Configuring and Tuning Archival Storage Systems

Reagan Moore, Joseph Lopez, Charles Lofton, Wayne Schroeder, George Kremenek
San Diego Supercomputer Center
Michael Gleicher
Gleicher Enterprises, LLC

Abstract

Archival storage systems must operate under stringent requirements, providing 100% availability while guaranteeing that data will not be lost. In this paper we explore the multiple interconnected subsystems that must be tuned to simultaneously provide high data transfer rates, high transaction rates, and guaranteed meta-data backup. We examine how resources must be allocated to the subsystems to keep the archive operational, while simultaneously allocating resources to support the user I/O demands. Based on practical experience gained running one of the largest High Performance Storage Systems, we propose tuning guidelines that should be considered by any group that is seeking to improve the performance of an archival storage system.

Introduction

The San Diego Supercomputer Center (SDSC) is the leading edge site for the National Partnership for Advanced Computational Infrastructure (NPACI). This National Science Foundation project supports academic computational research for researchers throughout the United States. NPACI has the aggressive long-term objective of providing a multi-teraflops capable compute engine linked to a petabyte sized archive that can support data movement at rates up to 10 GB/sec. The combined system will support computationally intensive computing in which terabyte-sized data sets are written to the archive and data-intensive computing in which multi-terabyte data sets are read from a data collection stored within the archive. The expectation is that the ability to analyze very large data sets will become a very important supercomputing application.

The critical element in the NPACI data-intensive computing system is an archival storage system that can meet the transaction and I/O rate demands of the compute engine. The long-term goal is to provide the same access to the archive as the compute engine has to its local disk. This will make it possible to decrease the amount of disk cache that must be provided to both systems. Based upon measurements made on Cray supercomputers [1,2], a teraflops capable computer is expected to generate data at the rate of 10 GB/sec. A sustained rate of 10 GB/sec is equivalent to the movement of a petabyte of data per day. Hence our desire to improve archival storage performance by understanding the configuration and tuning needed to keep any component of the system from becoming a performance bottleneck.

The difficulty is that archival storage systems are quite complex. They contain multiple subsystems that support data movement, transaction processing, event logging for meta-data backup, and data migration between multiple levels of a cache hierarchy. We examine each of these subsystems to understand how they can impact overall archival storage performance. We illustrate the configuration selection and associated tuning based upon the High Performance Storage System (HPSS) [3]. The SDSC HPSS system is one of the largest HPSS systems in production use [4]. It stores over 6.2 million files comprising 80 TB of data distributed between three tape silos, and internally moves up to two terabytes of data per day. The total capacity for uncompressed data is 180 TB, using 10-gigabyte tapes. SDSC plans to increase the size of the system by a factor of at least 10 over the next three years. We therefore also look at the scalability of the system to determine whether the goal of a high-performance petabyte archive that can be completely read in a single day is achievable.

Designing an archive

Archival storage systems typically provide resources to support long term storage of data on inexpensive media such as tape. To improve response times, the data is cached on disk. The performance of a system is then usually considered in terms of external usage metrics such as the average access time needed to retrieve data, the rate at which the system can ingest or export data, or the total capacity of the system. From this perspective, archive configuration is focused on the appropriate allocation of resources to disk caches and tape robots to meet user demand.

Of equal importance, however, are the resources that are dedicated to the archival storage infrastructure. These include the number of CPUs used to execute the storage servers, the amount of memory used to support executing processes, the disk space allocated to support internal meta-data directories and transaction logs, and the tape resources dedicated to backup of internal system tables. Tuning of an archival storage configuration must therefore address the allocation of resources to support internal archive functions as well as the external user load.

The configuration of an archival storage system can be understood in terms of three fundamental subsystems:

1. Transaction processing
   - Nameservice
   - Authentication
2. Meta-data management
   - transaction logging
   - directory backups
3. Data movement
   1. data migration across caching hierarchy
   2. load balancing across communication channels and servers
   3. network configuration (internal versus external data transfer)

Each of these subsystems has a maximum sustainable rate at which it can respond to user demands. To increase the performance of the overall system, either more powerful resources must be used to support a subsystem, or the subsystem must be parallelized across multiple hardware and software components. Both approaches were used during an upgrade of the HPSS system at SDSC in September 1998. SDSC examined in detail the tuning required to improve the performance of HPSS. The goal was to eliminate any bottlenecks that the internal subsystems were imposing on standard usage metrics such as sustainable I/O rate and response time. At the same time, we wanted to eliminate all interrupts that were caused by load dependencies within the internal HPSS subsystems, and improve our ability to maintain critical system meta-data.

