

Sun StorEdge™ QFS and SAM-FS Software

Technical Overview
March 2004



Table of Contents

Executive Summary	1
Advanced File Systems	3
Why Choosing the Right File System is Important	4
Consolidating Data	4
Better Performance	5
MetaData Separation	5
Variable Block Size	6
Flexible Stripe Options	7
Integrated Volume Management	8
Fast File System Recovery	8
Q-Write	9
Automatic Direct I/O	9
Pre-Allocating Disk Blocks	9
File Sharing in a SAN Environment	10
Managing Space Consumption	12
Enforcing Security	12
Solaris™ Operating System — Multi-Threaded for Performance	12
QFS Software Successes	13
San Diego Supercomputer Center	13
A Better Paradigm for Protecting Data	15
Traditional Backup Software Limitations	15
The Shrinking Backup Window	16
Time Consuming Restores	16
Costly Resources and Scalability Issues	16
SAM-FS Software — Beyond Backup	17
Reducing or Eliminating the Backup Window	17
Faster Restores	19
Scalability and Reducing TCO	20
Beyond Hierarchical Storage Management	20
Releasing Files to Free Disk Space	21
Staging Files Back to Disk	21
Recycling Disk Space	22

File System Differences Between QFS and SAM-FS Software	22
Customer Successes	24
Audi AG	24
Perlegen Sciences, Inc.	24
SAM-QFS Software — An End-to-End Solution for ILM	25
Customer Successes	26
TeraMEDICA, Inc.	26
University of Calgary	26
Developing Better Solutions for the Future	27
Modulization	27
Object-Based Storage	27
Conclusion	29
Scalable, High-Performance File Sharing	29
Resource-Efficient Archiving	29
Putting it All Together	30
References	31
Web Sites	31
Papers	31

Chapter 1

Executive Summary

The information that businesses and organizations need to manage and use today — from email to video to extremely large scientific data sets, to medical images and records — is growing at immense rates and promises to keep growing exponentially. Likewise, environments with smaller amounts of information to manage, face many of the same challenges as businesses generating volumes of data assets, in terms of storage utilization, data protection, and recovery. However, it is the quantity of information that is currently generated by today's organizations, and the increasing regulatory scrutiny governing how data is managed for long-term archiving, secure access, and proper disposal, that are driving the need to find better and more cost-effective solutions to manage multiple terabytes, and increasingly petabytes, of data.

As the importance of information changes over time, so does the need to manage it as a resource, similar to the way companies manage product lifecycles. This trend towards managing the lifecycle of data is referred to as Information Lifecycle Management or ILM. ILM solutions are designed to provide file services in which active data is maintained on faster disk and inactive data is migrated to secondary storage, but all data appears locally, and is equally accessible to multiple users and applications.

In addition to managing data over time, customers demand file access and sharing capabilities that truly scale in terms of performance, capacity, and utility. Demand is growing for multi-gigabyte files, and for file systems of tens or hundreds of terabytes, with a continuing expectation of resiliency and easy management that is available with smaller file systems.

Yet these new demands, coupled with storage subsystem complexity, channel connectivity options, and redundancy needs, strain the limits of traditional file systems and volume managers. Evolving existing file systems to meet requirements for high performance, resiliency,

recoverability, and manageability becomes even more challenging as file systems scale into millions of files and multiple terabytes of data. It is clear that along with advancements in hardware technology, advanced file systems are required to meet current and future demands for using and managing data on a larger scale.

Newer generation, advanced file systems — such as the one that forms the foundation of Sun StorEdge™ QFS software and Sun StorEdge SAM-FS software — are designed to minimize complexity, incorporating the best features of both volume managers and legacy file systems into a file system that leverages the underlying hardware device geometry, spans multiple devices for scalability, and shortens code paths and utilizes thread technology for performance.

Now, with an advanced file system — including built-in volume management — as its base, products such as QFS software can provide a highly scalable, file sharing over SAN environment with exceptional performance for accessing data. And to manage this vast amount of data, SAM-FS software offers cost-effective and efficient archiving, quick restore, and disaster recovery capability. QFS and SAM-FS software are unique in that the two products share a common advanced file system, which enables a tightly integrated data management solution designed to optimize the network and storage hardware (Figure 1-1), enabling devices to perform at virtually full rated speeds and scale as data storage needs grow without sacrificing system performance.

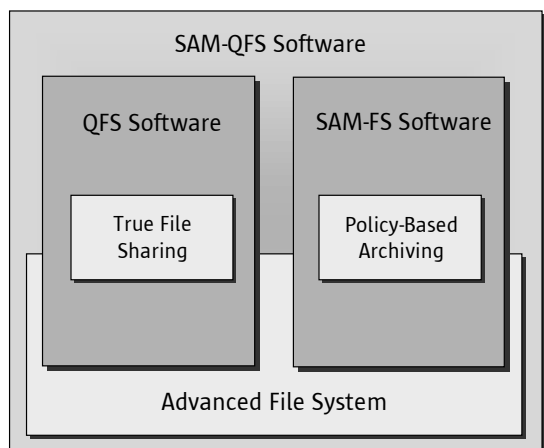


Figure 1-1: QFS and SAM-FS software share a common advanced file system and when combined provide an innovative file sharing over SAN and archiving solution addressing ILM from end-to-end

With the combination of Sun StorEdge QFS and SAM-FS software, known as SAM-QFS when both products are enabled, Sun offers a new approach — an innovative file system, file sharing, and archiving services — that meets the demands of today's information-on-demand environment and enables end-to-end data life cycle management based on the real value of the data to a business.

This paper details the limitations of current file systems and the advanced file system with automated archiving software Sun has developed to address these issues and the demands of data-intensive users and applications. It then provides a technical overview of Sun StorEdge QFS and SAM-FS software and how they provide a complete ILM solution for efficiently sharing data and cost-effectively utilizing resources to help reduce the total cost of ownership (TCO) of data management.

Chapter 2

Advanced File Systems

A file system's purpose is to ease access to online, nearline, and offline storage through a simple, commonly understood interface. When users need to store large amounts of data they typically choose either database management systems or file systems, depending on the desired method of access, performance and manageability requirements, and the volume of data. Database management systems often offer the option of storing data on raw devices, but most administrators prefer to construct database systems over file systems because the data is easier to manage and backup.

File systems provide a degree of abstraction from the hardware, enabling storage resources to be allocated appropriately to various users. However, the evolution of sophisticated computing platforms, coupled with unimagined uses and reduced component costs, have driven the demand for file services beyond the capabilities of traditional UNIX® file systems. Users are creating and accessing increasingly larger files and file systems, anticipating petabyte-sized file systems in the near future. They need the ability to share those files in a high performance environment without having to bear the expense of replicating them for each user. And they expect the same easy to use, if not better, data management and protection that is available with small file systems. Existing architectures easily become strained with these demands because they often inherit an outdated set of requirements and implementation details. For example, most file systems can be built on only one device that is limited in size, forcing administrators to split, grow, shrink, and reallocate storage between numerous small file systems.

Volume management software has been developed to overcome these limitations in file systems, aggregating disks together into single, larger volumes that contain the single-device file system. In addition, software volume managers can provide concatenation, striping, and RAID

functionality, and well as the ability to flexibly allocate storage to multiple file systems. But volume managers abstract the file system from the devices upon which the data resides. The file system must read and write into a pseudo-device without regard for the underlying hardware geometry or its performance characteristics. This additional interaction between two separate software components increases the length of the data path, therefore increasing processing complexity and negatively affecting scalability and performance.

Why Choosing the Right File System is Important

Ultimately, users choose a file system based on its ability to provide easy data access and manageability without reducing performance. With a properly designed advanced file system, files can exceed the size of a single device, quotas control the space a user can consume without having to resize volumes, Access Control Lists (ACLs) prevent unauthorized access to data, the file system can grow dynamically and essentially without bounds, and is capable of backing itself up.

As environments become more data-intensive, managing very large files and file systems, it is important to choose the right file system for the problems that need to be solved. Sun StorEdge QFS software is designed to provide consolidated, high performance, file sharing over SAN, where all data is consolidated into a pool and is accessible to multiple users, and applications run without needing to be modified.

The following sections on consolidating data and performance outline the inherent problems encountered when trying to support large environments with current solutions and detail how these challenges are overcome with various features within the advanced file system in QFS software.

Consolidating Data

The current trend towards increasing productivity and efficiency in computer technology is to physically and logically divide systems into pools of resources — CPU/memory, storage, and networking — that can be utilized at higher rates and scaled individually to quickly meet changing demands. For storage, this means moving all of the data that is on devices directly attached to individual servers on to network-attached devices, such as Network Attached Storage (NAS) or Storage Area Networks (SANs).

