White Paper

SGI® InfiniteStorage Shared Filesystem CXFS™: A High-Performance, Multi-OS Filesystem from SGI

Laura Shepard and Eric Eppe

Abstract

While storage area networks (SANs) give multiple computer systems high-speed connections to shared storage, they do not remove the data-sharing bottlenecks associated with direct-attached and network-attached storage. Even in a SAN, storage is still dedicated to a specific system. This being the case, when there is a need to share data among systems, traditional file-serving and file-sharing methods such as NFS, FTP, CIFS Samba®, and RAID controller-based copies are still being used. Each of these data-sharing mechanisms has performance, availability, and complexity penalties.

To completely remove these penalties and realize the promise of SAN technology, SGI has developed InfiniteStorage Shared Filesystem CXFS, a revolutionary shared filesystem for SANs that allows multiple heterogeneous systems to simultaneously access all data on a SAN. In combination with other storage virtualization technologies, CXFS offers heterogeneous systems unprecedented high-speed shared access to data on high-speed storage.

CXFS enables 64-bit IRIX® OS-based systems, Solaris™ 8 and Solaris™ 9, Windows NT® 4.0, Windows® 2000, 64-bit Linux® for SGI® Altix®, Red Hat® 7.3 for 32-bit Linux, and IBM® AIX® 5L systems to seamlessly and instantly share data with no copies to wait for, store, and administer. Windows® XP and Mac OS® X will be available in the second half of calendar year 2003. Additional UNIX® support is in development.

CXFS leverages the proven capabilities of SGI® InfiniteStorage Filesystem XFS®, the most scalable, high-performance, journaled 64-bit filesystem available. CXFS brings the full feature set and performance of XFS to heterogeneous SANs. This paper examines the issues surrounding high-speed shared data access and describes how CXFS overcomes these challenges while providing key advantages to customer applications.

For a detailed discussion of XFS, see Scalability and Performance in Modern Filesystems." [SGI, 2000].

1.0 CXFS: Fulfilling the Promise of SAN Technology

While SANs offer the benefits of consolidated storage, centralized management, and a high-speed data network, CXFS enables true data sharing by allowing SAN-attached systems direct access to shared filesystems. A shared filesystem provides data access speeds well beyond what is achievable via traditional methods such as NFS and FTP, solving data-sharing bottlenecks for a broad range of environments. This significantly boosts productivity when large files are shared by multiple processes in a workflow.

The explosive growth in storage deployment has created significant challenges for traditional storage architectures. Plummeting prices of storage media, increasingly data-intensive applications on ever more powerful platforms, and expanding data availability requirements are shifting the traditional paradigms of server-attached or network-attached storage. SANs were developed in response to these changing requirements. SAN technology allows multiple computer systems to share access to storage resources using high-speed Fibre Channel connections.

SGI uses a three-layer model to describe SAN technology. Layer 1, the SAN infrastructure, consists of the basic hardware and software components necessary to construct a working SAN.

Layer 1 is the basic hardware infrastructure of the SAN. In a SAN, connections between systems and storage are made through one or more Fibre Channel switches. The systems, storage, adapters, cables, and switches make up a SAN; the switches and cables alone are referred to as a Fibre Channel “fabric.” Today’s SAN technology allows up to 400MB per second (full duplex) per single Fibre Channel connection.

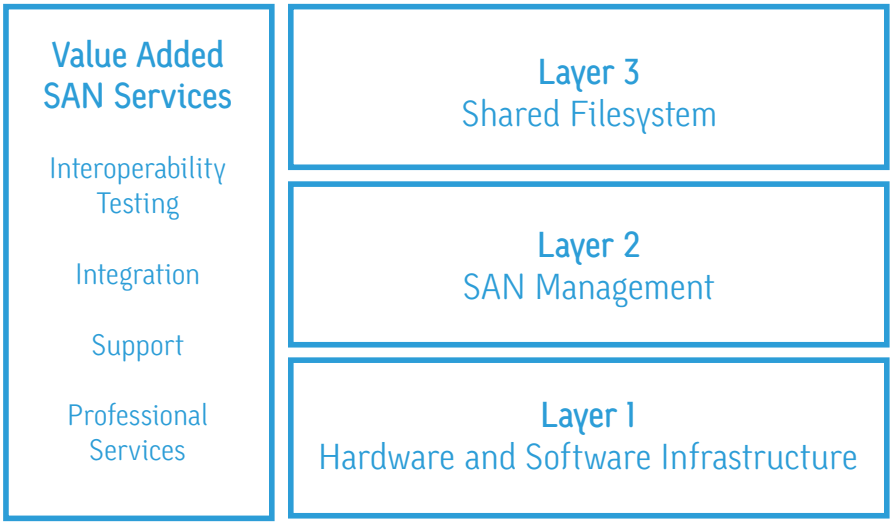


Fig. 1. Storage area network layer model

Layer 2, SAN management, includes tools to facilitate monitoring and management of the various components of the SAN and to protect SAN data. Most of the tools used in direct-attach storage environments are available for SANs. More comprehensive LAN-management-style tools that tie common management functions together are also becoming available.

SANs provide many of the same configuration options as traditional networks, but avoid the high overhead of traditional network protocols. SANs provide the ability to deploy flexible, fault-tolerant storage systems with exceptional performance. Fibre Channel SANs combine the time-tested SCSI protocol with high-speed serial data transmission for outstanding reliability, high performance, and increased flexibility.

The greatest potential of SAN technology lies in Layer 3, the shared filesystem. Layer 1 and Layer 2 components allow a storage infrastructure to be built in which all SAN-connected computer systems potentially have access to all storage but don t provide the ability to share data. Separate computer systems cannot access data in the same filesystem, let alone share the same files. Additional software is required to mediate and manage shared access; otherwise, data would quickly become corrupted. A shared filesystem is designed to meet this need, providing direct high-speed access to data at SAN speeds.

Even if consolidating storage is one of the primary drivers for building a SAN, because it greatly simplifies the administration of RAID arrays, a high-speed shared filesystem can reduce or eliminate the need for replication, freeing disk space for application use and dramatically reducing management associated with replicated data.

Thus, a shared filesystem can change the way that many customers work by changing and improving the efficiency of workflow between systems needing to share data. In applications such as digital media or oil and gas exploration, the output of one processing step becomes the input to the next. A shared filesystem eliminates the need to copy data and overcomes the bandwidth limitations of LANs.

