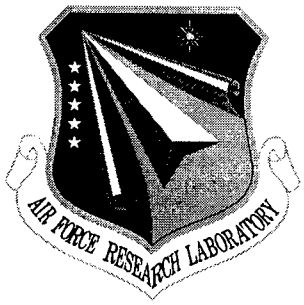
AFRL-IF-RS-TR-1998-119
Final Technical Report
June 1998



MASS STORAGE SYSTEM (MSS)

Synectics Corporation

John C. Rossi and Joseph J. Riolo

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-1998-119 has been reviewed and is approved for publication.

APPROVED:

ALBERT J. JAMBERDINO
Project Engineer

FOR THE DIRECTOR:

JOSEPH CAMERA, Deputy Chief
Info & Intel Exploitation Division
Information Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/IFED, 32 Hangar Road, Rome, NY 13441-4114. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1998	Final May 94 - Feb 98

4. TITLE AND SUBTITLE

MASS STORAGE SYSTEM (MSS)

5. FUNDING NUMBERS

C - F30602-94-C-0127
PE - 63260F
PR - 3480
TA - 00
WU - 87

6. AUTHOR(S)

John C. Rossi and Joseph J. Riolo

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Synectics Corporation
111 E. Chestnut Street
Rome NY 13440

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/IFED
32 Hangar Road
Rome NY 13441-4114

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-1998-119

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Albert A. Jamberdino/IFED/(315) 330-2845

12a. DISTRIBUTION AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The original effort had three main objectives. The first objective involved the development, test and delivery of a mass storage system using hierarchical storage concepts. This system design integrated both high-performance, high cost storage devices with low-performance, low-cost storage devices. This approach provides high-speed performance while minimizing overall system costs. The equipment implementation consisted of using both magnetic disk RAID and a large-capacity optical jukebox. The second objective concentrated on using benchmark tests to verify the true performance of candidate storage devices under a variety of operational conditions. The final objective developed a computer network model to assist the impact of various mass storage architectures and their impact on the user's overall computer network.

14. SUBJECT TERMS

Computer Storage, System Architecture, Hierarchial, Storage Management

15. NUMBER OF PAGES

84

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Standard Form 298 (Rev. 2-89) (EG)
Prescribed by ANSI Std. 239.18
Designed using Perform Pro, WHS/DIOR, Oct 94

TABLE OF CONTENTS

1.0 SUMMARY

1.1 CONTRACT

This document is the Final Technical Report for contract number F30602-94-C-0127 entitled, "Mass Storage System (MSS)." The MSS was originally a 24-month effort running from May 20, 1994 to May 20, 1996. Amendments and extensions stretched the end date to February 28, 1998. Synectics Corporation developed the MSS for the Global Information Base Branch (IFED) (formerly IRAP), of the Air Force Research Laboratory (AFRL, formerly Rome Laboratory) Information and Intelligence Exploitation Division (IFE).

1.2 OVERVIEW

The purpose of the MSS effort was to deliver a prototype capability to demonstrate and evaluate new and existing mass storage technologies and to test performance of these concepts in a distributed network environment. To accomplish this goal Synectics investigated existing storage technologies with an eye toward providing a prototype hierarchical mass storage system (PHMSS). These efforts led to an integrated system consisting of a Kodak ADL 2000 Optical Jukebox, Data General CLARiiON RAID, and Hewlett Packard 755 workstation hosting a modified Unitree+ Hierarchical Storage Management (HSM) software package. Additionally, a Mass Storage Evaluation Environment (MSE2) software tool was developed to integrate public domain benchmarks. MSE2's role was to facilitate generation and analysis of device and systems performance measurements. The PHMSS was hosted at three separate test sites to progressively evaluate the system in increasingly complex environments. Synectics facilities in Rome, NY provided the integration, familiarization, and preliminary data gathering testbed. The system then relocated to AFRL, Rome Research Site for integration into the IE2000 Lab network. This move provided for testing in an operational environment and increased the exposure of the system to potential customers. Finally the entire PHMSS was transported and installed in the 480 Intelligence Group (IG) facility at Langley AFB, Virginia for incorporation in their facility as a fully operational system. The using community at Langley identified the need for additional control over the PHMSS, which led to prototype development of the Mass Storage Management Interface (MSMI). Late in the life span of the effort, a prototype Optical Redundant Array of Inexpensive Disks (ORAID) became available via another project at AFRL, Rome Research Site. The testing and evaluation of this system was undertaken to investigate the potential advantages of this new technology for mass storage situations.

1.3 OBJECTIVES

The original contract had three objectives: (a) Develop, test, and deliver a mass storage system (MSS). (b) Develop and deliver a suite of benchmark testing software. (c) Develop user models and evaluate the impact of mass storage. As the contract progressed, the scope was expanded to investigate rewriteable optical jukebox and O-RAID concepts.

The first objective was achieved by using a hierarchical storage design. Based on statistical use, most users access only 20% of their total data 80% of the time. The remaining data needs to be kept on-line, but seldom requires high-speed access. The hierarchical design integrates both high-performance, high-cost storage devices with low-performance, low-cost storage devices. The hierarchical approach provides high-speed performance while minimizing overall system costs. Implementation of the hierarchical design consisted of using both magnetic disk RAID and a large-capacity optical jukebox.

The second objective concentrated on using benchmark testing to verify the true performance of candidate storage devices under a variety of conditions. Often product specifications are overly optimistic or specify performance under ideal conditions. Benchmark testing provides a common suite to ensure that different products are evaluated under similar operational conditions. The use of "freeware" test modules from the Internet helped to save cost and shorten the development schedule.

The third objective used computer network modeling as an aid to develop the MSS. A commercial software package was purchased for the modeling task. The goal was to assess the impact of various mass storage architectures and the MSS location within the user's overall computer network. Computer modeling insured that the MSS would provide improved user access to required data while lowering overall storage costs.

2.0 USER REQUIREMENTS AND TECHNOLOGY ASSESSMENT

This section discusses the motivation behind the MSS project and provides foundation information regarding the technology areas investigated and employed. The specific goals of the effort and a document outline are also provided.

2.1 USER REQUIREMENTS

The storage and retrieval of intelligence data, specifically digital imagery, places heavy demands on data storage and retrieval subsystems. The volume of data required for digital imagery clearly

requires high-density storage media and a low cost per byte of stored information. Unfortunately, there is an inverse relationship between the access and data transfer times of memory storage subsystems, and their data density/cost per byte. This relationship presents a real quandary for the developer of an imagery exploitation system. It is desirable to have the largest possible amount of imagery and related data resident and available to the intelligence analyst at any given time. However, it is also necessary that the imagery itself (to a lesser extent) and the imagery-related data (to a very great extent) be accessible in an expeditious manner. Currently there is no single mass storage alternative that is both suitably fast and cost effective. A solution to this dilemma is to use a hierarchical hybrid system to build a virtual device that is both fast and inexpensive. A feasible system design would consist of an optical storage system, such as a jukebox, which would serve as an imagery archive. This would be coupled with higher speed systems, such as magnetic disk farms or RAID arrays. These arrays would store more current data, such as information on political hot spots (or other areas of interest to analysts). This type of hybrid system would provide the most cost-effective balance between speed and storage capacity using state-of-the-art technologies that will be available throughout the 1990s and into 2000. However, the complex nature of such systems, combined with the changing nature of an analyst's activities, makes it imperative to determine the optimal configuration prior to implementing a mass storage system for operational use. With MSS Synectics attempts to validate the concept of a hybrid storage system and provide insight to the information required to predict specific customer mass storage needs.

2.2 UNDERSTANDING OPERATIONAL DEMANDS

In this section we describe our understanding of the operational demands for mass storage in terms of the number of very large databases that need to be on-line in an operational environment. This understanding is critical to this development effort in terms of providing realistic demonstrations of the technology to give users a "look and feel" of the value of having the mass storage capability to meet their operational storage needs.

Imagery, one of the highest quality sources of remotely collected intelligence, is the "eyes" of the commander. One reason for the high quality of imagery is that, in many cases, imaging systems are difficult to deceive. Another reason is that imagery supports a wide range of military needs. The following is a list of some military applications of imagery:

- Imagery Intelligence
- Image Measurements for Target Metrics and Location
- Target Materials
- Strike/Fire Support

Because of its wide range of applications, it is both resource and mission effective to be able to share an image with any military function that can use it. In the traditional environment, where reconnaissance systems rely upon hard copy film, sharing imagery requires film duplication and dissemination by means of physical shipping and/or hand carrying. This process is both resource intensive and time consuming. With the advent of electro-optical sensors and the use of digital systems it is now feasible to handle imagery in electronic form. The image is duplicated, when needed, through the use of a soft copy system and disseminated in near real time via transmission over communication lines. The availability of high quality imagery in near real time tremendously increases its value to the military commander.

The change from conventional hard copy exploitation to soft copy exploitation using digital systems is currently underway within the Air Force and other DoD organizations. This change from hard to soft copy exploitation impacts the organization and structure of on-line databases as well as each and every aspect of an image exploitation center. One of the biggest impacts will be altering the functional areas, and the support systems that will need to be modified. Even the products produced by an exploitation center will be new and different. As an example, for target materials production to be performed in soft copy all of the intelligence data (support, imagery, geographic, etc.) must also be in soft copy. Today, much of the intelligence data used in the production of target materials is still in hard copy form (e.g., country histories), and the soft copy data sources are not always on-line and immediately available to the target material developer. Thus the functional changes that must occur as a result of the transition from hard copy to soft copy will cause the greatest change in an exploitation center's data base needs. These functional changes will alter both the types of data and the form of the data that will be required.

The following is a list of the current soft copy imagery or imagery related databases that are necessary to support an image exploitation center.

- ❑ Geographic Data Base
- ❑ Imagery Data Base
- ❑ Point Positioning Data Base
- ❑ Reference Imagery Data Base

The 480 Intelligence Group at Langley AFB estimated future digital imagery storage requirements to be 33.5 Terabytes as long ago as 1992. Clearly, systems capable of handling these volumes of information need to be investigated and developed. Synectics hopes to provide a solution offering cost effective and efficient approaches to data storage and retrieval.

2.3 TECHNOLOGY ASSESSMENT

This section introduces the technologies and associated terminology used in the MSS effort. The following areas are addressed:

- ❑ Storage Technologies
- ❑ Hierarchical Storage Management
- ❑ Benchmarks

2.3.1 STORAGE TECHNOLOGIES

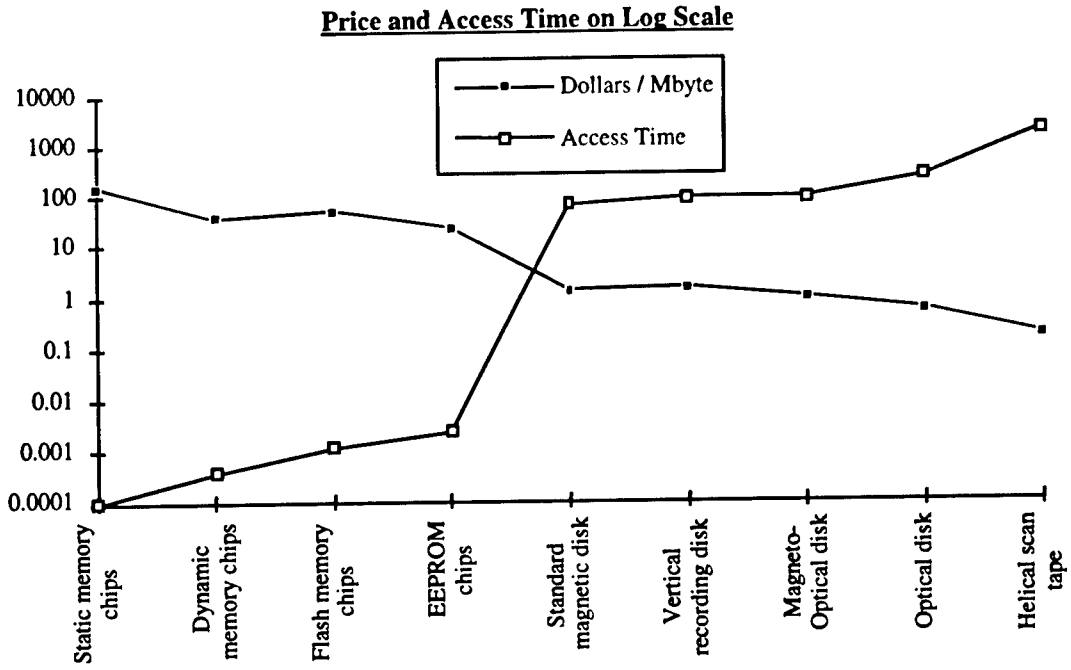
The storage and retrieval of large volumes of data places heavy demands on the associated data storage subsystems. Unfortunately the inverse relationship between the access and data transfer times of memory storage subsystems, and their data density/cost per byte requires careful consideration in the implementation of MSS. This relationship, illustrated in Exhibit 1, presents a real quandary for the developer of an imagery exploitation system. On the one hand it is desirable to have the largest amount of imagery and related data possible resident and available to the analyst at any given time. On the other hand, it is necessary that the imagery itself (to a lesser extent) and the related feature data (to a very great extent) be accessible in an expeditious manner.

As Exhibit 1 points out, there is not currently a mass storage alternative, which is both suitably fast and cost effective. Storage technologies which offer very high capacity, high access speed and low price, such as the 3-D volumetric memories, are not yet mature enough to consider as solutions to current needs. The solution for the MSS is to use a hierarchy of different storage technologies to build a virtual device, which is both fast and inexpensive.

It is important to find out which types of media, and how much of each type, should be in the mix. This section discusses the data density and access speed for some storage technology types that the MSS will be able to support.

The following is a brief overview of the storage technologies, which should be considered in the implementation of the MSS.

Exhibit 1. Comparison of Access Time and Cost per Byte for Storage Technologies



2.3.1.1 SEMICONDUCTOR MEMORY DEVICES

This class of memory and storage devices are lumped under the category of semiconductor-based devices and are of limited utility (at the current state of development) for the archival storage of imagery data. The extremely poor cost per bit, and storage/physical form factor ratios represented by these classes of storage devices preclude them from all but the most specialized functions. The following is a brief overview of these devices:

- Static memory chips – The access time for static memory units is in the sub-five nanosecond range. In addition to size and cost, static memory suffers from its volatile nature. Being volatile means that the information stored in the memory is not preserved when power is removed. This makes it unsuitable for archival storage.

- Dynamic memory chips (another volatile memory type) – These devices can be as much as an order of magnitude less expensive per byte compared with very high-speed static memory.

- Flash memory chips and Electrically Erasable PROM (EEPROM) chips are a special category of semiconductor memory which is not volatile. These are the Programmable Read Only Memory (PROM) These memories are non-volatile, fast, and are the most rugged and impervious of all memory technologies.

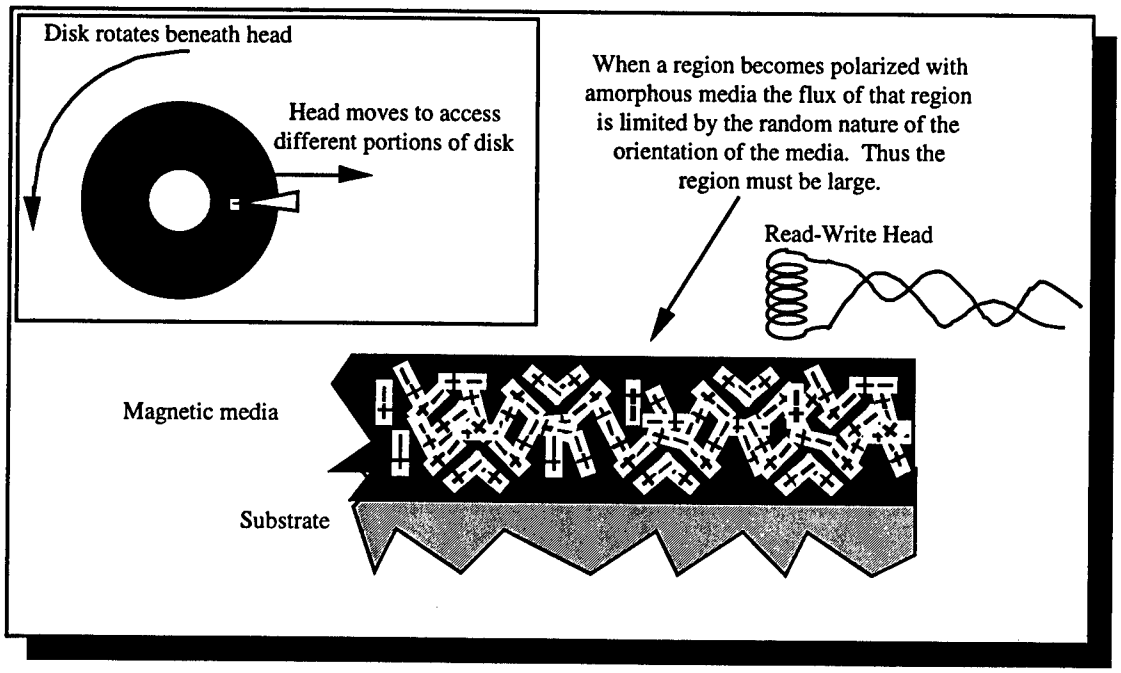
2.3.1.2 *DISK-BASED MAGNETIC MEDIA*

The magnetic recording disk is currently the most cost-effective device used to store data if that data is to be read-writeable with medium access times. Advances in magnetic disk storage evolution will not reach the limitations imposed by physical nature of the technique for a few years yet, and the magnetic disk (or some variant thereof) will probably remain predominate into the year 2000 and beyond. The following is a brief overview of these devices.

- Standard magnetic disk uses either a flexible Mylar (floppy disk) or fixed (hard disk) substrate onto which an amorphous magnetic medium has been deposited. Small regions of the disk are magnetized in one polarity or the other. As the disk spins under the pickup head the transition of one polarity to the other represents a one bit; the lack of that transition represents a zero. Exhibit 2 illustrates standard magnetic disk concepts.