SANs enable all storage to be consolidated into a pool of data on its own network, allowing multiple servers to connect to the same physical storage using Fibre Channel (FC) switches. However, simply sharing the same storage hardware does not mean that multiple servers are able to share the same physical disk area or access the same files and data without compromising the integrity of the data. For most environments, NFS has been the most popular choice among UNIX platforms for file sharing, lowering the technical complexity of sharing devices via physical channels. But network-based architectures that implement NFS are limited in two fundamental aspects, particularly when large amounts of data are concerned.

First, they depend on networks that are neither designed nor optimized for channel I/O. Parallel channel-based architectures such as SCSI (Small Computer system Interface) are designed to simply and efficiently transfer data between a single host interface and a few storage devices. Distances between devices are kept short in order to provide high speed transfers with low latency. Recent serial-based channel architectures, such as Fibre Channel, try to exceed the latency and performance capabilities of parallel channels by processing the transport overhead in

hardware and increasing the frequency of transmission to offer greater bandwidth. However, these channel-side networking protocols make many optimizations appropriate for small, short-distance channel-side networks that are impossible to implement when designing protocols and solutions for network-side file sharing and global Wide Area Networks (WANs).

The second limitation in the NFS architecture is that all data must pass through a single controlling host or NAS file server. The capacity and throughput of the entire configuration is therefore constrained by the performance of that single node. Scaling performance requires replicating the data and serving it from multiple servers, increasing management complexity and storage costs. In addition, higher latency introduced by accessing files through the NFS client's operating system, a network, and a remotely-attached host makes NAS architectures unsuitable for latency-sensitive applications such as transaction databases.

Although NFS is still the right choice for many applications, customers who demand file sharing with high transaction rates and throughput, or who cannot build large enough NFS servers for user requirements need another alternative. A QFS file system implemented on a SAN solves the issues encountered with an NFS/NAS architecture by coordinating file information across a TCP/IP network, leaving the Fibre Channel free to perform only I/O directly between the client and the storage system. In this way, the appropriate protocols can be used for both communicating and I/O.

Better Performance

Advanced file systems should be designed to minimize the number of instructions required for each I/O and to intelligently optimize data layout and I/O patterns on devices for optimal throughput. QFS software is designed to solve file system performance bottlenecks by maximizing the performance of the file system in conjunction with the underlying disk technology and hardware. It helps overcome other UNIX file system shortcomings such as the time required to create new file systems, file system growth limitations, lengthy file system checks after an unintended interruption, and limitations on the number of files in the file system.

To provide more optimized throughput and scalability, QFS software limits the movement of the physical head on the disk with new technologies such as metadata separation, variable block size, pre-allocation, and automatic direct I/O. Additionally, integrated volume management with performance options such as disk striping and disk allocation eliminate additional software layers and uses less system and CPU overhead than other UNIX file systems, while improving performance and scalability.

MetaData Separation

UNIX systems use inodes to contain the information (or metadata) required to access files, such as the size, owner, and location of the data block for the file. In traditional file systems, inodes are generated when the file system is created and are placed at the beginning of the disk. When a user requests access to the file the inode must be read in order to locate the physical blocks on the disk and to check for authorization. Read operations are interrupted when requests for other files are issued and the drive must reposition to locate the inode information. And, when writing files, the disk heads must constantly move to update the inode information at the beginning of the disk and then move back again to continue writing data. This continuous repositioning of the disk head severely limits data throughput and reduces overall performance of the system.

The QFS file system solves this problem by separating metadata from file data. Disk drives can now complete a read or write request without repositioning to locate or update the metadata.

Metadata can be accessed and updated simultaneously on a separate device, as shown in Figure 2-1, resulting in more optimized performance for reading and writing data.

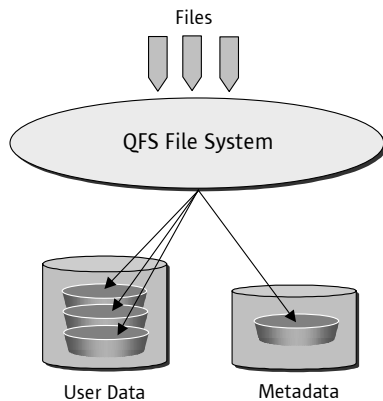


Figure 2-1: Writing metadata and data simultaneously with QFS software

Variable Block Size

The QFS file system optimizes I/O performance between the file system and hardware by allowing the administrator to specify a variable block size. Block devices, such as disks, store data and perform I/O in randomly accessible blocks. Setting a small block size can help enable better performance for millions of small files in a file system, whereas file systems that contain predominantly large files with greater I/O requirements benefit from larger block sizes.

Many customers implement some form of RAID (Redundant Array of Inexpensive Disks) level in order to provide larger file systems, redundancy, or both. The most popular RAID levels are RAID 0, RAID 1, and RAID 5. RAID 0 concatenates disk to create a larger *virtual* disk, while RAID 1 mirrors disk drives within the RAID array to provide redundancy. RAID 5 achieves both of these goals with fewer disks by writing parity information across all disks in the array. If any one of the disks fails, the parity data on the remaining disks can be used to reconstruct the data from the failed disk. In reality, this means that with RAID 5, actual data is never written across all disks in the array because one drive must write the parity data.

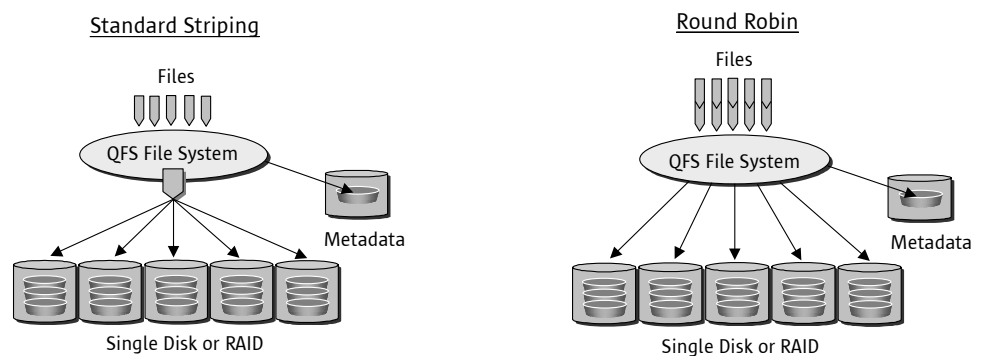
Every time a file is changed, the parity information must change as well, causing what is called a read-modify-write operation to occur in the RAID hardware or software. With read-modify-write, the parity information must be read and modified to reflect changes to the file and then written back to disk. If the block size is not properly aligned with the hardware, read-modify-write operations can generate a large overhead for the system. In addition, when data is written to a RAID 5 array, the controller switches between the available disks (minus one disk for parity) based on an internal setting called a “stripe width”. The stripe width is usually set to a value between 8 and 128 KB (in multiples of 8). For optimal performance, the block size, or Disk Allocation Unit (DAU), should be properly aligned with hardware. The variable block size feature in QFS software allows the I/O to be properly aligned with the hardware and the stripe width.

Sun StorEdge QFS software allows any DAU size from 4 to 65,528 KB. This allows the file system to match the block size or I/O to the optimal DAU size for each disk in the file system. For example, if the RAID5 array is configured with three data disks and one parity disk, and the stripe width is set to 16 KB, the block size should be set to 48 KB ($3 \times 16 \text{ KB} = 48 \text{ KB}$) or a multiple of 48 KB. This helps ensure that the 16 KB blocks on the disk are written contiguously, minimizing performance loss that occurs when the heads must reposition to a different area of the disk.

Flexible Stripe Options

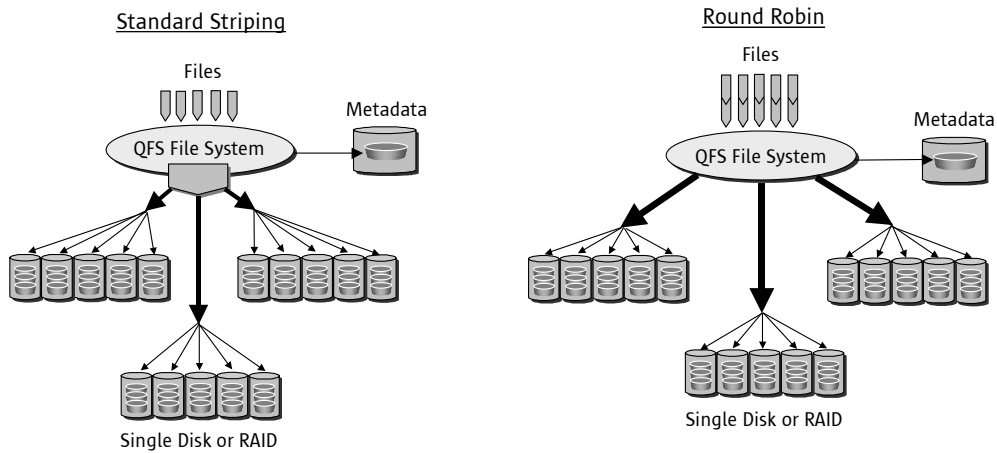
The QFS file system is designed to provide increased bandwidth to sequential I/O applications and multi-user applications with flexible stripe options, allowing multiple I/O streams to simultaneously write a file across multiple disks. The QFS file system can be set up as either a standard stripe (where a file is concurrently striped across all disk components) or as a round robin, where individual files are contained on a particular device. Both striping options are illustrated in Figure 2-2. Standard striping provides maximum throughput per single file. Round robin generally provides greater aggregate throughput for multiple, simultaneous streams of files.

