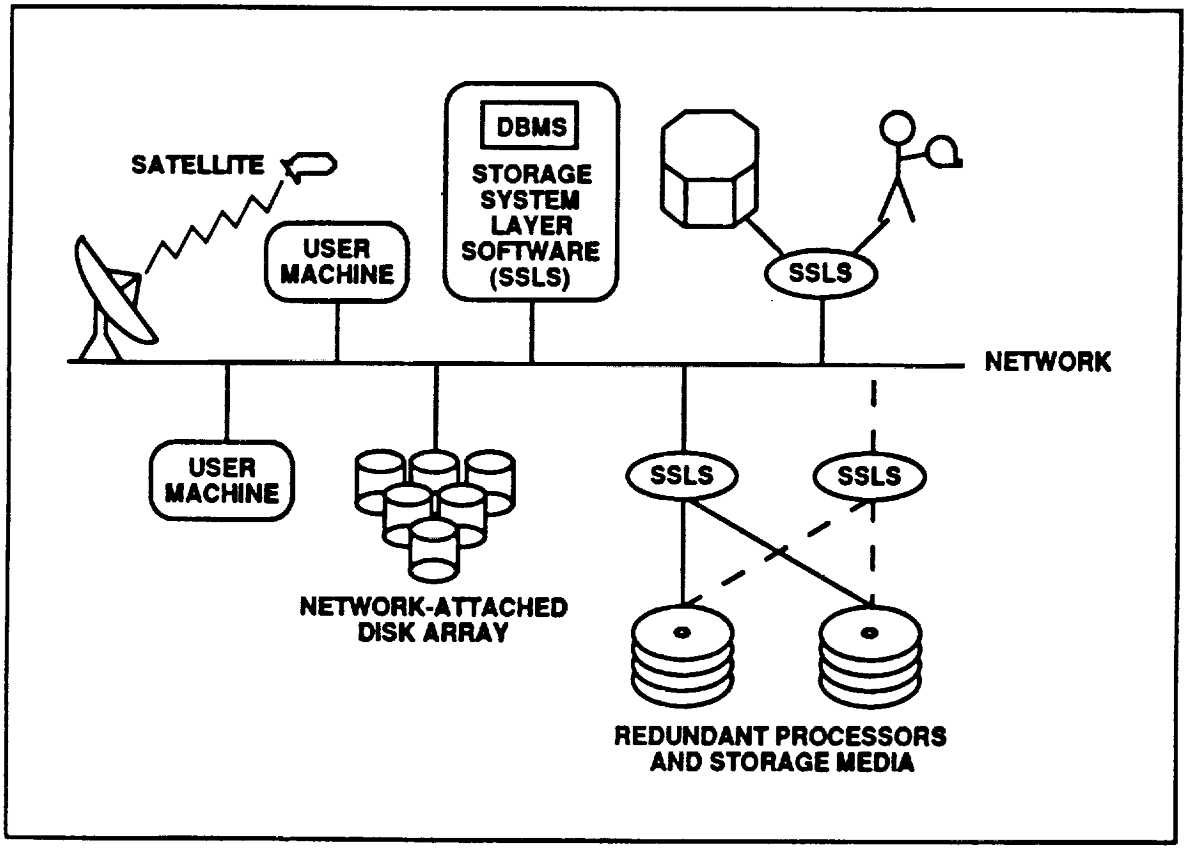
Software Flexibility is provided by several features of evolving storage system software, including a lack of software limitations, separate modules that map human file names to the files themselves, distributability and portability. Portability is the flexibility to run on a variety of software operating systems.

Reasonable Cost

Reasonable cost is provided by evolving storage system software through the features of storage media hierarchy management, automatic migration and caching, extensibility and portability. Managing data through a storage media hierarchy means that less money need be spent on expensive storage media because only active files will be stored on this higher layers in the hierarchy. Generally, active files are a small subset of all files in a storage system. Automatic migration and caching eliminates the cost associated with manual storage media management.

Extensibility allows cheaper storage media to be integrated into the system as it becomes available. Portability accommodates less expensive open system solutions to storage needs since the same storage software can run on a variety of software platforms that can be combined into a single storage system.

Figure 6 shows an example of an evolving storage system managing storage needs in a high-end, distributed computing environment. In this example, the naming mechanism is a database management system. The layers of the storage hierarchy consist of a disk array, optical disk and on-line and off-line tape. These layers are distributed among different processors in the network. Redundancy is provided in the optical disk layer of the storage hierarchy by a standby processor (dotted line) and redundant sets of optical disk.


