Title

Experience with the UniTree Mass Storage

Permalink

https://escholarship.org/uc/item/4c53n8rn

Authors

Holmes, H.
Loken, S.

Publication Date

1992-09-01

LBL-32863
UC-405

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

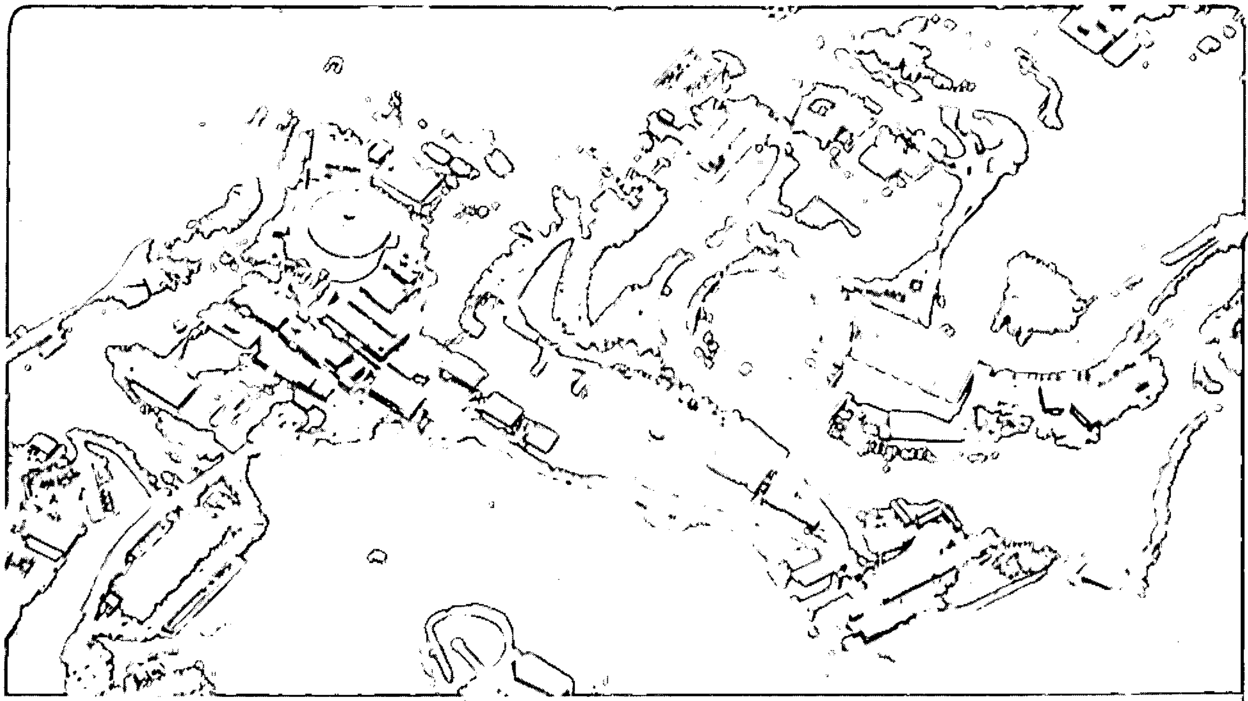
Information and Computing Sciences Division

Presented at the Conference on Computing in High Energy Physics,
Annecy, France, September 21–25, 1992, and to be
published in the Proceedings

Experience with the UniTree Mass Storage System

H.H. Holmes and S. Loken

September 1992



Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. Neither the United States Government nor any agency thereof, nor The Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or The Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or The Regents of the University of California and shall not be used for advertising or product endorsement purposes.

Lawrence Berkeley Laboratory is an equal opportunity employer.

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LBL-32863
UC-405

Experience with the UniTree Mass Storage System

Harvard H. Holmes and Stewart Loken
Information and Computing Sciences Division
Lawrence Berkeley Laboratory
One Cyclotron road
Berkeley, CA 94720

September 1992

Prepared for the 1992 Conference on Computing in High Energy Physics
Annecy, France, September 21-25, 1992

This work was supported by the Director, Office of Energy Research, Office of High Energy Physics and Nuclear Physics, of the U. S. Department of Energy under Contract No. DE-AC03-76SF00098.