Figure 2-2: Standard and round robin striping options



For environments with requirements for truly extraordinary file system performance, the QFS file system extends the concept of striping to support a “stripe of stripes”, creating a new feature known as *stripe groups*. A stripe group is a grouping of individual disks or RAID partitions to make each group appear as if it were a single logical disk. These logical disks can then be striped together and accessed as either a large stripe or in round robin, as depicted in Figure 2-3. The ability to link all of the disks together enables them to operate simultaneously and allows the file system to achieve predictable throughput speeds that can be greater than a standard UNIX file system. This is particularly important for data-intensive applications, such as ATM transaction processing or video-on-demand applications that have many users and are performance sensitive.

Figure 2-3: Striping with striped groups

Striping with Striped Groups**Integrated Volume Management**

The ability to stripe disks together is not a new concept. The advantage QFS software provides is it makes it easy to create a file system *based* on a disk stripe. With standard volume managers, disks or RAID partitions can be joined as a single file system. The embedded volume manager in QFS software goes further by creating larger files systems from multiple disks, slices, or subsystems. With the QFS file system, all reads and writes can be made simultaneously and directly to multiple disks. This has the advantage of removing the inherent file system bottleneck of traditional UNIX file systems, which force operations through only one meta device, or where additional volume manager software layers limit scalability and performance.

Fast File System Recovery

One aspect of using very large file systems that is often overlooked is the amount of time required to actually create the file system, to restart the file system in the event of a system interruption, or to reconstruct the file system in the event of a system loss. QFS software dramatically decreases the time it takes to grow a file system by dynamically assigning inodes — versus traditional files systems that pre-allocate inodes — so very little time is actually needed to establish the file system structure. In addition, by dynamically allocating inodes a file system can contain a virtually unlimited number of files and can grow in size without having to backup and restore the contents of the file system, which must be performed in a traditional file system in order to rebuild the file system to create additional inodes.

When creating, rebuilding, or recovering a UNIX file system, a traditional or non-journaled file system requires a time-consuming file system check (`fsck`) to verify and repair any inconsistencies in the disk components. For a terabyte-sized file system this can take hours to days to complete. In addition, when recovering from a system outage, even today's journaled file systems cannot guarantee that all transactions are recoverable. A QFS file system does not require an `fsck` or journaling because it keeps the file system consistent with features such as integrated error checking on all critical I/O, serializing critical metadata writes, and recording identification records on metadata, which can be dynamically detected and recovered. This approach helps ensure that all *completed* transactions are recoverable.

The file system should add little overhead to the system and be tunable to either allow highly optimized application to access devices at raw speeds, or to provide caching between applications and the devices for applications with less optimal designs. Advanced file systems should provide tuning parameters at various points in the I/O path and have the ability to automatically adapt to different application I/O behavior patterns. QFS software offers many tunables that enable the file system to adapt to particular environments. Two key examples are `qwrite` and automatic direct I/O.

Q-Write

The Q-Write feature of QFS software enables simultaneous reads and writes to the same file from different threads by disabling the POSIX write-lock mechanism. This benefits databases or thread-aware applications by allowing multiple threads of an application to update a single file in parallel, helping to improve productivity. Q-Write can be switched on by an API (Application Programming Interface) or set for the entire file system when the file system is mounted, and can be extended across multiple sharing hosts. With Q-Write, the application must be capable of controlling the multiple threads writing to the same file or data integrity can be compromised.

Automatic Direct I/O

Sun StorEdge QFS software provides several options to control I/O, either through buffer cache (paged I/O) or directly to disk (`directio`). For small files and small I/O, using paged I/O to read or write can improve performance dramatically. However, for large files or large request sizes, using paged I/O can create a considerable amount of overhead. For file systems with purely large files, the mount option `forcedirectio` can be employed to force every I/O in the file system to go directly to or from disk without using system memory for buffer cache.

If the file system contains a mix of small and large files, using direct I/O as the mount option decreases performance for small files and small I/O. For mixed environments, QFS software provides the ability to set direct I/O for individual files or directories. If direct I/O is set for a directory, the attribute is inherited by every file in that directory tree. This feature is flexible in that it enables users or applications to set or un-set the attribute for individual files or entire directory trees.

In addition, the QFS file system supports switching between paged I/O and direct I/O based on contiguous reads or writes, further improving performance for environments with a mix of random and sequential workloads. The file system can automatically detect the size of the request or file and assign the I/O method that is most efficient. Automatic detection may be configured for reads, writes, or both, allowing the file system to be optimally tuned to support the type of workloads presented by different applications.

Pre-Allocating Disk Blocks

Another performance enhancement in the QFS file system is the ability to pre-allocate disk blocks. When an application requests this feature, the file system pre-allocates a set of contiguous disk blocks for the write operation. Since these blocks are sequentially assigned, the application can write block after block to the disk without having to ask the file system to allocate additional blocks that may or may not be contiguous. This feature can be used in conjunction with `directio`, increasing performance significantly by writing directly and sequentially to disk.

File Sharing in a SAN Environment

Building a cost-effective, high performance data management system starts with consolidating disks and data for better utilization and to eliminate duplicate data. The second step is to employ an advanced file system in order to squeeze as much performance from the hardware as possible. The next challenge is to provide file sharing capabilities so that multiple users and applications can access the same data in a controlled, secure environment. This is where the innovative features of QFS software — beyond the file system that is common to both QFS and SAM-FS software — come into play. Sun StorEdge QFS software is designed to enable multiple readers, as well as multiple writers, to simultaneously access the same files, while providing quotas to set limits on space consumption and access control lists (ACLs) to define access for increased security.

Multiple Readers

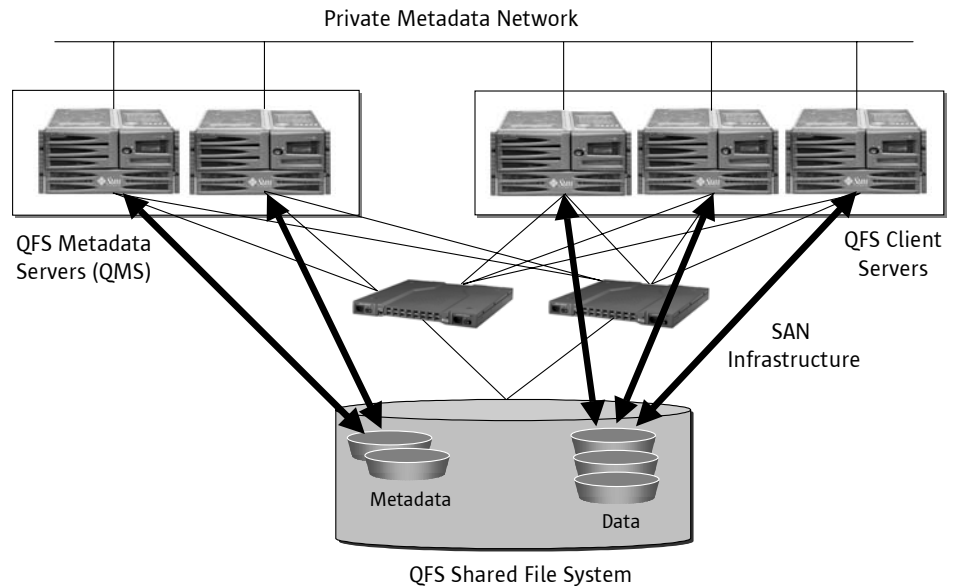
Sun StorEdge QFS software's Multi-Reader functionality enables the same physical file system to be mounted directly on multiple Solaris Operating System (Solaris OS) servers simultaneously. With Multi-Reader, one system can mount the file system as a reader/writer, while the remaining servers mount the same file system as readers. Although there can only be one writer at any given time, if it should fail, any of the reader systems can become the writer by simply remounting the file system as a reader/writer. Applications such as Web services serving static or semi-static pages can benefit by maintaining a single copy of the Web content that is readable by all Web servers and periodically updated by the reader/writer, rather than having to replicate the content on each server.

