

The Growth of the UniTree Mass Storage System at the NASA Center for Computational Sciences: Some Lessons Learned

Adina Tarshish

Mass Storage and Scientific Computing Branch
NASA/Goddard Space Flight Center, Code 931
Greenbelt, MD 20771
k3art@dirac.gsfc.nasa.gov
phone: +1-301-286-6592
fax: +1-301-286-1634

Ellen Salmon

Mass Storage and Scientific Computing Branch
NASA/Goddard Space Flight Center, Code 931
Greenbelt, MD 20771
xrems@dirac.gsfc.nasa.gov
phone: +1-301-286-7705
fax:+1-301-286-1634

Abstract

In October 1992, the NASA Center for Computational Sciences made its Convex-based UniTree system generally available to users. The ensuing months saw growth in every area. Within 26 months, data under UniTree control grew from nil to over 12 terabytes, nearly all of it stored on robotically mounted tape. HiPPI/UltraNet was added to enhance connectivity, and later HiPPI/TCP was added as well. Disks and robotic tape silos were added to those already under UniTree's control, and 18-track tapes were upgraded to 36-track. The primary data source for UniTree, the facility's Cray Y-MP/4-128, first doubled its processing power and then was replaced altogether by a C98/6-256 with nearly two-and-a-half times the Y-MP's combined peak gigaflops. The Convex/UniTree software was upgraded from version 1.5 to 1.7.5, and then to 1.7.6. Finally, the server itself, a Convex C3240, was upgraded to a C3830 with a second I/O bay, doubling the C3240's memory and capacity for I/O.

This paper describes insights gained and reinforced with the burgeoning demands on the UniTree storage system and the significant increases in performance gained from the many upgrades.

Introduction of UniTree at the NASA Center for Computational Sciences

The NASA Center for Computational Sciences (NCCS) provides services to more than 1200 space and Earth science researchers with a range of needs including supercomputing and satellite data analysis. The UniTree file storage management system first arrived at the NCCS on July 6, 1992. As UniTree was to be the primary system for mass storage management, the existing Convex C220 was upgraded to a C3240 with four CPUs, 512 megabytes of memory, and 110 gigabytes of disk. Also included in this initial configuration were 2.4 terabytes of robotic storage provided by two StorageTek 4400 silos. Although UniTree supported both NFS and ftp as access methods, access to UniTree was permitted only through ftp in order to meet the throughput demands of users of the NCCS's Cray Y/MP (UniTree's primary storage client), IBM ES9000, and workstation clients.

The mass storage contract under which Convex/UniTree was obtained required that it be able to handle 32 concurrent transfers while 132 other sessions supported users. The size of files transferred in acceptance tests was realistically large, about 200 megabytes each. The initial Convex UniTree system ultimately showed itself able to manage this workload, and by the third week in September it had passed acceptance.

In those first early months, the growth in UniTree usage was steady, but manageable. There was about 5 GB of new data being stored each day, about 10 GB a day total network traffic to and from UniTree. Ethernet access to UniTree was slow but generally reliable. As Convex UltraNet/HiPPI connectivity was not yet available, many users still preferred the block-mux channel speeds supported by the MVS Cray Station and continued to use the IBM/MVS legacy system to hold the bulk of their Cray-generated data.

In the course of the next two years we would observe repeated instances where UniTree usage would increase sharply and components of the software and supporting operating system services would fail under the heavy strain. We would note that upgrades to the NCCS's primary compute server would require corresponding upgrades to the mass storage system. We would become painfully aware of the relative immaturity of UNIX-based mass storage software in general and UniTree in specific when compared with other types of software in their availability of tools and ability to take advantage of high performance hardware. Nevertheless, contending with these obstacles, the NCCS's Convex/UniTree system has evolved to one of the most active worldwide, often transferring over 100 GB per day and over half a terabyte a week (Figure 3) while concurrently handling repacking tape activity to free over 150 400-MB tapes per day.

Effects of Compute Environment Upgrades on the UniTree System

With the arrival of UltraNet access for Convex/UniTree in January 1993, the UniTree usage curve took its first sharp upward turn. It was now routine for UniTree to receive 10 GB of new data each day, and for the total traffic to reach 20 GB a day. More and more Cray users began to use UniTree to store their data. In February 1993 the Cray Y-MP/4-128 was upgraded to double

its previous CPU power (Figure 2), and the rate of new data stored in UniTree also doubled to 20 GB/day. By the end of the month more than 7500 silo tapes out of an available total of 10,000 had been written with UniTree data.