The tuning steps revealed the complex set of interactions that occur between the archival storage subsystems. A number of potential problems had to be overcome:

Purge rate of migrated files. When files are stored on disk, a backup copy is made to tape within 60 minutes. When the disk cache fills up, a file whose backup copy has been completed can be deleted to make room for new files. Event logging is used to track the completion status of each archive task. Three different steps within this process need to be optimized: determination of which files to migrate; generation of unlink records for use of disk segments; and de-allocation of disk space. It is possible to queue up to 150,000 de-allocation requests under heavy load. The rate at which de-allocation occurs can end up controlling the rate at which files can be purged and, hence, the rate at which new files can be ingested into the archive.

Backup of meta-data. To ensure that user information about files stored within the archive is never lost, multiple mechanisms are used to replicate and back up file meta-data. This includes the mirroring of internal system tables, the storage of snapshots of the system tables, and the logging of all transactions that result in changes to the system tables. The snapshots and logs are written to tape to guarantee system recoverability. When tape resources are shared between meta-data backup and user data migration, it is possible through human error to overwrite a meta-data tape.

Dedication of tape drives to backup can help eliminate this vulnerability.

Meta-data preservation. Even when all backup systems work perfectly, it is possible to lose critical meta-data through human error, such as an erroneous explicit deletion of a meta-data backup file or transaction log file. Protection against this requires that two copies of the meta-data and transaction logs be kept at all times, stored on different media, under distinct path names.

Interference between communication protocols. One advantage of network attached HiPPI RAID systems is that they can support high-speed access (60-80 MB/sec) to large data sets using efficient communication protocols such as IPI-3. Archives must also support external communication utilities such as FTP that rely upon the TCP/IP protocol. Supporting both types of data movement through the same communication channel can lead to contention between large and small file access, and can cause instabilities in network drivers.

Load balancing between distributed servers. A second form of communication contention occurs when access to external networks is through a single I/O driver. Access to the disk cache has to be funneled through the single node to which the external communication channel is attached. This limits the sustainable I/O rate to the capability of a single I/O driver.

Transaction processing. Under heavy load, the number of simultaneous processes that must be run can exceed the size of memory or the compute power of a single node. Options include distributing processes across multiple nodes or using SMP nodes with a very large memory. Use of multiple nodes works well for separate daemons, such as supporting a large number of simultaneous FTP requests, while using large memory SMP nodes works well for servers that need to avoid communication overhead, such as maintaining meta-data directories.

Transaction processing. Under heavy load, access to HPSS internal system tables localized on a single disk can lead to large disk I/O queues and thus limit performance. Explicit distribution of system tables across multiple disks makes the backup procedure more difficult as a larger number of files must be managed. Options include using disk volume groups to create a logical volume that is distributed across multiple disks, or using mirrored disk files to support reading from either of the copies.

System stability. When very heavy usage of the system occurs, internal system tables can fill up faster than the rate at which the system data can be logged and backed up on tape. It is possible for an archival storage system to run out of the resources needed to manage its own system tables. At this point, manual intervention is required to rescue the system. This can be quite complicated, especially if the backup of transaction logs cannot be completed if there is inadequate disk space to assemble the logs.

System availability. A major design consideration was the amount of disk space to dedicate to storage of transaction logs. If a very large disk space is made available, problems associated with very heavy usage periods are minimized, as there will be enough space to log all the transactions. However, a large log space means that the time needed to validate the log and its backup to tape can be excessively long, resulting in increased down time while backups are done.

Improved support for the user I/O load. When the user input is quite varied (storage of large numbers of small files versus storage of very large files), resources must be identified to meet each of the possibly conflicting demands. Thus small files (less than 2 MB in size) should be kept on a large disk cache that minimizes the need to retrieve data from tape. Large files (greater than 200 MB in size) need to be stored on the disk system with the fastest I/O rates to minimize transfer time. This requires automatically separating the user load into multiple service classes to minimize contention, and the appropriate allocation of resources between the service classes. This task can require continual monitoring and modifications to the system, imposing a heavy tuning load on system administrators.

Configuration

The SDSC HPSS configuration is required to support a peak teraflops capable compute engine in 1999. The estimated load that must be sustained in

1999 is on the order of 10 TB of data movement per day, into an archive that holds up to 500 TB of data. The dominant concern in the configuration tuning was the development of a system that could meet the corresponding sustained I/O rate of 100 MB/sec. The approach taken was to first validate the ability of the underlying operating system (AIX) and hardware (IBM SP with eight Silver nodes) to handle the maximum expected load. We then evaluated the I/O capability of the system, and the ability of the archival storage system to drive a large fraction of the system I/O rate.