To satisfy the workflow requirements of cutting-edge applications such as these, SGI undertook the development of SGI® InfiniteStorage Shared Filesystem CXFS. CXFS is a first-of-its-kind shared filesystem for SANs that meets the exacting requirements of high-performance data sharing. CXFS fulfills the promise of all-SAN systems directly accessing

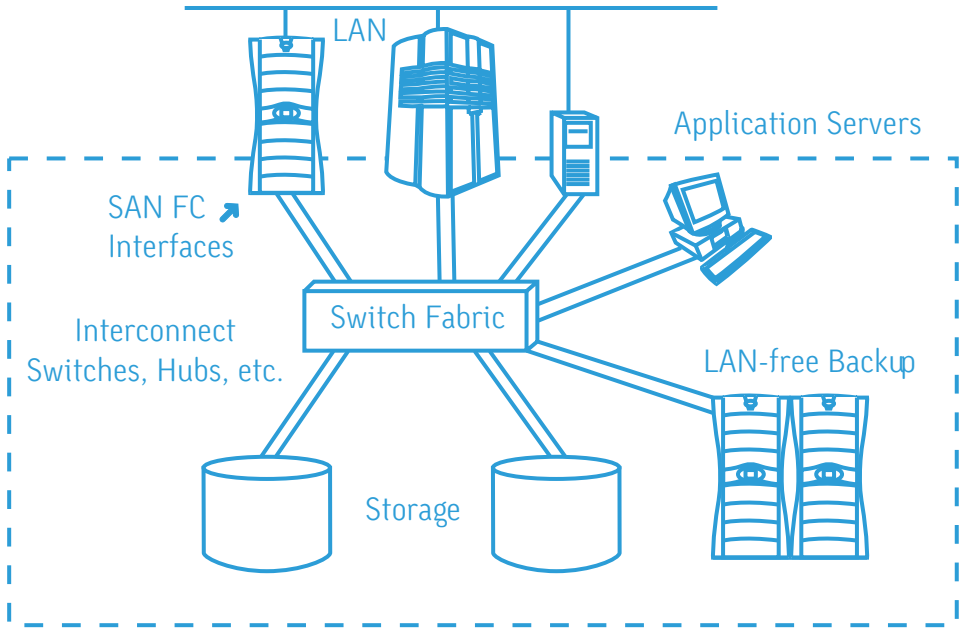


Fig. 2. Storage area network layer model

all files stored on SAN-attached devices. In combination with storage virtualization technologies such as hierarchical storage management (HSM), CXFS can provide heterogeneous system platforms with unprecedented high-speed access to SAN storage capacity.

2.0 Challenges of a Shared Filesystem

CXFS offers the performance and features of a traditional filesystem while providing the coordination needed for file sharing, distributed file locking, and fault isolation and recovery.

The challenges faced by a SAN shared filesystem are different from those faced by the traditional network filesystem. For a network filesystem, all transactions are mediated and controlled by a network file server. While the same approach could be transferred to a SAN using much the same protocols, this would not eliminate the fundamental limitations of the file server or take advantage of the benefits of a SAN.

The file server is the main impediment to performance and represents a single point of failure. The design challenges faced by the shared filesystem are more akin to the challenges of a traditional filesystem design combined with those of high-availability solutions. Traditional filesystems have evolved over many years to optimize the performance of the underlying disk pool. Application data is typically cached in the host system's memory to speed access to the filesystem. This caching—essential to filesystem performance—is the reason systems cannot simply share data stored in traditional filesystems.

If multiple systems assume they have control of the filesystem and write filesystem data they have in their cache, they will quickly corrupt the filesystem. On the other hand, implementing a filesystem that does not allow data caching would provide unacceptably slow access. Distributed data-cache management for shared access is one of the key challenges faced by the shared filesystem.

Like high-availability solutions, a shared filesystem should preserve data access and continuity in the event of failure of some of the systems or other components that make up the storage area network. Metadata management among multiple systems requires a level of control that must be protected and preserved in case of a failure to ensure continuing data integrity.

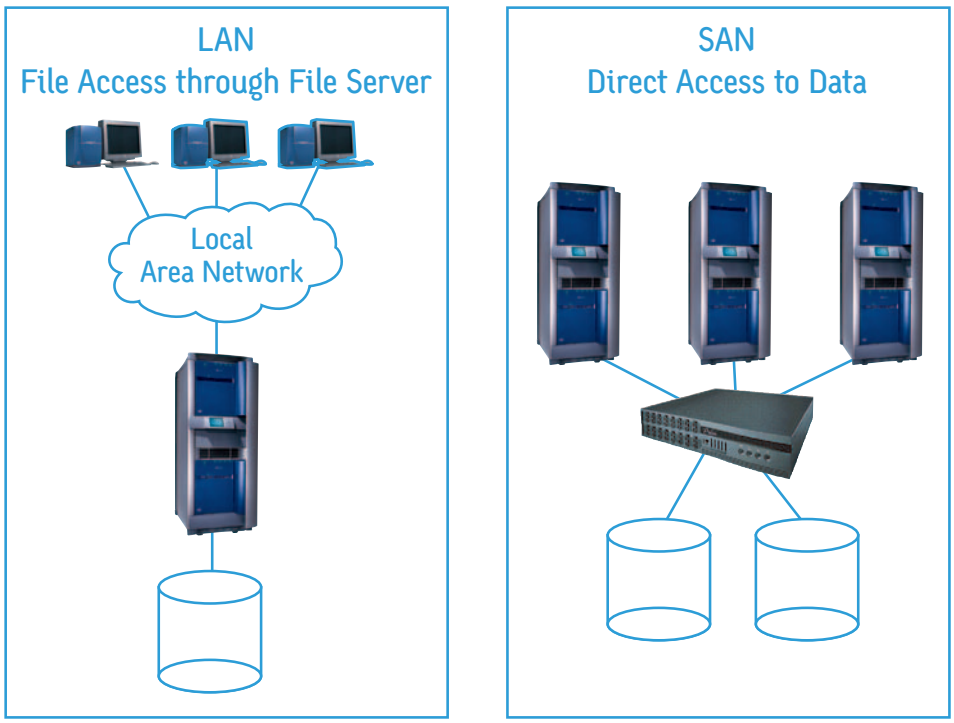


Fig. 3. Comparing LANs and SANs

CXFS is uniquely designed to overcome these problems. CXFS:

- Provides the performance and features of a traditional stand-alone filesystem
- Solves the metadata coordination problem
- Provides POSIX®-compliant file locking
- Provides fault isolation and recovery in case of system or SAN failures
- Supports memory-mapped files
- Allows execution of end-user applications without code changes
- Supports quotas
- Is a true, kernel-level, shared filesystem with no NFS component
- Is compatible with Trusted IRIX™ (Trusted IRIX is a secure form of the IRIX OS that has been granted a B1 certification)

The remainder of this paper explores how CXFS addresses these issues. In addition, because of their critical importance to the overall usability of the product, this paper also describes:

- The CXFS graphical user interface
- LAN-free backup using CXFS
- Real-world examples of CXFS benefits

3.0 Building on Proven Technology

CXFS is built on the proven foundation of the XFS filesystem. In addition, CXFS leverages technologies from other SGI® products such as FailSafe, thereby providing greater performance and robustness than software designed from scratch.