Experience with the UniTree Mass Storage System

Harvard Holmes and Stewart Loken

Information and Computing Sciences Division
Lawrence Berkeley Laboratory, Berkeley, CA 94720

Lawrence Berkeley Laboratory (LBL) is a beta test site for the UniTree mass storage system. Our initial configuration is based on a Sun workstation and includes 10 gigabytes (GB) of magnetic disk cache, 700 GB of Exabyte 8 mm tape storage, with two tape robots. We support a user community of 15 to 20 active users, about 250,000 files, and 33 GB of user data. The largest file stored is 1.5 GB. As of May 1992, we consider the system to be adequately stable and reliable for production use. As a beta site, we have worked on the Sun port, on the tape drivers for SunOS, and on integrating our tape robots into the UniTree software. File retrieval from tape usually takes less than five minutes. Continuing concerns are tape longevity and reliability, and improving performance to support 100 Mb FDDI.

UniTree Mass Storage System

The UniTree mass storage system is based on the emerging IEEE Mass Storage Reference Model [1]. As with any mass storage system, it provides the benefits of keeping track of the users' data, and (usually) providing ready access to that data without human intervention. The (future) benefits of standards for mass storage are advertised as the ability to "mix and match" from different vendors; we consider that a difficult proposition. But we agree that in the near term, the standard motivates modularity, which provides flexibility in configuration, and it allows practitioners to build on shared concepts and terminology.

UniTree provides two interfaces to the mass store: a Network File System (NFS) server, and a File Transfer Protocol (FTP) server. At LBL, our primary use is via the NFS server. New files are written to the magnetic disk cache, and when stable (an hour or so of no activity), the files are copied to magnetic tape. At LBL, we make two copies of each file on separate media. The number of copies is an installation parameter. Files remain on the magnetic disk cache until the space is needed; at that time the oldest and largest files are deleted to free up space. Files not on the magnetic disk cache are moved to the cache when they are referenced; after the transfer to the cache is complete, the user's request is satisfied. The FTP server provides a number of capabilities that are not possible using the NFS protocol. The two most important functions are the ability to include the file location (disk or tape) in a directory listing, and the ability to request that a file be staged from tape to disk for later use. Naive use of the NFS interface, e.g., a wildcard copy of files on tape, can cause severe inefficiencies in the mass storage system. In this case, the client operating system expands the wild card and requests the files one at a time from the mass storage system, causing a tape mount/load/unload sequence for each file. With FTP, a stage operation can queue up all the file requests and get them all in one pass over the tape. At LBL, we have developed a stage command which provides this function without requiring logging in and giving a password, as required by FTP. We have also developed a directory listing command which includes the file location, either disk or tape.

LBL's Mass Storage Configuration (and Costs)

LBL's initial configuration is based on a SUN SPARC 2 workstation with twelve 1.2 GB disk drives and a tape carousel with two Exabyte 5 GB drives and 45 tape slots. We are just now installing an Exabyte EXB-120 with 4 drives and 116 tape slots. This configuration will hold about 700 GB of user files in both robots. Approximate hardware costs were: SPARC station, $10,000; disk drives, $30,000; two drive tape carousel, $35,000; and four drive tape robot, $45,000. UniTree software ordinarily costs $35,000 for configurations up to 1000 GB. As a University of California DOE laboratory, LBL obtained the software under an agreement negotiated by Lawrence Livermore National Laboratory; this was a significant consideration at the time.