The IBM SP Silver nodes are SMPs, each containing four 604e processors. The nodes are interconnected by a TrailBlazer3 (TB3) switch, which has a peak I/O bandwidth of 150 MB/sec per node. The HPSS standard suite of load tests was run, using the SP to both generate the load and support the HPSS system. This limited the maximum load level to 7 terabytes of data moved through the HPSS system per day. When the load is generated by a separate compute engine, we expect the sustainable data movement to be at least 10 terabytes of data per day. The stress test included all of the service classes, effectively designed to support small, medium, and large files as shown in Table 1.

The tests revealed multiple hardware problems, some of which could only be seen at the highest load. Upgrades to the RAID disk, the SP switch, and the node hardware eliminated all of the problems. In some cases this required the next version of hardware or software. In other cases faulty hardware was replaced. The process eliminated all of the potential sources of hardware problems for the production use of the system. In fact, during the following month of November, the archival storage system was stable, with the only down time occurring during preventive maintenance periods.

The I/O capability of the system was measured by explicit timed movement of large data sets from disk to an SP node and between SP nodes. Two types of disk storage were used: HiPPI attached MAXSTRAT disk that had a measured I/O bandwidth of 60 MB/sec, and IBM SSA attached RAID disk that sustained up to 30 MB/sec for disk reads. An IBM High Performance Gateway Node (HPGN) was used to tie external communication channels into the IBM TB3 switch. To send data to an external computer, the data would flow from a Silver node, through the TB3 switch into the HPGN, and then across the external network. Data movement between nodes through the TB3 switch was measured at rates up to 130 MB/sec. Data movement between nodes connected through the HPGN was measured at rates up to 90 MB/sec.

The I/O rates that could be sustained using HPSS version 3.2 were then measured. The test configuration consisted of seven Silver SMP nodes. One held the HPSS core servers and three Encina SFS (meta-data) servers. Two nodes supported FTP clients and also served as bitfile mover platforms for accessing SSA RAID disk. Four additional nodes served as dedicated bitfile mover platforms. The nodes were interconnected through the TB3 switch. The core server machine had 3 GB of memory. Five of the remaining nodes had 2 GB of memory, with a 3-GB node assigned randomly as a client or mover.

The test consisted of reading a 2-GB file using the HPSS Parallel File Transfer Protocol (PFTP) interface. PFTP is one of the high performance user interfaces to HPSS and is functionally a superset of conventional FTP.

Case 1. The 2-GB file was read from one SSA RAID string on one mover node. Results were 24-26 MB/sec.

Case 2. The file was set up as a parallel file and was read from two SSA RAID strings on one mover node with HPSS striping between the two RAID strings. Results were 48-51 MB/sec.

Case 3. The file was read from four SSA RAID strings connected to two mover nodes, with two SSA strings on each mover node, and with HPSS striping between the four RAID strings. Results were 97-98 MB/sec. That is, the client node was receiving 97-98 MB/sec, while each mover node was sending about half that amount.

Case 4. Two simultaneous files were transferred as in Case 3, using two client nodes, four mover nodes, and eight SSA strings. Essentially, two Case 3 studies were run, each with its own client node, mover nodes, and SSA strings, but sharing a common HPSS core server and system image. Results were 192-194

MB/sec aggregate across the client nodes. The load on the HPSS core server peaked at 75% CPU utilization, implying that higher bandwidths will either require a faster processor or use of an eight-way SMP node.

The above numbers were obtained with no other processes running in the system. In particular there was no contention for the SSA disk strings. Since the tests were run within the bounds of a single SP, the SP's own TB3 switch served as the network. The network protocol was TCP/IP, which was tuned for the test. The TCP/IP tuning was critical to the success of the effort, with performance varying a factor of 2.5 depending upon the parameter settings. Given data sets large enough to stripe across at least four SSA RAID strings, the SDSC HPSS system should be able to support disk data movement of 10 terabytes per day.

Subsystem resource allocation

Resources were allocated to the HPSS subsystems to avoid internal performance bottlenecks. The HPSS configuration for transaction support is shown in Figure 1, for data movement support is shown in Figure 2, and for backup support is shown in Figure 3. In practice, all of the subsystems reside on the IBM Silver node SP, augmented with a High node, a Wide node, an RS/6000, and the HPGN.