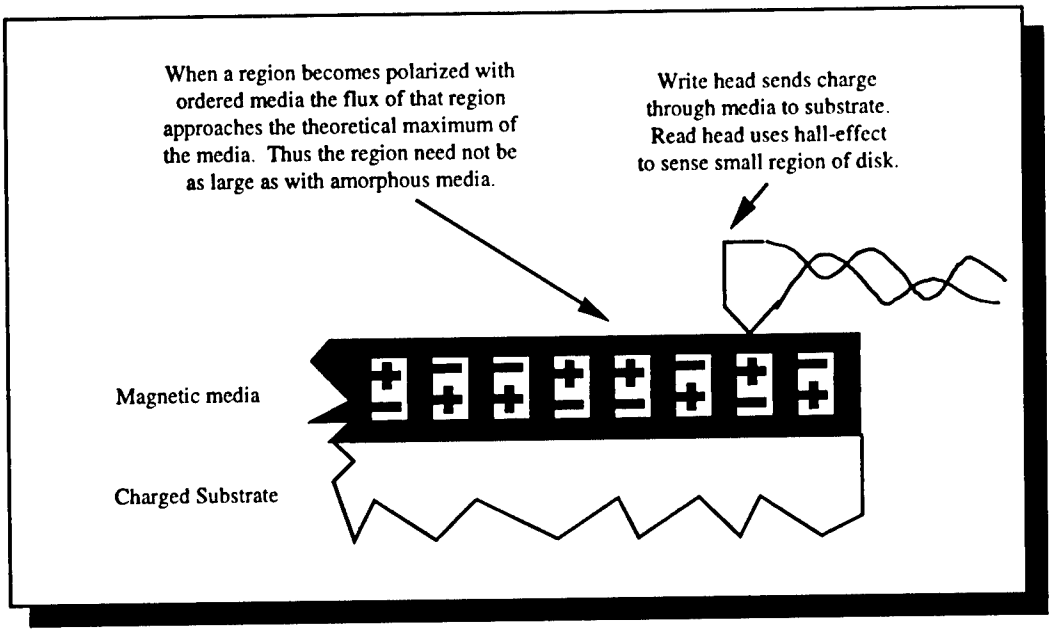
  The amorphous nature of the magnetic medium, and relative imprecision with which the read-write heads may be positioned on the disk surface with any degree of repeatable accuracy, requires that the size of the region needed to represent a bit is reasonably large (from a molecular size perspective). As a result the data densities represented by a standard magnetic disk are at the lower bound of the useful densities required for MSS applications.

Exhibit 2. Magnetic Disk



- Vertical magnetic recording is used to increase the data density of magnetic disk systems. Rather than an amorphous magnetic medium, the vertical recording disk uses a medium, which embodies a highly ordered lattice at the molecular level. Because of this organization the medium may be imparted with a flux density that is at least an order of magnitude greater than is ever possible with amorphous magnetic media. Exhibit 3 illustrates vertical recording.

Exhibit 3. Vertical Recording Disk



2.3.1.3 Optical Storage Media

Optical storage systems use photons and selectively reflective materials rather than electrons or magnetic flux to store information. This allows for very high data densities, and is one of the areas of data storage technology that is having a great impact on the storage and utilization of imagery. Optical storage systems form the focus of the MSS effort. The following presents a brief description of these devices.

- Standard optical disk uses optics to store and retrieve data. It is a technology that is currently in wide use for the archiving of information that is constant. The operation of an optical disk is similar to that of a magnetic disk. When reading, a highly focused, low power laser light source is focused onto an optical medium (reflective Mylar) which is laminated onto a carrying substrate. This medium either reflects the light (a one) or deflects the beam (a zero). The physical size of the disk surface required to store one bit, and the high accuracy with which a laser light source may be focused, conspire to make the data density of the disk very high.

The writing of an optical disk is performed by either "burning" a region of the disk with a high power laser, to cause that region to scatter light that strikes it, or by physically scoring that region with a stamping machine (used for mass production only). In either case that point on the disk is physically altered and cannot be changed. This leads to one of the major drawbacks of optical media as it currently exists; it can only be written once. This Write Once Read Many (WORM) mode of operation is fine for the archiving of imagery, but it is not well suited for some other aspects of imagery exploitation.

- Rewriteable optical media is an evolving technology, which allows for the restoration of regions of the reflective media through the application of mild heat and a dense magnetic field. To date these systems have proven prohibitively expensive, but recently market forces have begun to move the cost of these technologies downward. It is probable that a large difference will remain in both price and data density between these devices and the standard WORM and CD-ROM technologies for the near future.

- Optical tape is another currently emerging technology. The optical medium is applied to a flexible backing, and rolled onto a reel-like magnetic tape. This allows for an extremely high data density per cubic area of storage. This technique reduces the rate at which a particular portion of the medium may be accessed, but for achieving extremely high physical density, which may be the case for some image archival systems, this techniques is a currently available option.

- Magneto-optical disk. As discussed in the magnetic disk section, the ability to position the head precisely over the disk is critical to the maximum data density that the disk can achieve. There are a number of systems for using optical markings placed physically on the disk surface as landmarks to guide the magnetic head. This technique automatically compensates for wear in the bearings and other mechanical parts, and allows closer tolerances (and thus more data on a singe disk).

2.3.1.4 RAID AND SPECIAL SYSTEMS

There are a number of data storage solutions, which are not so easily classified as those discussed above. The first of these is the Redundant Array of Inexpensive Disks (RAID) system. RAID represents the state of the art in magnetic mass storage systems, and will be one technology under consideration in this effort.

In a RAID system a number of physical disk drives are ganged together such that they appear to the host to be a single large device. Individual data elements are spread among the physical devices in a redundant fashion such that the loss of any individual physical drive will not affect the integrity of the data. When a drive fails it can be removed and replaced with an empty drive (hot-swapped). In addition the controller is able to restore from the redundant locations all of the data that was on the failed drive. Within a short period of time the data is fully redundant again and safety is restored.

RAID systems come in many varieties, which are designated by RAID levels (0 through 5). RAID 0 denotes systems where each drive has an exact mirrored twin. The twin can be used in the event of a failure of the master, and there is no performance penalty for using such a system. There is, of course, a cost penalty since each piece of data is stored twice.

All other RAID levels store the data in pieces, generally getting larger from RAID 1 (bits), on up the line to RAID 5 (blocks). In addition an increasingly complex parity structure is stored on the drives. In the event of a failure of a disk the data is reconstructed using the parity codes. In such a case the data is not truly redundant, but the cost savings are obvious since the parity information is between 1/9th and 1/5th the size of the original data. Of course a performance penalty is paid due to the complex nature of the parity calculations and the disk head thrashing required to distribute the data.

Different RAID levels are best for different transaction types. RAID 1 and 2 are good for general operations. RAID 3 is well suited for large data blocks that are read often but written seldom. RAID 4 and 5 are used when complete fault tolerance is required, but performance is not crucial. The MSS will allow for experimentation into the appropriateness of various RAID levels of any RAID system installed.

2.3.1.5 HELICAL SCAN TAPE

Other mass storage considerations might include the use of helical scan tape as a substitute for archival (low access rate storage) memory. This technology is quite mature, and represents data storage densities that are very favorable. The physical properties of tape generally make it undesirable for random access mass storage, however the MSS system will allow for the inclusion of such devices to see whether they can be effective in the imagery exploitation environment.

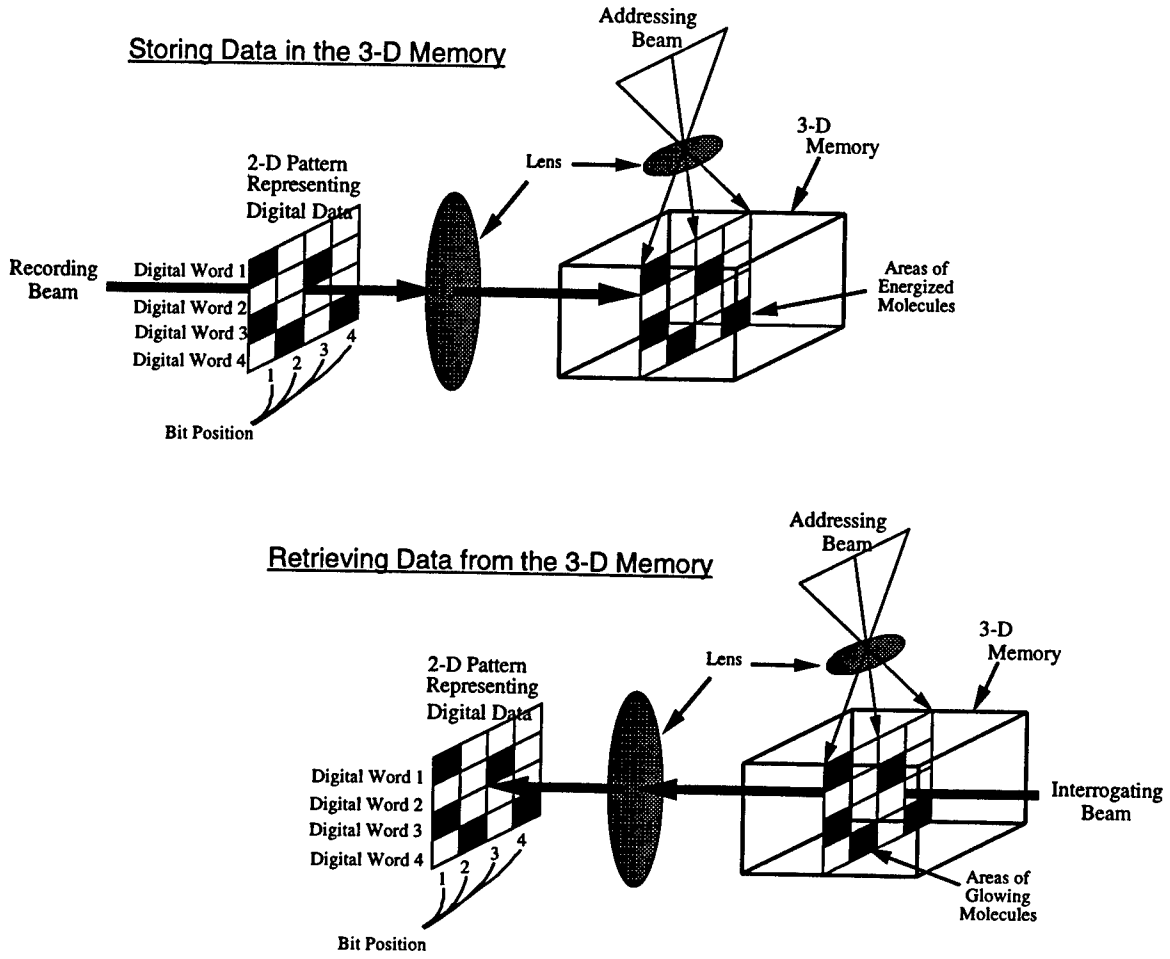
2.3.1.6 3-D VOLUMETRIC MEMORY

All of the current COTS recording technologies are basically two-dimensional recording systems (magnetic, optical and solid state), and these systems become very large when mass storage systems are created. To reduce the physical size of mass storage systems AFRL, Rome Research Site is currently investigating a three-dimensional (3-D) storage technique. Call/Recall, Inc., San Diego, California is developing this storage system for AFRL. This 3-D digital recording technology employs an organic molecule that changes its optical state when illuminated by two different optical sources. This state change can then be viewed as an orange glow when illuminated by an interrogating optical beam.

This innovative method of 3-D storage operates in the following manner (see Exhibit 4). Digital data is represented as a two dimensional (2-D) pattern which is projected into the cube by a recording beam. An addressing beam orthogonal to the recording beam intersects the recording beam in a very narrow 2-D plane or slice through the cube. This fixes the 2-D pattern by energizing the molecules in the area where the two beams intersect. Other 2-D patterns can be

recorded at other planar locations by refocusing the recording beam and intersecting that plane with the addressing beam to create a 3-D storage capability. To retrieve each 2-D pattern the addressing beam is used to illuminate a given-recorded plane, which causes each area of energized molecules to emit an orange glow. This orange 2-D pattern is then optically focused onto a 2-D detector and the optical data is then converted into digital data for digital computer processing.

Exhibit 4. 3-D Volumetric Memory



Despite its obvious advantages, 3-D volumetric memory technology has not yet reached a point where it can be included in the MSS project.

2.3.1.7 *Storage Technology Summary*

Table 1 is a summary of our review of the various methods for storage and retrieval of digital data. Shown in this table are the primary advantages and disadvantages of each technology as well as the reason for selection for integration into the MSS.

Table 1. Storage Technology Summary

Storage Technology	Primary Advantages	Primary Disadvantages	Potential for MSS
Semiconductor Memory Devices	Tolerant of rugged use	Volatile	None
Disk-Based Magnetic Media	Large commercial base	Not tolerant of rugged use	None
Optical Storage Media	Archival and tolerant of rugged use	Limited read/write and limited capacity, but improving	Good for large archive and rugged use
RAID	System reliability	Slow access and retrieval	Good for system reliability
Helical Scan Tape	Large capacity	Very slow access and retrieval	None
Magneto-Optical Disk	Archival and tolerant of rugged use	Limited capacity, but improving	Good for large read/write and rugged use
3-D Volumetric Memory	Very large capacity archival, and tolerant of rugged use	In development	The best, but still in development

The use of a hybrid system provides the most cost-effective balance between speed and capabilities with the technologies foreseen throughout the 1990s and into 2000. An example case would be to use optical storage for actual imagery archive data (perhaps a jukebox system), and higher speed magnetic disk farms or RAID arrays to store data on current political hot-spots (or other areas of interest to analysts). The complex nature of such systems, coupled with the changing nature of analyst activities, makes it imperative to determine the optimal configuration prior to actual implementation.

2.3.2 Hierarchical Storage Management

Ideally all information could be stored using high-speed devices providing near instantaneous availability. Unfortunately, current technology makes this approach prohibitively expensive in mass storage situations. Hierarchical Storage Management (HSM) systems automatically move data between storage media layers, monitoring data usage and using rules to determine which data and when should be moved to lower levels of storage. The desired effect of an HSM system

is to provide data management which maximizes the ability to balance high speed, high cost per density storage options with lower performance, low cost per density media. All the workings of an HSM are transparent to the user.

This section will cover the benefits of an HSM, terminology, architecture, and the misconceptions preventing HSMs from becoming more widely used.

2.3.2.1 OVERVIEW

Today's local area networks (LANs) are growing at an unprecedented rate. Increasingly larger numbers of users with highly sophisticated applications are creating files, which, from both a size and volume aspect, are straining the resources of data storage devices. Data are not only in paper form. Voice mail, electronic sticky notes, email, video clips, sound files, and photographs are among the data bogging down computers and networks. The results are huge storage repository and timely access requirements. Traditionally network managers threw more disk drives on the network, or with the advent of personal computers joining networks, adding another, larger hard drive. Even with the moderate, and ever declining, costs of disk drives this action is a very expensive "band-aid fix."

In addition to not being a very cost effective storage technique, throwing more storage on each local machine is not the answer. It causes more data management problems than it solves. For the network manager, distributed data management is problematic: routine backups across network, particularly mixed platforms, become more complicated; naming conventions are not standard across platforms; some drives will be published, while others are not. Furthermore, with a move from centralized data management to enterprise data management, the organization starts to lose tight control over its data. Some of the concerns and risks include file version management, excess storage capacity, disaster recovery, and security implications. With multiple users having multiple local copies, tracking which is the most current, most correct version is not a straightforward determination. Configuration management procedures need to be strictly enforced. Multiple copies scattered about the network create a good deal of wasted storage space. Additional disk drives required to maintain duplicate files translate into high, unnecessary costs. Multiple copies also have disaster recovery implications: due to accidental loss of data or system failure, determining which version is to be used becomes an issue. However, the security risks constitute the most serious concern of distributed data management. Files on local storage are harder to track, opening the door for theft and intentional, malicious corruption.

The reality is that data availability decreases as the amount of storage increases. Adding more centralized drives helps the management cost, but the media costs are high. Management costs increase and productivity decreases with the amount of stored data. Management costs include:

- ❑ User disk and file management
- ❑ Productivity losses from storage failures
- ❑ Disaster recovery costs
- ❑ Installation, repair, and service contract costs

The actual costs of the devices themselves are decreasing while management and associated costs are increasing. Yet, despite the need to maintain access to the data, an overwhelming majority of the data is rarely accessed again. The solution is a Hierarchical Storage Management system.

2.3.2.2 HSM BENEFITS

The benefits of hierarchical storage management are many. Most data is infrequently accessed. Migrating less active data to secondary or even tertiary, storage devices results in a more effective, efficient use of available space on the primary data disk drive and an optimized access time for the most important data. By freeing up space on the primary drives, HSMs with their use of comparatively slower secondary and tertiary storage devices, reduce the overall cost of data storage. Network managers are released from the onus of managing storage capacity and able to reallocate their time to network performance. The costs of network managers far outstrip the cost storage devices. With the use of file pointers (stubs), the user maintains transparent access to the data, despite its physical location. This results in higher productivity, which in turns lowers the cost of doing business.

2.3.2.3 HSM TERMINOLOGY

Hierarchical storage management is an automated storage management system that can be configured to intelligently place data in storage repositories, moving it through a series of storage devices according to immediacy of need. More frequently accessed data is placed on high performance, high quality disk storage. Mission critical data should remain in this active storage environment. Less frequently accessed data is moved to slower, less expensive rewriteable optical, tape, or CD-ROM. Because it is less actively accessed, the slower access speeds are a minor issue.

The following terminology is used to describe the workings of an HSM:

- HSM cache — fast, expensive hard disk storage used for critical data
- Migration — copying of data from the HSM cache to the level directly beneath the cache. All files are migrated at least once to guarantee copies exist. Migration does not remove any information from the cache; it prepares for data purging and insures data safety. Automatic migration generally occurs under two conditions: (1) when the cache contains a specified number of unmigrated files, and these files have remained resident and unchanged in the cache for a specified time; and, (2) when migration has not run in a specified period of time, regardless of the number of unmigrated files. Immediate migration allows for manual movement of specified files irrespective of the above conditions.
- Vaulting — migration to layers deeper than that directly below the cache. Vaulting provides the path to successively lower levels in the hierarchy. This ability allows for greater flexibility in the storage options in a hierarchical system and allows for the repacking of data in tape based systems to permit recovery of partially used media.

- High watermark — network administrator determination of when automatic migration begins. This is usually a specified percentage of the cache.

- Low watermark — network administrator determination of when automatic migration ends.

- Purging — process of deleting data from the cache; it exists as a method to regain space in the HSM cache. All files to be purged must have been previously migrated to assure the existence of a current copy. Purging occurs when the high watermark is met and continues until the low water mark is reached. The high and low watermarks are configurable to meet the demands of the situation.

- Staging — movement of files from layers beneath the cache to the level of the cache. The movement is direct from the resident level to the cache; no intermediate layers are involved. Common methods that trigger HSM staging include network file system (NFS) and File Transfer Protocol (FTP) data access.

- Stub — pointer to the new location of the file, which provides the appearance of transparency to the user.

The supported automation features of an HSM provide the flexibility required to configure a hierarchical storage solution to best meet the needs of a particular application.

2.3.2.4 HSM ARCHITECTURE

Hierarchical systems typically distinguish three distinct layers of storage referred to as on-line, near-line, and off-line. These storage levels provide the means by which mass storage systems can balance needs for performance and capacity against the desire to minimize cost. As Exhibit 5 shows, lower capacities of high cost per density, high speed storage are located at the upper end of the hierarchy while greater amounts of lower cost per density, lower performance media occupies the lower levels.