Upgrade I: UltraNet and Cray Y/MP

UniTree's growing popularity soon exposed a serious impending threat—we were running out of storage. The only production-level versions of UniTree that existed at that time did not allow for more than 10,000 tapes to be managed by the system, but the NCCS UniTree system had consumed three quarters that amount in its first five months of operation. At our prodding, in early March 1993 Convex developed and installed a modification to allow for up to 100,000 tapes, 18,000 of them for robotic storage and the rest for vaulting, or deep archive. A second modification allowing for 36,000 tapes in robotic storage was installed in mid-April. *Lesson: Find out hard-coded limits as early as possible; have them modified if necessary.*

UniTree vaulting and repacking remained a concern. Our version of Convex UniTree 1.5 included an executable to handle repacking, or removing the "holes" from tapes caused by deleted files, as well as vaulting, or the copying of little-used files onto free-standing tape for deep archive, but neither function worked properly at our site. It was apparent that the additional 8000 "robotic-controlled" tapes now defined by software as the top level in the storage hierarchy would not last for more than a couple of months; without repacking or vaulting, this newly added capacity would merely postpone the consumption of the entire top-level hierarchy. In addition, the two UniTree silos were nearly full: without vaulting, most of the additional 8000 tapes in the top level would not be mounted by robotics but by human operators. On active days, that would amount to hundreds of manual tape mounts a day to read and write users' most recent data. We did not have the operations staff necessary for such an undertaking, nor did we want to slow users' access to most recent files while humans located and mounted the tapes. For these reasons, the NCCS insisted on fully functional repacking and vaulting.

By April 5, 1993, we finally had a working tape repacker for UniTree 1.5. Immediately we began to repack in earnest, freeing hundreds of tapes for new data. By April 22, 1993, we had also succeeded in vaulting to free-standing tapes. Working with Convex, we developed utilities that operators could invoke to write an internal UniTree label on new free-standing tapes, so that they could be used for vaulting. Operators were soon mounting vault tapes 24 hours a day, in an effort to keep the silos from filling. *Lesson: Include tests for repacking and vaulting along with tests for all other essential functions in initial acceptance testing.*

Upgrade II: Cray C98

At the end of August 1993 the Cray YMP was replaced by a Cray C98 with six CPUs. Network traffic to and from UniTree increased to 40 - 70 GB a day, 25 - 35 GB of which was new data. Due to inefficiencies in tape writing, UniTree 1.5 could handle no more than 24 GB of new data in the course of a day. As a result, by November 1994 we began to experience periods when the disk cache would fill and users were unable to store or retrieve any more data. A full disk cache also meant that vaulting and repacking would come to a halt, eventually causing the silos to fill. When UniTree ran out of eligible silo tapes for new data it would simply crash. Attempts were made to facilitate the writing of new data to tape, thereby slowing the filling of disk cache, by isolating the channel paths used for writing. Patches were installed optimizing the order in which files were migrated to tape to free disk cache space sooner. Despite these measures, UniTree had to be scheduled unavailable to users on six separate occasions (totaling 140 hours) for standalone migration and vaulting. The tape writing inefficiencies were not significantly improved until UniTree+ 1.7.5 was installed in late March 1994. *Lessons: In data-intensive environments with storage systems already near maximum load, resource plans to upgrade supercomputers must include provisions to upgrade the storage system if the supercomputer is to be used effectively. Include performance requirements in acceptance testing.*

UniTree Stresses Supporting Subsystems

Heavily used mass storage systems stress the supporting operating system services and hardware in ways unlike those of the traditional compute-intensive applications run on the high-powered machines now serving storage. In the NCCS's experience, networking and tape subsystems are particularly vulnerable. Limitations in these systems have sometimes affected UniTree's ability to write retrievable data.

UltraNet and HiPPI/TCP

Although it capably handled 90% of Cray-UniTree traffic when it was working well, UltraNet's history at the NCCS was troubled. Testing it after it first arrived, we discovered several serious bugs and had to wait for microcode fixes and software patches. (Initially the UltraNet native path was limited to 16 concurrent transfers; use of the host-stack path would crash the Convex; and the Convex would hang if UltraNet executables were used for Ethernet transfers.) While waiting for a patch to fix the latter problem, Ethernet access was disallowed on the port used by the UltraNet executables, and Ethernet transfers were given a separate port. After these initial bugs were fixed, a subtle timing problem between Cray and Convex UltraNet transfers intermittently afflicted transfers, sometimes affecting over a thousand connections a day. None of the vendors involved had experienced these failures between machines on their own floors. Concerted efforts by Cray and Convex staff resulted in an improved, but not cured, situation. *Lesson: A high-performance product that works well in the homogeneous environment on your vendor's floor won't necessarily work well in your heterogeneous shop.*