The goals of the transaction support tuning were to improve stability and decrease the amount of system maintenance. The individual steps in the process included:

Migration to HPSS version 3.2. This eliminated known bugs and decreased the amount of event data that was logged. This decreases the size of the meta-data log files, which decreases the amount of processing required to validate the migration of data from disk to tape.

Migration to the Silver node SMP. This increases the CPU power available to support the Encina SFS file system log file and transaction processing system.

Parallelization of the logging process. In the tuned configuration, the meta-data logs stored on the Encina SFS file system are split across three individual servers to balance the load. Effectively, separate log servers are used for each of the three data file sizes. Log clients for gathering event information are run on each node that supports a bit-file mover.

Identification of the amount of disk resources needed to support the meta-data. Approximately 2.5 GB of meta-data disk space is used per million archived files. A total of 72 GB of disk space was allocated, to allow room for growth of the system to 28 million files.

Identification of the amount of disk resources needed to support transaction logs. Approximately 100 MB of transaction log data is written per thousand user requests. The logs contain information for data stores, migrations, back-ups, and accesses for each of a data set's segments. Effectively five days worth of transaction logs are kept on disk, two copies of the current day's transactions, and one copy for each of the prior three days. Every day the transaction logs from the previous four days are synchronized with the meta-data volumes. This ensures a complete and consistent set of meta-data and log files, with no transaction logs having to be kept more than five days. A total of 7.5 GB of disk space is allocated for transaction log data.

Use of mirrored AIX volume groups to eliminate meta-data access bottlenecks. Meta-data directories are segmented across 16 disk drives. Writes to any segment can be done independently without having to lock the segments on the other disks. Since the meta-data directories are also mirrored, reads can be done from either copy of the data.

Consolidation of servers on an SMP node. To minimize system overhead, the core HPSS services (storage servers, nameservice, bitfile server, and meta-data monitor) are run on a single four-way SMP node.

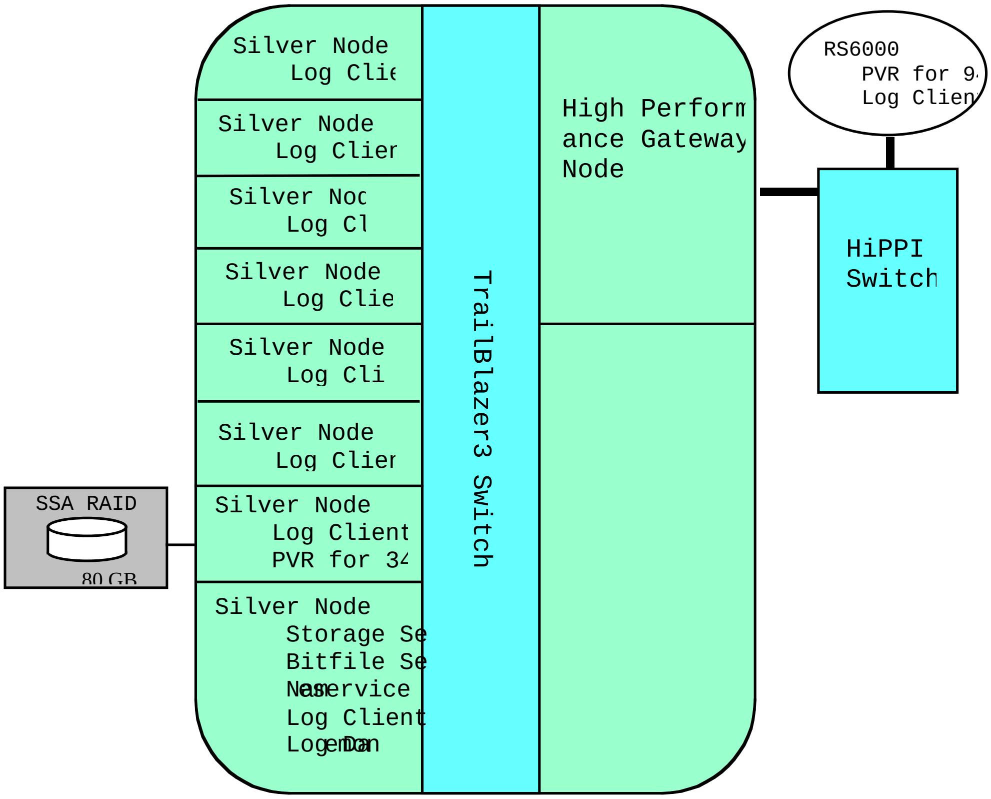

Figure 1. Transaction support subsystem for HPSS.