On-line storage refers to the highest level in the hierarchy and usually consists of magnetic disk-based media. On-line data is accessible to the user without any intervention by the HSM system. Once information is positioned at this layer, HSM overhead factors no longer impact data storage or access times. A user's local or network accessible hard drive storage provides the most common example of on-line storage.

The term near-line references the layer beneath on-line storage. While HSM intervention is required to access information stored at this level, most hierarchical storage configurations strive to minimize potential throughput degradation by employing higher performance magnetic or optical media. This is the level at which HSM attempts to anticipate data needs. Every HSM imposes overhead requirements at this level for data management information. An example of near-line storage would be a magnetic disk based RAID serving as a moderately high

performance, and fault tolerant, layer between high performance disk and lower performance archival media.

Off-line storage occupies the layer (or layers) below near-line and consists of the slowest least expensive storage media in the hierarchy. Typically reserved for archival usage, off-line storage's greatest benefit is its ability to offer large capacity at lower cost/density. Historically this level required manual intervention and management (e.g., tape storage/loading and inventory management). The introduction of robotic jukebox systems has supplanted the need for human intervention to a large degree. Additional successively larger but slower layers of off-line storage may be implemented to maximize usage of available storage options and further decrease storage costs. Off-line storage options consist primarily of high capacity optical jukebox or tape-based systems.

Exhibit 5 also demonstrates the variety of data movement between hierarchical layers available in a typical HSM configuration. These manipulations generally occur automatically when triggered by user defined parameters. Optionally, system administrators can implement the actions manually to meet special needs such as maintenance or testing situations.

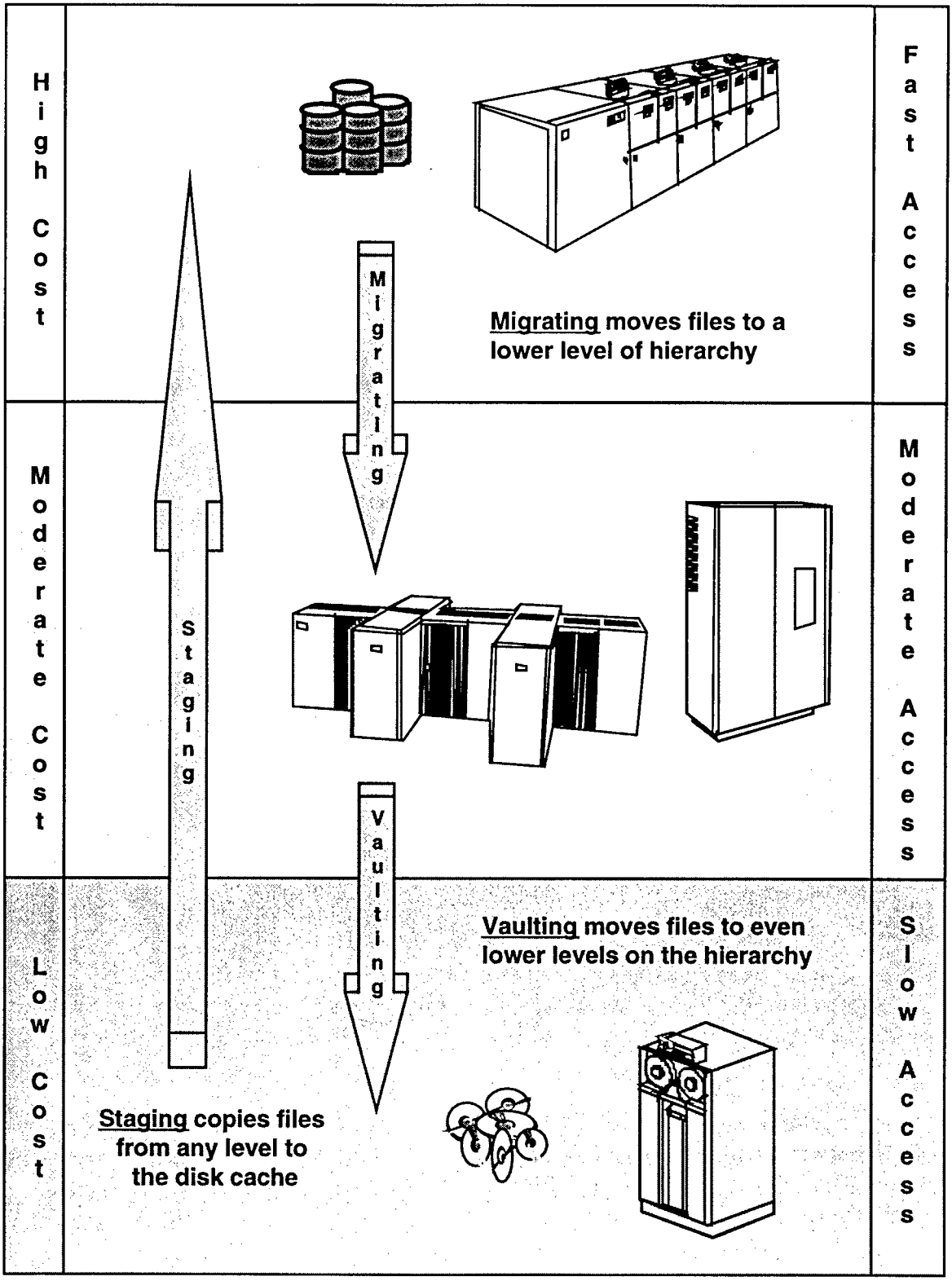
Network managers establish policies that define when the data is migrated in the high watermark and low watermark attributes. Most define the criteria for files selected based on such attributes as the last access time, minimum file size, file owner, file group, and files exempt from migration such as executables or DLLs, which can result in network OS performance damage if removed from the original locations.

2.3.2.5 HSM Misconceptions

The HSM's strongest selling point is that it maintains the user's references to the data, automatically recalling data as files are accessed. To the end user it is transparent, albeit slightly slower when the data is located off-line. So why isn't an HSM system installed in every data-intensive organization? There are three main reasons. First, hierarchical storage management systems have their roots in the mainframe world. The dynamics of the mainframe/data center are not the same as that the more modern distributed client/server environment. Issues not considered when implementing a hierarchical storage management system include network bandwidth, varying tape speeds, and heterogeneous platforms and operating systems. Today's typical distributed computing environment does not consist of homogeneous platforms and operating systems. Many organizations are mixing personal computers, WindowsNT servers, and UNIX workstations. While platform-independent HSM software is beginning to emerge, it is still in its infancy.

The second reason is due to confusion between hierarchical storage management, backup, and archiving. Confusion between the three data management strategies have resulted in organizations not moving to an HSM, believing it to be redundant and therefore unnecessary. While HSM is complimentary to the backup and archiving of data, they are not identical.

Exhibit 5. Data Movement between Hierarchical Layers



- Backup — the process of storing copies of data on media, while leaving the original in place. Network managers perform backups on a regular schedule of the entire file system. Backups ensure that data can be retrieved in the event of a system failure, disaster, or simple carelessness.

- Archival — the process of storing data to which regular access is no longer needed. Typically, archival and retrieval are a manual process. The media used for archiving must be safe and durable. Files are removed from the system after being archived, freeing up disk space.

The goals of archival, backup, and HSM strategies differ.

- Archival — provides extra protection in the event of data loss and conserves on-line storage space. There is no regular schedule implied with data archival. Most commonly, it is performed at the end of a project.

- Backup — protects against accidental data loss or damage. Backups are done on an established schedule and are automatic.

- HSM — conserves network storage, while providing access to all data. Migration and recall of data is automatic.

With an HSM system in place, however, data recovery systems manage only the active data set, making full backups quicker and able to be schedule on a nightly basis if required.

The third barrier to the acceptance of HSM is the entry cost. The initial cost of purchasing HSM software and multiple levels of storage devices is quite high. This is especially true when installation costs are considered. Any organization that doesn't look at the full lifecycle costs of their mass storage architecture will conclude that HSM is prohibitively expensive. However, those that perform this analysis will conclude that the savings, due to reduced media and management costs, are substantial over the mid-term and long-term.

2.3.3 BENCHMARKING

Benchmarks provide the predominant method for evaluating computational systems from a performance standpoint. These programs use simulated workloads to quantify actual system metrics. Extrapolation of benchmark results can predict actual system operating characteristics.

2.3.3.1 TYPES

Benchmark programs fall into two general categories that target component and system (or application) performance. Model and synthetic workloads comprise the two different approaches to evaluating system performance.

Component benchmarks focus on the performance of isolated aspects of a larger system. Common component types are CPU, memory, I/O, graphics, and network resources. Component benchmarks are popular because the results are usually easy to interpret and provide clear comparisons between components in the same category. Care must be taken in extending the meaning of component benchmark results so as not to infer overall system performance from isolated measurements. System benchmarks evaluate the performance of many components working together. They generally provide a more realistic picture of the performance that the user will encounter.

Model workloads are based on actual executable code appearing in the applications making up the test scenario. They accurately mimic usage of a system and therefore provide a high degree of fidelity and predictive accuracy. Model workloads apply to very specific situations and achieving the proper mix of system demands requires careful planning and implementation. Synthetic workloads attempt to simulate resource consumption by creating software that statistical portray actual situations. Because the workload level can be varied while maintaining a constant operation mix synthetic workloads are particularly adept at determining peak performance figures.

2.3.3.2 PITFALLS

A variety of situations exist that can lead to incorrect conclusions when evaluating systems based on benchmark information. In some cases these errors are caused by misinterpretation of results and in others the culprit is misunderstanding of system configuration impact. Some brief examples of common mistakes follow.

The effect of competing processes in a multi-tasking environment often causes inconsistent or undependable benchmark results. This problem impacts evaluation of both individual platforms and 'machine versus machine' testing. Situations as obvious as working with a word processor during the running of a time consuming benchmark to the more subtle effects caused by background processes (e.g. terminate and stay resident programs such as virus software) can lead to incorrect assessment of system performance. Providing a quiescent or reproducible environment is paramount in benchmark driven evaluations.

Care must be taken to match selected benchmarks to the scenario under test. Basing decisions on results generated by an I/O intensive benchmark makes little or no sense if the primary focus of testing is to determine the best candidate for running a graphically intensive application. Similarly, the makeup of a benchmark's workload should reflect the conditions actually expected.

Wide variations in performance results can often indicate an unfair comparison caused by testing systems or components that are fundamentally different due to underlying technology. Speed comparison between SCSI and non-SCSI devices illustrate an example of this 'apples to oranges'

situation. In such a case, the competing technologies are not comparable and other factors, for example cost, should weigh more heavily in the decision making process.

Performance measurement is a complicated issue and should not be described by a single number. Approach benchmarks that average results into a single score very carefully. Considerable information can be lost or obscured when averaging individual performance statistics. While it may mean more work, evaluation of specific results focusing on the areas of interest will provide a more valuable and accurate picture.

2.4 PURPOSE

The objective of the Mass Storage System task is to provide AFRL, Rome Research Site, with a capability to develop concepts to overcome the current lack of mass storage alternatives that are both suitably fast and cost effective. Our solution to this dilemma uses a hierarchical hybrid system to build a virtual device to provide a mass storage capability in an open systems architecture. We also propose to test and evaluate the performance of these concepts for storing and retrieving large volumes of data such as digital imagery, signal data, multimedia data, etc.

This type of hybrid system provides the most cost-effective balance between speed and storage capacity using state-of-the-art technologies that will be available throughout the 1990s and into 2000. The complex nature of such systems makes it imperative to determine the optimal configuration prior to implementing a mass storage system for operational use. Providing a modeling capability to postulate various candidate systems and user performance requirements in a distributed network environment will meet this demand and also insure that the laboratory performance environment of the MSS is not misleading.

2.5 DOCUMENT OUTLINE

Section 3 of this document details the methods, assumptions, and procedures for accomplishing the objectives of the MSS effort. Section 4 contains results and related discussion pertaining to testing phases of the project. Section 5 presents Synectics conclusions. Section 6 describes recommendations for further courses of investigation and action relating to Mass Storage System development.

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

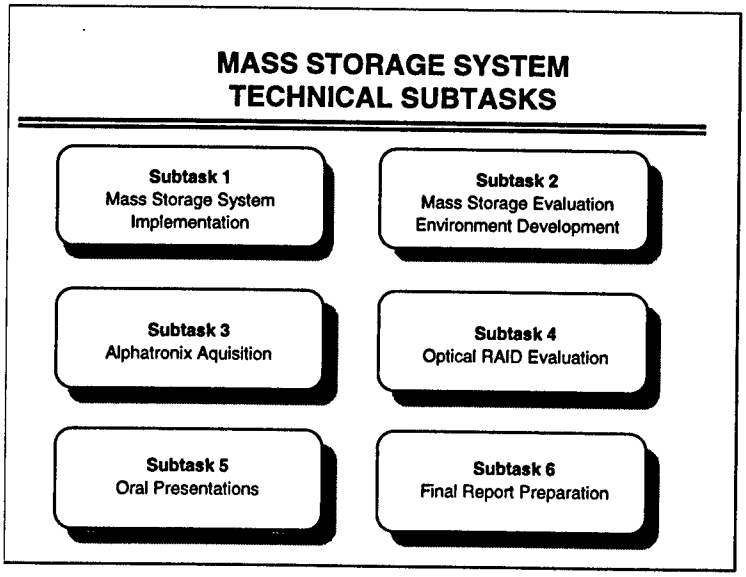
This contract consisted of two primary tasks. First, identify and integrate the components needed to construct a prototype hierarchical mass storage system (PHMSS), and second, develop a capability to effectively test the resulting system. Data collected from these tasks provide performance statistics for analytic and comparative evaluation of the PHMSS and other available systems.

Our approach to the first task centered on assembling an optical jukebox to serve as an archive, coupled with a higher speed RAID array, to store more current data. Completing the system was a workstation that hosted both the hardware and HSM software needed to manage the system as a cohesive unit. The overriding goal for this task was proving that high speed, high cost storage could be blended with lower speed, lower cost storage to provide an effective balance between cost and performance. The result of Synectics efforts was a mass storage system capable of handling up to 1/2 terabyte of data.

The second task consisting of the test and evaluation portion of the MSS effort centered on development of a Mass Storage Evaluation Environment (MSE2) software tool. This custom application was designed to provide both a testing capability and a tracking and analysis capability. MSE2's goal was to incorporate benchmark programs with an underlying database.

Six distinct subtasks, as shown in Exhibit 6, were undertaken to achieve this goal. Technological tasks 1 and 2 were developed concurrently throughout the life of the project.

Exhibit 6. MSS Subtasks



The remainder of this section describes each subtask and the methods used to develop the required storage and testing capabilities. Procedures used and assumptions made are also documented.

3.1 MASS STORAGE SYSTEM (MSS)

Implementation of a prototype hierarchical mass storage system required the careful analysis of existing technologies to determine an architecture that would provide the desired balance between cost and performance. Once the system was developed, extensive testing was accomplished to determine the systems operating characteristics. Less technically oriented but equally important was the identification of issues and concerns important to operational users when discussing mass storage needs and the potential of the PHMSS as a solution. The following discussion centers on the implementation of the architecture, the three distinct phases of testing and evaluation, and the Mass Storage Management Interface (MSMI) effort. Please refer to Exhibit 7 for the MSS development timeline.

3.1.1 ARCHITECTURE

The theoretical design of the PHMSS was based on an optical jukebox, to serve as an archive, coupled with a higher speed RAID array, to store more current data. Completing the system was a workstation to host both the hardware and the HSM software needed to manage the system cohesively. Once the overall architecture had been defined investigation of existing technologies was accomplished.

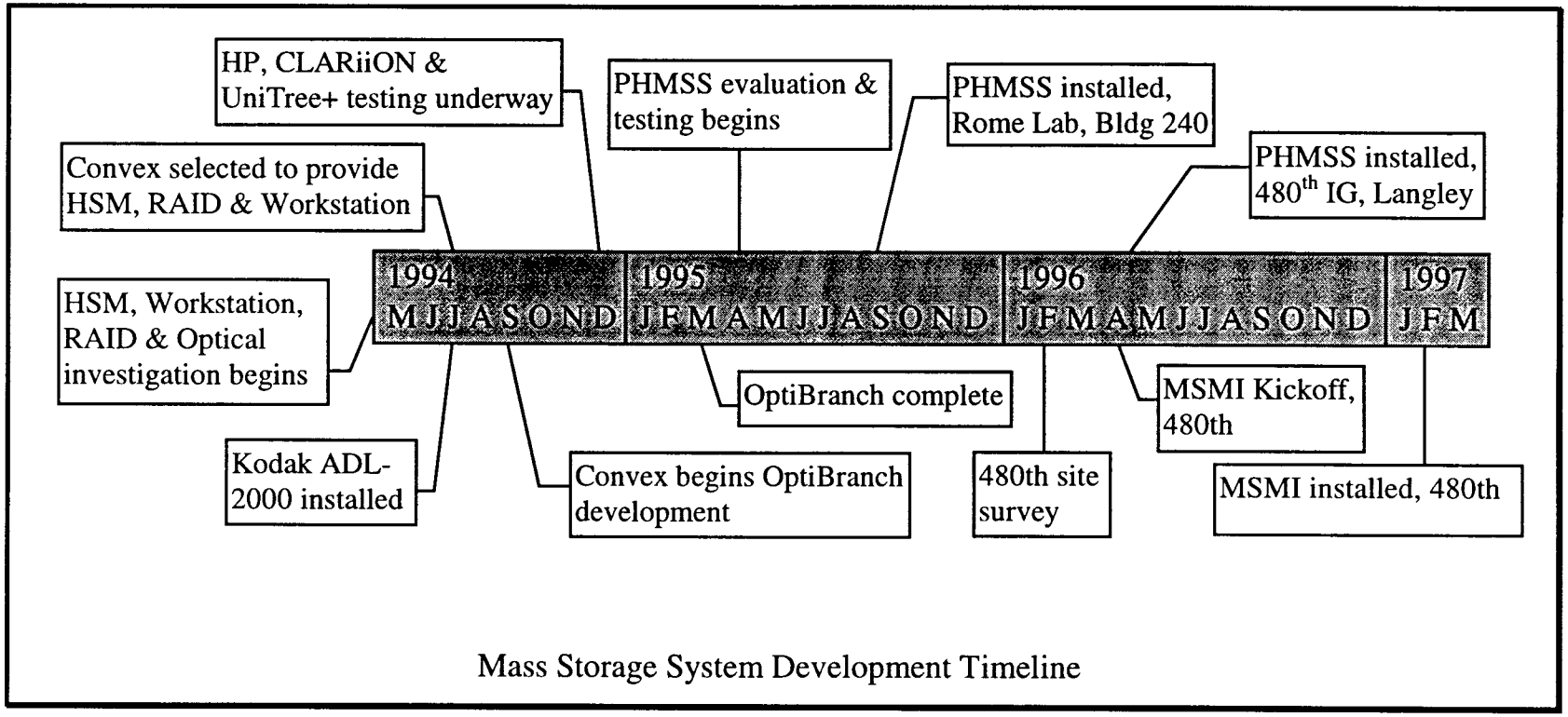
3.1.1.1 HARDWARE

A large number of vendors were contacted for preliminary information. This led to the scheduling of meetings with those companies offering products in areas applicable to MSS. Final hardware and software selections were made after evaluation of the potential candidates. The decisions were driven by a number of requirements including:

- Interoperability
- Scalability
- Cost

Selection of the Kodak ADL-2000 was driven by the lack of any competing optical jukebox systems that provided comparable storage capacity or product maturity. Synectics procured a jukebox that initially provided ½ Terabyte of storage to be used in the implementation and testing of the PHMSS.