Under UniTree+ 1.7.5 we discovered that an abrupt abort of a single Cray UltraNet transfer would cause all other UniTree transfers to hang. Such an abort was regularly caused by a Cray user's deleting an NQS job that was actively transferring to UniTree. Attempts were made to have NQS job deletion and the "kill" command terminate processes less abruptly on the Cray, but with mixed results. Again Cray and Convex staff worked together to mitigate the problem, but their efforts were impeded by the difficulty in finding expertise from CNT/UltraNet. The problem was encountered during a period of financial uncertainty for the UltraNet corporation, before its acquisition by CNT, and many key UltraNet experts had left the company. *Lesson: Especially for relatively small markets and exotic architecture's, your vendor's company or critical staff may go away; encourage interoperating/dependent vendors to present alternatives.*

UltraNet interoperability problems were not limited to Cray/Convex transfers. The UltraNet hub adaptor repeatedly "autodowned" whenever transfers over a certain size were attempted from the IBM/MVS mainframe. This and related MVS/UltraNet problems were severe enough that the planned transfer via UltraNet of over 500 GB of data from the legacy MVS/HSM system to UniTree was instead detoured via the Cray. Block-mux Cray station transfers moved MVS data sets from IBM/MVS to Cray disk, then the legacy files were transferred via HiPPI to Convex/UniTree. While this was not the preferred use for pricey Cray disk, the duration of this workaround was limited and use of these C98 resources was favored over burdening an already saturated Ethernet with an additional 500 GB in transfers. UltraNet connections on the Convex and Cray were ultimately replaced with a HiPPI/TCP connection to an 8 x 8 HiPPI switch in September 1994. *Lesson: Significant systems problems sometimes require creative short-term contingency plans that use resources in unconventional ways.*

Initial experiences with a point-to-point HiPPI connection between Cray and Convex were also inauspicious. These initial problems were resolved after it was determined that the two vendors had been adhering to different parts of the standard. *Lesson: Despite acceptance of standards, interoperability between vendors cannot be taken for granted because the standards are subject to interpretation.*

Network Resource Allocation

Difficulties also arose when, to add a point-to-point HiPPI connection between the Cray and Convex, we upgraded the ConvexOS operating system from release 10.2 to 11.0. Aiming to maximize network performance, we increased certain UniTree networking parameters to values that had produced best results in testing at Convex, and noted promising performance during testing. Running with these parameters in production mode, we began to see numerous networking allocation failures, a phenomenon not observed during the HiPPI point-to-point stress testing. In addition, some users reported discovery of certain UniTree files that had been corrupted. We immediately reduced the networking parameters values to minimize the

occurrence of the allocation failures. Convex staff identified the problem as a mishandling of the allocation failures and worked steadily on a patch to prevent the data corruption when these failures recurred. Evidence pointed to heavy Ethernet traffic as a primary factor in the allocation failures, as the slower Ethernet transfers tie up resources for a longer period of time than do HiPPI transfers. After painstaking analysis of the UniTree log files, the NCCS identified and published the list of all files at risk of having been corrupted by the problem. We installed and tested the ConvexOS patch as soon as it was available, and, although network allocations continue to fail under heavy Ethernet loads, the failures are now handled properly with no further data corruption. However, periods of these network allocation failures result in some user transfers failing, migration and repacking slowing to a crawl, and the annoying inability to use UNIX pipes and sockets. *Lesson: Stress tests aimed at pushing high-speed interfaces won't catch all systems problems; include stress tests with lower-performance interfaces in your test suites and add tests for new "gotchas" as you find them..*

Tape Driver Travails

In February 1994, the discovery was made that a flawed Tape Library Interface (TLI) driver was causing thousands of consecutive tape marks to be imbedded within UniTree data files, making those files unretrievable by UniTree. Detection and resolution of the problem was belated because this behavior apparently occurred only with UniTree 1.5, and not with any other application. The workaround for the excessive-tape-mark problem was a Convex-written utility designed to wade through the reading of up to half a tape's worth of tape marks before reading data. Attempts to add this tolerance to UniTree's tape system failed because other UniTree processes still timed-out waiting for the files to be read. The suspect driver also caused some internal tape labels to be overwritten by tape marks after the tapes had been written with data. Some of these tapes were recoverable simply by relabeling them (sans end-of-tape mark), but large blank areas following initial tapemarks on other tapes made the data beyond unreadable. With assistance from Convex, we copied and reconstructed these tapes manually. Installation of the patched tape driver, when it became available, ensured that no new tapes would be written with either of these problems. *Lesson: Mass storage applications may reveal system flaws not exposed by other testing; encourage vendors to include characteristics of mass storage systems under load in their system quality assurance test suites.*