As in standard UNIX file systems or a file system mounted with NFS, there is no locking mechanism between the QFS writer and reader systems. When a file is created or modified on the writer, the inode information, and perhaps the data for the file, is cached in memory until the file is closed or the Solaris OS issues a `sync` operation. Since the readers access the metadata first when opening a file, the metadata (or inode) may not have been updated, or in the case of a newly created file, may not exist yet. Once the inode has been written to disk, the readers can access the file locally, taking full advantage of the Fibre Channel network in the SAN.

Multiple Readers and Writers

As discussed previously, there are several limitations with using NFS to share large files and file systems. The QFS shared file system addresses these issues by enabling multiple Solaris OS hosts mount the file system simultaneously for both read and write, as illustrated in Figure 2-4. To accomplish this, the QFS shared file system introduces the concept of a QFS Metadata Server (QMS). Each shared file system must have a QMS to provide essential file allocation and file locking services to all hosts mounting the share file system, and to make sure all access is coordinated and secure. Only one QMS server can be active at any one time. The QMS communicates with the other hosts via a TCP socket, taking advantage of optimized communications protocols and leaving the Fibre Channel free to actually transfer data.

Figure 2-4: QFS shared file system



The metadata in a QFS shared file system is placed on metadata devices, which are separate from the data devices, have dedicated connections, and are accessed only by the QMS. This provides an added level of security and performance benefits, as metadata lookup and data I/O operations do not interfere or cause contention.

A primary function of the QMS is to coordinate multiple hosts accessing the same file at the same time. When a Solaris OS application issues a read or write function to a file, the operating system obtains a lease from the QMS for that file. There are three types of leases — read, write, and append. Each lease is given out for a period of time — with a default of 30 seconds — and may be tuned to adapt to specific applications. When a lease expires a grace period is provided, and if not renewed may be given to another host. Multiple read and write leases can be given out for the same file, but only one host may append a file at a time. This allows multiple hosts to write to the same file concurrently. With multiple writes, all I/O is performed directly to disk (direct I/O), bypassing the Solaris OS page cache in order to avoid having to coordinate cache coherency across multiple hosts. In general, leases help QFS software maintain POSIX semantics on any hosts at any given time and the entire file system has locking semantics consistent with NFS.

The QMS is also responsible for providing block allocation services for the file system. When a host wishes to write or append a file it communicates with the QMS and requests blocks to which it can write. The QMS sends the location these blocks via the TCP socket connection and the client host writes directly to them and then updates the QMS so that it may be recorded into the metadata for that file.

To provide a level of redundancy, multiple QMS systems may be defined for a given file system, but only one may be active at a time. If the QMS server should fail, an alternate may be activated by issuing the `samsharefs` command via a command line interface. This failover scenario may also be integrated into a cluster facility by allowing the cluster framework to issue the command after it detects a hardware failure.

Heterogeneous Read/Write with SANergy

Sun StorEdge QFS software supports the Tivoly SANergy environment, allowing the file system to be shared with heterogeneous hosts over a SAN via the SANergy client application. The SANergy Meta-Data Controller (MDC) is installed on the active QMS. Heterogeneous clients that wish to access the shared file system have the SANergy client software installed. The QFS file system can be exported via NFS or CIFS (Common Internet File System), whether the clients are QFS software clients or SANergy clients.

For clients using SANergy, I/O directed to the file system is actually intercepted by the SANergy client software, which communicates the I/O request to the SANergy MDC. The SANergy MDC communicates with the QFS file system via an API to define a list of blocks to read, or free blocks to write. All other SANergy features work as documented. This solution can also be combined with SAM-FS software, providing a high-performance, heterogeneous, shared data SAN with continuous, automated protection for on-line data.

Managing Space Consumption

To efficiently manage and utilize large file systems, the capability to control and limit the amount of space that can be consumed by a user or application becomes imperative. QFS software includes the ability to set quotas to allow fine-grained control of the user environment to limit space consumption and improve resource utilization. Quotas are based on mount point, and can be created for individual users, groups, or administration sets.

Enforcing Security

Now that a community of users is able to access the same data, it is important to have the ability to enforce some security measures for the data. The Access Control Lists (ACLs) feature within QFS software provides a flexible means of preventing unauthorized access to data that is globally accessible. File access can be defined for a file or user, or group of files or users, and is implemented in the same manner as standard Solaris Operating System ACLs.

Solaris Operating System — Multi-Threaded for Performance

The Solaris Operating System achieves superior scalability by using threads as the fundamental unit employed to allocate processors. A *thread* is an independent sequence of program instructions that can be executed by a processor. Many threads can be executed in parallel by separate processors in multiprocessor systems. The Solaris OS responds to interrupts, performs driver and background activities, and handles application requests using threads, and the number of possible threads is limited only by the amount of available memory.

By using the same mechanism for scheduling both kernel activities and user processes, improvements in threads improve all aspects of performance for applications built to utilize multiple processors, including the advanced file system in QFS and SAM-FS software. In an environment where both the operating system and file system support multi-threading, multiple threads can access a file at the same time, a capability that is extremely beneficial for applications such as transaction processing databases, video streaming applications, and high performance computing (HPC).

QFS Software Successes

Sun StorEdge QFS software is a mature product, currently is its fourth major revision, and has been successfully implemented in high performance technical computing (HPTC) environments, as well as government, military, aerospace, education, research, and media. One excellent example of an organization that solved its growing data needs with QFS software is the San Diego Supercomputer Center.

San Diego Supercomputer Center

San Diego Supercomputer Center (SDSC) is one of the leading research and supercomputing facilities in the United States. They acquire, deploy, and manage the highest performance computer systems, storage systems, and network environments to support academic researchers and the National Science Foundations. This industry used to be focused primarily on compute-intensive applications, but is becoming much more data intensive. In addition to processing large amounts of data, SDSC needed the ability to manage, move, and analyze rapidly growing scientific data sets in order to support the growing reliance on data intensive applications and reduce their operational TCO.

SDSC turned to Sun™ Services for help in designing and implementing a SAN with over 250 terabytes of heterogeneously shared storage and 365 x 24 x 7 availability that is expected to operate at efficiency levels over 90 percent and sustain data movement at 2.2 gigabytes per second. They chose the Sun SAN environment because it gives them the ability to manage storage as a network versus a direct attached storage environment, resulting in enhanced productivity and lower costs. For SDSC, the capability of a storage environment that can be geographically distributed is critical in terms of managing storage in a grid environment and operating the IT infrastructure as a network, providing support for their databases, data management, data mining, and many users.

Chapter 3

A Better Paradigm for Protecting Data

With the advanced file system and file sharing capabilities of Sun StorEdge QFS software, organizations can now efficiently utilize storage resources and provide file access to multiple users, increasing performance and productivity, and decreasing TCO. The next step is protecting that data from loss or system failures and managing its lifecycle so it consumes a minimal amount of resources. The limitations of full and incremental backup strategies to deal with the growth of enterprise data, coupled with limited IT resources and compliance regulations, have forced companies to seek alternatives that lower costs, mitigate risks and reduce complexity. This requires a better paradigm for managing and protecting data, based on its importance to the company. Sun StorEdge SAM-FS software is designed to provide this capability, combining the advanced file system discussed above with the storage and archive management (SAM) utility.

Traditional Backup Software Limitations

Historically, backup has been used to protect data from data loss. Data is copied using different backup techniques at specific times to secondary media, usually tape. The main reason for backup is to have the ability to restore data in case of a disaster, user error, hardware failure, or to recover old versions of a file. Traditional backup presents three inherent problems when applied to large file systems: the need for a backup window, the time it takes to restore the data, and the amount of resources required (storage, network, tape, and administration).

The Shrinking Backup Window

Due to the global economy, many companies today can no longer afford to take data offline for a long period of time to backup data. This means that the time window for backup is constantly shrinking, while the amount of data to be backed up is growing rapidly. Corporations cannot wait many hours or days to be back online and to continue business as usual in case of a disaster or hardware failure. This limits the time window for backup and recovery and creates an even bigger problem for current backup and recovery strategies.

A common backup strategy with traditional backup software is to back up every file in the enterprise at least once a week, whether they have changed or not, which can be very time consuming. Full backup strategies can include keeping many backup sets, that is, versions for backups collected over a specific time period. Because of the time required to complete a full backup, it is usually performed during the weekends. This implies that files created or changed during the week are not protected. To protect and restore files between full backups, organizations can deploy incremental backup strategies. With incremental backup, only new or changed files are copied.