SGI gained a significant advantage in the development of CXFS by leveraging existing technologies. CXFS was designed as an extension to the XFS filesystem, widely recognized as the most scalable and highest performance 64-bit filesystem available today. Many years of development effort went into creating the rich feature set of the XFS filesystem. CXFS has successfully retained important XFS features, including:

- Journaling for reliability and fast (subsecond) recovery
- 64-bit scalability to support extremely large filesystems and individual files
- Industry-leading performance
- Dynamically allocated metadata space
- User, group, or project quotas

- Filesystem reorganizer (defragmenter) to restore the performance of heavily used filesystems
- Direct or asynchronous I/O and memory-mapped files

CXFS preserves these underlying XFS features while distributing I/O directly between disks and hosts. The XFS I/O path uses asynchronous buffering techniques to avoid unnecessary physical I/Os by delaying writes as long as possible. This allows the filesystem to allocate disk space efficiently. Space tends to be allocated in large contiguous chunks, yielding sustained high bandwidths.

XFS makes extensive use of B-trees in its internal structures, including directories, which allow XFS and CXFS to maintain excellent response times, even as the number of files in a directory grows to tens of thousands or hundreds of thousands of files. The primary differences between XFS and CXFS involve only configuration tasks.

CXFS disk volumes are constructed using XVM, SGI® InfiniteStorage Volume Manager. XVM augments the earlier XLV volume manager by providing disk striping, mirroring, and concatenation in any possible combination. Advanced recovery features are fully integrated, and XVM can accommodate dynamic LUN expansion and other features of advanced RAID arrays. The low-level mechanisms for sharing disk volumes between systems are provided, making defined disk volumes visible across multiple systems. XVM can combine a large number of disks across multiple Fibre Channels into high-transaction-rate, high-bandwidth, and highly reliable configurations.

XVM is designed to handle mass-storage growth far into the future. It will handle many orders of magnitude more disks than the largest disk farms of today. It allows one to configure millions of terabytes (exabytes) of storage in one or more filesystems across thousands of disks.

4.0 Metadata Management and File Locking

While file data flows directly between systems and disk, CXFS metadata is managed using a client/server approach. The algorithms used provide superior file performance, guarantee data integrity, and are more robust than those used in network filesystems.

Filesystem metadata includes the information that describes individual files (file name, size, location, permissions, etc.) and information about the filesystem itself, such as which disk blocks are available for allocation to files. Coordinating access to a shared filesystem requires ensuring that each system accessing the shared data has a consistent and up-to-date view of the metadata. There are several general approaches to solve this problem. Theoretically, metadata can be fully distributed so that each active computer system has direct access to the metadata it requires. Such a scheme requires complex mechanisms to ensure integrity.

Alternatively, a client/server model in which a single server is designated as a clearinghouse for all metadata access may be used. Systems still have direct access to disk to read and write data. Note that this is substantially different from the network filesystem model in which all information, metadata and data, uses the same data path. CXFS uses the client/server model to manage metadata. Only IRIX and Altix hosts are allowed to be metadata servers.

The level of complexity required by fully distributed metadata (a la DFS) is difficult to manage, and the necessary technology has not been developed to a point suitable for robust commercial products. SGI continues to collaborate on this approach with university partners. In the interim, great care has been taken to provide the greatest possible performance from a centralized metadata server.

While data moves directly between the hosts and disks (peer-to-disk), metadata is coordinated through a server for each CXFS filesystem. This metadata server acts as a

central clearinghouse for metadata logging, file locking, buffer coherency, and other necessary coordination functions.

CXFS metadata servers are highly available by design, with built-in failover and recovery. Metadata servers provide high transaction rates, fast metadata queries, and rapid recovery if a failure does occur. The CXFS metadata server is based on case-hardened XFS transaction logging (journaling) and other proven XFS technologies. The CXFS metadata path is divided between the metadata clients and the metadata server, but the underlying technology for all filesystem transactions is proven XFS filesystem code.

CXFS delivers local or near-local file access performance because only the metadata passes through the server; data reads and writes are direct to disk. All CXFS requests for metadata are routed over a TCP/IP network and through a metadata server, and all changes to metadata are sent to the metadata server, which uses the advanced XFS journal features to log metadata changes. Since the amount of the metadata is typically small relative to the amount of user data, the bandwidth of a Fast Ethernet LAN is sufficient for the metadata traffic, although Gigabit Ethernet may also be used. A dedicated LAN is required for metadata traffic.

Great care has been taken to ensure the speed of metadata transactions to minimize the impact of the metadata server on metadata-intensive workloads. Several underlying XFS features enhance metadata performance:

- Fast metadata algorithms, including excellent buffering
- Sophisticated structures and algorithms for fast lookups
- Ability to allocate large extent chunks, minimizing metadata transactions for space allocation

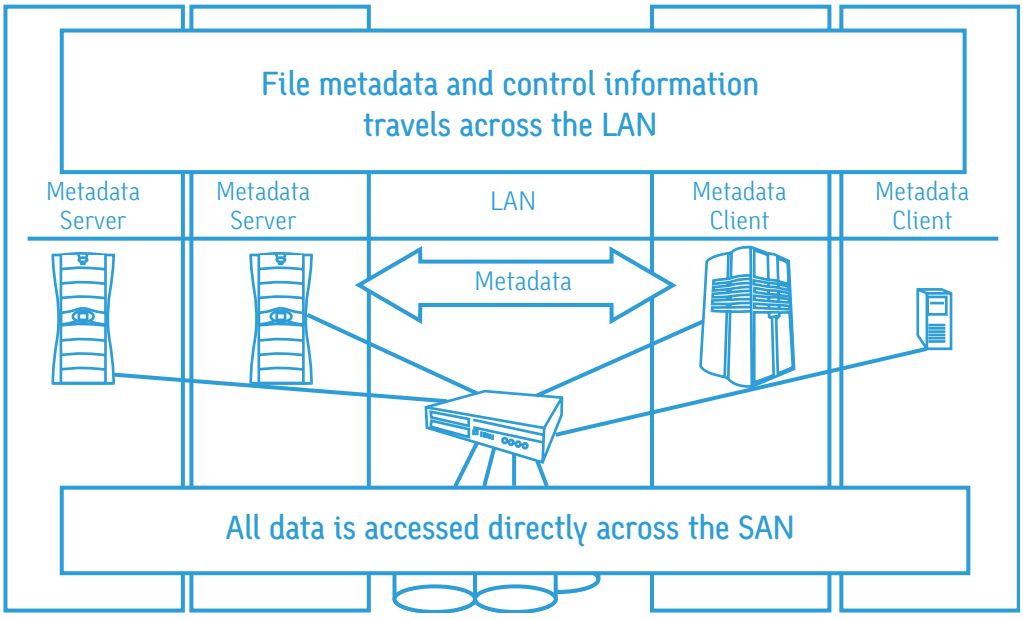


Fig. 4. Flow of data and metadata in CXFS

In addition, CXFS includes the following enhancements to streamline metadata accesses:

- Specialized thin RPCs
- Excellent buffering of both data and metadata on the clients
- Multiple metadata servers: one per filesystem
- Built-in metadata server failover or relocation
- Ability to bypass the CXFS layers on the metadata server host
- Ability to dedicate small hosts as metadata servers, if necessary

Under most workloads, CXFS performance far exceeds that of traditional network filesystems and approaches that of stand-alone XFS. Even in the worst case, where the workload is almost entirely metadata access, performance is typically more than double that of a network filesystem.