Also troublesome were problems eventually attributed to the interaction between an older Convex TLI driver and our freestanding Memorex tape drives, which were used to write least recently used files to operator-mounted tapes. 7.5 percent of tapes written on the Memorex drives with this older version of the TLI driver were discovered to have one or more "null bytes" prepended to the beginning of data blocks. The additional imbedded bytes prevented UniTree's retrieving many files on tapes with this problem. The Convex-written utility that enabled the retrieval of files with embedded multiple tape marks included provisions to retrieve files with "null bytes" as well. This transparent handling of spurious prepended null bytes was successfully added to a customized-for-NCCS version of UniTree tape executables. While the exact cause of the extra null bytes has not been pinpointed, evidence suggests that differences in interpretation of the FIPS-60 standard was a factor. A more serious problem with no known cause occurred on 199 of 27,000 Memorex-written tapes (i.e., fewer than 1 in 1000): entire

blocks of data were missing. UniTree retries unsuccessful writes (on a new tape, if necessary); apparently the driver had not notified UniTree of some unsuccessful block writes. Affected files could not be recovered at all; if a driver problem had caused something extra to be written to UniTree tape, a method could be devised to reconstruct users' files. But there was no way to reconstruct missing data blocks that had no copy on disk.

The Tape daemon/ACSLs silo software saga

As the data under UniTree's control increased, so did the number of requests to retrieve data from UniTree tape. The Convex's tape daemon, used to allocate and deallocate tape drives, was frequently overwhelmed by the load, and communication timeouts and failures between it and the STK ACSLS silo-control software abounded. UniTree 1.5 aggravated the situation considerably by re-requesting the entire list of unsatisfied tape mounts every 2 minutes. There was some discussion about differences in packet addresses and versions being used by the two vendors, and engineers made numerous modifications to both ACSLS silo software and the tape daemon in an effort to mitigate this problem. In addition, the Sun server running the ACSLS silo software was also isolated on a private subnet to eliminate effects of extraneous network traffic on tape daemon/ACSLs communications. Ultimately we were forced to disallow the UniTree "stage" subcommand, which users had been using (and abusing) to request scores of tape mounts simultaneously.

The measures above have significantly reduced the frequency of severe tape daemon/ACSLs communications failures, but another intermittent tape daemon problem persists. Several times a week the tape daemon exhausts its available file descriptors and must be killed and restarted, causing loss of the state of current tape drive allocation and often requiring careful monitoring to restore normal tape allocations while ensuring minimal impact on UniTree. The problem's cause remains elusive after some investigation, and Convex has elected to use its resources to work on the ConvexTMR system which will replace the tape daemon instead of pursuing the file descriptor problem. Delivery of the TMR replacement has been delayed, resulting in some frustration at the prolonged exposure to tape daemon shortcomings—but also some solace in knowing these resources are being applied to resolve remaining TMR problems before its insertion into a production environment.

Science User-Driven Storage System Performance Requirements

In early summer 1993, the NCCS UniTree system was handling about 20 GB new data per day, with some effort. We anticipated delivery of a Cray C98 with more than twice the CPU power of the Cray Y/MP at the end of the summer. The NCCS's users and staff expressed concern about the ability of the UniTree system to handle the additional storage load from the C98. Convex asserted that with the right hardware and software configuration, the NCCS would be

able to meet the users' requirements. Science users were canvassed to determine specific mass storage needs for the foreseeable future (in essence, until augmentation or replacement of the C98). Their responses formed the basis of our acceptance requirements (Table 1) for the upgrades proposed by Convex and the project integrator, FDC Technologies. Although performance requirements appeared strenuous compared to production traffic in summer 1993, we have subsequently seen many instances where production usage approaches the peak loads artificially sustained during acceptance testing.