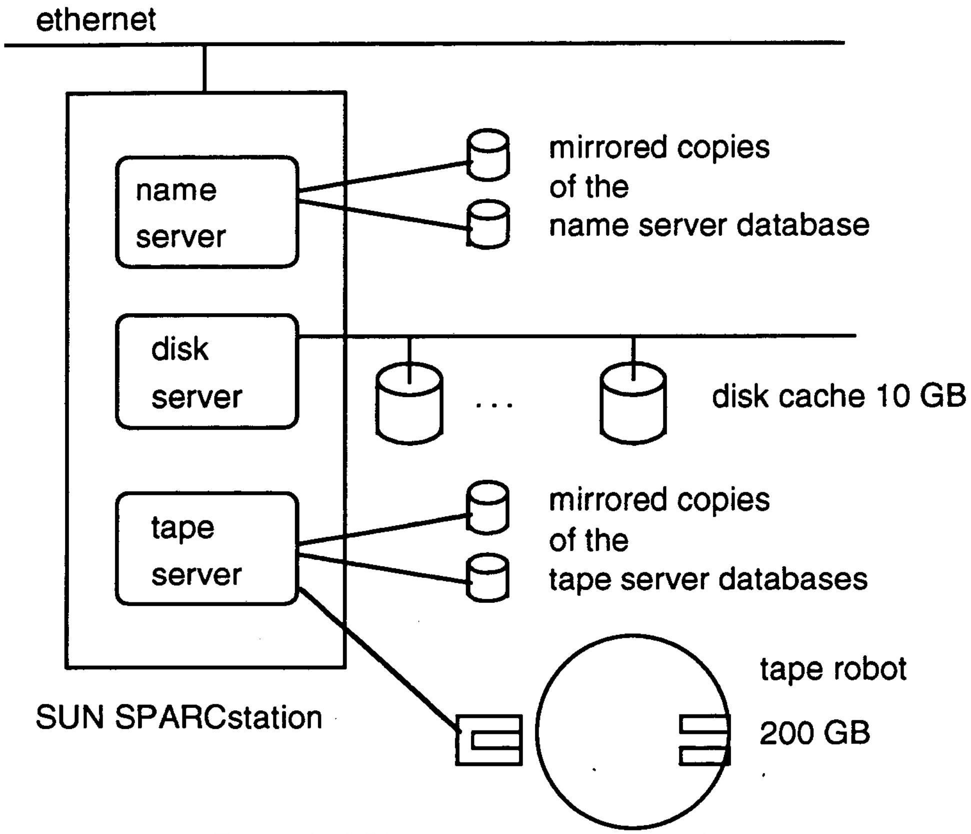


Figure 1. LBL mass storage configuration

LBL has dedicated one full time programmer to develop our mass storage system. He has spent about 50% of his time installing UniTree versions, working to get Exabyte tape drives and the robots integrated with UniTree, and tracking down problems in the UniTree software. Neither SUNs nor Exabyte tape drives nor our tape robots are supported configurations for UniTree, but the software has been able to adapt. We are now working on our fifth major revision of the software.

Operational issues consume another 30% of our programmer's time. These include monitoring the operation of the system, repairing the effects of hardware and software crashes, duplicating tapes for storage off site, adding new accounts, organizing directory structures, monitoring security, and evaluating and ordering new hardware.

During development, we focused on supporting a few groups as test users. Each group had an informal principal contact. As usage has expanded, we have continued to find principal contacts within most groups, as groups seem to centralize this responsibility. Working with these principal contacts has been very effective for our programmer.

Performance

The most important performance issue for a mass storage system is that it not lose files. UniTree is good in this regard; we lost only a few files in beta testing, and we believe we (or the UniTree developers) found and fixed the cause of each loss. We have not lost files since we declared ourselves in "production" status. Occasionally UniTree is not able to accept a file and returns an error message to the client -- this is not considered a lost file. UniTree does not lose files across system crashes or hardware malfunctions -- except disk crashes when the file has not been copied to tape (four to eight hours after receipt), since files are not duplicated on the disk cache. We have not lost files in this way in production, but we advise users not to delete their copy of files for a day or so. Ailing disk drives can be taken out of service without loss of files; the utility which accomplishes this moves files to another disk if they are not on magnetic tape yet.

To obtain the best media performance, we use data grade tapes; we consider it inexpensive insurance. We keep track of the number of loads for each tape, and (attempt to) log tape errors. So far we have not learned how to get the tape error statistics out of the tape drives/drivers, which would allow us to monitor tape error rates. A block by block utility copy program can be used to duplicate a tape that is going bad. We expect that the archival life of the tape will be good enough that we will go to the next generation of storage device before the tape shelf life is reached.

UniTree is quite stable and rarely crashes. Crashes or hangs due to tape problems will usually not affect the disk operations; users are able to continue work, or will see a delay for tape requests, while the cause of the problem is repaired and the tape system is brought back on line. Crashes due to power failures still require an operator to bring the system back up. We and the UniTree developers are working on this.

Data transfer rates to and from UniTree are generally limited by the ethernet connection, 200 to 500 KB/second, but there is not much CPU left when UniTree is busy. Until we install FDDI, we will not worry about efficiency, but we know several areas where UniTree is inefficient, e.g., message passing uses sockets rather than shared memory, asynchronous I/O is not implemented, etc. Access times to tape average two to three minutes if a tape drive is free; see the section "Lessons Learned" for the changes necessary to achieve this speed.