Exhibit 7. MSS Development Timeline



Mass Storage System Development Timeline

After careful consideration, Convex Computer Corporation was chosen to provide the RAID and workstation components for the PHMSS. Convex provided a number of benefits such as existing mass storage testbed facilities and their ability to provide a highly scaleable server solution. The Data General CLARiiON RAID, with a 20GB capacity, was selected to provide the magnetic cache for the PHMSS. The CLARiiON provides unmatched performance, fault tolerance, flexible configuration, and industry acceptance. The Hewlett Packard (HP) Series 755 with 2GB hard disk, 192 MB RAM, and support for Fast and Wide SCSI II, Ethernet, and FDDI network connectivity, was chosen as the host platform for PHMSS. Convex provides an upward compatible product line starting with the HP and going all the way to parallel supercomputer solutions.

3.1.1.2 Hierarchical Storage Management (HSM) Selection

In mid 1994 only two HSM packages, Kodak's MultiStore, and Epoch Corporations Infinistore supported the Kodak ADL-2000. Both of these solutions were deemed to be unsuitable primarily due to product immaturity (Kodak) and lack of desirable support (Epoch). Subsequent discussions and evaluation (please refer to Appendix A for details) led to the choice of OpenVision's UniTree as the HSM product for MSS. UniTree was chosen because of its high speed performance, its ability to overcome file and file system size limitations imposed by most operating systems, its position as the de facto standard in large mass storage systems, and its good product stability. OpenVision markets UniTree through a number of Value Added Resellers. The Convex Computer Corporation version of UniTree, UniTree+, was chosen due to the fact that OpenVision touted it as the most stable and highest performing implementation. Other factors included the scalability and interoperability provided by a Convex solution, and the high level of continuity gained given the choice of Convex as the PHMSS host and RAID supplier.

To solve the problem of UniTree not supporting the Kodak ADL-2000, Synectics coordinated an agreement between Synectics, Kodak, and Convex to enhance Convex's version of UniTree+ to incorporate the need functionality to support the Kodak jukebox. The resulting capability, known as OptiBranch, provided the first implementation of UniTree to support large format optical media. OptiBranch also demonstrated a great case of dual use development and technology transfer.

3.1.2 Test and Evaluation

Test and evaluation of the PHMSS was accomplished in three stages. The initial phase was carried out at Synectics' Rome, NY facility. More formal testing was done at AFRL, Rome Research Site facilities. Finally the PHMSS was relocated to Langley, AFB for inclusion into an operational setting.

3.1.2.1 *INITIAL TESTING*

By the end of 1994 the individual components of the PHMSS had arrived at Synectics facilities in Rome, NY. Following installation of the ADL-2000 by Kodak and delivery and setup of the CLARiiON RAID and HP 755 workstation by Convex, and while awaiting completion of the UniTree+ modification by Convex, system familiarization and preliminary network modeling and simulation work took place.

Synectics' development personnel utilized the 20 GB RAID as an NFS-accessible storage device to obtain a preliminary notion of its abilities. Removing RAID drives and disabling redundant components during operation induced simulated failures. The HP 755 was also put through its paces to gain background in the HP-UX operating system. A number of SunOS based benchmarks were ported and compiled under HP-UX to provide I/O performance statistics prior to installation of the HSM. Throughout this phase the equipment performed flawlessly and was able to cope with every attempt to expose weaknesses.

Network modeling and simulation was also undertaken during this period. Modeling the performance of a virtual device based on heterogeneous components is far more difficult than with traditional storage devices. COMNET-III from CACI Products Company was selected as the tool to enable effective prediction of performance characteristics of proposed configurations and the impact of changes to existing systems without having to actually implement the considered changes.

After a 60-day trial period, which included a 4-day training course, initial modeling of the PHMSS began focusing on a number of activities such as:

- ❑ Device load balancing
- ❑ Network planning
- ❑ File migration strategy planning
- ❑ Cost/Benefit analysis
- ❑ Performance prediction/validation

The results of this modeling and simulation work provided a modeling scenario to provide preliminary modeling task results for the March 1995 480IG status update briefing.

Upon the completion of OptiBranch, Convex delivered and installed the modified UniTree+ HSM software. Synectics then received training in the operation and configuration of UniTree+ and the Kodak ADL-2000. Storage and retrieval of Synectics operational files confirmed the operational status of the PHMSS as a fully integrated mass storage system under HSM system control. With confidence in the basic operational characteristics and the proven ability to access

and utilize the optical jukebox, final preparations began for moving the PHMSS to AFRL, Rome Research Site.

3.1.2.2 INTERMEDIATE TESTING

Kodak was hired to perform an official de-install and install of the ADL-2000 jukebox for the move to the Intelligence and Cartographic Facility, Building 240, AFRL, Rome Research Site. Several minor problems were encountered upon reinstallation including a bad Read/Write head in the optical drive. These discrepancies were resolved and proper operation of the jukebox was verified. The remaining PHMSS components were transported and reinstalled by Synectics uneventfully. AFRL provided dedicated Ethernet and FDDI LANs including two Sun workstations for use in evaluating the PHMSS.

More formal performance measurement tests were developed and undertaken during this phase of the effort. The focus was primarily on the effect of the HSM and the access speed of the ADL-2000 on the overall system. A test suite of variously sized data files were generated and accessed using all available network paths to assess the NFS and FTP data throughput rates. These figures were recorded and formed the foundation for much of the technical data presented to the 480IG during demonstrations and briefings.

During performance test and evaluation the UniTree+ software exhibited a random tendency to spawn runaway processes. Synectics worked in concert with the Convex UniTree+ modification team to isolate and resolve the problem. A bug was discovered in the modified UniTree+ software and a patch was written and installed. Thorough exercising of the system was accomplished to verify that the problem no longer existed.

In anticipation of final relocation of the PHMSS to the 480IG, Langley AFB, Va., a pre-installation site survey of their facility was accomplished. Data was collected on available space, power, physical access, and cooling. There were no factors discovered that would impact delivery and installation of the PHMSS.

3.1.2.3 OPERATIONAL TESTING

The proven system was trucked by Synectics to the 480IG, Langley AFB, Virginia to undergo final integration in an operational environment. Kodak technicians performed the on-site installation of the ADL-2000. No problems were discovered following the move. The remainder of the system was assembled and tested by Synectics with no discrepancies noted.

Synectics provided the designated PHMSS system administrator with initial hands-on demonstration and training. It was recommended that the administrator attend the Convex-sponsored UniTree+ training program to gain a more robust understanding of the system. Synectics provided the necessary technical expertise needed to fully integrate the PHMSS within the 480th operational scheme.

3.1.3 MASS STORAGE MANAGEMENT INTERFACE (MSMI)

The 480th expressed concern that the command line interface for accessing the UniTree+ HSM functionality could hinder integration of the PHMSS into their operational environment. This issue was raised toward the end of intermediate testing of the PHMSS. Synectics proposed development of a graphically oriented application to provide a Mass Storage Management Interface (MSMI) to alleviate this perceived deficiency. The 480th agreed to the idea and initial concepts and requirements were discussed during the PHMSS delivery trip.

At the MSMI kickoff briefing held at the 480th, Synectics presented the MSMI preliminary design that consisted of an HTML based interface which would enable access to all administrative functions of the HSM. Additional features such as the capability to track and manage files stored in the PHMSS through the use of user supplied metadata was also discussed. The concept for metadata management of files would also be available for use in the management of non-PHMSS resident files. The briefing was well received and the 480th promised to provide Synectics with additional user needs and ideas as they arose.

Synectics developed and implemented a Beta level version at their Rome facility. MSMI was developed as a browser based Client/Server application utilizing the NCSA HTTPD web server, a Sybase database, and Netscape Navigator browser. All PHMSS users would be able to contribute to, access, and manipulate the metadata based upon ownership and security rules. These rules were to be specified by the 480th as they defined their need. The PHMSS administrator would also be able to access the UniTree+ facilities to provide manual control of data stored on the PHMSS as well as perform all required routine administrative tasks.

When initial capabilities were achieved, Synectics returned to the 480th to brief the status of and install the initial release of MSMI on the PHMSS. The 480th PHMSS administrator assisted in the installation and received instruction regarding the design, structure, and use of MSMI. Synectics also delivered a draft MSMI user guide.

3.2 MSE2

The second largest task in support of MSS was development of a Mass Storage Evaluation Environment (MSE2). A tool was needed to measure performance statistics at both the device and network levels for mass storage components. To facilitate this task Synectics developed software to integrate a wide range of public domain benchmark programs with an underlying database. The details of this effort are described in the following sections.

3.2.1 DESIGN

Overall design of MSE2 consisted of three primary components.

- The graphical user interface (GUI)
- The underlying database
- Public domain benchmarks

The goal was to provide a system capable of providing an environment to run a variety of benchmark software while eliminating the need of requiring the user to become familiar with the intricacies of each individual measurement package. In addition, the ability to store and later retrieve previous test results for comparison and analysis was desired.

The identified target system for application deployment was the SunOS 4.3 operating system running on Sun workstations. The decision was made to use the 'C' language for overall program development. Motif and Sybase were selected to support the GUI and database components of MSE2. All development and testing was accomplished at Synectics' Rome facility utilizing existing hardware and software.

With the fundamentals in place, high level design of the system began. The first task was to identify candidate benchmark programs. Over 150 public domain packages were evaluated for suitability to the project. Fifty of these were identified as having functionality applicable to the testing of the MSS. They were downloaded from FTP sites, unbundled, and sorted by functional category. After compilation and testing the 19 most suitable benchmarks providing test capabilities for data throughput, processor performance, graphics performance, and network performance were selected.

After the benchmarks were chosen (please refer to appendix B for additional details) and their respective outputs analyzed, the design of the user interface and underlying database started. Besides facilitating the selection, configuration, and execution of benchmarks, and storage of the generated results, effective means for establishing and tracking system architecture profiles were considered. These factors directed the focus of independent rapid prototyping of both the GUI 'look and feel' and database schema.

3.2.2 IMPLEMENTATION

Once the basic design philosophy was decided, actual implementation of MSE2 was undertaken. This consisted primarily of integrating the user interface, database, and benchmark applications into a single cohesive system. The I/O benchmark Bonnie was chosen as the first test case due to the programs relative ease of use, representative configuration choices, and output structure,

The source code for Bonnie was modified and recompiled to redirect output to the database. Additional code was incorporated into the GUI portion of MSE2 to tie everything together. Testing of MSE2 was performed using locally available storage devices and MSS equipment. While the successful completion of this test case provided the basic procedure for integration of the remaining 18 benchmarks, modifications and design changes were made to the entire application throughout the remaining benchmark integration phase which lasted until mid 1995.

3.2.3 USE

MSE2 was used primarily while the PHMSS resided at the Synectics facility. During the initial equipment installation and test phase, MSE2 proved to be a valuable tool not only for verifying performance but also for quantifying the impact of configuration changes to the PHMSS. The lack of a readily available platform and logistical problems with Sybase access hindered the relocation of the MSE2 to AFRL, Rome Research Site. Possible solutions for this and other issues regarding MSE2 are provided in section 6.

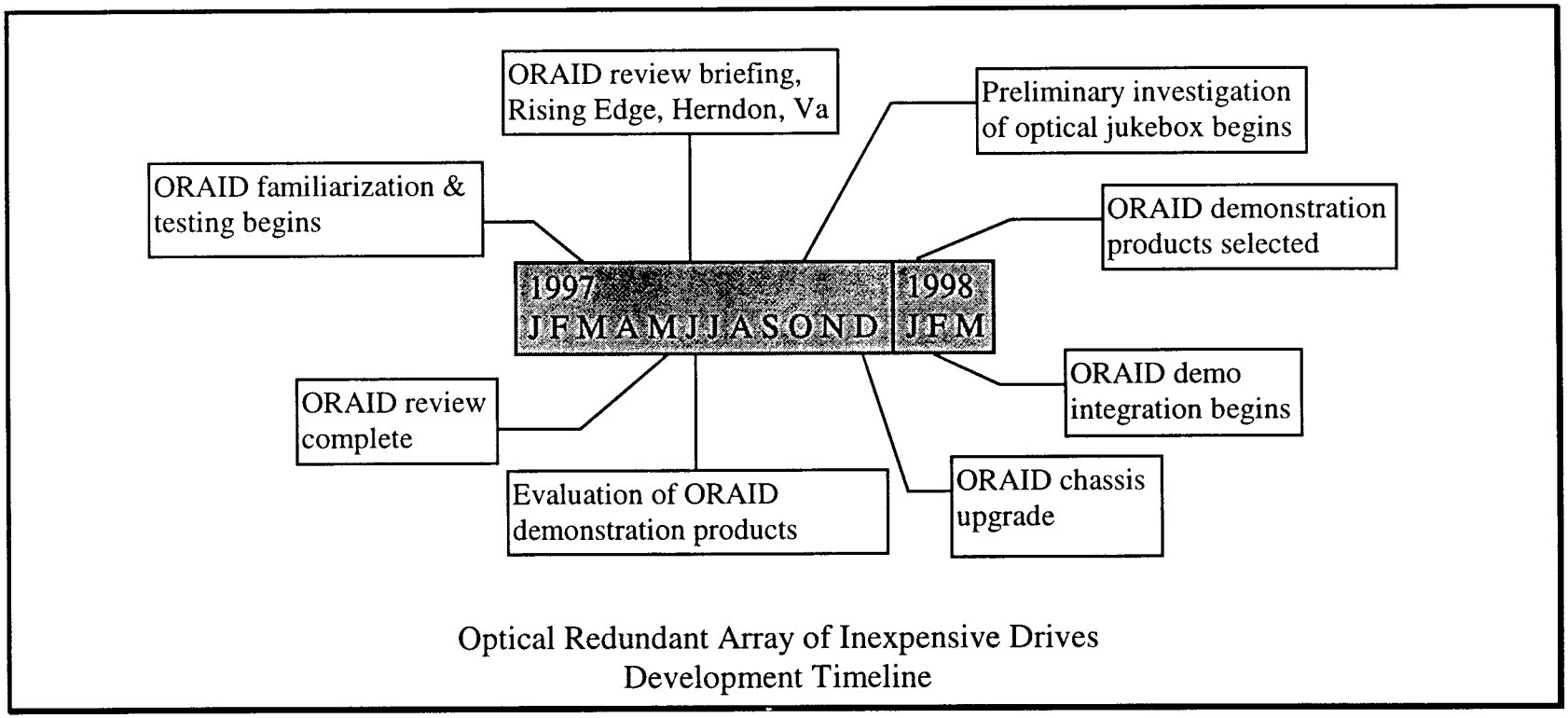
3.3 ORAID EVALUATION

In early 1997 Synectics evaluated a prototype Optical RAID system developed by Rising Edge Technologies Incorporated (RETI). This innovative application of optical storage in a RAID device offered a potential new component for the PHMSS as well as a new area in which to apply testing knowledge gained in other phases of the MSS effort. The ORAID effort centered on two primary components, assessment of the units' functionality and development of a portable system demonstration capability. Please refer to Exhibit 8 for the ORAID development timeline.

3.3.1 ORAID FUNCTIONALITY

Determining ORAID functional characteristics consisted of system familiarization, performance testing, and observation of overall robustness. System familiarization and the majority of this testing took place in the ICF at AFRL, Rome Research Site. Verification of results gathered at AFRL was accomplished at the Synectics Rome facility. The actions and observations performed during functionality testing led to publication of a report and meetings with RETI to brief Synectics results and conclusions.

Exhibit 8. ORAID Development Timeline



Optical Redundant Array of Inexpensive Drives
Development Timeline

3.3.1.1 FAMILIARIZATION

RETI personnel provided Synectics initial training and exposure to the Optical RAID. After a brief hands-on demonstration and tutorial, a period of familiarization took place to gain experience with the characteristics, limitations, and operation of the device. Once a sufficient level of confidence was acquired preliminary performance measurements were made using the Bonnie and Iozone benchmarks from the MSE2 suite. The results of these tests indicated that more rigorous testing was a worthwhile next step. Synectics developed a three phase structured approach to accomplish this task. The first phase concerned verification of the ORAID test plan provided by RETI. The second task addressed ascertaining data throughput figures for the Optical RAID over its entire range of operation by running detailed performance tests. Concurrent with these tasks, assessment of the ORAID's overall robustness took place.

3.3.1.2 PERFORMANCE TESTING

Measuring throughput of the ORAID in a setting replicating actual usage comprised Synectics' thrust in the performance phase of testing. Selection of two public domain benchmark programs, Bonnie, and Iozone, facilitated repeatable I/O testing. The selected benchmarks were run on the RETI ORAID at each RAID level (0, 3, and 5). To provide some level of comparison, testing was also performed on the internal magnetic hard drive of the test platform. ORAID testing was accomplished with 1, 64, and 256 MB files, with each RAID level treated as a separate case. Tests were run under conditions that excised available configurable parameters for the device. No attempt was made to measure or verify maximum attainable device capabilities.

All testing was accomplished on a Sun Sparc 5 running the Solaris 2.4 operating system. Repeating each test five times under the same setup provided a degree of insulation from performance anomalies such as potential interference from background processes and the possible effects of memory caching.

3.3.1.3 RETI TEST PLAN VERIFICATION

Synectics' primary goal for the verification phase of testing was confirmation of the correctness and usability of the RETI Test Plan. While not intended as a replacement for a formal user's manual, the test plan also provided a satisfactory method for gaining additional insight to the system's usage and capabilities. Each action in the provided test plan was taken and the results documented. The two areas receiving the most attention were the Host Interface Tests and Fault Tolerance tests.

3.3.1.4 ROBUSTNESS

Concurrent with the performance testing, operation of the ORAID was observed in an effort to try and understand the device's usability under a variety of conditions. The systems ease of installation, configuration, and ability to recover from errors were targeted. Running the tests in an all UNIX environment provided the opportunity to monitor the ORAID's reaction to an environment that differed markedly from its IBM PC compatible development environment. A

positive picture of system robustness justified investigation into future use and development of the ORAID.