Reliability:
<ul style="list-style-type: none"> • The Convex/UniTree system must be available 95% of the total scheduled time as well as 95% of the prime shift • No data loss is acceptable • Performance and reliability requirements must be measurable within a normal production environment

Table 1a: Acceptance Requirements—Reliability

Performance (Phase 1):
<ul style="list-style-type: none"> • Store (put) and migrate 85 GB/day; retrieve (get) 300 GB/day from disk and tape, and free 85 GB/day through repacking and vaulting, all operations simultaneously occurring • Demonstrate 96 concurrent transfers of 32 MB each plus 64 "idle" sessions (doing a "dir" or "pwd") • Sustain an average aggregate transfer rate of 9.75 MB/sec • Demonstrate a migration rate of .98 MB/sec

Table 1b: Acceptance Requirements—Performance (Phase 1)

Performance (Phase 2):
<ul style="list-style-type: none"> • Store and migrate 100 GB/day, retrieve 300 GB/day, and free 100 GB/day through repacking and vaulting, all operations occurring simultaneously • Sustain an average aggregate transfer rate of 13 MB/sec • Demonstrate a migration rate of 1.32 MB/sec

Table 1c: Acceptance Requirements—Performance (Phase 2)

Acceptance Testing

The proposed configuration included a Convex C3800 series machine running Convex/Unitree+ 1.7.5. It became clear that peripheral hardware resources required for acceptance testing (UltraNet/HiPPI or HiPPI/TCP connections to the Cray C98, multiple robotic tape drives and controllers) were only available in the NCCS production environment. The NCCS user community was briefed on the need to make the UniTree production system unavailable during acceptance testing; although they preferred 24-hour/7-days-a-week access to UniTree, they recognized the sacrifice would result in longer-term benefits. Testing progressed more slowly than anticipated, complicated by the critically saturated UniTree 1.5 production system and problems discovered in then-Beta UniTree+ 1.7.5 software. Acceptance tests completed in early June, 1994, using production-released Convex/UniTree+ 1.7.6. Performance results are shown in Tables 2 through 5.

Test 1	1.5 Production Observed	Phase 1 Requirements	UniTree+ 1.7.6 Testing	Phase 2 Requirements
ftp "puts" (stores)	58.3 GB/day (0.691 MB/sec)	85.0 GB/day (1.007 MB/sec)	1.183 MB/sec	100 GB/day (1.185 MB/sec)
migration rate	36.0 GB/day (0.427 MB/sec)	85.0 GB/day (1.007 MB/sec)	1.016 MB/sec	100 GB/day (1.185 MB/sec)
ftp "gets" (retrieves)	34.1 GB/day (0.404 MB/sec)	300 GB/day (3.56 MB/sec)	11.558 MB/sec	300 GB/day (3.56 MB/sec)
vault./repack rate	20 GB/day (0.237 MB/sec)	85.0 GB/day (1.007 MB/sec)	1.0528 MB/sec	100 GB/day (1.185 MB/sec)

Table 2: Performance test #1

Test 2	1.5 Production Observed	Phase 1 Requirements	UniTree+ 1.7.6 Testing	Phase 2 Requirements
total ftp sessions	128	160	168	
ftp transfer sessions	32	96	100	none
"idle" ftp sessions	96	64	68	

Table 3: Performance test #2

Test 3	1.5 Production Observed	Phase 1 Requirements	UniTree+ 1.7.6 Testing	Phase 2 Requirements
aggregate network transfer rate	6.5 MB/sec	9.75 MB/sec (150% of observed 1.5 baseline; test system must include tape activity)	12.7417 MB/sec	13.0 MB/sec (200% of observed 1.5 baseline; test system must include tape activity)

Table 4: Performance test #3

Test 4	1.5 Production Observed	Phase 1 Requirements	UniTree+ 1.7.5 Testing	UniTree+ 1.7.6 Testing	Phase 2 Requirements
migration rate	0.658 MB/sec observed on a quiet system	0.98 MB/sec (150% of observed 1.5 baseline)	1.33 MB/sec	1.016 MB/sec	1.32 MB/sec (200% of observed 1.5 baseline)

Table 5: Performance test #4

Current Storage Hardware

The machine that completed acceptance was a 3-CPU Convex C3800 configured with 2 I/O bays. The C3830 has double the memory of the C3240 and more than twice the I/O bandwidth. The addition of the second I/O bay increased the maximum number of channel control units (CCUs) from 8 to 16; 12 CCUs are currently installed, including 2 enabling HiPPI/TCP connections to the Cray C98. Figure 1 shows this storage configuration.

UniTree disk cache has increased from the initial 50 GB to 155 GB for user data. We also obtained 40 GB of disk for RAID, after experiencing disk failures that caused repeated disk process crashes days later, during attempts to access a file with a fragment on the failed disk. *Lesson: RAID has successfully protected user files from disk hardware problems on a number of occasions, and has proven a valuable investment we consider to be worth the reduction in space available for user files.*

NCCS robotic storage has increased to 5 STK 4400 silos with 24 transports. Eight operator-mounted tape drives have been added for vaulting of least-recently-used files. 28 of these 32 transports have been upgraded from 18-track to 36-track. In addition, 22,000 cartridges of 3480 and 3490 tapes are being replaced by 3490E cartridges, which hold approximately 800 MB per tape. Movement of existing files to denser media is accomplished by creative use of repacking.