Time Consuming Restores

The reason backups are performed is to be able to recover files when needed. Due to the ever-increasing amount of data, disaster recovery cannot be achieved easily with current full and incremental backup strategies. First, the initial full backup must be restored. Then, every incremental backup must be restored sequentially in order to restore the file system to its accurate state. This can be very time consuming and complex and can take many hours or days. Users do not have access to their data until all files are restored. And valuable time and money is lost during the time consuming recovery process, making this strategy financially unacceptable for many companies.

Because backup of large amounts of data is very time consuming, many companies can only afford to make one copy of the data to tape, making the data vulnerable. To solve this issue, companies make multiple copies of backup tapes, which are usually moved off-site into a vault. In case of a catastrophe and loss of the original backup media, the tape copies can be used to restore data once they have been retrieved from the vault. Copying tapes is very time consuming and adds tremendous media cost and administration overhead.

Costly Resources and Scalability Issues

With the quantity of data increasing steadily, full backups and restores can no longer be accomplished in the time allowed without adding an unreasonable amount of hardware. For example, to complete a full backup or restore of a 10 terabyte file system in 24 hours, the backup application must be able to stream data at a rate of 113 megabyte per second for 24 hours, to or from local tape drives. With current tape technology, streaming data at this rate requires about 15 tape devices with an average performance of 10 megabytes per second, assuming the application can fully stream data to all 15 devices and including time for loading, unloading, and positioning. If the backup window is subsequently reduced to 6 hours, it would take four times as many tape drives, or 60, to complete a full backup. This is not feasible due to the limited number of tape drives in a tape library, limited number of I/O channels in a server, limited scalability of the backup application, and the inability to keep all tape drives streaming at one time.

The example above assumes that all tape devices are local to the server. Enterprise-wide backup must utilize the LAN to backup data from many servers or workstations, which creates

additional problems. For example, the same 10 terabytes of data would require a minimum of ten 100 megabit Ethernet networks to perform the transfer in 24 hours, but the reality is that no one achieves 10 megabytes per second on 100 megabit Ethernet, nor can sites afford 24 hour backup windows, making this type of solution obsolete.

And finally, current media strategies require much more media space than actual data space. Each full backup creates a new copy of a file. If a file exists at the beginning of the year, has not changed, and the customer keeps one full backup every week, this file will have 52 copies on 52 different pieces of media at the end of the year. This wastes an enormous amount of media and adds significantly to media cost and administration overhead.

SAM-FS Software — Beyond Backup

Sun StorEdge SAM-FS software can manage many petabytes to exabytes of data, easily keeping up with data growth by continuously, automatically, and transparently making copies of new or changed files to tape or other secondary media. There is no need to stop access to the file system, or files, while SAM-FS software copies files, and no backup window is required. The software can make multiple copies of the files simultaneously to multiple locations, including remote sites. Unlike traditional backup solutions, which copy all files again and again, whether they have changed or not, SAM-FS software copies only changed or new files. This dramatically decreases the amount of data copied and uses less media much more efficiently.

SAM-FS software can be used with existing file systems (today the archiving function can only be used with a SAM-FS file systems or with QFS software). However, since QFS and SAM-FS software provide the same advanced file system, any data that resides on SAM-FS file systems will enjoy most of the performance benefits discussed in the previous chapter. SAM-FS software offers a complete solution to both the advanced file system in SAM-FS software and the shared file system over SAN and performance of QFS software by also providing data protection and recovery.

Reducing or Eliminating the Backup Window

Sun StorEdge SAM-FS software protects files with continuous archiving services that automatically create one copy of each file to nearline media on local or remote disk, tape, and magneto optical libraries, as illustrated in Figure 3-1. The software can automatically make up to four copies of each file to different types of media. In addition, the software features the capability to perform a disk-to-disk copy and archive, allowing companies to develop more resilient disaster recovery scenarios by copying to remote sites.

After a file has been successfully copied to media, all information concerning the file copy is entered into the inode of the file, including the tape volume serial number of the media used, block location of the file on media, and the media type used. With current backup technologies, files are copied again and again at each full backup cycle even if they have not changed. With SAM-FS software, a file never needs to be copied again unless it changes, because it is protected by the copies made to different media.

Should a file change after it is copied, the advanced file system in SAM-FS software will automatically make new copies to new media or new locations, protecting the new versions of this file again, without having to wait for a full or incremental backup as with traditional backup software. After the new copy is made successfully, the inode of the file is updated to reflect the new location and media of the new copy. Automatically copying files only after they have been created or modified effectively eliminates the need for a backup window and can significantly decrease administrator overhead.

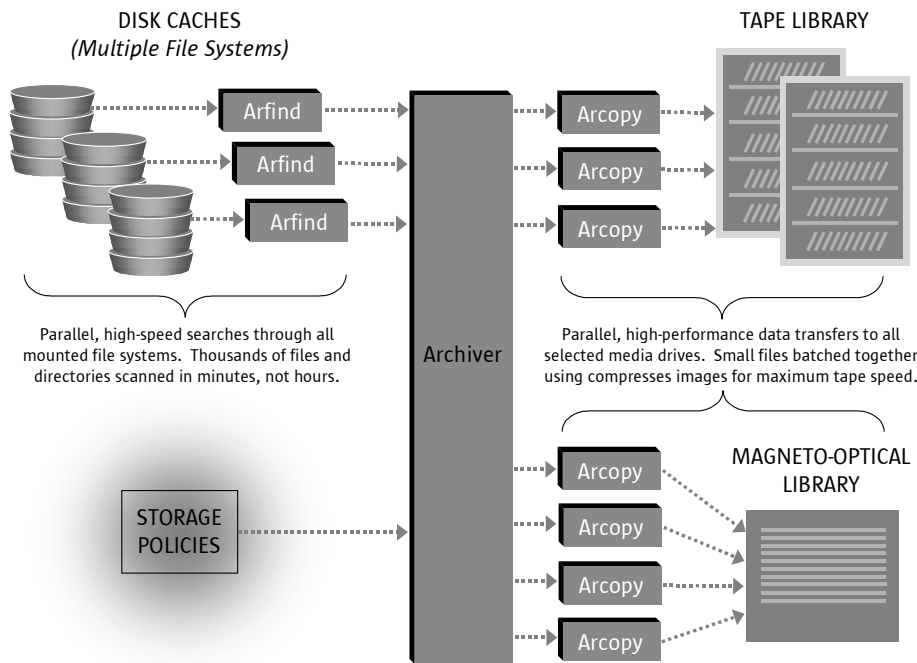


Figure 3-1: SAM-FS software's continuous archiving functionality enables administrators to set automatic archiving policies that determine when, where, and how data is stored

Automated Management

Sun StorEdge SAM-FS software provides sophisticated capabilities for defining automated archiving policies, enabling administrators to group related files into common archive sets based on attributes such as file size, ownership, or file extension, providing fine-grained control over how and where the data is archived. It also allows them to specify files that should never be archived, such as temporary files, thus saving time and resources.

SAM-FS software's automated policies can use a combination of criteria to identify files that are eligible to be migrated to tape or other less-expensive storage. For example, an administrator could create a migration parameter that includes the following criteria: data, time, activity, class, file system fill level, minimum time, minimum size, and priority.

In addition, SAM-FS software supports truncation policies — through the release feature — that use independent criteria to determine which data blocks can be safely removed from disk in order to create more free space for new files. Truncation policies may be applied by file system, minimum time, and minimum size. Other products reside on top of the standard file system, and release files by truncating them to zero length, leaving the name space but no user data. These are also called stub files. When a user requests a file, the software sees that the file is zero length, checks the external database, and stages the file. Some products also allow the stubs to be greater than zero in length, which is equivalent to a partial release in SAM-FS software. With a partial release, a number of bytes stay on the disk and the remainder are released. SAM-FS software supports variable sizes for partial release, settable file-by-file, from 8 KB to 2 GB. SAM-FS software can also trigger staging before the application hits the end of the on-disk stub data. SAM-FS software can also set release policies by file system, time since access, and time since staged. All of these parameters can be biased against one another to calculate a floating point eligibility ranking for migration.

The automated policies in Sun StorEdge SAM-FS software are extremely useful in helping customers comply with the increased regulatory scrutiny governing how data is managed for long

term archiving, secured access, and proper disposal. Many industries are required to keep certain types of data stored and accessible for a period of months or years. SAM-FS software helps organizations comply with these requirements in a very efficient and cost-effective manner.

Using Segmentation for Larger Files

Utilizing the segmentation feature of SAM-FS software for large files can further decrease the amount of data to be copied. Because backup files can be extremely large, it can take a long time to read or write a single large file to a single tape drive. SAM-FS software segmentation can divide large files into smaller file segments transparently. Individual segments or parts of a file can now be managed and protected independently by only copying the changed segments of a large file instead of the entire file. This allows the software to stripe multiple segments of a single file across multiple tapes or tape drives in parallel, thus drastically increasing performance of copies and restores for extremely large, multi-terabyte files.