The goals of the data movement subsystem tuning were to increase sustainable I/O rates and decrease the latency of file access. The individual steps in the process included:

Distribution of bitfile movers across seven Silver nodes, a High node, and an RSz6000. The aggregate I/O rate from the bitfile movers exceeds 300 MB/sec. To sustain this rate requires 12 SSA RAID strings, each having eight disks (six data disks, one parity disk, and one spare disk), for a data capacity of 650 GB.

Increased node memory size to handle simultaneous FTP requests, disk bitfile movers, data migration movers, and tape transfers. Up to 1,500 MB of memory is used to support the multiple servers, with each FTP daemon using a 16 MB buffer. Since each of the Silver nodes has at least 2 GB of memory, this allows each node to handle up to 90 simultaneous FTP requests. The maximum number of simultaneous requests is limited to 512 by the number of per-process parallel threads supported by the current AIX release.

Tuning of the internal SP communication channels. The large memory on each node is used to improve inter-server communication between nodes on the SP. The maximum socket buffer size was expanded to 1 MB, the amount of MBuf memory was set to 64 MB, and 64 kB TCP send/receive buffers were allocated for the SP switch. In addition, the TB3 switch buffer pool size was set to 16 MB for both sends and receives. Thus on the order of 100 MB of memory per node was dedicated to communication buffers.

Separation of communication protocols onto different I/O channels. Previously, SDSC ran TCP/IP and IPI-3 over the same HiPPI channel through a single I/O driver. An IBM High node is used to drive the IPI-3 protocol over a separate HiPPI connection to the MAXSTRAT RAID disk. The HPGN is used to support the TCP/IP protocol. This reduces contention and decreases communication interrupts.

Addition of HPGN support for external HiPPI access. This allows multiple bitfile mover nodes to send data through the SP TB3 switch to the HPGN, and then onto a HiPPI channel connected to a HiPPI switch, increasing the sustainable I/O rate.

Expansion of the cache hierarchy levels by increasing the size of the MAXSTRAT disk cache to 830 GB of useable space. This allowed the physical separation of storage of small data sets from the disk storage for medium sized data sets. Separate I/O channels were provided for each type of data set.