SATELLITE
USER MACHINE
DBMS
STORAGE SYSTEM LAYER SOFTWARE (SSLS)
SSLS
NETWORK
USER MACHINE
SSLS
SSLS
NETWORK-ATTACHED DISK ARRAY
REDUNDANT PROCESSORS AND STORAGE MEDIA

(Figure 6)

Conclusion

Users in computing environments with high-end storage requirements must turn away from using conventional storage system software if these requirements are to be met. Conventional software, designed for simpler computing environments with simpler storage requirements, cannot satisfy today's high-end storage needs.

Evolving storage system software, influenced by the IEEE Mass Storage System Reference Model, is designed to meet high-end storage requirements for the foreseeable future. It is in these systems that today's high-end users will find solutions to their storage software needs.

Acknowledgement

The contributions of Michael Hardy, Products Manager, Distributed Computing Solutions, to this paper are gratefully acknowledged.

References

1. Sam Coleman and Steve Miller, "Mass Storage System Reference Model: Version 4 (May, 1990)," Tenth IEEE Symposium On Mass Storage Systems, Monterey, California, May, 1990.

Storage System Solutions

NSSDC Conference on Mass Storage Systems

Carole B. Hogan

Director of UniTree Technical Development
Distributed Computing Solutions (DISCOS)

July 23, 1991

General Atomics
DISCOS Division

STORAGE SYSTEM SOLUTIONS

Overview:

- Storage system definition
- Storage needs
- Conventional storage systems
- Evolving storage systems

PRECEDING PAGE BLANK NOT FILMED

STORAGE SYSTEM SOLUTIONS

Storage System Definition:

- Facility that
  - manages

    large amounts of data

    over long perlods of time
  - provides

    convenient use of data

STORAGE SYSTEM SOLUTIONS

Storage Needs:

- Reliable storage
- Large amounts of data
- Rapid response
- Hardware flexibility
- Software flexibility
- Reasonable cost

STORAGE SYSTEM SOLUTIONS

Reliable Storage:

- Capability to provide system availability and data integrity
  - real-time applications
  - long-term archive

STORAGE SYSTEM SOLUTIONS

Large Amounts of Data:

- Capacity to store and provide access to increasing amounts of data
  - satellite-based data acquisition
  - faster processors

STORAGE SYSTEM SOLUTIONS

Rapid Response:

- Capability to provide access to data as quickly as possible
    - high-speed networks
    - high-speed storage devices

STORAGE SYSTEM SOLUTIONS

Hardware Flexibility:

- Capability to adapt to new storage media
    - optical-based
    - helical scan tape

▶ STORAGE SYSTEM SOLUTIONS

Software Flexibility:

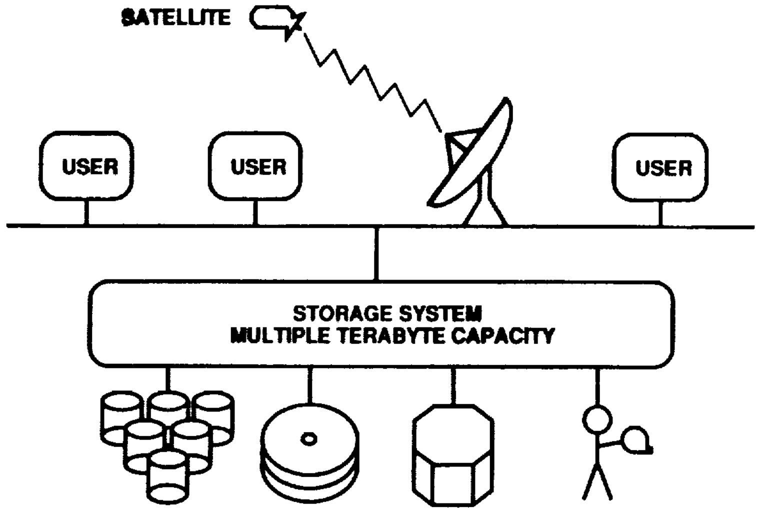
- Capability to operate on several software platforms
- Capability to adapt to new application requirements
  - database management system access

▶ STORAGE SYSTEM SOLUTIONS

Reasonable Cost:

- Provide maximum amount of storage within budget guidelines
  - automated storage media management

STORAGE SYSTEM SOLUTIONS



STORAGE SYSTEM SOLUTIONS

Conventional Storage Systems:

- **Description**
- **Example**
- **Shortcomings**

STORAGE SYSTEM SOLUTIONS

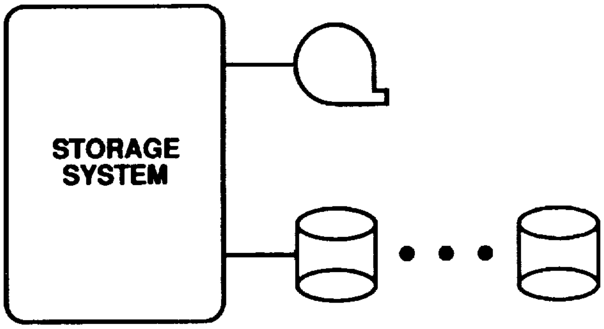
Conventional Storage System Description:

- Provides
  - long term storage and retrieval of user data
  - file abstraction for users to manage data
  - directory pathname as the mechanism to name files
  - open, write, read, and close operations on files
  - management of magnetic disk storage media for files

STORAGE SYSTEM SOLUTIONS

Conventional Storage System Example:

- UNIX File System

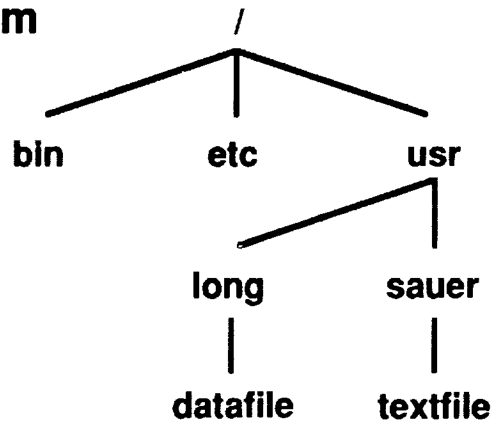


STORAGE SYSTEM

STORAGE SYSTEM SOLUTIONS

Conventional Storage System Example:

- UNIX File System



STORAGE SYSTEM SOLUTIONS

Reliability:

- Only as good as
  - reliability of magnetic disk
  - frequency of backup

STORAGE SYSTEM SOLUTIONS

Conventional Storage System Shortcomings:

- Reliability
- Capacity
- Performance
- Hardware flexibility
- Software flexibility
- Cost

STORAGE SYSTEM SOLUTIONS

Capacity:

- Manages single storage media type
  - few per machine
- Provides no mechanism to continue writing to another disk which has available space
  - limited file size
- Uses software feature that limits total capacity
  - 100s of gigabytes

STORAGE SYSTEM SOLUTIONS

Performance:

- Manual storage media management
  - operator off-loading data from disk to tape and back again

STORAGE SYSTEM SOLUTIONS

Hardware Flexibility:

- Magnetic disk as the only storage media type

STORAGE SYSTEM SOLUTIONS

Software Flexibility:

- No capability to adapt to new application requirements such as new naming conventions

STORAGE SYSTEM SOLUTIONS

Cost:

- More and more expensive disk
- Labor-intensive storage media management

STORAGE SYSTEM SOLUTIONS

Evolving Storage Systems:

- Philosophy
- Key features
- Architecture
- Meeting storage needs
- Example

STORAGE SYSTEM SOLUTIONS

Philosophy:

- Flexible storage system that meets current application needs and is easily adapted to changing application needs and hardware technology advances

STORAGE SYSTEM SOLUTIONS

Key Features:

- IEEE Mass Storage System Reference Model
- Redundancy
- No software limits
- Storage media hierarchy
- Automatic migration and caching
- Distributability
- Separation of control and data
- Extensibility
- Separate naming mechanism
- Portability

STORAGE SYSTEM SOLUTIONS

Redundancy:

- The ability to continue service in the event of hardware failure such as processor or storage media failure

STORAGE SYSTEM SOLUTIONS

No Software Limits:

- **Unlimited length files**
- **Unlimited number of files**
- **Unlimited length file names**
- **Unlimited number of directories**

STORAGE SYSTEM SOLUTIONS

Storage Media Hierarchy:

- **Different types of storage media configured in a layered or hierarchical fashion and managed as a single storage system**

STORAGE SYSTEM SOLUTIONS

Automatic Migration and Caching:

- **Movement of data between the media layers of the storage hierarchy according to administrative policy**
- **Migration is data movement downward in the hierarchy**
- **Caching is data movement upward in the hierarchy**

STORAGE SYSTEM SOLUTIONS

Distributability:

- **Storage system components running on different processors within a network**

STORAGE SYSTEM SOLUTIONS

Separation of Control and Data:

- The ability to transmit control and data over separate network paths

STORAGE SYSTEM SOLUTIONS

Extensibility:

- The ability to easily integrate additional types of storage media into the storage hierarchy

STORAGE SYSTEM SOLUTIONS

Separate Naming Mechanism:

- The ability to easily adapt to new naming conventions

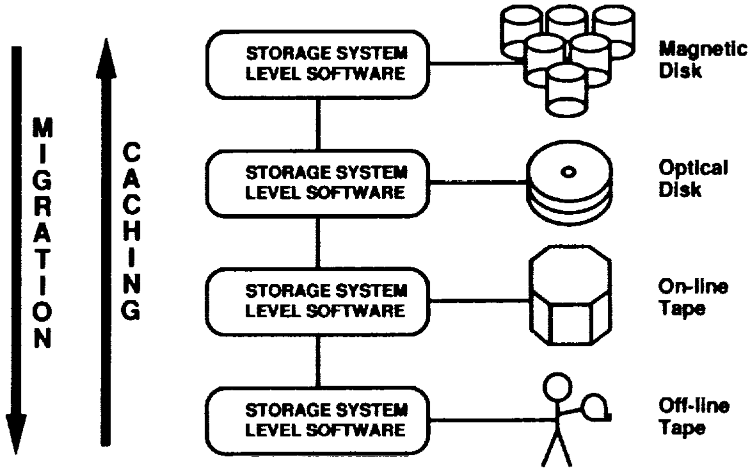
STORAGE SYSTEM SOLUTIONS

Portability:

- The ability to run on a variety of software platforms

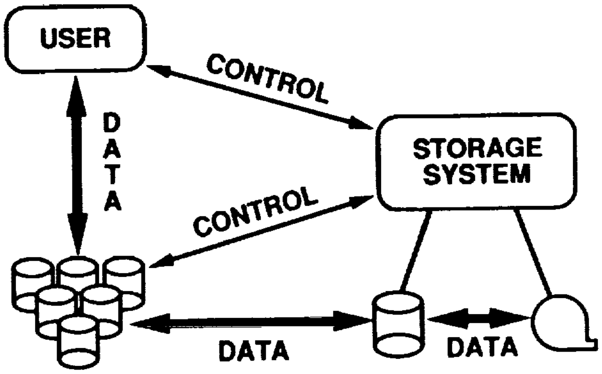
STORAGE SYSTEM SOLUTIONS

Architecture:


MIGRATION
CACHING
STORAGE SYSTEM LEVEL SOFTWARE
Magnetic Disk
STORAGE SYSTEM LEVEL SOFTWARE
Optical Disk
STORAGE SYSTEM LEVEL SOFTWARE
On-line Tape
STORAGE SYSTEM LEVEL SOFTWARE
Off-line Tape

STORAGE SYSTEM SOLUTIONS

Architecture:



STORAGE SYSTEM SOLUTIONS

Meeting Storage Needs:

- **Reliability**
- **Capacity**
- **Performance**
- **Hardware flexibility**
- **Software flexibility**
- **Reasonable cost**

STORAGE SYSTEM SOLUTIONS

- **Reliability:**
  - **Redundancy**

- **Capacity:**
  - **Storage media hierarchy**
  - **No software limits**

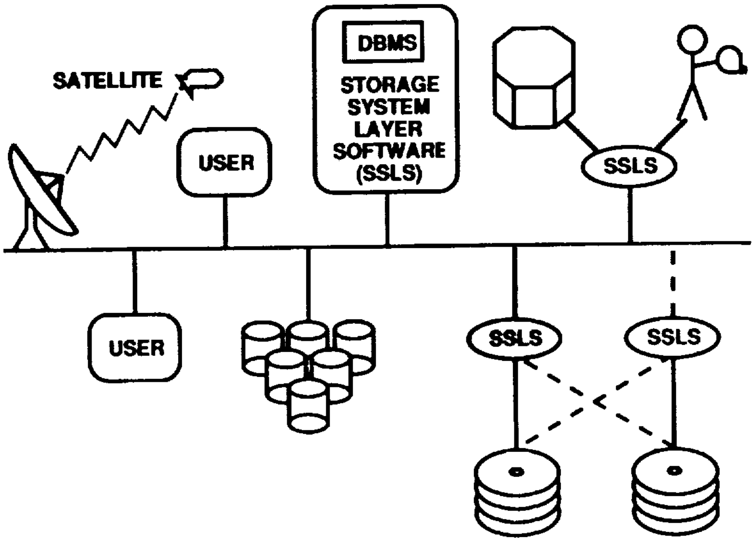
STORAGE SYSTEM SOLUTIONS

- Performance:
  - Automatic migration and caching
  - Separation of control and data
  - Distributability
- Hardware Flexibility:
  - Extensibility

STORAGE SYSTEM SOLUTIONS

- Software Flexibility:
  - No software limits
  - Separate naming mechanism
  - Distributability
  - Portability
- Reasonable Cost:
  - Storage media hierarchy
  - Automatic migration and caching
  - Extensibility
  - Portability

STORAGE SYSTEM SOLUTIONS


SATELLITE
USER
DBMS
STORAGE SYSTEM LAYER SOFTWARE (SSLS)
SSLS
USER
SSLS
SSLS