Tests were performed while inducing a variety of failures to verify that the ORAID provided expected levels of data protection provided by more traditional magnetic RAID units. Of major concern was assessment of the likelihood of data loss through the taking of reasonably foreseeable actions. Of secondary interest was the timing of the various processes required for the operation of the ORAID.

The following intentional actions were taken to see if actual data loss could be induced.

- ❑ Using improper shutdown sequences
- ❑ Simulated drive failure via removing and replacing drives
- ❑ Shuffling optical media comprising data sets
- ❑ Skipping requested operational/usage steps

Processes required for system initialization, formatting media sets, data reconstruction, etc. were timed to provide information of potential value for workflow scheduling.

3.3.1.5 RESULTS

Results of ORAID functionality testing were published by Synectics in a report titled "MASS STORAGE SYSTEM - Review of Rising Edge Technologies, Inc. Optical Redundant Array of Inexpensive Drives". The details of this report were summarized and briefed to RETI at their Herndon, Virginia offices. Discussions held during and after this meeting generated interest in pursuing the possibility of configuring the ORAID prototype as a robotically controlled jukebox device. These discussions led to the award of a MSS subcontract to Rising Edge Technologies for the purpose of providing a preliminary analysis of an optical jukebox system.

3.3.2 PORTABLE ORAID DEMONSTRATION CAPABILITY

Additional tasking to begin development of a portable ORAID demonstration capability was formulated under the umbrella of MSS. Besides offering a solution to a specific situation, namely that of effectively demonstrating the RETI ORAID to potential users and customers, there is a wide spectrum of additional scenarios to which the proposed system applies. For example, the same components provide a high technical ability to showcase other non-related systems, particularly software, and as an added bonus, the system serves as a software installation testbed and delivery mechanism.

Requirements to incorporate the MSE2 software in this capability were also investigated. Delivering MSE2 in a 'ready to run' format insures immediate usability while eliminating a number of logistical concerns. Identifying hardware, configuring software, and dealing with on-site installation problems become non-issues. The ability to bring the benchmarks to the peripherals, rather than the other way around, greatly expands the number of devices that can be tested.

Synectics prepared a product assessment and cost analysis document to provide details on the following areas. The government approved the ideas proposed in the demonstration plan and we proceeded to purchase the desired components. In support of this phase of the program RETI repackaged the ORAID resulting in a more compact and standardized form.

3.3.2.1 ACQUISITION PHASE

Acceptance of our recommendations resulted in the purchase of the following items. All purchases included full maintenance agreements and documentation when applicable.

A Tadpole 3TX Portable SPARCbook series workstation was chosen as the host system to integrate the software and hardware components. This platform provides a 170MHz TurboSPARC processor with SPARCstation 20 performance in a portable form. The purchased unit is configured with 32MB RAM, a 2.1 GB Internal hard drive, 800x600 SVGA display, and the Solaris 2.5.1 operating system.

Paragon Imaging Inc.'s ELT series of software was requested as the primary demonstration software. Data on three versions of the software, ELT 3000 and ELT 4000/All Source, and ELT 7000 were compared. As ELT 4000 contains all of the functionality of ELT 3000 as well as Internet integration, greater NITF capability, and a host of other features, ELT 4000 was purchased to fill this role.

To fill the need for a mobile, large-audience display capability a number of high-resolution LCD projectors were investigated. The Epson PowerLite 7000 was selected after comparing features such as price, resolution, portability, and brightness. A wheeled porter case for the projector and accessories was also obtained.

Protection of the ORAID during shipping and transportation required acquisition of a Air Transport Association specification 300, Category 1 standard (ATA Spec 300) custom made case. A suitable XHA Transport case was located and procured through Rising Edge Technologies, Inc. during their modification to the ORAID.

The integration of MSE2 requires Sybase SQL Server and Sybase Open Client/C. As the database standard for government projects, Sybase would also fill database requirements for other potential systems installed for demonstration or testing purposes. Sybase Adaptive Server 11.5 and Open Client/C 11.1.1 were procured.

An additional requirement imposed by MSE2 is the availability of the Motif Window Manager (MWM). Again, MWM is a standard used by a wide variety of UNIX based software packages and would also fulfill potential future needs. Integrated Computer Solutions (ICS) Motif version 1.2.4 runtime package was purchased to fulfill this need.

3.3.2.2 INTEGRATION PHASE

As the above components arrived they were installed, configured, and tested. The first stage of development was configuring the Tadpole workstation to host the ORAID. After successful configuration the ORAID was attached to the system and preliminary ad hoc testing was performed to confirm their compatibility. This phase of integration proceeded very smoothly.

Next the Sybase Adaptive Server and Open Client/C software was installed. During testing it was discovered that Sybase makes no attempt to maintain backward compatibility with previous versions of their products. So, while the proper operation of the database and libraries was accomplished, integration and testing of MSE2 was made impossible due to software incompatibility. A potential solution to this situation is presented in section 0.

Installation and setup of Paragon ELT presented its own set of difficulties. The symbology tool packaged with ELT 4000 failed to operate in an acceptable fashion. After a number of conversations with Paragon technical support, it turned out that the software depends on Motif libraries available only with the full Development kit. This additional software was purchased from ICS and the all ELT problems were resolved.

Additional integration, testing, and scenario generation for demonstration activities was not accomplished before the end of the contract term. Recommendations for follow on actions are presented in section 0.

3.3.2.3 ADDITIONAL PURCHASES

Shortly before the end of the contract, the decision to provide additional capability to the ORAID demonstration effort was made. Synectics proceeded to purchase a Gateway 2000 Pentium 166Mhz PC and a Matrox Video, Graphics, and TV Kit to host a video capture and editing system.

These components will enhance the ability to generate realistic demonstration scenarios by utilizing video-based products pertinent to potential end users of the ORAID. In addition, demonstration of the compatibility of the ORAID device with PC based systems can be accomplished. This system will also provide a delivery capability for the COMNET III modeling and simulation software utilized during the MSS effort.

3.4 ALPHATRONIX

An Alphatronix 38.4 GB Optical Jukebox was included in this contract for evaluation and use by the IE2000 ICF. Final delivery to AFRL, Rome Research Site, and installation occurred during the week of 22 August 1994.

3.5 FINAL TECHNICAL REPORT DELIVERY

This document satisfies the requirements for subtask 5.

3.6 ORAL PRESENTATIONS

Synectics personnel conducted oral presentations at the times and places scheduled by the AFRL representative.

4.0 RESULTS AND DISCUSSION

This section will present a summary of performance figures for the PHMSS and ORAID as well as the results of the ORAID functionality tests. Overall impressions and comments pertinent to both systems are also addressed.

4.1 MSS

The network data transfer performance of the system is impressive. NFS and FTP data transfers of large (15 MB) files approach 58% of the theoretical bandwidth of the 10 Mb Ethernet. This compares very favorably with the 35% - 37% values normally seen for FTP and NFS. On the down side, the low data transfer rate of the Kodak ADL-2000 makes staging of large files fairly slow. Data transfer rates in the 600 KB/sec to 720 KB/sec for reading and 300 KB/sec to 400 KB/sec for writing are typical. These rates, combined with the average 6 second pick time attributable to the robotics, result in prohibitively long access times for near-line data situations.

The overall proof of concept for MSS went well. A large-scale storage system was assembled that integrated representative storage technologies from all areas of the cost/density range. An important lesson was learned during the installation of the PHMSS within the 480th operational

environment. It is paramount to accurately define the user's requirements so that the system performance can be precisely aligned. The MSS experience demonstrates the need to thoroughly accomplish requirement analysis and specify exact customer tolerances prior to assembling a multi Terabyte storage system.

4.2 ORAID

Results of the testing done with the RETI ORAID provided the following average throughput statistics.

- Read Rate – 2.8 MB/sec to 4 MB/sec
- Write Rate – 2 MB/sec to 3 MB/sec

These rates reflect the average utilizing RAID levels 0, 3, and 5 run across multiple tests.

Generated results demonstrate the expected trade off of speed for data redundancy and safety. The achieved rates do compare favorably with other storage devices in similar situations. As is the case in most performance benchmark situations, the data generated provides only a portion of the information needed to draw hard conclusions. RETI achieves I/O rates of 3.125MB/sec sustained data transfer and 6.25MB/sec burst data transfer based on testing which communicates directly with the SCSI hardware. Synectics made no effort to demonstrate or determine isolated hardware obtainable rates of data transfer. By including overhead associated with the operating system, file system, and background processes, throughput rates generated by Synectics testing represent what the user is likely to experience.

Overall the RETI ORAID is a solidly built and well thought out data storage system. It fills the requirements of mid- to large-scale storage and provides responsive throughput with the assurances of RAID technology.

Synectics work with the Rising Edge Technologies Inc. ORAID determined that the unit provided a number of unique advantages over traditional magnetic based RAID devices. The ability to establish and maintain multiple datasets gained by the ability to remove the optical media provides flexibility not easily achievable by magnetic systems. This characteristic makes the ORAID suitable not only for 'area of interest' on-line storage but also as a very flexible archive instrument. Performance results (both statistical and user observed) bear out the unit's suitability for large-scale storage requiring the safety afforded by RAID technology. The use of rewriteable optical media overcomes the deficit of WORM technology opening up many more potential applications.

5.0 CONCLUSIONS

A number of valuable lessons were learned during this effort. The PHMSS concept provides a viable approach to meeting mass storage needs. The effort was successful in a majority of areas although some, such as modeling and user integration, proved more difficult than anticipated. The following conclusions have been made based on the Mass Storage System program.

- ❑ The proof of concept PHMSS provides a large-scale storage system capable of storing up to 1/2 TB of data. Overall the design and integration of the system contained no major roadblocks or surprises.

- ❑ The impact on data access times caused by the archive layer of a hierarchical storage system cannot be underestimated. Under an established archival concept of operations the users are generally aware of this limitation. Situations outside the typical archival realm require advanced planning strategies to facilitate preloading of data to offset access time deficiencies.

- ❑ Thorough end user requirement analysis and education need to be accomplished prior to development of specific mass storage solutions.

- ❑ Difficulties faced while integrating the PHMSS into the 480IG operational environment stemmed primarily from:
  - Insufficient understanding of the hierarchical storage concept.
  - Insufficient understanding of the long-term cost savings provided by HSM for multi-terabyte storage.
  - Shifting of 480th focus and priorities due to a change in operational and financial situation.

- ❑ The performance and fault tolerance of the Data General CLARiiON RAID proved exceptionally impressive.

- ❑ The lack of rewriteable media for the Kodak ADL-2000 hindered acceptance of the PHMSS for situations requiring access to dynamic information.

6.0 RECOMMENDATIONS

This effort resulted in a number of recommendations that are summarized below.

6.1 UPGRADE MSE2

Modernizing and porting MSE2 to function under current operating system and database revisions will provide an ongoing capability to test existing and emerging storage technologies.

- ❑ Port MSE2 to obtain compatibility with Solaris OS 2.5.x and Sybase System 11.
- ❑ Leverage the ORAID demonstration hardware and software to provide greater flexibility and portability.
- ❑ Upgrade existing integrated benchmark software to newest revisions.
- ❑ Expand the variety of I/O based benchmarks included in MSE2.
- ❑ Investigate optional GUI methodology to provide more portable interface.

6.2 DEVELOPMENT OF ORAID JUKEBOX

The enhancement of the existing ORAID prototype to expand its capacity and versatility provides the logical next step in developing mass storage capability. Additional tasks for consideration are listed below.

- ❑ Replace standalone Magnetic Optical (MO) drives with MO jukeboxes to increase capacity and provide automated archival capability.
- ❑ Develop and integrate hardware solution to HSM by utilizing magnetic disk based transactional cache.
- ❑ Develop test plans and scenarios.
- ❑ Collect and categorize sample test data.
- ❑ Perform system evaluation leveraging MSE2 and COMNET III.
- ❑ Collect and analyze user requirements to develop effective demonstration approaches.
- ❑ Provide demonstrations at user locations.

6.3 DEMONSTRATE ORAID PROTOTYPE

Continued exposure of the existing ORAID prototype tested during the MSS effort would provide an excellent opportunity to increase user awareness of both general mass storage principles and the availability of a innovative storage solution.

- ❑ Continue development of ORAID portable demonstration capability.
- ❑ Calculate the effect of the RAID process via comparison to a standalone optical device.
- ❑ Utilize demonstrations to present ORAID prototype as:
  - Standalone storage solution
  - Lead in to ORAID Jukebox
  - Mass storage educational tool
- ❑ Identify customer(s) for ORAID prototype

APPENDIX A – ACRONYMS

AFRL	Air Force Research Laboratory
COTR	Contracting Officer's Technical Representative
COTS	Commercial Off The Shelf
CPU	Central Processing Unit
DLL	Dynamic Link Library
DoD	Department of Defense
EEPROM	Electrically Erasable Programmable Read Only Memory
FDDI	Fiber Distributed Data Interface
FTP	File Transport Protocol
GB	Gigabyte
GUI	Graphical User Interface
HSM	Hierarchical Storage Management
HTML	Hyper Text Markup Language
HTTPD	Hyper Text Transport Protocol Daemon
I/O	Input/Output
ICF	Intelligence Cartographic Facility
IFE	Information and Intelligence Exploitation Division
IFED	Global Information Base Branch
IG	Intelligence Group
KB	Kilobyte
LAN	Local Area Network
MB	Megabyte
MO	Magnetic Optical
MSE2	Mass Storage Evaluation Environment
MSMI	Mass Storage Management Interface
MSS	Mass Storage System
MWM	Motif Window Manager
NFS	Network File System
ORAID	Optical Redundant Array of Inexpensive Drives
OS	Operating System
PHMSS	Prototype Hierarchical Mass Storage System
PROM	Programmable Read Only Memory
RAID	Redundant Array of Inexpensive Disks (Drives)
RETI	Rising Edge Technologies Incorporated
SCSI	Small Computer Systems Interface
WORM	Write Once Read Many

APPENDIX B – REFERENCED DOCUMENTS

The following documents of the exact issue shown form a part of this document to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this document, the contents of this document shall be considered superseding requirements.

NASA 1994	Third NASA Goddard Conference on Mass Storage Systems and Technologies
480th AIG July 1992	480th Air Intelligence Group (AIG) Digital Storage Concept of Operations
Synectics WH-93-QW-00 15 January 1996	Mass Storage System Installation Site Survey Specifications
Rising Edge Technologies 20 May 1997	Optical RAID Installation and User's Manual
Synectics WH-93-QW-00 07 July 1997	Optical RAID Demonstration System Product Assessment and Cost Data

Copies of specifications, standards, drawings, and publications should be obtained from the contracting agency or as directed by the contracting officer. Technical society and technical association specifications and standards are generally available for reference from libraries. They are also distributed among technical groups and using Federal Agencies.

APPENDIX C – HSM EVALUATION PAPER

C1.0 INTRODUCTION

Historically, most sites have used magnetic tape as a digital image archive medium to reduce the cost of on-line storage of data. In so doing, they not only sacrificed fast and easy access to the data, but also incurred the operational costs associated with handling a tape library.

Today's operational costs of manually archiving data are rapidly becoming less and less justifiable. As CPU power increased exponentially with the advent of RISC-based systems and the shift to a client/server Information Technology (IT) paradigm began to take hold the types and amounts of data being stored also began to increase dramatically. At the same time, the costs associated with many forms of storage media have plummeted.

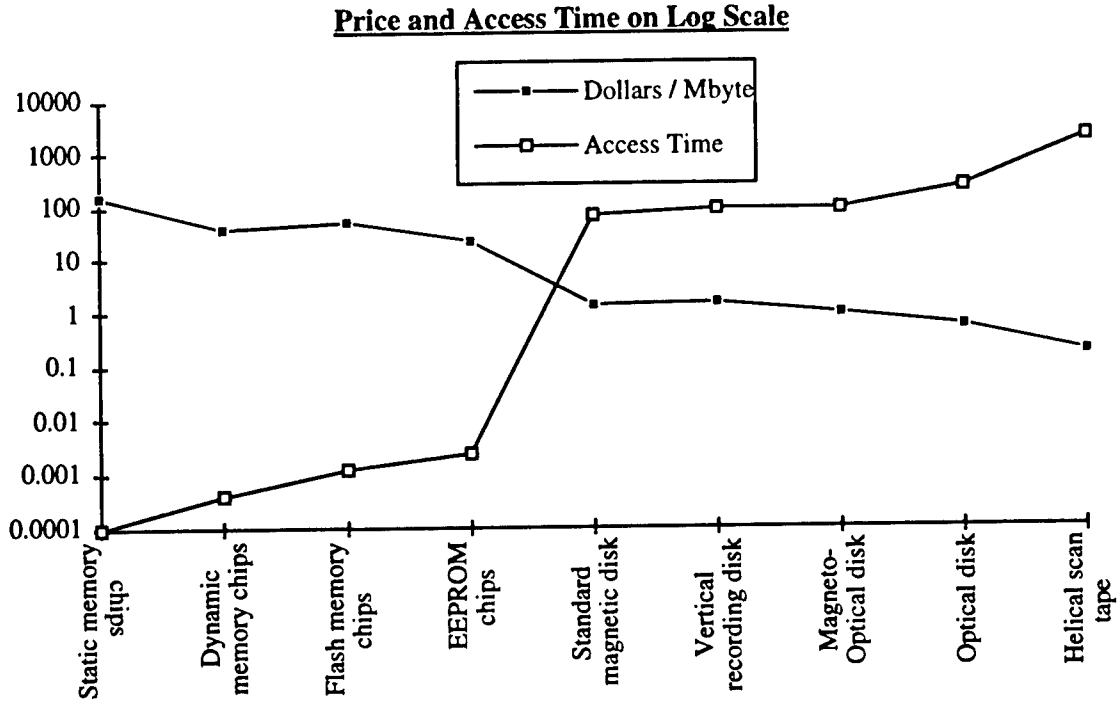
As a result of these changes, the need to automate the shelving of data has become a growing concern. Hierarchical Storage Management (HSM) was once envisioned as the moving of data between different classes of magnetic disk drives in a datacenter before archiving that data on an off-line medium. Technology changes, however, demand a different scenario. HSM today looks to move data between nodes on a network before shelving the data on to a near-line (on-line, but not directly accessible) medium.

C2.0 HIERARCHICAL STORAGE SYSTEMS

The storage, retrieval, and exploitation of imagery and related feature data place heavy demands on the associated data storage subsystems. With its voluminous data sizes imagery clearly requires very high densities of storage media and a low cost per byte of stored information. Unfortunately it is generally true that there is an inverse relationship between the access and data transfer times of memory storage subsystems, and their data density/cost per byte. This relationship is illustrated in Exhibit C1.