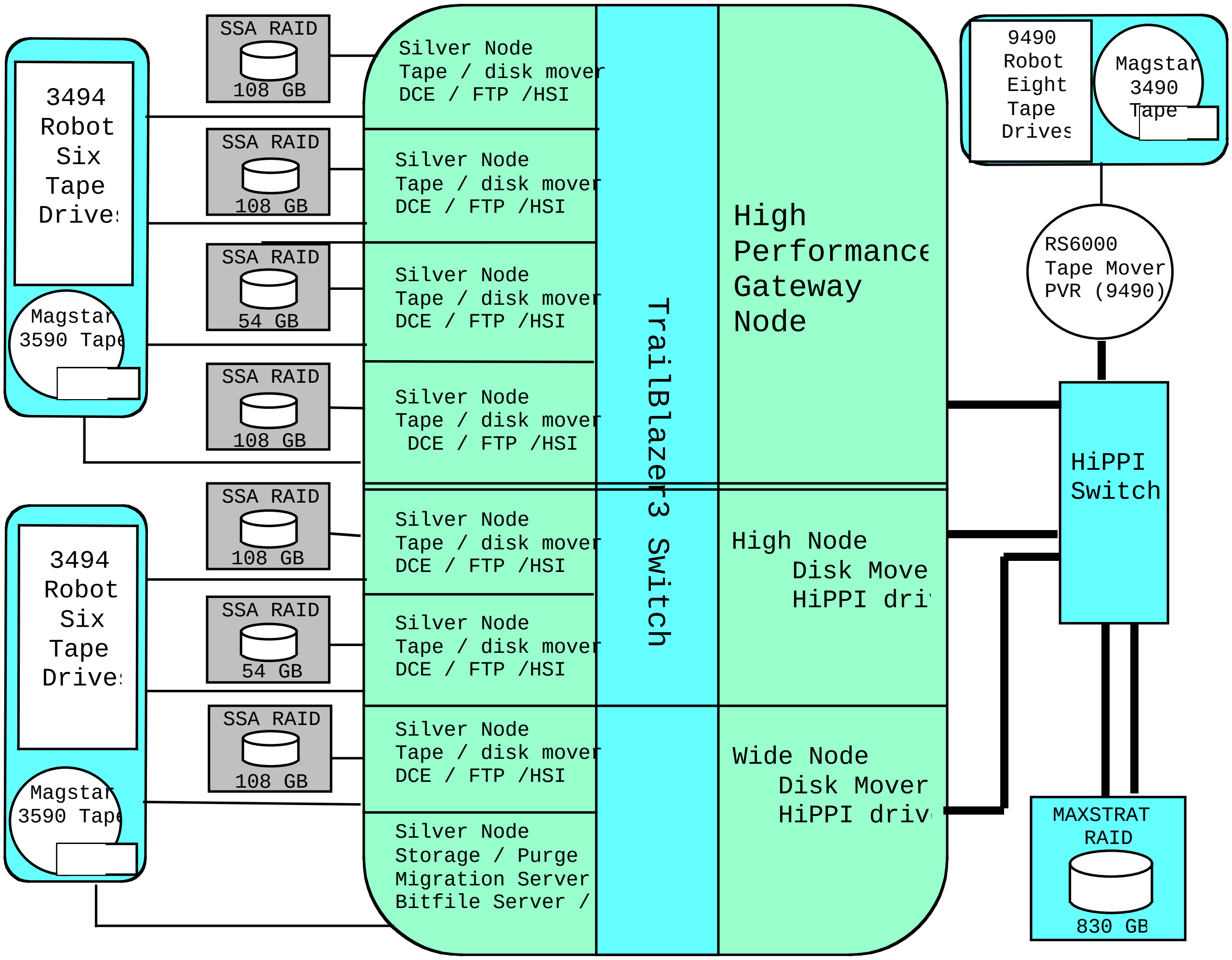

Figure 2. Data movement subsystem for HPSS.

Allocation of separate disk resources to each service class. The small file access was segregated from the large data set movement by caching all small files on SSA RAID disk, while caching all medium-sized one-copy files on MAXSTRAT RAID disk. Using separate bitfile movers for each service class also reduced contention between data movement for large versus small files.

The goals of the backup subsystem tuning were to increase reliability and remove all contention for backup resources. The individual steps in the process included:

Development of a comprehensive meta-data backup script, which validated all steps of the backup process. This makes it easier to verify completion of backups, and monitor allocation of resources to each step of the backup procedure.

Mirroring of all meta-data directories and transaction log files. The total amount of space made available for mirrored data is about 80 GB, of which about 21 GB is in current use.

Dedication of tape resources to the backup environment to eliminate human error. Two tape drives are dedicated within the IBM 3494 tape robots to the backup process to eliminate all manual operations associated with meta-data backup. Two copies are written of each transaction log file using different path names to protect against human error.

Direct connection of the tape drives to the node doing the backups. This allows more efficient transfer mechanisms to be used that have lower overhead. Use of DD on the SP allows overlapped reads from disk and writes to tape through shared memory.

Partitioning of the transaction logs. This allows snapshots to be taken of a partition without having to stop writes to other partitions, increasing the availability of the system. The creation of a complete snapshot still requires that writes be locked out while the snapshot is generated. About 1.5 GB of data are backed up to tape each day.

Weekly complete snapshots of transaction logs. During preventive maintenance each week, a complete back up of the transaction logs is created. This takes about 90 minutes and provides another level of recovery against loss of data.

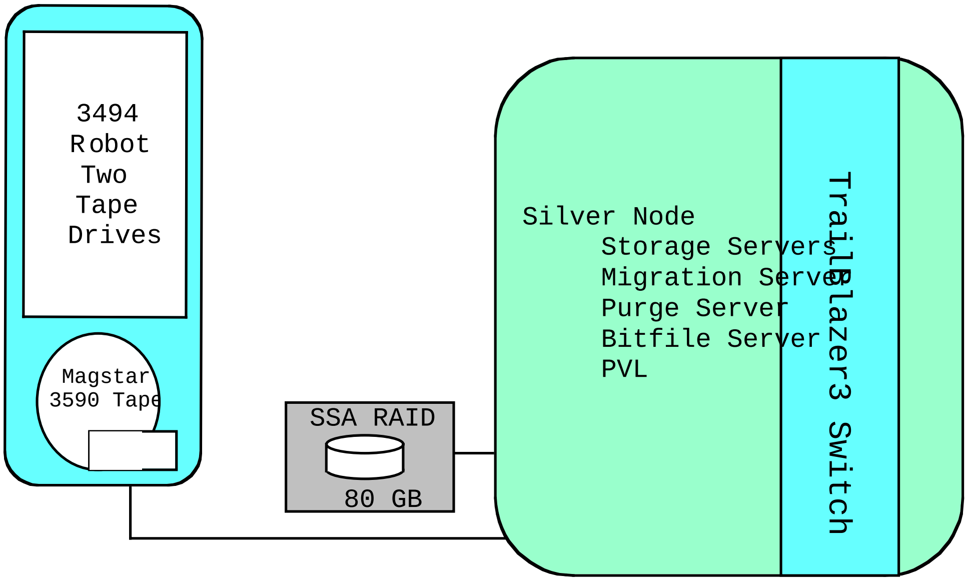

Figure 3. Backup support subsystem for HPSS.