Operations to the CXFS metadata server are typically infrequent compared with data operations directly to disk. For example, opening a file causes a request for file metadata from the metadata server. After the file is open, a process may read or write the file many times without additional metadata requests. When the file size or other metadata attributes for the file change, this triggers a metadata operation.

CXFS uses tokens to manage and control file metadata and file access. There are a number of tokens for each file being accessed via CXFS. These represent different aspects of the file—timestamps, size, extents, data, etc. Possession of a token gives a client the right to cache information associated with that token and in some cases to modify that information. When a client gets a token it also receives the associated metadata from the server—file data is accessed directly from disk.

In the case where a client is the only one accessing a file, once it has the tokens and the associated metadata it does not need to talk to the server unless it needs to change the file in some way. When writing to the file, the client needs to talk to the server only to allocate new space. If unused space—a file extent—has already been allocated, the file can be written as necessary.

Different operations require different sets of tokens. Once a client acquires a token, it keeps the token until either another client wants a conflicting token or it is done with the file. There are read and write versions of each token. While multiple clients can have a copy of the read token for a given file, only one can have the write token and there can be no read tokens on clients when the write token is out. Losing a token means a client must stop looking at the cached information associated with it. If a client is holding modified information it must return cached metadata to the server and cached file data to disk. Token management for the cases of multiple readers or writers is illustrated in figure 5.

Multiple hosts can cache buffered file data at once—provided no one is writing to the file. (This is ideal for Web serving, allowing many hosts to access, cache, and serve the same pages.) Only one host at a time can have dirty buffers for a file. Multiple hosts can do direct reads and writes to a file in parallel with the same guarantees as multiple local processes accessing the same file. CXFS, through its token mechanism—provides much stronger guarantees of correctness than does NFS, even when file locking is not used.

Because file locking has been a perennial source of problems for NFS (the most commonly used protocol for network file sharing), great care has been taken to ensure that file locking works correctly in CXFS. POSIX, BSD, and SVR4 file locks are implemented by forwarding client lock requests to the metadata server with RPCs. The server maintains information about all locks being held. Locks behave the same whether the locking processes run on a single host or multiple hosts. No changes are required for an application to use locking with CXFS.

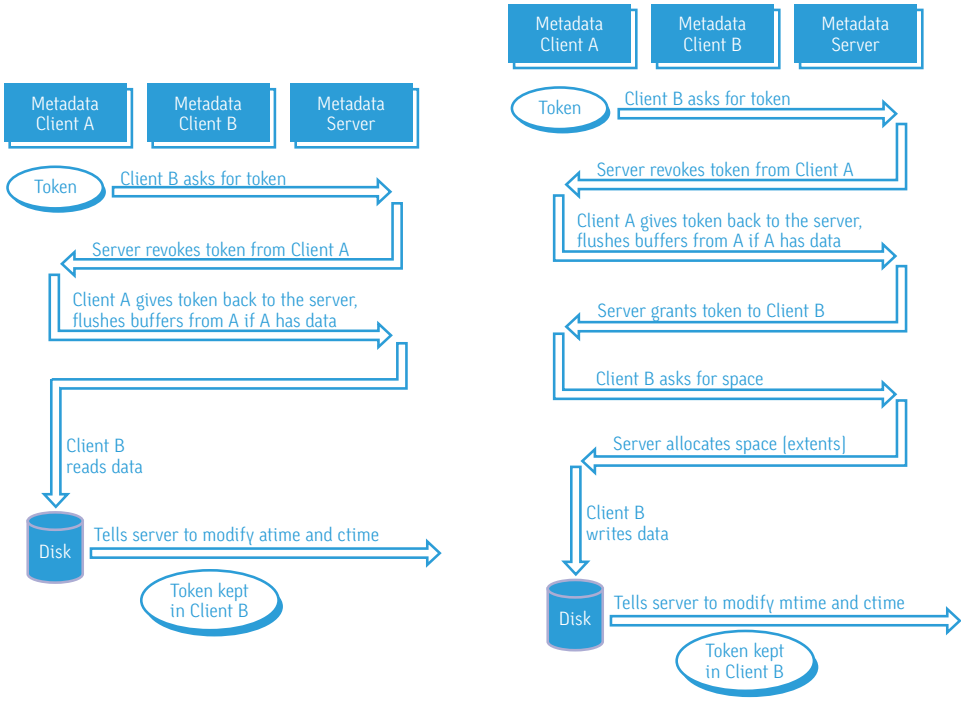


Fig. 5. Metadata flow for read and write when multiple clients are accessing a file

5.0 Fault Isolation and Recovery

CXFS provides robust features to ensure that data is always available, even in the event of system or component failures.

The CXFS physical configuration can be designed to provide exceptional resilience. Disks are typically protected in RAID arrays. Redundant Fibre Channel fabrics can be used to provide each host multiple paths across the SAN to stored data. A dedicated TCP/IP network is provided for metadata and other CXFS traffic. Redundant networks will be supported in an upcoming release. Remote reset across serial lines can also be configured so that misbehaving systems can be restarted. The resilience of the hardware can readily be tailored to the sensitivity of the data and specific site requirements. Minimally redundant configurations are ideal for shared high-speed data access, while fully redundant configurations can be used for high availability applications.

Because CXFS allows a filesystem to be shared between independent computer systems, great care must be taken to provide failure detection and recovery to ensure data integrity. CXFS must be able to detect and recover from single-system failure, failure of the active metadata server, failure of SAN or TCP/IP network components, and possible partition of the SAN or the TCP/IP network.

In addition to metadata traffic, the health of all systems sharing a filesystem is monitored across the TCP/IP network using standard messaging and heartbeat. Should a system failure be detected, the metadata server will automatically take steps to recover. The metadata server inspects the XFS journal and rolls back any incomplete transactions the client had in progress. This process is exactly analogous to the recovery process on a stand-alone system that crashes and reboots, except that the metadata server performs the recovery on behalf of the client.