This relationship presents a real quandary for the developer of an imagery exploitation system. On one side, it is desirable to have the largest amount of imagery and related data possible resident and available to the analyst at any given time. On the other side, it is necessary that the imagery itself (to a lesser extent) and the related feature data (to a very great extent) be accessible in an expeditious manner.

Exhibit C1. Cost vs. Time



Hierarchical Storage Management (HSM) provides a solution to this problem by using "layers" of each media type. RAM (very fast) is used the cache the magnetic media (reasonably fast) which is in turn used to cache the optical media subsystem (extremely slow).

The remainder of this white paper is devoted to a discussion and comparison between a selection of the leading HSM products available on the commercial market.

C3.0 OVERVIEW OF COMMERCIAL PRODUCTS

There are a number of commercial products that provide HSM and which are applicable for the role of archiving and managing digital imagery.

C3.1 ALPHATRONIX EMISSARY/HSM

Alphatronix Inc.'s Emissary/HSM is a hierarchical storage management (HSM) software package that is designed to operate with their Inspire optical jukebox system. This combination will allow system managers to create a storage hierarchy that puts the most frequently used data on

the fastest and most expensive storage medium, and the less frequently used data on a less expensive, slower storage medium. An example of a relatively fast medium is a local magnetic disk drive. A slower storage medium could be a removable optical disk drive on the storage server.

Emissary/HSM uses a concept of "storage banks," which are multiple optical jukeboxes attached to the host, and volume sets, which are collections of optical cartridges used to store data logically; each storage bank has several volume sets assigned to it.

The program is designed to operate without operator intervention in a networked client/server environment. Emissary/HSM costs $4,800 for a server license. A 16-slot optical jukebox and jukebox software cost $14,900.

C3.1.1 OPERATION

The Emissary/HSM software package is able to function entirely in a true lights-out manner. As such, it provides a means to define global triggers to initiate or suppress file migration based on system load conditions as well as file-specific parameters for choosing the files to be migrated. In addition, the package is able to function in a networked client/server environment as well as in a stand-alone mode.

With Alphatronix's Emissary/HSM software and Inspire optical jukebox, system managers can set up a storage hierarchy. In this system, the "hot" data would reside on the fastest and most costly storage medium, a local magnetic disk drive, while less frequently used data are automatically migrated to a less costly and slower storage medium, a removable optical disk drive residing on a storage server.

C3.1.2 INSTALLATION

Adding new devices to a Unix-based system is a non-trivial task that partly involves rebuilding the Unix kernel. Therefore, when Alphatronix sells a copy of Emissary/HSM, it sends a field engineer to the customer site to assist with installation at no extra charge. Currently the installation task is somewhat simplified by the limitation that the system be used only with an Alphatronix Inspire jukebox.

The first step in the installation process is to edit the SunOS system configuration file /sys/sun4c/conf/GENERIC. (We used the generic kernel; the last part of the path could be different on your system.) We had to start by editing the configuration file to add entries for the jukebox and its optical disk drive at the proper SCSI addresses.

Once the kernel is rebuilt and the system rebooted the installer mounts an optical cartridge that contained the installation scripts. Using a tar command moves the scripts onto the hard disk and runs the installation script to complete the installation.

One noticeable drawback of the Alphatronix Emissary/HSM product is the lack of support for any third-party optical jukeboxes. This should change in the near future however, as Alphatronix says it intends to unbundle its Inspire jukebox from its Emissary/HSM software to make it more versatile.

C3.1.3 USE

With all of the hardware and software installed to operate the Inspire jukebox, it is necessary to start building a storage hierarchy. First, load the optical cartridges into the jukebox by running jb [underscored] admin, an X Windows application used to manage the jukebox.

Running the insert/remove utility will make the system aware of a cartridge inserted into the jukebox. (Loading a cartridge using the front panel of the jukebox is possible but the jukebox software will not be aware of it.)

After the cartridge is inserted, the optical drive attempts to read the volume label. If successful, the label is displayed. If the cartridge is new, the format utility must be run to assign a new volume name to the cartridge and to place a SunOS file system on it.

Unlike some HSM software packages, Emissary/HSM does not install or create a foreign file structure on the host system. As a result, it is not necessary to dedicate the jukebox to Emissary/HSM. In fact, you can have some volumes assigned to Emissary/HSM and others to the host system, to be used with any other file system.

Emissary/HSM works entirely with normal SunOS file systems. Even with only one drive in the jukebox, the Emissary/HSM will take care of loading and unloading the correct optical disk cartridge whenever I/O was requested to a file system residing on that cartridge.

When used as a client/server application, Emissary/HSM comprises a client migration manager, which migrates disk files according to user-defined rules, and a server application, which controls access to an attached SCSI optical jukebox. Client and server processes communicate directly. They do not require NFS to run.

C3.1.4 EXPANSION

Emissary/HMS works with storage banks and volume sets. A storage bank is simply an optical jukebox attached to a host. You can have multiple hosts with jukeboxes attached to them, and Emissary/HMS will see multiple storage banks.

In turn, each storage bank has a number of volume sets assigned to it. A volume set is just a collection of optical cartridges used to logically store data. A set might be defined for each user on a system, or each department in a company. It's also possible that a single volume set be all that's needed for all the data on an entire system. Emissary/HMS neither dictates how many volume sets are needed, or how many cartridges there are in each volume set.

Once the volume sets are defined in Alphatronix's Data Migration Manager (DMM) application, the next step is to structure the rules that will govern file migration. These rules determine which files are to be migrated, the storage bank, and volume set to where files are to be migrated, and when the migration from magnetic disk to optical disk will take place.

Defining the files that will be subject to Emissary/HMS's migration process is done using a rule set definition window within DMM. Doing this enables the system manager to browse through all disk file systems and choose the directories that a particular rule will apply to. When Emissary/HSM is looking for files to migrate, it will check all the files in these directories and all the files in any subdirectories below.

After designating the files that will be migrated, it is then necessary to set up the criteria for that migration, based on four attributes: mode, file ownership, file size, and files system watermark.

There are two modes of migration: by date and by age. In migration by date, a date is set and any file with a time stamp before this date in migrated. In migration by age, an age is set and any file that is older than this age is migrated.

The file-ownership criteria are based on defining a list of file owners and file groups. If a file matches any of the groups or owners listed, it will be migrated.

The file-migration process can be further tuned by setting upper and lower file size limits for migrated files. The upper limit is useful in preventing large files that will take a long time to retrieve from being migrated based on mode or ownership. The lower limit is equally useful for screening small files for which the overhead that migration would impose on the system would be not be compensated for by any gain in on-line storage space.

Finally, the pace of file migration can be throttled upwards or downwards via the setting of high and low watermarks for a disk's file system. By setting a high watermark, early migration can be triggered when there is no space to create new files on a disk. By setting a low watermark, Emissary/HSM can be prevented from migrating too many files at once or migrating so many files that the optical jukebox becomes a system bottleneck.

Finally, once the file-migration criteria are set, it is necessary to set the migration attributes: frequency of migration, migration destination, and any migration options. Migration frequency can be set for immediately, one time only, or at a specific interval. Intervals can be set to daily, weekly, or specific times.

The storage bank and volume set where the files are to the sent can be independently set for rule set. The migration attributes screen provides pull-down menus with names of all storage banks and volume sets.

C3.1.5 FILE MIGRATION OPTIONS

There are a number of additional file migration options that can be set. For example, the DMM software can be set to follow symbolic file links when looking to migrate files. Alphatronix suggests setting the follow symbolic links option to no, as it could cause trouble should an unexpected symbolic link point to a system file.

Careful system administration is required to prevent possible ill effects of enabling this option. For example if the administrator designated a file with symbolic links to the system library for migration, the result would be the migration of a file from the system library needed to retrieve files that had been migrated to a storage bank

Allowing this file to migrate essentially destroys the operating system. When attempting to access data in a file that had been migrated to an optical disk, the operating system references the system library file in order to retrieve the original file. This, in turn, creates an irresolvable pointer reference, as the system then tries to retrieve the library file that is needed to retrieve itself.

As an added convenience for system managers in a production environment the DMM can be set to report on which files it would migrate if it were to run immediately. This option is very useful to test the effects of rule sets before Emissary/HSM actually moves any files.

It is also possible to tell Emissary/HSM to compress the files as it moves them to the optical disk. This option saves space on the optical disks, but sacrifices performance, as data compression is done on the fly. Emissary/HSM uses the standard Unix compress and uncompress commands, which means that data access is not compromised even when the software is not running, since files can always be copied and uncompressed manually.

C3.1.6 PROS AND CONS

PROS

- Foreign file formats or pre-written disks supported.
- Unlimited control of migration rules (to the point of being dangerous).
- Unlimited control of compression scheme used.
- Multiple jukeboxes supported as a single virtual device.

CONS

- Multiple jukebox vendor's hardware not supported.
- Direct hooks for data base integration not part of basic package.
- Limited number of operating systems supported.

C3.2 EPOCH

The ability to handle optical and magnetic media in a heterogeneous configuration (so that the actual location of data is invisible to the subscribers) is a complex task. The manipulation of the optical storage device(s) and juke box(s), the caching of data (both read and write), and the use of pre-fetch strategies to accelerate perceived performance all require extensive processing.

There are off-the-shelf systems that will perform these tasks. One of the most mature of these is the Epoch Inifinistore system. This is a full-featured HSM system that compares with the Alphatronix system as described in detail in section 3.1. EPOCH uses a dedicated processor (SUN SPARC) which acts as a "clearing house" for all data. This clearinghouse system controls, either directly or via FDDI, all of the media on the system to form one giant virtual media.

The EPOCH system uses statistical techniques to tailor itself to the environment in which it is located. Access patterns, read-after-write latency, and a host of other data items are monitored. The EPOCH then modifies its operation to provide the optimal performance.

The differences between the Epoch and Alphatronix systems are threefold.

C3.2.1 FOREIGN FILES AND FORMATS

Unlike the Alphatronix solution, the Epoch will not handle disks formatted in "foreign" ways, or which were written on another system. In effect each Epoch must be installed as a "virgin" (empty) system and populated from the beginning with all of the data that it will manage.

Although this system increases the access rate and hit rates for data it means that sites with existing catalogs of imagery or other information on optical media must transcribe that data to the librarian.

C3.2.2 JUKEBOXES SUPPORTED

The Epoch system is currently capable of supporting all of the jukebox architectures currently on the market. Unlike some other systems (notably offerings from Alphatronix and Kodak) which are married to jukebox systems manufactured by the offeror.

C3.2.3 MULTIPLE OS SUPPORT

Currently the Alphatronix system is only available for Unix platforms, preferably running SunOS or Solaris. The Epoch system is also available for IBM, Novell, and VMS architectures, although mixed mode operation is not supported.

C3.2.4 PROS AND CONS

PROS

- Multiple jukebox vendor's hardware supported.
- Multiple jukeboxes supported as a single virtual device.
- Multiple operating systems supported.
- Direct hooks for data base integration.

CONS

- No foreign file formats or pre-written disks supported.
- Limited control of migration rules.
- Limited control of compression scheme used.

C3.3 PINNACLE VIRTUAL FILE SYSTEM 1.0

The Pinnacle Virtual File System 1.0 is a hierarchical management system designed for use with Pinnacle $9,995 Alta-20GB, $19,995 Aspen-40GB, $49,995 Alpine-120GB and $79,995 Mammoth-186GB, Magneto Optical Jukebox systems only. Currently the system is available only for Netware and Appleshare servers. A release is scheduled for early 1995 for Unix platforms. Additional information for this offering has not been released.

C3.3.1 PROS AND CONS

PROS

- None.

CONS

- No multiple jukebox vendor support.
- No Unix support.

C3.4 KODAK

Kodak offers a full-featured HSM system for use in conjunction with its high-end 14-inch jukebox, the System 2000. The System 2000 is now being shipped with two expansion cabinets. The main unit features a two-slot robot for 30 percent faster disk changing, plus front- panel diagnostics that can be remotely accessed. The base unit with a single drive and SCSI controller is expandable to 50 slots. The new expansion boxes, the Archivist and the Performer, can add up to 84 more slots with no drive or 50 more slots with a drive. With the expansion, the System 2000 can store 1.3 terabytes on existing platters. Prices start at $143,000.

Kodak is bidding its 14-inch format as a subcontractor to IBM Corp. for the Internal Revenue Service's document imaging procurement and with Hughes for and FBI fingerprint imaging project. In 1994, Kodak plans an increase in capacity of its 14-inch disks from 10.2 gigabytes to 13G via a new write laser. Existing Model 6800 drives will be upgradeable to read or write disks of both capacities and to read the older 6.8 GB platters.

C3.4.1 KODAK HSM PACKAGE

The Kodak HSM package allows the System 2000 to operate as a virtual device in a client server environment. The migration of data between the magnetic and optical elements is automatic and seamless to the users.

As expected the Kodak HSM system operates only with their optical jukebox system. The system will automatically, or manually, create a storage hierarchy that puts the most frequently used data on the fastest and most expensive storage medium (local magnetic disk drive), and less frequently used data on the storage server. The program is designed to operate without operator intervention in a networked client/server environment.

C3.4.2 OPERATION

The Kodak HSM software package is able to function entirely in a true lights-out manner. As such, it provides a means to define global triggers to initiate or suppress file migration based on system load conditions as well as file-specific parameters for choosing the files to be migrated. In addition, the package is able to function in a networked client/server environment as well as in a stand-alone mode.

C3.4.3 JUKEBOXES SUPPORTED

One of the biggest drawbacks of the Kodak HSM product is the lack of support for any third-party optical jukeboxes. It supports only the Kodak jukebox system. As a result though the system is highly optimized for that platform and provides excellent performance in such an environment.

C3.4.4 EXPANSION

Kodak HMS can work with multiple jukeboxes to create an arbitrarily large virtual file system. In addition the administrator has a broad number of options in defining the files that will be subject to the migration process. This is done using a rule set definition process which the system manager to browse through all disk file systems and choose the directories that a particular rule will apply to. When system is looking for files to migrate, it will check all the files in these directories and all the files in any subdirectories below.

After designating the files that will be migrated, it is then necessary to set up the criteria for that migration, based on four attributes: mode, file ownership, file size, and files system watermark.

There are two modes of migration, by date and by age. In migration by date, a date is set and any file with a time stamp before this date is migrated. In migration by age, an age is set and any file that is older than this age is migrated.

The file-ownership criteria are based on defining a list of file owners and file groups. If a file matches any of the groups or owners listed, it will be migrated.

As with other systems the file-migration process can be further tuned by setting upper and lower file size limits for migrated files. The upper limit is useful in preventing large files that will take a long time to retrieve from being migrated based on mode or ownership. The lower limit is equally useful for screening small files for which the overhead that migration would impose on the system would be not be compensated for by any gain in on-line storage space. Also as with other systems the pace of file migration can be throttled upwards or downwards via the setting of high and low watermarks for a disk's file system. By setting a high watermark, early migration can be triggered when there is not space to create new files on a disk. By setting a low watermark, Emissary/HSM can be prevented from migrating too many files at once or migrating so many files that the optical jukebox becomes a system bottleneck.

Finally, once the file-migration criteria are set, it is necessary to set the migration attributes: frequency of migration, migration destination, and any migration options. Migration frequency can be set to immediately, one time only, or at a specific interval. Intervals can be set to daily, weekly, or specific times.

The storage bank and volume set where the files are to be sent can be independently set for rule set. The migration attributes screen provides pull-down menus with names of all storage banks and volume sets.

C3.4.5 Pros and Cons

Pros

- Unlimited control of migration rules (to the point of being dangerous).
- Highly optimized for Kodak 2000 jukebox.
- Multiple jukeboxes supported as a single virtual device.

Cons

- Foreign file formats or pre-written disks not supported.
- Multiple jukebox vendor's hardware not supported.
- Limited number of operating systems supported.

C3.5 LEGATO / CHEYENNE

The Legato NetWorker HSM (also marketed by Cheyenne software as the ARCserve) is an archive management product which was originally designed as a backup management system for large Novell networks. Recently the system has been expanded to provide HSM functionality and to support semi-automatic data migration.

As a traditional backup-software makers Legato Systems Inc. and Cheyenne Software Inc. are newcomers to the hierarchical storage-management software market. The systems currently produced allow a network administrator to manage file storage across a network, setting up a system that migrates old files from on-line servers to cheaper, off-line storage which may include optical jukeboxes and tape drives from a number of manufacturers.

C3.5.1 BACKUP ORIENTED

Unlike the other systems described in this white paper the Legato / Cheyenne system is only automated in the downward migration path. The restoration of data is not invisible and requires manual intervention by either an administrator or the user.

C3.5.2 VIRTUAL ACCESS NOT SUPPORTED

In addition the archived data may not be used directly on the archival media, it must be restored, in its entirety to the magnetic media for use. In this regard the Legato offering is unlike any of the other systems which all present as "virtual" devices and whose operation is seamless and invisible to the operator.

C3.5.3 PROS AND CONS

PROS

- Inexpensive.
- Multiple device types (tape, disk)
- Multiple operating systems.

CONS

- Manual intervention required.
- Not a virtual device.

C3.6 PALINDROME

Palindrome and archive device manufacturer Conner Peripheral offer a system which is similar to the Legato / Cheyenne offering in that it is a migration of what was originally a backup engine. Originally a Netware product the Palindrome offering is currently available for Unix systems as well. Palindrome Corp.'s Backup Director, Version 2.1 product is a script based migration

system. Capable of managing a number of archival devices, including optical jukeboxes from a number of manufacturers, Backup Director uses rules set by the administrator (in the form of a script) to transfer unused data to archive.

C3.6.1 No Virtual Storage

As with the Legato system, Palindrome's offering is not virtual. The users of archived data must retrieve the data in its entirety to utilize it. In addition the data must be transferred back to working storage, it can not be used directly on the archive media.

C3.6.2 Semi-automatic Archival

Unlike all of the other systems the Palindrome system uses a script as the only decision-making mechanism for archival. In addition the archival is non-hierarchical, the deletion of the original is not automatic.

C3.6.3 Pros and cons

Pros

- Inexpensive.
- Multiple device types (tape, disk).
- Multiple operating systems.

Cons

- Manual intervention required.
- Not a virtual device.
- Not a hierarchical device

C3.7 Amdahl

The Amdahl UniTree (and its successor the A+ UniTree) are HSM products originally developed by Lawrence Livermore Laboratory to handle their tremendous libraries of archived data. The system has evolved to become a stable, multi-platform, and multi-OS HSM system for the very high-end (enterprise level and higher) markets.

C3.7.1 Distributed Architecture Support

The UniTree product has all of the migration and auto-archiving features found in all of the other HSM products addressed in this paper with the addition that it can operate in a highly distributed environment. UniTree is designed to unify all of the mass storage devices on an entire enterprise-wide WAN into a singe virtual memory subsystem, and to automatically migrate data not only based on use, but on the location of that use so that data tends to be local when accessed.