Two systematic stress tests have been run on our UniTree configuration. In the first test we gradually increased the number of processes reading and writing files on the mass storage system to see if some resource limit would be reached. About 330 processes were created before the client workstation nearly exhausted its swap space and became so slow that it was rebooted. The mass storage system slowed down and clients saw "NFS server not responding" messages, but no other ill effects were noted. The mass storage system still had plenty of CPU time left. In the second test, a process was periodically started which asked for a randomly selected file from the MSS. The file could have been on disk or on tape (most likely). This test was run three times, with inter-request intervals of 5 minutes, 1 minute, and 15 seconds. At 5 minute intervals, the tape robot was able to keep up with requests, rarely requiring even a second drive. At one minute intervals, the tape robot was able to keep up with requests, with the queuing mechanisms in the tape server accumulating only a few file requests for each tape mount. (This performance is a strong function of the number of drives available and the number of tapes in the test; in this case, two drives and about 15 to 20 tapes were in use.) At 15 second intervals, the tape robot eventually reached equilibrium with the request rate. Some statistics from this test:

- 2 drives and ~ 15 tapes were in use
- ~ 2700 files were requested
- ~ 150 tape mounts were processed
- the number of requests waiting fluctuated between ~100 and 272
- the average number of requests waiting was 115
- the maximum response time to a request was ~ 4 hours
- the average response time to a request was ~ 22 minutes

Lessons Learned

We have learned a lot about Exabyte 8 mm tapes and how they work. The three things that caused us the most grief were poor tape writing performance, long tape positioning times, and the limitations on switching from reading to writing. Poor tape writing performance was caused by the need to force data onto the physical media before marking that data as positively migrated. This is typically accomplished by writing a file mark and then backing over it to continue writing more data; this takes a prohibitively long time on the Exabyte. The solution is to send a write-file-mark command with a count of zero to the drive, which flushes all buffers without actually writing a file mark. This required modifying the UniTree software and the SunOS device driver. Long tape positioning times were caused by the software sending each skip-tape-mark command separately; both the UniTree software and the device driver in the SUN OS had to be modified to send one command with a count to the drive. The third problem is that the Exabyte drive can only switch from reading to writing if it is immediately before or after a tape mark. The implication is that an aborted write operation cannot be successfully retried, since the tape mark that originally preceded the transition to write mode has been overwritten! After the retry fails, UniTree will go on to the next tape, leaving the rest of the original tape unused. There is currently no way to recover the lost media space, but the next version, due soon, will have a tape repacking function which will recover this lost media. Having conquered these difficulties, we now find that the tape system is at least as reliable as the software.

Since we wanted to integrate several new components into UniTree, our role became more developer than beta tester. It would have been impossible to do what we wanted without source code. In some cases, having the source code allowed us to fix things that were low priority for the vendor; this is a nice thing to be able to do.

The NFS protocol has no way to ask for files in advance. Consequently, users must be provided a way to stage files. Otherwise, by the time the user gets his file, the rewind/unload command has been sent to the tape drive, and the next file must wait for the full sequence of tape loading, threading and seeking. We have developed a simple utility to stage files. While file staging is in the FTP interface, FTP requires a password, and so cannot be placed in a script (otherwise we would have just used FTP). This may be solved when Kerberos or other systematic authentication scheme is put into wide use.

Finally, a mass storage system creates a wide area file system, one that at least spans an entire laboratory or campus. This requires a campus wide set of user identifiers (UIDs), and attention to the related security issues. And it must be easy to use with existing file systems. At LBL, we began with a directory hierarchy that followed administrative boundaries; but this adds one level of complexity that users don't need (or use). We are now just putting users in a master directory, or putting them in subdirectories if a large number of directory entries would cause inefficiency. We eagerly await the next generation of wide area file systems that will give users better performance and give us better administrative control.

Conclusions

Bringing up a relatively new mass storage system has proven to be much more work than expected. While there seemed no other systems at the time with the desired flexibility, other systems have made significant advances during our longer-than-expected development period. Someone investigating the market now has a wide variety of choices, with two of the leading contenders being UniTree and Epoch (which will soon have a software only system). At the moment, UniTree is a contender, but not necessarily in the lead, in the race for features, performance, price and reliability.

Acknowledgment

This work was supported by the Director, Office of Energy Research, Office of High Energy and Nuclear Physics, of the U. S. Department of Energy under Contract No. DE-AC03-76SF00098.

Reference

[1] Colman, S., and Miller, S., eds., Mass Storage System Reference Model: Version 4 *IEEE Technical Committee on Mass Storage Systems and Technology,* 1990

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
TECHNICAL INFORMATION DEPARTMENT
BERKELEY, CALIFORNIA 94720