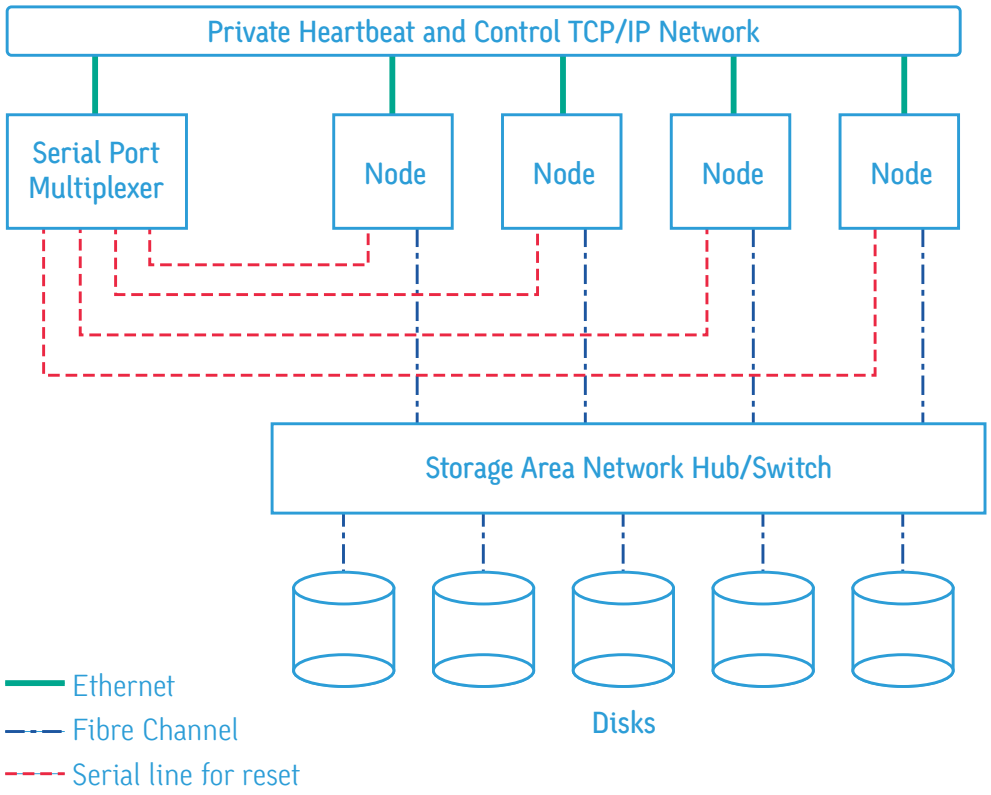


Fig. 6. A storage area network illustrating optional reset circuitry

Should the metadata server on a particular host fail or be disabled, control will move to a host designated as a backup metadata server. This shift in control is transparent to user processes using CXFS files. They proceed normally through this transition. Multiple hosts can be designated as backup hosts for metadata servers.

If no backup is designated, the filesystem will recover when the metadata server comes back online. Client access to the filesystem is suspended until the server is active.

CXFS takes advantage of the resiliency of the XFS log based filesystem (journaling). Metadata transactions are logged by the metadata server and mirrored for rapid and reliable recovery. XFS filesystem recovery avoids the need for time-consuming filesystem checks during a recovery cycle after an ungraceful shutdown. Using the transaction log, XFS is typically able to update and validate a filesystem in a fraction of a second.

Network partitioning is the most difficult type of failure faced by shared filesystems. In the worst case, the TCP/IP network is partitioned into two separate pieces, and a potential metadata server exists in each network partition. If these servers were both to become active, the filesystem would quickly be corrupted.

CXFS prevents problems arising because of possible network partitioning by using a cluster membership quorum. The cluster membership is the group of nodes that are actively sharing resources in the cluster. In order for the filesystem to be active, a simple majority or quorum of the nodes must be available. This effectively prevents two metadata servers from becoming active.

With the availability of new CXFS clients running on other operating systems, such as Sun™ Solaris™ or Windows NT 4.0, SGI has developed a new complementary fault isolation mechanism named “I/O fencing,” which prevents data corruption in case of a client failure

detection. CXFS "I/O fencing" provides the ability to fence immediately any node out of the SAN fabric, suppressing any access to the I/O path and thus preventing potential data corruption.

For more information, see The CXFS Software Installation and Administration Guide.

6.0 User Interface

CXFS can be easily configured, monitored, and managed using Java™ language-based tools. The CXFS Manager has Web-browser-like functionality and provides the main configuration interface. It organizes all CXFS tasks into three different categories: nodes and cluster, filesystems, and diagnostics. The Guided Configuration Mode provides sets of tasks (tasksets) to help the new user through initial configuration and setup of the CXFS cluster (Set Up a New Cluster, Set Up a New Filesystem). The "Find a Task" category lets the user type keywords to find a specific task.

CXFS Cluster View provides the main monitoring interface, presenting an iconic view of the CXFS cluster showing nodes and filesystems. Cluster View is used to monitor an active CXFS cluster. Cluster members with error conditions blink red, items that have been defined but not activated appear gray, and items that are active and normal are green. Detailed information about any cluster element can be obtained through menu-based operations on a selected icon.

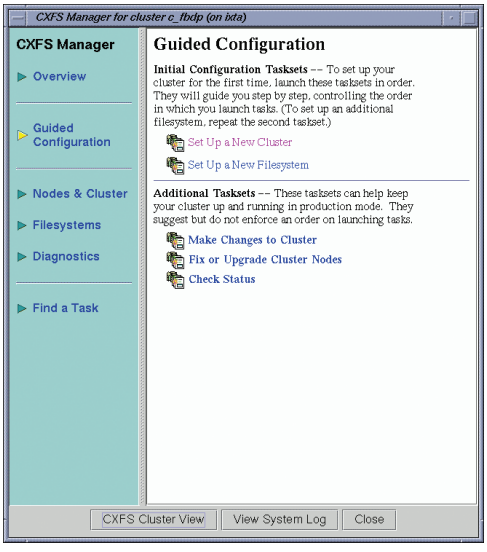


Fig. 7. CXFS manager shown in the guided configuration mode

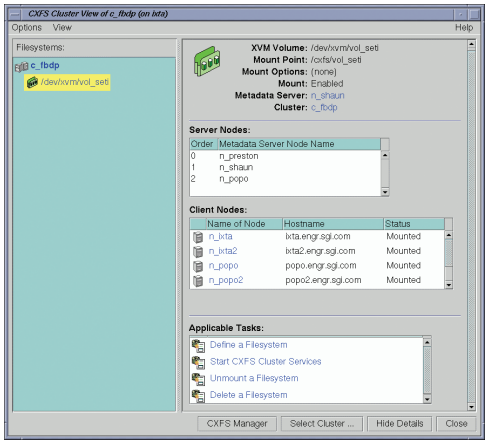


Fig. 8. CXFS cluster view showing details for a single filesystem

Because the tools are written in Java, they are not limited to execution on an IRIX console. They can be run in any Web browser supporting Java or on any platform supporting a Java run-time environment, making remote management of CXFS convenient and easy.