Cray - Convex/UniTree System

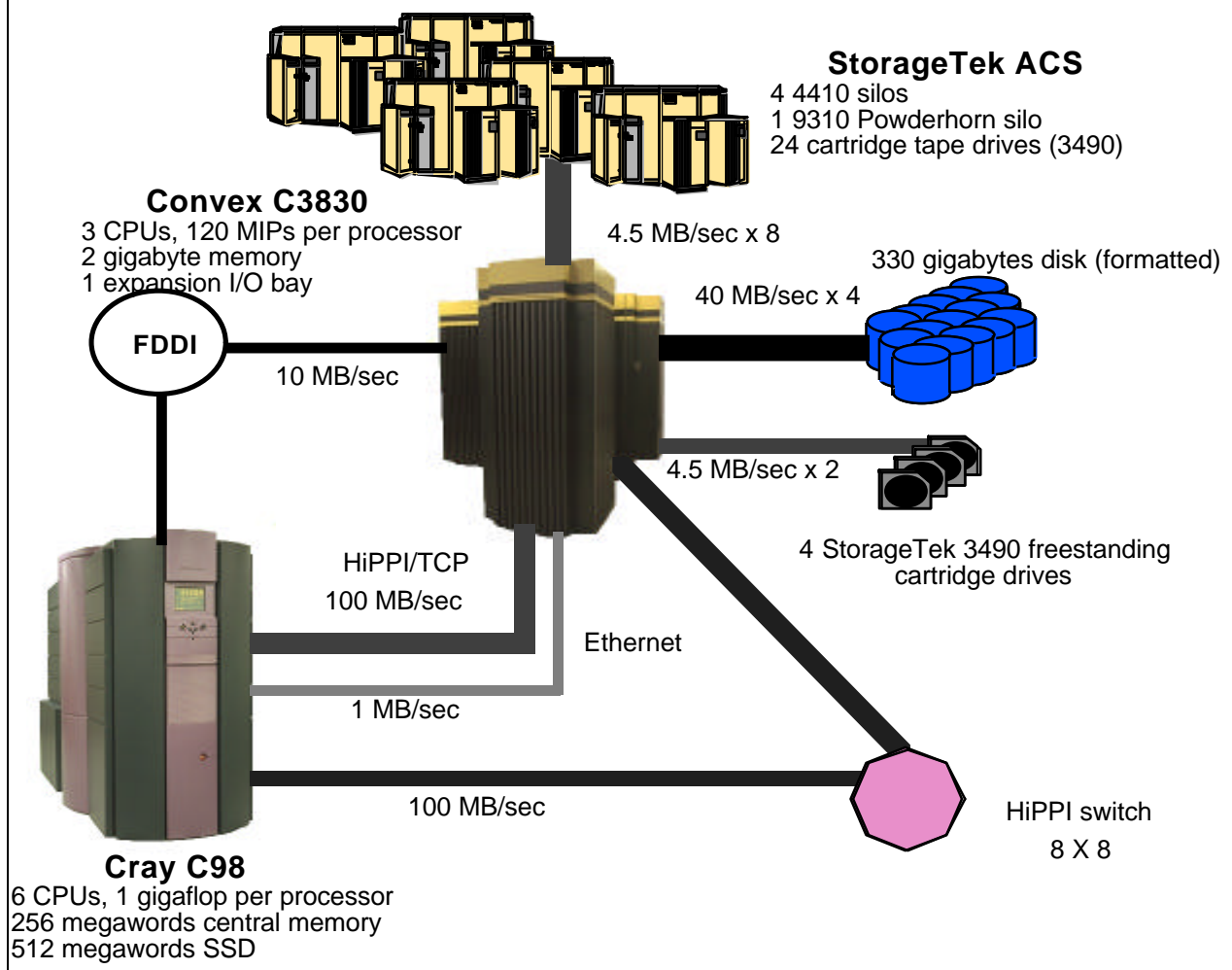


Figure 1: Cray - Convex/UniTree Configuration

Conclusion

The Convex UniTree system in production use at the NCCS today has seen significant improvements since its installation in 1992, and today meets or satisfies most of our expectations, and most of our users' current needs. From a system that could comfortably handle only 25 GB in transfers a day in early 1994, we now routinely handle over 100 GB/day with a high degree of user confidence. Robotic storage capacity has increased an order of magnitude,

from 2.4 to 24 terabytes, with minimal down time due to problems. We are now beta-testing a release of UniTree with features that anticipate our future requirements, unlike 12 months prior when we anxiously awaited a release that would meet our current needs. The process of reaching this current state, however, was not without considerable problems and frustrations. From experiences gathered during the last two years, three themes seem to dominate:

- *Users' input can be a valuable resource.* Their input on future requirements is essential for planning and justifying future acquisitions and for performance requirements in acceptance testing. Our users' feedback and cooperation during critical load times and acceptance testing was crucial to the evolution of performance and capacity improvements on our floor today.
- *Standards don't guarantee interoperability.* At least four problems cited above resulted from several vendors' different interpretations of standards. The standards/interoperability issue also applies to the mass storage software itself. UniTree was among the first UNIX-based mass storage systems to be ported and licensed on a wide variety of platforms. In light of delays on bugfixes and new releases from the previous UniTree originator, and demands for improvement from their customers, individual vendors have made significant modifications to UniTree. Some of these modifications affect a site's ability to move their UniTree tapes and databases to a different vendor's platform. Leveraging strength in numbers, the UniTree Users' Group has gotten vendors and the new originator of UniTree to agree to work together to resolve portability issues.
- *Stress testing:* Include high performance and low performance interfaces in stress testing, and add tests for "gotchas" to the suite as new problems are discovered. If it's possible in your environment, have vendors run acceptance testing with your equipment on your own floor, because it's virtually impossible for vendors to duplicate your environment. If practical, set up a test instance of your mass storage system and Beta/stress test new releases so that problems are detected and resolved before the product is installed on your production system.

The NCCS's science users project the need to transfer 2 terabytes a day by 1999. Up-and-coming high performance media, networks, and the like will achieve the rates required by our high-performance computing users, although the lag between the introduction of new hardware and the operating system and mass storage software's full utilization of its capabilities remains a concern. Our current beta test of Convex UniTree+ 2.0, which better exploits hardware via its enhanced tape resource configurability and multiple migration writes, should provide some insights on system behavior with higher-performance peripherals. But the increased sharing of data fostered by national and global information infrastructure efforts is already broadening the needs and the nature of the NCCS user community. Consequently, the NCCS is investigating interim methods to accommodate the "long haul," lower-speed needs of numerous remote users while sustaining high levels of service to local high-performance computers, although we anticipate researchers' and vendors' eventual development of more elegant means to handle these divergent needs. Current NCCS study involves creative use of UniTree families, tapetypes, and callout scripts to control the impact of many simultaneous remote sessions on high-demand

needs of Cray processing. Our storage system progress to date, although not without its turbulence, induces great optimism about our future ability to meet the needs of both our lower-speed and high-performance science users, whose research activities drive one of the most active mass storage sites world-wide.