Resource allocation for user data

We analyzed the usage of HPSS by logging statistics about every transaction done to access data. The transactions are maintained in an Oracle database, which allows general queries to be issued to compose usage statistics over arbitrary time periods [5]. During the month of November 1998, the HPSS system ran stably. A daemon pinged the system every 15 minutes and recorded no down times except during the weekly preventive maintenance period. This method of recording availability includes effects due to network reliability, application client reliability, as well as HPSS reliability. During November, a total of 6.7 TB of data was moved between HPSS and external clients.

The dominant use of the HPSS system at SDSC is for backup of files, constituting 30% of the data movement and 64% of the files stored. This provides a relatively uniform background load on the archive. User demand for storing or analyzing very large data sets is the next major use of the system. User demand for storing data from numerical simulations appeared to be relatively constant, and only decreased during holidays.

The user load is separated into three classes of service for small, medium, and large files. Separate disk space is allocated for each service class. Files are automatically assigned a service class based upon their size, as specified in Table 1. Note that separate disk space is allocated to support service classes that request a second copy of the data. The second copy is made when the data sets are written to tape, as all of the disks are RAID. The backup systems at SDSC dominate the usage of the small service class, and are a substantial component of the usage of the medium service class.

The actual user load was calculated by summing across all transactions that took place over the month of November. The total amount of data read or written into HPSS during November was 6.7 TB, giving an average data movement of 223 GB per day. The peak daily data movement was also calculated for each service class, and is listed in Table 2. The largest amount of data moved in a single day was 642 GB. Note that this is a measurement of the external I/O load on HPSS. The total I/O load is typically twice as much, since the data are written to disk and then to tape. During the peak data movement day in November, HPSS supported 1.4 TB of both internal and external data movement. Support for service classes in which data are automatically replicated can increase the total data movement to a total rate three times that of the external data request rate.

Table 1. HPSS service classes at SDSC.

Class of Service	File Size	Disk Space (GB) Single Copy	Disk Space (GB) Two Copies
Small	size < 2 MB	77	30
Medium	2 MB < size < 200 MB	831	222
Large	200 MB < size < 6 GB	193	115

Table 2. HPSS usage during November 1998.

Class of Service	Average # files per day	Average file size (MB)	Average daily data movement (GB)	Peak daily data movement (GB)
Small	11,406	0.2	1.9	7
Medium	4,373	28	123	390
Large	353	280	99	267

The peak loads were caused by higher than normal storage of data into HPSS. This was driven by the migration of data from other computer centers into the SDSC archive. During November, the total amount of data written to the archive was 4.4 TB, and the total amount of data read was 2.3 TB. The peak amount stored in a single day was 380 GB, while the peak amount read was 264 GB. The peak daily data movement occurred on different days for each service class.

The effectiveness of the disk cache can be estimated by calculating the average length of time that a data set could remain on disk, before being purged to make room for new data. This requires knowing the average hit rate of the disk cache. During November, 62% of the file accesses were satisfied from disk, implying that the data were already resident on disk before the request was made. The average daily rate at which the disk cache is filled is then estimated as 38% of the read rate (representing data cached onto disk from tape) added to the average storage rate into the archive. By dividing this estimate of the average daily load into the disk space reserved for each service class, the average residency time of a file in the cache can be estimated. Using the peak daily data movement rate gives the minimum cache life. Since this procedure assumes all reads and writes are done against different data sets and that no rewrite of data is done within the cache, this gives a lower bound on the cache lifetime.

Table 3. HPSS file disk cache lifetimes.

Class of Service	Average File Cache Life (Days)	Minimum File Cache Life (Days)
Small	40	11
Medium	11	2
Large	2	0.7

The resource allocation to the Small service class is meeting its dominant requirement, that small files can be retrieved from disk without having to mount tapes. The target for the Small service class is to hold data for at least a month on disk. The target for the Medium service class is to store data for a week to support research projects more effectively, while the Large service class should hold data for at least two days. When the load on the system increases, the amount of disk assigned to each service class will need to be expanded proportionally.