CXFS provides a flat, single-system view of the filesystem; it is identical from all hosts sharing the filesystem and is not dependent on any particular host. The path name is a normal POSIX path name; for example, /u/username/directory. This path does not vary if the metadata server moves from one host to another, if the server name is changed, or if a server is added or replaced. This simplifies storage management for administrators and users. Multiple processes on one SMP host and processes distributed across multiple hosts have the same view of the filesystem, with similar performance on each host.

This differs from typical networked filesystems, which tend to include the name of the file server in the path name. This difference reflects the simplicity of the SAN architecture

with its direct-to-disk I/O, compared with the extra hierarchy of the LAN filesystem that goes through a named server to get to the disks.

Unlike most networked filesystems, the CXFS filesystem provides a full UNIX filesystem interface, including POSIX, System V, and BSD interfaces. This includes semantics such as mandatory and advisory record locks. No special record-locking library is required Some semantics are not appropriate and not supported in shared filesystems. For example, root filesystems belong to a particular host, with system files configured for each particular host's characteristics. In a root filesystem, one defines devices for a particular host configuration, such as /etc/tty and /etc/tape. These device definitions are not appropriate in a shared filesystem. In addition, named pipes are supported in shared filesystems only when all the processes using them reside on the same system. This is because named pipes are implemented as shared-memory buffers.

CXFS supports XFS filesystem extensions, such as extended attributes and memory-mapped files. On some operating systems other than IRIX, these features may be limited by the particular operating system's capabilities.

7.0 LAN-Free Backup with CXFS

Using CXFS, backup traffic can be shifted from the LAN to the SAN, reducing network congestion and decreasing backup time—using CXFS with standard backup packages can reduce backup windows from hours to minutes. Existing backup applications work seamlessly in the CXFS environment.

Consolidated network backups are performed in many computer installations. Data is transferred over a LAN to a dedicated backup server, minimizing the number of required tape devices and centralizing operations. LAN backup has many advantages, but—not surprisingly given the tremendous growth in the amount of data being stored—has also led to new problems. Despite tremendous increases in network bandwidth, LANs frequently fail to keep up, and the increased network traffic created by backup data may also interfere with high-priority user traffic.

At the same time, backup windows—the available time in which backups can be performed—have been shrinking. This is due to such forces as globalization and e-commerce, which require data to be available around the clock. Administrators typically pick periods of low network utilization and low server load to perform backup.

CXFS provides the capability to perform backups using a method that not only moves data traffic off the LAN and onto the SAN to increase backup performance, but also removes the load from busy compute servers and moves it to a dedicated backup server. Existing time-tested backup applications such as Legato NetWorker® work seamlessly with CXFS without modification.

A backup server attached to a SAN running CXFS reads data directly from the filesystems to be backed up and writes it to storage just as if the filesystems were local to the backup server. The backup server therefore assumes the entire workload associated with backup. The backup server can be implemented using a relatively inexpensive workstation, analogous to the workstations used in other LAN-free backup methodologies. Multiple backup servers can be deployed if necessary.

Integration into a SAN allows backup bandwidth to scale to meet the needs of very large data sets. The number of disk channels, the size or number of backup servers, and the number of tape channels can all be scaled to provide backup bandwidth far in excess of what can be achieved with other methods.

The easiest migration path to server-free backup with CXFS requires only that the backup server be attached to the SAN and CXFS installed. Since the installed base of tape devices, libraries, etc., typically uses the SCSI interface, these devices remain directly attached to the backup server, as they are in a

network backup scenario. The backup server runs the backup application, accesses all filesystem data directly using CXFS, and transfers data to backup media as normal. The simplicity of this solution lies in the fact that the only new software introduced is CXFS. The backup server can continue to provide

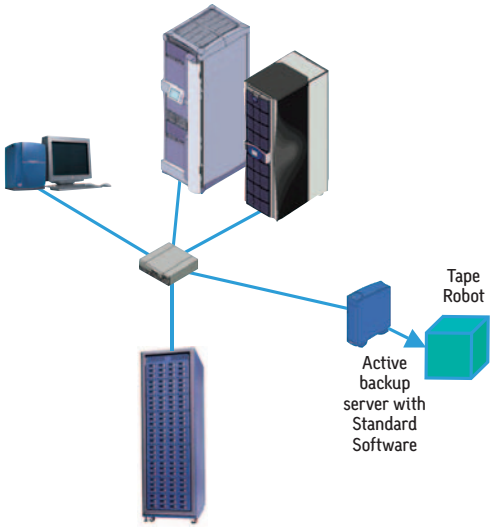


Fig. 9. Server-free backup using CXFS and standard backup software

backup services for network clients as required.

8.0 High Availability and Unlimited Storage: Using CXFS with FailSafe and DMF

By combining the unique features of CXFS with the SGI high-availability software product FailSafe and HSM product Data Migration Facility (DMF), SGI offers a complete data management storage architecture.

This storage architecture meets the most critical needs of data access and data management:

- Data availability with XFS, XVM, and CXFS
- Application availability with FailSafe
- Data virtualization with DMF and XVM

FailSafe provides application monitoring and failover to a secondary system in the event of a hardware or software failure. FailSafe clusters can consist of up to 16 nodes protecting one or many applications. Specialized agents are available for such applications as DMF, NFS, e-mail, Web serving, and popular databases. Scripts can be used to tailor FailSafe to other applications.

Using CXFS and FailSafe together combines the advantages of high-availability applications with a shared high-performance filesystem. As soon as an application fails over from one system to another, it can again begin accessing data in shared CXFS filesystems. FailSafe is not used to protect the availability of CXFS filesystems, since that is an intrinsic capability of CXFS. FailSafe and CXFS clusters are set up using the same procedures, dramatically simplifying use of the products together.

DMF is used to manage vast quantities of data. The HSM capabilities of DMF provide a flexible framework for automatically migrating less frequently used online data to less expensive near-line or offline media until it is needed. In environments where a large amount of data must be stored but only a limited subset is used at any given time, CXFS and DMF together provide an economical virtual storage environment that can accommodate huge amounts of data (tens or hundreds of terabytes). Online disk capacity can be maintained as a modest percentage of total storage capacity. Total storage capacity is increased through the addition of inexpensive tape cartridges rather than disk capacity (more expensive in terms of purchase cost, space usage, and power consumption). A small delay occurs when an offline file is accessed while the file is located and staged back to online storage. (An application can begin accessing the file before the copy to disk completes.) This method has an added advantage—since an accessed file is migrated from tape to relatively unconstrained disk resources, the file will in most cases be allocated space in a very efficient manner, increasing the performance of file access while it is online.