In addition, SAM-FS software is developed utilizing the multi-threading technology in the Solaris Operating System. Each copy or restore is fully independent, allowing each thread to utilize resources efficiently. This technology enables the software to stream data to any tape technology at its rated device speeds.

Faster Restores

The inode information of all files in the file system can be saved using the `samfsdump` utility, which stores only the inode information for files, not the data portion, into a single file image representing a snapshot of the file system. Unlike backup, there is no time window required to run the utility, so it can be run at any time, even while data is accessed in the file system. Because the file contains only the inode information, it is very small, making it easy and cost-effective to create a new copy every night or even more frequently. The inode information stored in the `samfsdump` file, together with the media used for the copies represents the full backup of the file system, and can easily be used to reconstruct the directory structure and file names of the file system in case of a failure or disaster.

Restoring a file system protected by SAM-FS software is extremely fast when compared to traditional backup software because only the metadata needs to be restored before the file system can be mounted and used. This can take minutes rather than the hours or days it would take for a full restoration from tape. Once the metadata is restored and the file systems mounted, all references to the data are satisfied from the SAM-FS archive, so that files are migrated back to online storage as they are accessed, providing transparent access to the actual file data from near-line storage. And, SAM-FS software's **read behind** feature enables users to begin reading the file even before it is fully restored, significantly benefiting users who need to access large files.

Data Integrity

With the ability to restore metadata, the system can be returned to operation in almost no time. But even that capability can be hindered if the media on which the data is stored is defective in some way. To help maintain data integrity, SAM-FS software includes a system error facility (SEF) reporting mechanism. The SEF allows system administrators to capture and compile a report from the log sense pages of tape devices. This report can help notify administrators if tapes have aged, should be replaced, rotated, etc.

In addition, SAM-FS software supports media changer and tape drive SCSI-3 TapeAlert. TapeAlert refers to the capability of a tape device to provide detailed diagnostic information using

the ANSI standard interface. TapeAlert detects errors on TapeAlert-supported drives by constantly checking the drive for potential failures, and reporting these problems before they occur.

TapeAlert issues log sense data at these points:

- At the beginning of a write or read application
- After an un-recovered error
- At the end of each tape when an archive spans more than one cartridge
- At the end of a write or read SAM-FS software operation
- And, it polls idle devices every 60 seconds

When an error is detected, a dialog is displayed explaining the error, and offering instructions on how the problem can be resolved. TapeAlert records are reported by the device through the log sense page. Each TapeAlert record contains the following fields:

- Log parameter code
- Flag name (message)
- Flag type (severity)
- Recommended application client message
- Probable cause

Scalability and Reducing TCO

Because SAM-FS software needs to copy only new and changed files, only the tape space for these files is necessary. The software can easily keep up with the explosion of data in today's economy by never having to perform a full backup of *all* old and new data. Not only does this save costs by eliminating the need to add media for old and stale data, it also saves valuable time and money by limiting administrator overhead and reducing the number of tape devices that would be required to accomplish a full backup within a given window of time.

Tape Drive Sharing

To further improve resource utilization, SAM-FS software includes the ability to share tape drives between multiple SAM-FS software processes on different servers. In SAN environments it is possible to allow tape drives with Fibre Channel transports to be configured on multiple hosts simultaneously, enabling multiple SAM-FS software servers to share tape drive resources. This feature is available only when the robot that contains the tape drives is managed by a network-attached controller, such as StorageTek ACSLS or Sony Petaserver.

Beyond Hierarchical Storage Management

Unlike traditional Hierarchical Storage Management (HSM), SAM-FS software archives continually instead of waiting for a "high water-mark" to be reached before archiving begins. SAM-FS software continuously archives only those files that have changed, saving resources and generating little-to-no impact on overall system performance. But SAM-FS software can help organization save even more by managing and storing the data according to its importance to the business, creating a transparent and virtually unlimited file system.

Most storage management software products copy a file to removable media once the high water mark (HWM) is reached on the disk cache. This approach may result in reliability and performance problems. Reliability may be compromised if there is a disk failure before the HWM is reached and the file has not been copied to removable media. Performance can also be affected if the HWM is reached during a time of high user activity, because the large effort of copying can

slow system performance. SAM-FS software can automatically copy files based on time rules (minutes, hours, and days). SAM-FS software daemons are continuously searching for files that can be copied to removable media. After the file is copied, the original file remains on disk cache until the pre-specified high water mark is reached, at which point it is released from cache.

SAM-FS software provides more than HSM functions, with more features and tunability than other HSM software, including:

- Direct access to removable media — SAM-FS software can read files directly from removable media without staging the file into disk cache. Direct access allows efficient access to large, near-line databases.
- Tunable design — SAM-FS software can be tuned to provide optimized operation for each customer environment.
- Fast recovery — When a disk fails, the SAM-FS software system is capable of recovering all archived data, usually in a matter of minutes. To recover, metadata must be loaded onto a new disk using the `samfsrestore` utility. The archived data on removable media is complete and does not have to be staged into the new disk. If a file is not archived, the data resident on the magnetic disk is lost, and the SAM-FS file system marks the file as damaged. A damaged file notifies end users that their file is unusable and should be recreated. A damaged file can only be removed. Detection of damage at the time the incident occurs significantly increases the chances that the lost data can be recovered, reconstructed, or regenerated by other means.
- Integration with QFS software
- Integrated volume management

Along with continuous archiving, SAM-FS software provides three other components responsible for actually removing and restoring data from disk cache so that more expensive disk can be better utilized. These components are the releaser, stager, and recycler.

Releasing Files to Free Disk Space

Releasing is the process of freeing primary disk storage that is used by files. There are two threshold values — expressed as a percentage of total disk space — used to manage online disk usage. They are the high water mark and the low water mark. When online disk space exceeds the high water mark setting, the system automatically begins releasing the disk space of archived files until the low water mark is reached. Files selected for release depend on the file's size, age, and other configurable settings. It is also possible to set certain files to *release never* so that they always remain on disk.

Staging Files Back to Disk

When an offline file is accessed, the stager automatically stages the file back to disk cache. For a sequential read of an offline file, the read operation tracks directly behind the staging operation, allowing the file to be immediately available to an application before the entire file is restored.

The stager processes request errors automatically, attempting to find the next available copy of a file if an error, such as media error, unavailability of media, or unavailability of an automated library, is returned.

The *stage never* option can be used when SAM-FS software is employed as a backup cache to enhance a traditional backup environment, allowing requests for backup data sets to be delivered directly to the backup application upon request and never be staged back online. If a customer is using SAM-FS software as a backup cache, the files can be written to the cache very fast, so it is

possible to fill the cache over the high water-mark. When using *stage never* it should be possible to stage files that need to be restored directly from tape to the backup applications that is requesting the files. Two additional features that differentiate SAM-FS software from other HSM products are:

- **Pre-staging** — Large files can be automatically pre-staged to disk cache. Pre-staging can be very beneficial for video and audio broadcast applications, where specified files must be transmitted from high-performance disk arrays. Pre-staging can also batch multiple file requests on a given media volume to provide better media and robot usage.
- **Associative staging** — Associative Staging is an attribute that can be assigned to a file or a directory. When one file is accessed (staged), all other files with the attribute enabled are also staged while the user is working with the initial file (as a transparent background process). When the user requests any of these additional files, they are immediately available. Associative staging is particularly useful for situations where a number of related files comprise a project. Associative staging reduces manual intervention by the user, provides faster access to the related files, and reduces robot motion and media shuffling.

Recycling Disk Space

As users modify files, archive copies associated with the old versions can be purged from the system. The recycler identifies the volumes with the largest proportions of expired archive copies and moves non-expired copies to different volumes. When only expired copies exist on a given volume, a site-defined action can be taken. For example, the volume can be relabeled for immediate reuse or exported to off-site storage, thus keeping an historical record of file changes.

In addition, the partial feature allows a specified portion of a file to be retained on disk cache after the file is released. This enables applications like the filemanager to access the necessary header information without staging the file back to disk cache.

File System Differences Between QFS and SAM-FS Software

There are some differences between a QFS and SAM-FS file system. One main difference is performance — the QFS and SAM-QFS (combined QFS and SAM-FS software) file systems provide the ability to attain raw, device-rated speeds with the administrative convenience of a file system. Other differences are:

- **Disk allocation unit (DAU)**. The SAM-FS file system uses several sizes of DAU. The small DAU is 4 KB. The large DAU is configurable into 16-, 32-, or 64-bit units. The available DAU size pairs are 4/16 (default), 4/32, and 4/64.