C3.7.2 COMPLEXITY AND EXPENSE

As might be expected this power and flexibility come at an extreme cost. Not only is the software product expensive but the setup and maintenance are complex in the extreme. According to Amdahl an average installation on a Solaris server costs $175,000, including A+ UniTree and A+ User Access software. However an additional installation, setup, and optimization contract will typically cost $40,000.

C3.7.3 OTHER UNIQUE FEATURES

UniTree's distributed nature is enhanced by the recent addition of embedded ATM support. UniTree is compatible with a number of different operating systems on a heterogeneous network. The aforementioned asynchronous transfer mode (ATM) technology also makes long-distance management and backup of an entire enterprise-computing environment truly feasible.

C3.7.4 PROS AND CONS

PROS

- Highly flexible and extensible.
- Multiple device types (tape, disk).
- Multiple operating systems.
- Distributed architecture supported.
- Location migration supported.

CONS

- Very complex.
- Very expensive.
- Overkill for a centralized server configuration.

C3.8 LAWRENCE LIVERMORE LABORATORY

Lawrence Livermore National Laboratory (which also developed UniTree [see Section 3.7]) has signed a licensing agreement for its internally developed HSM storage system with General Atomics Distributed Computing Solutions (DISCOS) division in San Diego. The lab has spent the last six years developing a storage system, the Livermore Integrated Network Computing System (LINCS). LINCS is hierarchical, uses C and UNIX and stores data on disks, optical jukebox systems, and tapes.

Due to its C implementation it is portable from one platform to another. LINCS was developed on an Amdahl 5860, and General Atomics ported it to the VAX in less than two months.

C3.8.1 AUTO MIGRATION

Data not recently accessed is stored on disk. Newer data is on disks served by automated loaders, which lets a user get the requested data in 30 seconds. Data usually is first stored on disk, but extremely large files are stored directly to the archive medium (a unique feature of LINCS).

Once a file is stored on archival media, the disk version will be destroyed. Large files that are not accessed might be removed within a few hours, while a small file of a memo could stay on the disk for years, as long as it is accessed once a month. Data that is several years old is migrated off of the loader and may be placed in a vault, and an operator must load the media manually.

C3.8.2 DEVICE SUPPORT UP TO USERS

Originally LINCS was developed for IBM 3480 tape cartridge loaders, the Storage Technology Corp. automated library, and IBM-compatible disks on the Amdahl. However, users can write software drivers to allow other storage devices to work with LINCS.

The system works well but is tailored for IBM systems only. The product is very cost effective, and having been developed by a national lab, would be GOTS, however the setup and configuration would be difficult as the system does not support the devices, OS, or jukebox intended for MSE2.

C3.8.3 PROS AND CONS

PROS

- GOTS.
- Source available for addition of performance measures.

CONS

- Significant integration required.
- Device drivers must be written.
- Not a supported product at this time

C3.9 ZITEL

Zitel Corporation's Hierarchical Storage Management (HSM) software is targeting Digital Equipment Corp.'s HSJ family and other SCSI, CI, and DSSI storage subsystems.

C3.9.1 HARDWARE / SOFTWARE HYBRID

The Zitel File Manager (ZFM) system is unique among those described in this paper in that it is (in part) based on the company's Cached Actuator Storage Device (CASD).

The device includes spindle-level write-back cache, located between the disk drive and the SCSI bus. A proprietary algorithm manages data movement between the cache and the disk drive based on access frequency.

In this respect the hierarchical management aspect of the operation is taken one level farther than other systems. By combining the CASD with a licensed version of Software Partners/32's Hierarchy and HotSwapper HSM software. The system manages the migration of hot (frequently accessed) and cold (infrequently accessed) files between RAM cache, high speed disk, standard disk drives, and secondary storage devices like tape and optic al libraries.

With ZFM, the hottest data is kept in the CASD module, called the Hot File Device (HFD), while the coldest files are migrated to less expensive tape or optical devices. The CASD caching algorithms dynamically swap the hottest logical block numbers in and out of cache memory. Warm files are stored on non-cached disk drives.

C3.9.2 Maximum Performance / Cost Ratio

Zitel claims that the combination of CASD technology with HSM software minimizes the number of warm files in caches, which frees up more space for hot files. Compared to solid state disks, CASD offers slower I/O rates and access times but higher capacity and a much lower cost per MB. For example, Digital's solid state disks cost about $95 per MB and deliver a peak I/O rate of 800 I/Os per second with an average access time of less than 1msec. CASD costs $4.90 per MB, and delivers 60 I/Os per second with access times under 100msec.

Currently the system is available for Netware and OpenVMS, but delivery is scheduled for Unix in the later part of 1995.

C3.9.3 Pros and Cons

Pros

- Very high performance.

Cons

- Not currently available for our environment.
- Devices supported are limited

C3.10 E-Systems

The E-Systems EMASS is a HSM product tailored for use with its DataTower optical storage system. A fairly typical HSM system the distinguishing characteristics of the EMASS are its ability to address and manage well in excess of 10,000 terabytes of data.

Currently the Army Engineer Waterways Experiment Station in Vicksburg, Mississippi has installed DataTower with EMASS. The system is so high performance that the controller is a CRAY-64 dedicated to media control. Besides the DataTower archive, which is about the size of a phone booth, the EMASS will perform automated backup to the company's ER90 high-speed helical scan tapes drives. The tape drives can deliver data to a user at 15 megabytes/sec and are used to free up room on the MO drives for more "warm" data.

C3.10.1 PROS AND CONS

PROS

- Very high performance.
- Large memory and indexing model.

CONS

- Not currently available for our environment.
- Devices supported are limited.
- Extraordinarily expensive.

APPENDIX D – MSE2 BENCHMARK EVALUATIONS

Mass Storage Evaluation Environment (MSE2)

Evaluation of Benchmarks Selected for Implementation

Preface: Text displayed in `Courier 9 point` font represents information copied verbatim from the FAQ archives. If this information is to be included in any form of documentation, the appropriate reference information will need to be included.

Name: Bentley - Unix Review Metrics/Timers

Category: CPU Performance -- General C Constructs.
Language: C
Source: toklab.ics.es.osaka-u.ac.jp --> /unix/benchmarks

Description:

This benchmark returns a Mics/N value for a number of basic operations. This benchmark is used by Unix Review in their evaluation of Unix based systems.

Sample Output:

```
   Operation                          Clicks for each trial     Mics/N
Null Loop (n=5000000)
  ()                                 218  219  218  217  217      0.73
Int Operations (n=5000000)
  i1++                               326  327  327  327  326      0.36
  i1 = i2                            302  303  303  303  306      0.29
  i1 = i2 + i3                       351  351  351  351  352      0.44
  i1 = i2 - i3                       351  351  351  351  352      0.44
  i1 = i2 / i3                      1040 1040 1040 1041 1041      2.74
  i1 = i2 % i3                      1040 1041 1040 1041 1040      2.74
Float Operations (n=5000000)
  f1 = f2                            304  303  303  302  303      0.28
  f1 = f2 + f3                       595  594  594  595  594      1.26
  f1 = f2 - f3                       594  595  594  594  596      1.26
  f1 = f2 * f3                       628  629  630  628  629      1.37
  f1 = f2 / f3                      1304 1306 1304 1307 1311      3.63
Numeric Conversions (n=5000000)
  i1 = f1                            374  375  375  375  375      0.52
  f1 = i1                            436  436  435  435  436      0.73
Integer Vector Operations (n=5000000)
  v[i] = i                           525  460  432  441  437      0.80
  v[v[i]] = i                        576  584  534  526  506      1.09
  v[v[v[i]]] = i                     625  643  646  625  611      1.37
Control Structures (n=5000000)
  if (i == 5) i1++                   294  289  291  290  290      0.24
  if (i != 5) i1++                   411  412  411  411  412      0.65
  while (i < 0) i1++                 290  289  291  290  290      0.24
  i1 = sum1(i2)                      496  496  495  497  497      0.93
  i1 = sum2(i2,i3)                   675  677  676  676  678      1.53
  i1 = sum3(i2,i3,i4)                835  836  836  836  837      2.06
Input/Output (n=50000)
  fputs(s,fp)                         29   30   31   30   34      9.54
  fgets(s, 9, fp)                     27   25   25   27   26      7.94
  fprintf(fp, sdn, i)                163  163  163  162  163     53.54
  fscanf(fp, sd, &i1)                208  198  210  209  196     67.34
Malloc (n=50000)
  free(malloc(8))                     61   61   61   60   61     19.54
  push(i)                             38   40   38   39   36     12.01
  i1 = pop()                           8    9    8    8    9      2.07
String Functions (n=500000)
  strcpy(s, s0123456789)             126  126  126  126  126      3.47
  i1 = strcmp(s, s)                  144  145  145  145  145      4.10
  i1 = strcmp(s, sa123456789)         93   94   93   93   93      2.38
String/Number Conversions (n=50000)
  i1 = atoi(s12345)                   17   16   16   16   16      4.67
  sscanf(s12345, sd, &i1)            203  202  201  202  211     67.21
  sprintf(s, sd, i)                  158  157  164  164  158     52.67
  f1 = atof(s123_45)                1162 1155 1156 1155 1155    384.81
  sscanf(s123_45, sf, &f1)          1057 1056 1056 1056 1056    351.34
  sprintf(s, sf62, 123.45)          1101 1099 1101 1100 1099    365.94
Math Functions (n=50000)
  i1 = rand()                         14   16   14   15   14      4.14
  f1 = log(f2)                        32   32   31   32   32      9.87
  f1 = exp(f2)                        37   38   38   38   38     11.87
  f1 = sin(f2)                        45   45   45   45   46     14.34
  f1 = sqrt(f2)                       61   60   61   62   61     19.61
```

Name: Bonnie

Category: I/O
Language: C
Source: toklab.ics.es.osaka-u.ac.jp --> /unix/benchmarks

Description:

Bonnie is the V2.0 version of the I/O throughput benchmark, filesys (FSX). Bonnie measures file system performance under conditions designed to resemble operations on large text databases using a 100MB file. It attempts to quantify the performance of several file system operations that have been observed to be bottlenecks in I/O intensive applications. This evaluation is achieved by performing a series of tests on a file of known size and displaying the results relative to CPU utilization. These tests fall into three categories and are divided as follows.

Sequential Output

(1) Per Character uses the putc() function to write the file.
(2) Block uses the write() function to output the file.
(3) Rewrite uses the read(), write(), and lseek() functions to evaluate rewriting a file. (Note: The read block is dirtied prior to being rewritten to insure that it is physically rewritten to disk.)

Sequential Input

(1) Per Character uses the getc() function to read the file.
(2) Block uses the read() function to input the file.

Random Seeks

This test performs 1000 seeks to random locations within the test file and uses the read() function to read in a block of data. 10% of the blocks read (100 blocks) are dirtied and rewritten to disk.

Sample Output:

```
File './Bonnie.391', size: 104857600
Writing with putc()...done
Rewriting...done
Writing intelligently...done
Reading with getc()...done
Reading intelligently...done
Seeker 2...Seeker 3...Seeker 1...start 'em...done...done...done...
              -------Sequential Output-------- ---Sequential Input-- --Random--
              -Per Char- --Block--- -Rewrite-- -Per Char- --Block--- --Seeks---
Machine    MB K/sec %CPU K/sec %CPU K/sec %CPU K/sec %CPU K/sec %CPU  /sec %CPU
          100   491 98.2  1578 48.6   359 23.8   409 95.4  2125 83.4  40.5 13.7
```

Name: BYTE Unix Benchmark -- V3.0

Category: General Unix
Language: C
Source: ftp.uu.net --> /published/byte/benchmarks/unix
qiclab.scn.rain.com --> /pub/bench

Description:

The BYTE Magazine Unix benchmarks (last updated in July, 1991) includes measurements of double precision arithmetic (dhrystone 2 with and without register variables), 7 arithmetic measures, system call overhead, process creation (fork and execl), file copy throughput, pipe throughput and context switching, and a recursive Tower of Hanoi. These benchmark scores are used to calculate a BYTE bench relative index. Many of the tests found in the BYTE benchmark are adapted versions of existing benchmarks, in fact, the musbus benchmark appears to have been a major influence on the writers of the BYTE bench. The benchmark suite itself is made up of the following tests. These tests may be run independently, by group, or as entire suite.

Test	Description
double	Test the performance of double precision floating point arithmetic.
dhry2	Perform the Dhrystone 2 benchmark test without using registers.
dhry2reg	Perform the Dhrystone 2 benchmark test using registers.
execl	Evaluate system performance as a function of executing programs using the Unix *execl* command.
fstime	Evaluate the performance of File System Throughput.
shell	Evaluate System performance by loading the system with concurrent shell scripts.
arithoh	Calculate the overhead associated with Arithmetic Processing.
pipe	Evaluate the throughput performance of the Unix pipe.
context1	Evaluate the performance of Unix context switching.
spawn	Evaluate the system performance with respect to process creation.
register	Test the performance of register arithmetic.
short	Test the performance of short integer arithmetic.
int	Test the performance of integer arithmetic.
long	Test the performance of long integer arithmetic.
float	Test the performance of single precision floating point arithmetic.
syscall	Evaluate the overhead associated with Unix system calls.
C	Evaluate system performance with relation to the time required to compile and link a test program.
dc	Evaluate system performance using the *dc* utility to calculate the square root of 2 to 99 decimal places.
hanoi	Evaluate the performance of recursive processing by solving the Towers of Hanoi puzzle.

Sample Output:

```
   BYTE UNIX Benchmarks (Version 3.11)
   System -- bart
   Start Benchmark Run: Thu Jun 23 13:49:12 EDT 1994
    5 interactive users.
Dhrystone 2 without register variables     22388.2 lps   (10 secs, 6 samples)
Dhrystone 2 using register variables       22242.8 lps   (10 secs, 6 samples)
Arithmetic Test (type = arithoh)           75964.2 lps   (10 secs, 6 samples)
Arithmetic Test (type = register)           2541.2 lps   (10 secs, 6 samples)
Arithmetic Test (type = short)              2365.9 lps   (10 secs, 6 samples)
Arithmetic Test (type = int)                2534.4 lps   (10 secs, 6 samples)
Arithmetic Test (type = long)               2538.0 lps   (10 secs, 6 samples)
Arithmetic Test (type = float)              1862.6 lps   (10 secs, 6 samples)
Arithmetic Test (type = double)             2529.9 lps   (10 secs, 6 samples)
System Call Overhead Test                   4573.8 lps   (10 secs, 6 samples)
Pipe Throughput Test                        2736.7 lps   (10 secs, 6 samples)
Pipe-based Context Switching Test           1206.1 lps   (10 secs, 6 samples)
Process Creation Test                         46.7 lps   (10 secs, 6 samples)
Execl Throughput Test                         17.5 lps   (9 secs, 6 samples)
File Read  (10 seconds)                     9919.0 KBps  (10 secs, 6 samples)
File Write (10 seconds)                      800.0 KBps  (10 secs, 6 samples)
File Copy  (10 seconds)                      287.0 KBps  (10 secs, 6 samples)
File Read  (30 seconds)                    10136.0 KBps  (30 secs, 6 samples)
File Write (30 seconds)                     1299.0 KBps  (30 secs, 6 samples)
File Copy  (30 seconds)                      289.0 KBps  (30 secs, 6 samples)
C Compiler Test                               33.6 lpm   (60 secs, 3 samples)
Shell scripts (1 concurrent)                  29.6 lpm   (60 secs, 3 samples)
Shell scripts (2 concurrent)                  16.0 lpm   (60 secs, 3 samples)
Shell scripts (4 concurrent)                   8.0 lpm   (60 secs, 3 samples)
Shell scripts (8 concurrent)                   4.0 lpm   (60 secs, 3 samples)
Dc: sqrt(2) to 99 decimal places             626.5 lpm   (60 secs, 6 samples)
Recursion Test--Tower of Hanoi               598.8 lps   (10 secs, 6 samples)


                    INDEX VALUES
TEST                                        BASELINE     RESULT       INDEX

Arithmetic Test (type = double)               2541.7     2529.9         1.0
Dhrystone 2 without register variables       22366.3    22388.2         1.0
Execl Throughput Test                           16.5       17.5         1.1
File Copy  (30 seconds)                        179.0      289.0         1.6
Pipe-based Context Switching Test             1318.5     1206.1         0.9
Shell scripts (8 concurrent)                     4.0        4.0         1.0
                                                                    =========
     SUM of  6 items                                                      6.6
     AVERAGE                                                              1.1
```

Name: cWhetstones

Category: Scalar Performance
Language: C
Source: toklab.ics.es.osaka-u.ac.jp --> /unix/benchmarks

Description:

This benchmark is a translation of the Whetstones benchmark from Fortran to C. The bench evaluates a system's scalar performance and calculates a relative Whetstones value for the machine in terms of MIPS.

Sample Output:

```
Whetstone MIPS = 5.555556
```

Name: Dhrystone 2.1

Category: CPU -- Integer Processing
Language: C
Source: qiclab.scn.rain.com --> /pub/bench
from netlib@ornl.gov; "send index from benchmark"

Description:
Dhrystone is a synthetic workload developed by R.P. Wecker in 1984. It is patterned after the Whetstone benchmark but reflects a systems rather than scientific workload. This benchmark package consists of two versions of the Dhrystone benchmark, one that uses registers in its processing and one that does not. Modified versions of these benchmarks are part of the BYTE magazine Unix Test suite among others.

This benchmark strictly tests the system (integer) processing. By design, it makes no use of file I/O as part of its processing and has been coded to attempt to make the results independent of the underlying operating system. Results are given in terms of Dhrystones. The greater the Dhrystone number, the better the overall performance. The program gives a display of output to ensure correct processing. It the gives the number of microseconds per Dhrystone and the number of Dhrystones per second.