CXFS integrates with DMF and other HSM applications using the industry-standard data management application programmer's interface (DMAPI). The CXFS metadata server is responsible for mediating all DMAPI requests for HSM services. Therefore, DMF must be installed on the system that is designated as the metadata server for the CXFS filesystem. Most customers deploying DMF use FailSafe to ensure DMF availability. Used together, CXFS and DMF provide a single view of local and migrated files to all CXFS clients or NFS over CXFS clients. Any member of a CXFS cluster can transparently access files managed by DMF. The addition of FailSafe ensures the availability of data that has been migrated by DMF.

9.0 Multi-OS Platform Support

CXFS is available for 64-bit IRIX OS-based systems, Solaris 8 and Solaris 9, Windows NT 4.0, Windows 2000, 64-bit Linux for SGI Altix, Red Hat 7.3 for 32-bit Linux and IBM AIX 5L. CXFS for Windows XP and Mac OS X will be available in the second half of calendar year 2003. Additional UNIX . support is in development. IRIX systems supported include the Silicon Graphics® Octane® series, Silicon Graphics Fuel® and Silicon Graphics® Tezro®, SGI® Origin® 200, Silicon Graphics® Onyx2®, SGI® Onyx® 3000 series, SGI® Origin® 2000 series, and SGI® Origin® 3000 and SGI® Origin® 300 series systems. CXFS works with all RAID storage devices and SAN environments supported by SGI, including 1Gb and 2Gb switches, multiswitch fabrics, and hub-based SANs. A maximum of 48 CXFS nodes is supported today—64 nodes will be supported by the end of 2003, with plans to extend the limit beyond 64 in the near future.

While CXFS provides significant benefits for a homogeneous SAN, the new frontier in SAN virtualization is the ability for heterogeneous platforms running different operating systems to share data at high speed. There are several important technical considerations when implementing a shared filesystem with heterogeneous platform support.

Different processors use different representations for common data types. In order for heterogeneous platforms to use and understand filesystem metadata (and not corrupt it) a common data representation must be provided. This data representation is the XFS data format.

Most UNIX architectures make use of 64-bit addressing while Intel® architecture-based operating systems like Windows NT are 32-bit. A shared filesystem must address these differences in a standard way to ensure consistency. CXFS has been designed to handle these issues by storing metadata in a standard format across all platforms.

Implementing a shared filesystem with heterogeneous platform support is easier when done in user space rather than in the system kernel, but such an implementation may rely on features of the underlying operating system that are not appropriate for a shared data environment. User-space implementations also impact performance. CXFS is implemented in the kernel on all supported architectures (current and future), providing proven reliability, scalability, and the highest performance possible.

A few other vendors have implemented distributed filesystems on multiple platforms using existing OS-specific software packages and filesystems. These implementations are done primarily in user space, using NFS to manage metadata. NFS lacks the metadata performance and strict data correctness guarantees of CXFS. None of these products provides the performance and availability of CXFS.

10.0 CXFS Scalability and Performance

For common workloads, CXFS provides the same unsurpassed scalability and performance as the industry-leading XFS filesystem, enhancing the responsiveness of applications that use shared data.

CXFS has the features and performance to meet the challenges of distributed computing, combining the robustness and performance of

XFS with the convenience of data sharing. These benefits will be extended even further through the support of Mac OS X and other UNIX platforms.

CXFS continues to function at levels where many stand-alone filesystems fail. For instance, CXFS has been tested with up to 1 million files per directory, and it still works correctly and efficiently. At this level, most other filesystems stop working or are too slow to be useful.

In direct comparisons of CXFS on a single Fibre Channel Host Bus Adapter versus NFS on a single Gigabit Ethernet (both providing nominally the same bandwidth of 100MB per second), CXFS achieved an average practical transfer rate of 85MB per second. By comparison, NFS averaged only 25MB per second for the same workload, a workload that was optimized for NFS. This is attributable to the greater network and protocol overhead of NFS running on a TCP/IP LAN. Typically, Ethernet payloads are limited to 1,512 bytes per packet. Fibre Channel allows the application to negotiate a transfer window varying in size from a few bytes to 200KB, allowing the application to accommodate small and bulk data transfers.

In addition, CXFS and SANs can be scaled in ways that no network filesystem can. For instance, although Fibre Channel and network technologies like Gigabit Ethernet nominally provide the same bandwidth today (100MB per second), CXFS filesystems spanning multiple Fibre Channels on multiple HBAs can be created, effectively aggregating the available bandwidth to a single filesystem. A network filesystem is limited to the bandwidth of a single network and cannot be similarly aggregated across multiple networks.

CXFS takes the place of network filesystems where high-speed data access is most critical, while still allowing other network systems access to data. CXFS cluster members can be configured to run NFS or SAMBA CIFS, serving cluster data to clients outside the SAN.

For most workloads, CXFS provides performance approaching that of a non-shared XFS filesystem. With multiple Fibre Channel connections and multiple RAID disks, achievable bandwidths can reach many hundreds of megabytes per second or even many gigabytes per second. CXFS exhibits exceptional performance in many situations such as:

- Reads and writes to a file, opened by a single process
- Reads and writes to a file where all processes with that file open reside on the same host
- Multiple processes on multiple hosts reading the same file
- Multiple processes on multiple hosts reading and writing the same file, using direct I/O

Metadata and buffer coherency may become a bottleneck for some workloads. For instance, when multiple systems are reading and writing the same file, maintaining buffer coherency between systems is time consuming. These applications typically benefit from direct I/O, where all write data is flushed immediately to disk rather than being cached in memory buffers. Direct I/O eliminates the steps on the metadata server that would otherwise be required to flush dirty buffers from clients. Direct I/O is typically favored when file sizes approach or exceed the size of system memory, so many applications already take advantage of it.

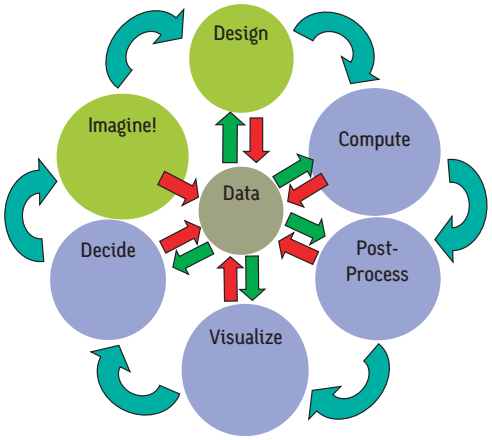
Other metadata-intensive operations may be slower on CXFS than they would be for a stand-alone filesystem. Operations that perform random access to numerous small files or those that repeatedly look up file and directory information may be noticeably slower. In all cases, however, these operations will be faster than they would be over common network filesystems.