The QFS and SAM-QFS file systems support a fully configurable DAU from 16 KB to 65,528 KB.

- **Metadata storage**. The QFS and SAM-FS file systems maintain file metadata information in a separate file. This enables the number of files, and the file system as a whole, to be enlarged dynamically. However, the SAM-FS metadata file resides on the same device as the file data. The SAM-FS file system can span multiple partitions by using disk striping, thereby improving access for large files.

The QFS and SAM-QFS file system separates the file system metadata from the file data by storing it on separate devices. The file system allows user to define one or more separate metadata devices in order to reduce device head movement and rotational latency, improve RAID cache utilization, or mirror metadata without mirroring data.

- **Support for multiple stripe groups.** To support multiple RAID devices in a single file system, striped groups can be defined in QFS software. Disk block allocation can be optimized for a stripe group, reducing the overhead for updating the on-disk allocation map. The SAM-FS file system does not support this feature.
- **Device interoperability.** When possible, QFS and SAM-FS software uses standard Solaris OS disk and tape device drivers. For devices not directly supported under Solaris OS, such as certain automated library and optical disk devices, a special device driver called `samst` is included in the SAM-FS software package.
- **Shared file system support.** A shared file system can be implemented in either a QFS or SAM-QFS software environment, but not in a SAM-FS software environment. Shared file systems do not support blocked special files, character special files, or FIFO (named pipe) special files.

Table 3-1 provides a summary of the key differences and similarities between the QFS and SAM-FS file systems.

Table 3-1: Summary of the Common File System Features and Key Differences

File System Feature	QFS Software	SAM-FS Software
Vnode Interface	Yes	Yes
Enhanced Volume Management	Yes	Yes
Support for Paged and Direct I/O	Yes	Yes
Preallocation of File Space	Yes	Yes
Unlimited Capacity	Yes	Yes
Fast File System Recovery	Yes	Yes
Variable DAU	Yes	Yes, but limited
Metadata in Separate File	Yes, and on separate devices as data	Yes, but on same devices as data
Multiple Stripe Groups	Yes	No
Shared File System Support	Yes	No

Customer Successes

Sun StorEdge SAM-FS software is a mature, proven technology that has been implemented by many companies to provide fast, scalable data management solutions at a reasonable price.

Audi AG

Audi AG in Ingolstadt, Germany manufactures high-tech automobiles, generating approximately 6 terabytes of new data each day. Audi's existing HSM system was being pushed to the limits of its hardware and software. They needed to migrate to a more scalable, faster solution to manage the production data generated daily and help ensure compliance with German regulations for record retention. Data also needed to be copied to a remote site for business continuance in the event of a disaster.

Audi implemented SAM-FS software and can now handle unlimited capacities and set policies to comply with regulations. And the environment requires only one person-day per week to manage. Because SAM-FS software needs fewer drives for data storage than conventional solutions and always stores data at maximum drive speed, the data storage solution at Audi is not only fast, but also operates with uniform capacity 24 hours a day. And, SAM-FS software allows them to store multiple copies at different locations, giving users access to data at all times.

Perlegen Sciences, Inc.

Perlegen Sciences, Inc. uses microarray technology to scan entire human genomes. Scanning whole genomes generates a terrific amount of data, in the realm of 120 terabytes for 50 genomes. Although the genome data does not need to be stored on high performance disk after the initial weeks of study, all 120 terabytes must be preserved and easily accessible from lower-cost media because it would be too costly to regenerate the data and new algorithms may be developed to allow more information to be gained from the data.

Perlegen chose Sun because of its leadership in high performance, 64-bit technology and cost-effective data management software. The company built a comprehensive file management system with SAM-FS software. Each file is logically organized according to its importance and frequency of use. Frequently accessed files are stored on the fast, hard disk cache, while older, rarely used files are migrated to the tape storage libraries for permanent storage.

SAM-FS software allows Perlegen's engineers to access data from and write data to tape without administrative intervention — keeping costs down by utilizing lower-cost media and minimizing administrative man-hours. It also enables Perlegen to set policies that automate the system to make three backup file copies across different media and geographic locations for added protection.

Chapter 4

SAM-QFS Software — An End-to-End Solution for ILM

Because Sun StorEdge QFS and SAM-FS software share the same advanced file system, they can be easily combined to provide a tightly integrated, cost-effective, high-performance, end-to-end solution for ILM. This combination is known as SAM-QFS. With SAM-QFS the file system manages capacity on its own, reducing the need for risky and labor-intensive reallocation of storage into and out of existing file systems as their sizes change. The disk cache is sized for the expected working set of files, rather than the required total capacity. Business rules describe the relative importance of user data to the system, and the system manages the data appropriately, moving it to offline storage when its importance diminishes. And all of this happens on a high performance file system that can operate and be shared at device speeds.

The combined benefits of SAM-QFS serve to differentiate Sun from other vendors in:

- Performance — With tightly integrated volume management and metadata separation.
- Extreme scalability — Performance or storage capacity can be increased simply by adding components as needed. Additional storage can be incorporated into the SAN, introduced into the file system, and contribute immediately to the performance and capacity of the environment. Additional servers can each have read and write access to the same data, simply by mounting it, and can therefore contribute immediately to the aggregate throughput and performance of the system.
- Innovative implementation of functionality — Such as file sharing, transparency, and virtual file systems.

- Flexibility — With the ability to tune for buffered reading and direct I/O to provide optimal performance for different application types.
- Robustness — With a proven, widely implemented product.
- Storage agnostic — with support for most major manufacturers of tape and optical libraries, as well as massive disk-based archives.
- Tightly coupled end-to-end data availability and protection — Reducing or eliminating back-up windows and restoring business much faster than traditional backup.

Customer Successes

When companies need high performance file sharing and cost-effective archiving capabilities, they turn to Sun. Here are a few customers who have implemented SAM-QFS software to provide a complete solution for ILM, with the benefits of increasing performance and decreasing TCO.

TeraMEDICA, Inc.

TeraMEDICA is a company dedicated to introducing new image management solutions that can support better patient care, improve productivity, and reduce costs. With over 600 users who need simultaneous read/write access to nearly six billion images at any given time, TeraMEDICA required a solution that could seamlessly grow to over a petabyte of storage.

TeraMEDICA chose Sun because of its complete storage solutions, differentiated by advanced, shared file system technology and archival system software. The company's Enterprise Image Management system, with Sun StorEdge SAM-QFS software, is designed to handle multiple medical image libraries that capture and store 800 megabytes to 1.8 terabytes of data annually, with the potential of scaling to several petabytes. SAM-FS software helps provide fast access and easy archival management of these huge medical images, regardless of the type of media they are stored on.

University of Calgary

The University of Calgary is a research and teaching university of growing national and international stature. Its Faculty of Medicine is making major contributions to the world's medical community, particularly in the field of bioinformatics. In the system that recently entered production for the university, a Sun Fire™ 6800 server runs the department's research applications, accessing data stored on five terabytes of Sun StorEdge T3 arrays, as well as a 20 terabyte Sun StorEdge DLT tape library.

Sun StorEdge SAM-QFS software allows data stored on disk and tape to be treated in the same manner, so that tape becomes essentially an extension of disk, anticipating when tape-resident data will be required and loading it onto disk for fast access as soon as it is needed. Intelligent storage management keeps related files together, further minimizing delays. Many automatic features optimize the disk-tape trade-off to provide fast access to data without wasting storage, and allowing online data to grow as large as one billion terabytes before a file system reconfiguration is required. And, all of this data management is transparent to the users, who never know or care whether the data resides on disk versus tape.

Developing Better Solutions for the Future

Sun is constantly developing better solutions for the future, listening to customer needs and innovating on industry standards that can help enterprises stay competitive in the emerging global economy. The future of QFS and SAM-FS software beyond traditional server-centric approaches includes modulization and distribution of the file and archiving services across Sun's integrated stack, along with the evolving standards for object-based storage and devices.

Modulization

As hardware technologies become more modular — creating pools of resources that are more easily managed and efficiently utilized — it is only logical that data management software will follow the same trend. A look into the future for SAM-QFS software includes a move towards modulizing the file system, enabling greater scalability, resource utilization, and leverage, by distributing file system and archiving services across IT resources to best satisfy quality of service requirements. For example, by moving allocation onto an intelligent switch such as Sun's N1™ Grid Services Platform it may be possible to improve overall performance in the environment for data access. Modulizing file system services can also enable dynamic on-line reconfiguration. For example, the ability to modify archiving services without adversely affecting the other file system services.