Sample Output:

```
Dhrystone Benchmark, Version 2.1 (Language: C)

Program compiled without 'register' attribute

Please give the number of runs through the benchmark: 100000

Execution starts, 100000 runs through Dhrystone
Execution ends

Final values of the variables used in the benchmark:

Int_Glob:            5
        should be:   5
Bool_Glob:           1
        should be:   1
Ch_1_Glob:           A
        should be:   A
Ch_2_Glob:           B
        should be:   B
Arr_1_Glob[8]:       7
        should be:   7
Arr_2_Glob[8][7]:    100010
        should be:   Number_Of_Runs + 10
Ptr_Glob->
  Ptr_Comp:          37104
        should be:   (implementation-dependent)
  Discr:             0
        should be:   0
  Enum_Comp:         2
        should be:   2
  Int_Comp:          17
        should be:   17
  Str_Comp:          DHRYSTONE PROGRAM, SOME STRING
        should be:   DHRYSTONE PROGRAM, SOME STRING
Next_Ptr_Glob->
  Ptr_Comp:          37104
        should be:   (implementation-dependent), same as above
  Discr:             0
        should be:   0
  Enum_Comp:         1
        should be:   1
  Int_Comp:          18
        should be:   18
  Str_Comp:          DHRYSTONE PROGRAM, SOME STRING
        should be:   DHRYSTONE PROGRAM, SOME STRING
Int_1_Loc:           5
        should be:   5
Int_2_Loc:           13
        should be:   13
```

```
Int_3_Loc:            7
        should be:    7
Enum_Loc:             1
        should be:    1
Str_1_Loc:            DHRYSTONE PROGRAM, 1'ST STRING
        should be:    DHRYSTONE PROGRAM, 1'ST STRING
Str_2_Loc:            DHRYSTONE PROGRAM, 2'ND STRING
        should be:    DHRYSTONE PROGRAM, 2'ND STRING

Microseconds for one run through Dhrystone:   65.2
Dhrystones per Second:                      15345.3
```

Name: diskio

Category: I/O
Language: C
Source: toklab.ics.es.osaka-u.ac.jp --> /unix/benchmarks

Description:

This benchmark is part of the NOSC Benchmark Suite. It is used to evaluate Disk I/O Performance. It calculates read and write times and calculates an average I/O Performance Rate.

Sample Output:

```
Run this on an idle system.
blocksize = 8192
write time = 9.787, speed = 1.635 Mbytes/sec
read time = 6.680, speed = 2.395 Mbytes/sec
Average I/O performance = 1.943 Mbytes/sec
```

Name: flops

Category: CPU/Floating Point
Language: C
Source: ftp.nosc.mil --> /pub/aburto

Description:

This benchmark calculates MFLOPS ratings for specific floating point operation mixes. Calculates a peak rating by using primarily registers, minimizing main memory access.

Sample Output:

```
FLOPS C Program (Double Precision), V2.0 18 Dec 1992

 Module     Error        RunTime      MFLOPS
                          (usec)
   1      4.4764e-13      6.4750      2.1622
   2     -8.3933e-14      4.1325      1.6939
   3      1.2879e-14      5.2400      3.2443
   4      4.4187e-14      4.9925      3.0045
   5     -3.9857e-14     11.3750      2.5495
   6      1.2323e-14      9.3350      3.1066
   7      7.9751e-11     10.4725      1.1459
   8     -1.7431e-14      9.5900      3.1283

Iterations      =    4000000
NullTime (usec) =     0.1200
MFLOPS(1)       =     2.0075
MFLOPS(2)       =     1.9166
MFLOPS(3)       =     2.5400
MFLOPS(4)       =     3.1210
```

Name: iostone

Category: I/O Performance
Language: C
Source: toklab.ics.es.osaka-u.ac.jp --> /unix/benchmarks
from: nistlib@cmr.ncsl.nist.gov; "send index"

Description:

This program, developed by Arvin Park at Princeton in 1986, evaluates I/O performance. It measures file system performance for a specific mix of I/O sizes and operations and returns an IOStones rating.

Sample Output:

```
Total elapsed time is 145 seconds and 491 milliseconds
This machine benchmarks at 13747 iostones/second
```

Name: IOzone

Category: I/O Performance
Language:
Source: qiclab.scn.rain.com --> /pub/bench

Description:

This benchmark is a highly portable I/O performance benchmark developed by Bill Norcott to test sequential file I/O. It writes an X megabyte file in Y byte chunks, rewinds the file, and re-reads it. It can also be used to test raw devices, which bypasses Unix buffer caching.

The benchmark can also be run in an auto mode where the test is run using 1-, 2-, 4-, 8-, and 16-megabyte files using 512, 1024, 2048, 4096, and 8192 blocks. This mode can also be made configurable.

Sample Output:

```
        IOZONE: Performance Test of Sequential File I/O  --  V1.16 (10/28/92)
                By Bill Norcott

        Operating System: SunOS -- using fsync()

IOZONE: auto-test mode

        MB      reclen  bytes/sec written   bytes/sec read
        1       512     290044              2405774
        1       1024    317857              3660167
        1       2048    324408              5050677
        1       4096    329517              6199531
        1       8192    1314818             6989807
        2       512     300977              2338647
        2       1024    317710              3700638
        2       2048    312189              5056790
        2       4096    327638              6176812
        2       8192    1482711             6964946
        4       512     300555              1838931
        4       1024    310957              2499790
        4       2048    323351              3025741
        4       4096    321698              3332822
        4       8192    1554280             3737579
        8       512     298028              1247575
        8       1024    312087              1915933
        8       2048    318202              2157534
        8       4096    322069              3244207
        8       8192    1602874             3609049
```

```
        16      512     296362              1166862
        16      1024    311774              1722031
        16      2048    318586              2173327
        16      4096    323088              3197083
        16      8192    1633648             3551316
Completed series of tests
```

Name: IPbench

Category: CPU -- Image Processing Performance.
Language: C
Source: cmr.ncsl.nist.gov --> /nistlib/ipbench

Description:

IPbench is an image processing benchmark developed by Mark T. Noga at Lockheed. It measures the performance of 50 common image processing operations on 512x512 images with 8-bit pixels, emphasizing integer operations, Boolean operations, and program flow.

The bench generates the following information for each test and generates totals for the entire suite at the end of processing. These values are calculated using the *getrusage* Unix system call.

Column Header	Description
o	Total Time Spent Executing the function only (excludes allocation, setup, and writing output) in User and System Mode.
u	Total Time Spent Executing in User Mode.
s	Total Time Spent Executing in System Mode.
t	Total Time Spent Executing in both User and System Mode.
mr	The maximum resident set size (in pages).
pr	The number of page faults serviced which didn't require any physical I/O activity.
pf	The number of page faults serviced which required physical I/O activity. (This could include page ahead operations by the kernel.)
sw	The number of times a process was swapped out of main memory.

Sample Output:

```
create_image1: o = 0.28 u = 0.25 s = 0.18 t = 0.43 mr = 115 pr = 87 pf = 0 sw = 0
create_image2: o = 8.07 u = 8.02 s = 0.23 t = 8.25 mr = 123 pr = 91 pf = 0 sw = 0
create_image3: o = 2.22 u = 2.17 s = 0.21 t = 2.38 mr = 115 pr = 89 pf = 0 sw = 0
AND: o = 0.26 u = 0.26 s = 0.27 t = 0.53 mr = 183 pr = 88 pf = 0 sw = 0
EXOR: o = 0.26 u = 0.21 s = 0.32 t = 0.53 mr = 174 pr = 84 pf = 0 sw = 0
OR: o = 0.24 u = 0.21 s = 0.28 t = 0.49 mr = 186 pr = 89 pf = 0 sw = 0
average1: o = 0.34 u = 0.34 s = 0.36 t = 0.70 mr = 250 pr = 95 pf = 0 sw = 0
average2: o = 0.28 u = 0.30 s = 0.38 t = 0.68 mr = 245 pr = 86 pf = 0 sw = 0
complement: o = 0.24 u = 0.22 s = 0.27 t = 0.49 mr = 178 pr = 84 pf = 0 sw = 0
complement1: o = 0.24 u = 0.24 s = 0.25 t = 0.49 mr = 186 pr = 84 pf = 0 sw = 0
complement2: o = 0.27 u = 0.25 s = 0.26 t = 0.51 mr = 177 pr = 84 pf = 0 sw = 0
dot_diff: o = 22.85 u = 22.46 s = 0.65 t = 23.11 mr = 430 pr = 91 pf = 0 sw = 0
east_edge: o = 0.81 u = 0.78 s = 0.31 t = 1.09 mr = 180 pr = 82 pf = 2 sw = 0
enlarge2x: o = 0.75 u = 0.59 s = 0.62 t = 1.21 mr = 370 pr = 82 pf = 2 sw = 0
enlarge3x: o = 1.77 u = 1.31 s = 1.37 t = 2.68 mr = 692 pr = 92 pf = 72 sw = 0
floyd: o = 5.21 u = 4.96 s = 0.53 t = 5.49 mr = 432 pr = 83 pf = 1 sw = 0
highpass1: o = 0.53 u = 0.51 s = 0.29 t = 0.80 mr = 186 pr = 82 pf = 2 sw = 0
highpass2: o = 0.79 u = 0.78 s = 0.24 t = 1.02 mr = 176 pr = 82 pf = 2 sw = 0
highpass3: o = 0.78 u = 0.75 s = 0.31 t = 1.06 mr = 181 pr = 82 pf = 3 sw = 0
hist_slide: o = 0.31 u = 0.31 s = 0.25 t = 0.56 mr = 183 pr = 83 pf = 1 sw = 0
hist_slide1: o = 0.26 u = 0.24 s = 0.28 t = 0.52 mr = 187 pr = 82 pf = 2 sw = 0
hist_stretch: o = 0.49 u = 0.44 s = 0.30 t = 0.74 mr = 183 pr = 83 pf = 2 sw = 0
```

```
histogram: o = 0.18 u = 0.22 s = 0.17 t = 0.39 mr = 117 pr = 86 pf = 2 sw = 0
hline_edge: o = 0.94 u = 0.92 s = 0.27 t = 1.19 mr = 185 pr = 83 pf = 1 sw = 0
horizontal_edge: o = 0.33 u = 0.31 s = 0.30 t = 0.61 mr = 182 pr = 83 pf = 2 sw = 0
laplacian1: o = 0.75 u = 0.73 s = 0.28 t = 1.01 mr = 184 pr = 83 pf = 2 sw = 0
laplacian2: o = 0.53 u = 0.51 s = 0.27 t = 0.78 mr = 180 pr = 83 pf = 2 sw = 0
laplacian3: o = 0.79 u = 0.74 s = 0.31 t = 1.05 mr = 186 pr = 84 pf = 1 sw = 0
lowpass1: o = 1.41 u = 1.38 s = 0.28 t = 1.66 mr = 177 pr = 83 pf = 2 sw = 0
lowpass2: o = 1.71 u = 1.70 s = 0.28 t = 1.98 mr = 180 pr = 84 pf = 1 sw = 0
lowpass3: o = 1.10 u = 1.06 s = 0.30 t = 1.36 mr = 176 pr = 84 pf = 1 sw = 0
lrline_edge: o = 0.98 u = 0.95 s = 0.27 t = 1.22 mr = 177 pr = 83 pf = 2 sw = 0
median: o = 6.59 u = 6.56 s = 0.27 t = 6.83 mr = 176 pr = 83 pf = 2 sw = 0
north_edge: o = 0.81 u = 0.75 s = 0.32 t = 1.07 mr = 180 pr = 83 pf = 2 sw = 0
northeast_edge: o = 0.81 u = 0.80 s = 0.25 t = 1.05 mr = 188 pr = 83 pf = 1 sw = 0
northwest_edge: o = 0.83 u = 0.80 s = 0.30 t = 1.10 mr = 178 pr = 83 pf = 2 sw = 0
order_dither: o = 3.68 u = 3.48 s = 0.45 t = 3.93 mr = 439 pr = 83 pf = 2 sw = 0
reduction2x: o = 0.06 u = 0.06 s = 0.22 t = 0.28 mr = 139 pr = 83 pf = 2 sw = 0
reduction3x: o = 0.03 u = 0.04 s = 0.18 t = 0.22 mr = 118 pr = 84 pf = 1 sw = 0
rlline_edge: o = 0.94 u = 0.90 s = 0.30 t = 1.20 mr = 177 pr = 84 pf = 1 sw = 0
shift_left: o = 0.24 u = 0.20 s = 0.28 t = 0.48 mr = 177 pr = 82 pf = 2 sw = 0
south_edge: o = 0.81 u = 0.77 s = 0.28 t = 1.05 mr = 185 pr = 83 pf = 2 sw = 0
southeast_edge: o = 0.81 u = 0.78 s = 0.30 t = 1.08 mr = 180 pr = 84 pf = 1 sw = 0
southwest_edge: o = 0.81 u = 0.75 s = 0.31 t = 1.06 mr = 179 pr = 84 pf = 1 sw = 0
subtract: o = 0.35 u = 0.35 s = 0.33 t = 0.68 mr = 251 pr = 84 pf = 1 sw = 0
threshold: o = 0.28 u = 0.27 s = 0.27 t = 0.54 mr = 177 pr = 84 pf = 1 sw = 0
translate: o = 0.50 u = 0.47 s = 0.28 t = 0.75 mr = 176 pr = 84 pf = 1 sw = 0
transpose: o = 0.22 u = 0.18 s = 0.31 t = 0.49 mr = 175 pr = 83 pf = 2 sw = 0
unsharp: o = 2.20 u = 2.17 s = 0.28 t = 2.45 mr = 188 pr = 83 pf = 2 sw = 0
vertical_edge: o = 0.33 u = 0.30 s = 0.31 t = 0.61 mr = 182 pr = 83 pf = 2 sw = 0
vh_edge: o = 0.32 u = 0.34 s = 0.25 t = 0.59 mr = 180 pr = 85 pf = 0 sw = 0
vline_edge: o = 0.94 u = 0.89 s = 0.31 t = 1.20 mr = 185 pr = 85 pf = 0 sw = 0
west_edge: o = 0.81 u = 0.77 s = 0.29 t = 1.06 mr = 179 pr = 85 pf = 0 sw = 0
totals: o = 78.34 u = 75.75 s = 16.98 t = 92.73
This benchmark run using compiler options: cc -O
```

Name: Logicbc

Category: CPU - Logical Bit Operation Performance
Language: C
Source: cms.ncsl.nist.gov --> /nistlib/export

Description:

This benchmark evaluates CPU performance with respect to logical bit operation.

Sample Output:

```
PERFORMANCES IN MLOPS:
V = S a V: 12.71
V = S o V: 9.12
V = V a V: 19.07
V = V o V: 19.07
V = V o (S a V): 36.47
V = V o (V a V): 27.96
V = (V a V) o (V a V): 20.97
```

Name: Netperf

Category: Network Performance
Language:
Source: col.hp.com --> dist/networking/benchmarks
sgi.com
ftp.csc.liv.ac.uk (and mirrors)

Description:

A networking performance benchmark/tool. The current version includes throughput (bandwidth) and request/response (latency) tests for TCP and UDP using the BSD sockets API. Future versions will support additional tests for DLPI, XTI/TLI-TCP/UDP, and WINSOCK; in no particular order, depending on the

whim of the author and public opinion. Included with the source code is a .ps manual, two manpages, and a number of example scripts.

Name: nhfsstone

Category: NFS Performance
Language:
Source: qiclab.scn.rain.com --> /pub/bench
toklab.ics.es.osaka-u.as.jp --> /unix/benchmarks
available from: nhfsstone-request@legato.com
"send unsupported nhfsstone"

Description:

A measure of Network File System (NFS) performance developed and maintained by Legato Systems. It measures server response time and server load (calls per second).

This benchmark generates an artificial load with a particular mix of NFS operations.

Name: tfftdp

Category: CPU
Language: C
Source: ftp.nosc.mil --> /pub/aburto

Description:

This benchmark performs Fast Fourier Transforms (FFTs) using the Duhame-Hollman method.

Sample Output:

```
FFT benchmark - Double Precision - V1.0 -  05 Jan 1993

FFT size   Time(sec)   max error
      16      0.0000     8.9e-16
      32      0.0000     2.7e-15
      64      0.0100     1.2e-14
     128      0.0100     2.9e-14
     256      0.0200     5.9e-14
     512      0.0200     1.7e-13
    1024      0.0700     3.4e-13
    2048      0.1300     9.1e-13
    4096      0.2600     1.8e-12
    8192      0.5300     5.5e-12
   16384      1.1200     1.1e-11
   32768      2.3500     2.2e-11
   65536      4.8800     5.8e-11
  131072     10.1500     1.5e-10
  262144     21.1400     3.2e-10

BenchTime (sec) =    51.3400
VAX_FFTs =      2.740
```

Name: ttcp

Category: Network Throughput Performance
Language: C
Source: sgi.com --> sgi/src/ttcp

Description:

The most commonly used measure of TCP/IP performance. Measures throughput on a single TCP or UDP circuit. Results are not verified or audited, and like TP1 the benchmark is frequently "enhanced".

Sample Output:

```
--> Sender Data <--

bart%  ttcp -t -s -v bart
ttcp-t: buflen=8192, nbuf=2048, align=16384/0, port=5001  tcp  -> bart
ttcp-t: socket
ttcp-t: connect
ttcp-t: 16777216 bytes in 16.01 real seconds = 1023.57 KB/sec +++
ttcp-t: 16777216 bytes in 7.63 CPU seconds = 2147.31 KB/cpu sec
ttcp-t: 2048 I/O calls, msec/call = 8.00, calls/sec = 127.95
ttcp-t: 0.1user 7.5sys 0:16real 47% 0i+66d 33maxrss 0+1pf 4085+96csw
ttcp-t: buffer address 0xc000


--> Receiver Data <--


bart%  ttcp -r -s -v
ttcp-r: buflen=8192, nbuf=2048, align=16384/0, port=5001  tcp
ttcp-r: socket
ttcp-r: accept from 192.9.200.1
ttcp-r: 16777216 bytes in 16.28 real seconds = 1006.38 KB/sec +++
ttcp-r: 16777216 bytes in 5.28 CPU seconds = 3103.03 KB/cpu sec
ttcp-r: 4192 I/O calls, msec/call = 3.98, calls/sec = 257.49
ttcp-r: 0.0user 5.2sys 0:16real 32% 0i+58d 29maxrss 0+1pf 97+4115csw
ttcp-r: buffer address 0xc000
```

Name: x11perf

Category: X-Windows Performance Monitor
Language: C
Source: ftp.uu.net --> /published/open-system-today

Description:

This benchmark evaluates X-windows performance. When run fully for both CXcopy and GXxor, the output from this bench can be piped to a file that can then be used as input to the Xmark program and result in the generation of an Xmark for the system. The test itself consists of a number of specific tests that can be run individually or as part of the entire suite.

Name: xbench

Category: X Graphics Performance
Language: C
Source: qiclab.scn.rain.com --> /pub/bench

Description:

This bench calculates the Xstone rating for a computer's client server. It consists of a number of tests that can be run individually or as a test. When run its entirety, the suite will calculate an Xstone rating for the system.

Name: Xmark

Category: Utility
Language: C
Source: ftp.x.org --> /R5contrib/Xmark1.15

Description:

This module generates an Xmark score based on the results of the x11perf benchmark. The program does not evaluate systems by itself, it is simply a score evaluation.

Name: Xwinstones

Category: X Graphics Performance
Language: C
Source: qiclab.scn.rain.com --> /pub/bench

Description:

This benchmark suite performs a variety of tests used to evaluate the X-windows performance for a machine. These tests can be run individually or as a group to calculate a number of winstone values.

*U.S. GOVERNMENT PRINTING OFFICE: 1998-610-130-81002