The real measure of the value of a shared filesystem is how it improves the productivity of scientific and creative users. Working with large data sets, information is stored in large central repositories and copied using FTP or NFS to other systems for local processing.

These data sets may be upwards of 2TB of data per day. With CXFS, this expensive, time-wasting, and inefficient process can be eliminated, dramatically increasing operational efficiency and moving the focus to innovation and insight.

11.0 CXFS in Production

CXFS gives SGI customers the opportunity to change the way data and information flow, allowing users to simultaneously learn from the information and improve its value, leading



to more powerful scientific and creative innovation. CXFS excels in environments where workflow results in large amounts of data moving between systems.

Oil and gas exploration and digital media applications depend on a workflow-processing model in which multiple machines work serially to process a data and information stream. Output from a system is passed to another system for post processing and so on. This processing model requires sharing large quantities of data. Many sites are still using inefficient methods such as FTP or NFS to copy data between machines and limit the size of the data sets they process because of inadequate bandwidth for transferring data. By allowing applications on different machines to share data at high speed without copying, CXFS can save users a tremendous amount of time and money. This section provides examples where CXFS has improved the efficiency of several applications.

11.1 CXFS in Oil and Gas Exploration

A large oil and gas company is using CXFS in its seismic data analysis operation to help discover new petroleum fields. Specialized applications have been developed in-house to process data from field studies and image geological features below the earth's surface. Compute-intensive applications like this one typically generate so much data that the data set has to be segmented into smaller pieces to keep data transfer times between systems manageable.

CXFS was extensively tested and benchmarked in this environment to determine how much improvement could be made to the overall operation. To increase parallelism, applications were modified to allow them to synchronize, an optimization that was not possible prior to the introduction of CXFS.

The main application begins processing a data set and the output is directed to a file that resides on a CXFS shared filesystem. Once a set amount of output has been created, a second application running on a separate system begins processing the output without waiting for the first application to complete. The second application synchronizes with the first to ensure that it does not read past the current end of file. The second application directs its output to a new output file in the shared filesystem, and the process repeats through several additional processing steps until completion.

The use of CXFS along with the modest application changes described has decreased the time required for start-to-finish processing of a data set by as much as 35%. The customer has also been able to process data sets up to three times larger than were previously possible. A relatively modest investment in CXFS software to improve workflow dramatically increased the customer's ability to complete useful work. A much larger investment in raw computing power would have been required to achieve similar results otherwise.

11.2 CXFS in Video Post-Production (Media)

The introduction of digital technology in the film production and post-production industry has created dramatic changes. Computers are already used at many stages of the production of a movie, but it is generally accepted that in the future, production will be totally digital, from shooting to projection in theaters. Cutting-edge filmmakers are already shooting films entirely with new HD digital cameras.

Working with high-quality digital video assets requires applications to manage huge amounts of data. A single HDTV frame consumes a minimum of 8MB, and a movie displays 24 frames per second (192MB per second). Post-production involves many complex tasks, such as digitizing analog content (35 mm movies), nonlinear editing, digital effects, and compositing. These tasks are usually performed in a workflow process in which data moves from one computer to the next in a sequence.

Post-production houses have been relying on cumbersome methods to manage digital assets and move data from one host to another:

- Online storage is moved manually from system to system: A direct-attached RAID array on one machine is disconnected after processing on that system completes and then connected to another machine to carry out additional steps in the process. This is an efficient process because it avoids slow copying and offline media, but data availability could be impacted by RAID failures while moving the array back and forth. Also, only one host at a time can access the latest asset.
- An asset is transferred between systems using tape: Once a processing step is completed, the output of that step is manually copied to tape, carried to the next machine in the processing sequence, and copied back to disk on that system. The efficiency of this process is limited by the bandwidth of the tape device.
- An asset is copied over a network: Traditional network-based file sharing, like NFS and/or CIFS, or proprietary point-to-point networks are used. Due to limited network bandwidth and protocol overhead, it can take hours or overnight to transfer files between machines, seriously impeding work.

A leader in post-production responsible for more than 30 productions a year (analog and digital) was confronted with these challenges in its workflow. With more than 15 machines involved in the process, sharing data had become too complex and too slow, even using Gigabit Ethernet.

A CXFS SAN was installed in August 2000 on all existing SGI IRIX OS-based machines, including Origin 200, Origin 2000 series, Onyx2, Onyx 3000 series, and Octane systems. Each machine or set of machines is responsible for a different aspect of processing using specialized applications.

According to the customer, CXFS permitted the staff to work faster than ever before possible. It now takes 10 minutes to process an asset that previously required all night to copy from one system to the next. The customer is using the CXFS configuration in 3x8-hour cycles nonstop, dramatically increasing the return on investment.

Additional customer success stories demonstrating how a shared filesystem is being used to improve workflow can be found at *www.sgi.com/products/storage/*.

12.0 Summary

Storage area networks have become as essential to data communication as TCP/IP is to network communication, providing benefits such as connectivity, manageability, and shared infrastructure. SGI CXFS is the first robust multi-OS shared filesystem for SANs. Based on the field-proven XFS filesystem, CXFS provides a highly available, scalable, high-performance environment for sharing data between multiple operating systems on a SAN. CXFS supports IRIX, Windows NT and Windows 2000, Solaris, IBM® AIX®, and Red Hat® Linux® and will support other platforms, enabling an unprecedented level of data sharing between platforms.

Use of the CXFS filesystem increases availability, resilience, scalability, bandwidth, and transaction rates while reducing storage costs and streamlining backups compared to traditional networked filesystems. These benefits allow for users to simultaneously learn from information contained in their data and improve its value, leading to more powerful scientific and creative innovation. CXFS has demonstrably increased workflow and efficiency in applications such as reservoir simulation and visualization (oil and gas), fluid dynamics modeling in HPC clusters (weather forecasts), and video post-production (media) as well as in other areas of scientific and creative endeavor.

The storage architecture of the future is one in which all systems have access to all data without concern for where it is located, how it is stored, or how it is formatted. By combining the capabilities of CXFS with SGI Data Migration Facility and FailSafe as an integrated solution, SGI customers can realize these benefits today.

For more information on SGI InfiniteStorage Shared Filesystem CXFS or other SGI prod-

Corporate Office
1500 Crittenden Lane
Mountain View, CA 94043
(650) 960-1980
www.sgi.com

North America +1 800.800.7441
Latin America +55 11.5509.1455
Europe +44 118.925.7500
Japan +81 3.5488.1811
Asia Pacific +1 650.933.3000

2691 [06/16/2004] J14634