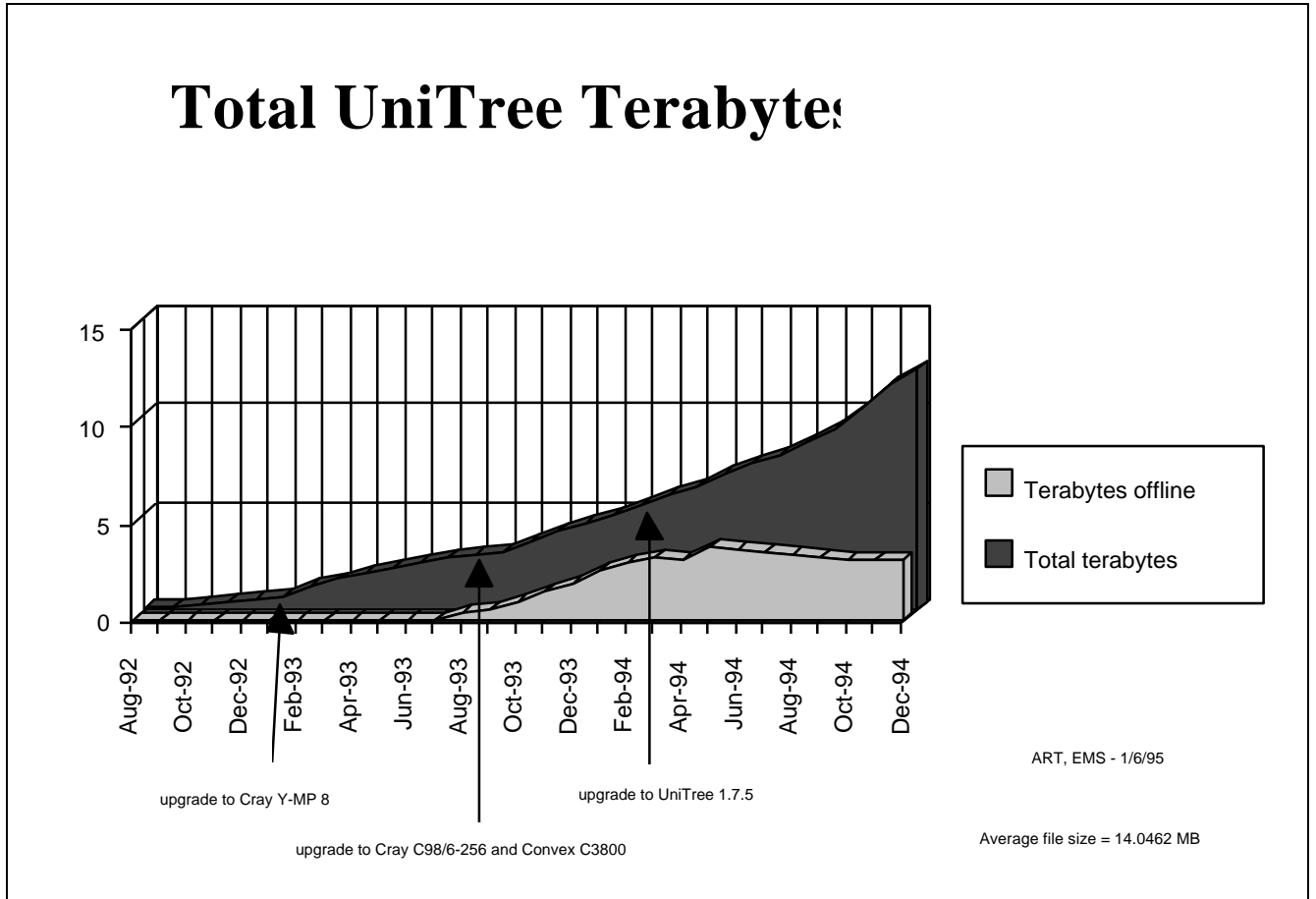
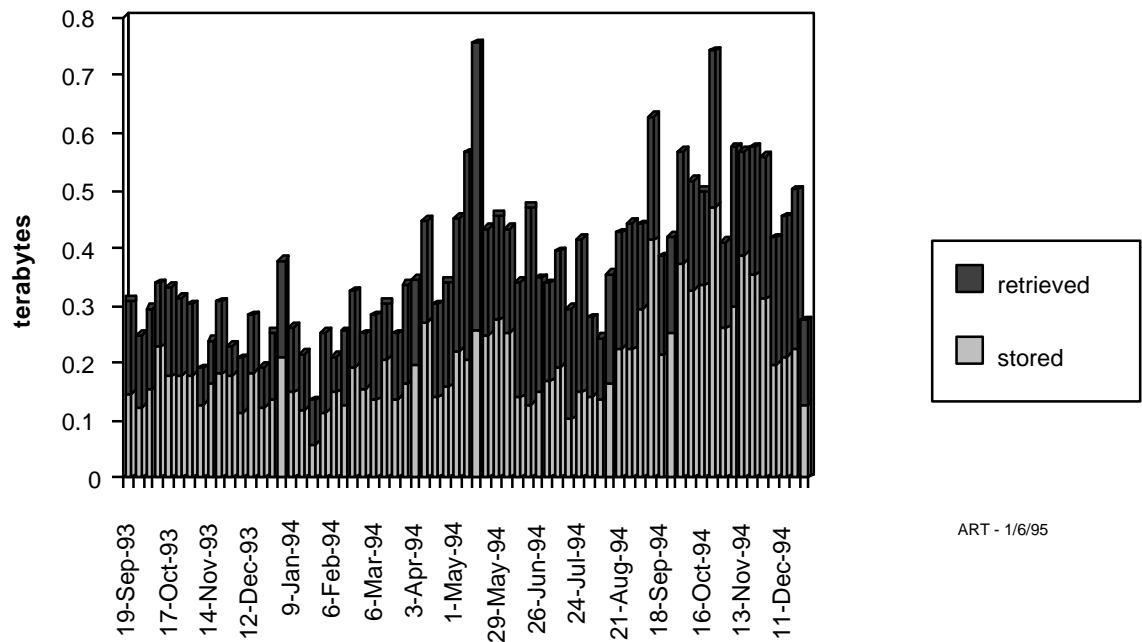


Figure 2: UniTree storage growth at the NCCS

Weekly UniTree Traffic



ART - 1/6/95

Avg stored: 43.63 GB/day avg retrieved: 31.03 GB/day (averages since 10/1/94)

Figure 3: UniTree weekly network activity