By comparing the total data movement per day against the average tape speed, we can estimate the number of tape drives that must be provided to support migration of data off of the disk cache. This computation includes the average tape access latency which is dominated by the time needed to spin the tape forward to the file location. We assume that the data access latency is incurred primarily on data retrieval from the archive. Writes are assumed to be done directly to an available tape. This implies that a workload dominated by tape reads will require more tape drives than one dominated by tape writes.

Table 4. HPSS tape drive utilization.

Average I/O Rate (MB/sec)	**Peak I/O Rate (MB/sec)**	**Number of tape drives (average load)**	**Number of tape drives (peak load)**
2.6	7.4	2	8

Thus on average usage days, we keep two tape drives continually busy migrating data off of disk and retrieving data sets. When a tape drive is in use, it spends roughly 15% of the time reading or writing data. The rest of the time is spent positioning the tape to read the file. To achieve a higher effective tape speed, the size of the data sets will have to be increased.

In practice, a substantially larger number of tape drives are needed to support multiple service classes. If all SDSC service classes are accessed simultaneously, a total of 12 tape drives would be needed.

Scalability

Based on the analysis of the actual load on the HPSS system at SDSC, and the corresponding utilization of the HPSS resources, the scaling needed to achieve a sustained data rate of 100 MB/sec can be estimated. This is a factor of 40 times as much data movement as is presently supported in production. The transaction processing subsystem is expected to support external data rates of 100 MB/sec with the present configuration, based upon benchmark tests. Note that this implies an internal data rate of at least 200 MB/sec within HPSS. The benchmark tests indicated the present system is capable of sustaining this rate.

The data movement subsystem will require expansion of the disk data cache size to 40 TB to maintain a comparable file cache life. A smaller disk cache will decrease the hit rate, causing a larger fraction of the data sets to be read from tape and increasing the number of tape mounts that are needed. Even with a 40 TB disk cache, at least 80 tape drives would be needed to support the migration of data. However, if the average size of the data sets increases by a factor of 40, the effective speed of the drives increases by a factor of five. The larger transfer decreases the fraction of the time devoted to tape manipulation and improves the effective transmission rate. The total number of drives that are needed is then only 18 to sustain the average I/O rate. If the disk cache is decreased in size by a factor of two, the number of tape drives that is needed is expected to double. Thus a 20 TB disk cache and 36 tape drives may be sufficient to support a teraflops supercomputer. It is interesting to note that the critical element for the disk cache is storage capacity and associated file cache lifetime, while the critical element for the tape system is average file size and effective bandwidth.

The backup subsystem is expected to support the desired I/O rate, but may require parallelization of the transaction logging across more servers.

Conclusion

The SDSC production HPSS system has been configured and tuned in anticipation of the I/O loads expected from a teraflops-capable computer. The process has illustrated the necessity to adequately characterize and support the internal subsystems of the archival storage system, as well as the need to characterize and support the user I/O requirements. Successfully implementing a scalable archival storage system requires assigning sufficient hardware resources to keep all system components from failing when under heavy load.

The next set of challenges will be how to increase the data rate to the 10 GB/sec range needed to be able to move a petabyte of data per day. This requires much cheaper disks, and much faster tape drives. Cost effective disk storage capacity is becoming available at the rate needed to build data caches that can handle the storage of hundreds of terabytes. High-performance tape storage will be harder to acquire. Increasing bandwidths to the desired level for tape subsystems will either require major advances in tape drive I/O rates, or striping over a much larger number of peripherals.

Acknowledgements

This work was supported by the National Science Foundation Cooperative Agreement ACI-9619020.

References

1. Moore, Reagan W., File servers, networking, and supercomputers, *Adv. Info Storage Systems*, vol. 4, SDSC Report GA-A20574, 1992.
2. Vildibill, Mike, Reagan W. Moore, Henry Newman, "I/O Analysis of the CRAY Y-MP8/864," *Proceedings, Thirty-first Semi-annual Cray User Group Meeting*, Montreux, Switzerland (March 1993).
3. HPSS Web site, http://www.sdsc.edu/HPSS
4. SDSC HPSS production statistics, http://www.sdsc.edu/hpss_sdsc/cgi-bin/hpss_database
5. Schroeder, Wayne, Richard Marciano, Joseph Lopez, Michael K. Gleicher, George Kremenek, Chaitan Baru, Reagan Moore, "Analysis of HPSS Performance Based on Per-file Transfer Logs," 16th IEEE Symposium on Mass Storage Systems, San Diego, 1999.