Object-Based Storage

Today, the two primary methods for consolidating and sharing storage are NAS and SAN. NAS is typically employed when cross-platform sharing of files is required (i.e., front-end Web servers), and SAN is used for applications such as distributed databases that require high performance. However, users are increasingly requesting a storage solution with both scalable performance and cross-platform capabilities that is also secure and easy to manage. They also require efficient backup operations and better utilization of storage resources within clusters.

Object-based storage (OSD) is one potential method for providing both cross-platform and high performance capabilities by abstracting storage functions into objects. This higher-level storage abstraction enables the creation of self-managed, heterogeneous, shared storage by moving low-level storage functions into the storage device itself. The device is then directly accessed via a standard object interface rather than a traditional block-based interface such as SCSI or IDE, enabling high-performance by eliminating bottlenecks between servers and storage devices.

When combined with NAS (NAS-OSD), object-based storage offers direct access to storage, providing a high-performance file serving architecture. In this environment, the NAS files servers can be designed so they essentially become file managers that own and manage storage but do not serve files.

An object-based storage SAN (SAN-OSD) environment will benefit by a greater ability to operate in a heterogeneous storage and system environment because many of the system-specific storage management routines will reside in the devices.

A standard for OSDs is currently being defined by the Storage Networking Industry Association (SNIA). The standard includes a command set designed for the iSCSI protocol, providing extensions to the SCSI block command set. The object specification, along with the command set, define a new architecture for intelligent storage devices that can be integrated into massively parallel, high performance, IP-based storage environments.

This emerging technology is being combined with a scalable metadata management layer that provides a file system interface to applications, managing information such as directory membership, ownership, and permission. In addition, this layer is responsible for striping files across OSDs, as well as reliability and availability.

The development of OSDs complements other SNIA management projects by empowering storage devices to actively participate in and contribute to storage management automation. For example, an OSD may be able to automatically identify and access storage objects and deliver them to backup devices. It may also be possible to implement device-level quality of service to optimize utilization at the object level. Finally, an ODS may have object attributes that can define flexible data management policies such as how long to keep the data before it can be destroyed.

OSD is one potential method that Sun is investigating for achieving both high performance and cross-platform features for the advanced file system in QFS and SAM-FS software. For more information on the emerging OSD standard see: www.snia.org/tech_activities/workgroups/osd/

Chapter 5

Conclusion

Building an efficient and cost-effective data management, or ILM solution for large file systems requires three capabilities that traditional file systems and backup software are incapable of providing — performance, fast file sharing, and resource-efficient archiving.

Scalable, High-Performance File Sharing

Sun StorEdge QFS software is designed to provide maximum scalability, performance, and throughput for the most data-intensive applications. Complex data sets no longer need to be spread across multiple file systems — they can be stored in a single, scalable file system, significantly reducing administrative overhead without degrading performance. Multiple applications and users can share the same files and volumes on the network, dramatically improving productivity and scalability from the workgroup through the data center. Key benefits of Sun StorEdge QFS software include the ability to:

- Easily share files for both reading and writing purposes, across heterogeneous clients with third party software
- Easily distribute files, i.e., Web content distribution
- Grow file systems without compromising performance
- Consolidate and easily manage large amounts of data in one file system

Resource-Efficient Archiving

Sun StorEdge SAM-FS software helps organizations manage information assets according to their business value. The software enables dynamic archiving, reduced backup windows, and fast recovery to help enhance productivity and improve resource utilization. SAM-FS software

consolidates innovative archiving and backup methodologies in a high-performance file system with virtually unlimited scalability. The software replaces traditional backups to improve storage resource utilization for applications where data needs to be available continuously and quickly restored in the event of a business disruption. Administrators can set automatic archiving policies to determine when, where, and how information is stored, managing large volumes of data cost-effectively. Metadata archiving and read-behind features help enterprises recover from business disruptions in minutes or hours, as opposed to days, and let users begin reading files even before they are fully restored. Enhanced policy-based administration and security features include quotas and access control lists (ACLs) to control space consumption and data access. Sun StorEdge SAM-FS software enables additional key benefits, including the ability to:

- Significantly reduce or eliminate backup windows by providing continuous archiving
- Quickly backup/archive and restore data from local or remote storage due to the high performance design of the product
- Grow file systems without limit, and without compromising performance
- Easily manage data by managing a single file system rather than multiple smaller file systems
- Easily manage very large files with SAM Segment
- Extend continuous archiving capabilities to remote sites with SAM Remote
- Assist with regulatory compliance issues affecting how long data should be stored, ensure data integrity and security and the proper disposal of data to meet the increasing regulatory scrutiny of business information
- Lower overall TCO by virtualizing the file system so that infrequently used data can be stored on less expensive media but can be accessed as easily as if it were local and online

Putting it All Together

SAM-QFS software puts it all together to form a new approach for ILM, enabling enterprises to get the most value from their information and meet demanding business requirements across a wide array of applications, regulations, user needs, and corporate policies, while delivering lower overall costs.

To help customers implement an ILM solution specific to their needs, Sun Services offers architecture, implementation, and management services for QFS and SAM-FS software and related data management solutions. A partial list of services includes:

- Fast restore for LANless backups
- Fast restore for NAS
- Email management (Sun Infrastructure Solution for Infinite Mailbox)
- Migration services
- Data warehousing
- Disaster recovery
- Archiving to a remote site for disaster recovery using SAM-Remote

Appendix A

References

Sun Microsystems posts product information in the form of data sheets, specifications, and white papers on its Web site at: www.sun.com/. Please also refer to the following resources for more information on topics discussed in this paper:

Web Sites

- www.sun.com/storage/software/performance (QFS software)
- www.sun.com/storage/software/utilization (SAM-FS software)

Papers

- *File System Technology — Thinking Outside the Box*, Lance Evans, SAM-QFS File Systems Group, Sun Microsystems
- *Improving Backup and Recovery Strategies*, A Technical White Paper, September 2003
- *Sharing Data with Sun StorEdge Performance Suite: QFS 4.0 Software*, A White Paper by Cliff Prescott, EMEA Data Management ISC
- *Profiling Data — A Method for Understanding and Justifying Data Storage Expenditures*, An Executive Brief

Organizations can also contact a local Sun sales representative to learn how Sun can help build competitive advantage with Sun StorEdge products that match data delivery needs from the workgroup to the data center.

SUN™ Copyright 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, CA 95054, U.S.A. All rights reserved.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS

Sun, Sun Microsystems, the Sun logo, Solaris, Sun StorEdge, Sun Fire, and Sun N1 are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.



Please
Recycle



Adobe PostScript

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA Phone 1-800-555-9786 or 1-800-555-9SUN Web sun.com



Sun Worldwide Sales Offices: Africa (North, West and Central) +33-13-067-4680, Argentina +5411-4317-5600, Australia +61-2-9844-5000, Austria +43-1-60563-0, Belgium +32-2-704-8000, Brazil +55-11-5187-2100, Canada +905-477-6745, Chile +56-2-3724500, Colombia +571-629-2323, Commonwealth of Independent States +7-502-935-8411, Czech Republic +420-2-3300-9311, Denmark +45 4556 5000, Egypt +202-570-9442, Estonia +372-6-308-900, Finland +358-9-525-561, France +33-134-03-00-00, Germany +49-89-46008-0, Greece +30-1-618-8111, Hungary +36-1-489-8900, Iceland +354-563-3010, India-Bangalore +91-80-2298989/2295454; New Delhi +91-11-6106000; Mumbai +91-22-697-8111, Ireland +353-1-8055-666, Israel +972-9-9710500, Italy +39-02-641511, Japan +81-3-5717-5000, Kazakhstan +7-3272-466774, Korea +822-2193-5114, Latvia +371-750-3700, Lithuania +370-729-8468, Luxembourg +352-49 11 33 1, Malaysia +603-21161888, Mexico +52-5-258-6100, The Netherlands +00-31-33-45-15-000, New Zealand-Auckland +64-9-976-6800; Wellington +64-4-462-0780, Norway +47 23 36 96 00, People's Republic of China-Beijing +86-10-6803-5588; Chengdu +86-28-619-9333; Guangzhou +86-20-8755-5900; Shanghai +86-21-6466-1228; Hong Kong +852-2202-6688, Poland +48-22-8747800, Portugal +351-21-4134000, Russia +7-502-935-8411, Singapore +65-6438-1888, Slovak Republic +421-2-4342-94-85, South Africa +27 11 256-6300, Spain +34-91-596-9900, Sweden +46-8-631-10-00, Switzerland-German 41-1-908-90-00; French 41-22-999-0444, Taiwan +886-2-8732-9933, Thailand +662-344-6888, Turkey +90-212-335-22-00, United Arab Emirates +9714-3366333, United Kingdom +44 0 1252 420000, United States +1-800-555-9SUN or +1-650-960-1300, Venezuela +58-2-905-3800