Functionality and Performance Evaluation of File Systems for Storage Area Networks (SAN)

**Martha Bancroft, Nick Bear, Jim Finlayson,
Robert Hill, Richard Isicoff and Hoot Thompson**

Storage Technologies Knowledge Based Center,
Department of Defense

Patuxent Technology Partners, LLC
11030 Clara Barton Drive
Fairfax Station, VA 22039
703-250-3754 Voice
703-250-3742 Fax
hoot@ptpnow.com

The Eighth NASA Goddard Conference on Mass Storage Systems and Technologies held in cooperation with the Seventeenth IEEE Symposium on Mass Storage Systems.

Abstract

The demand for consolidated, widely accessible data stores continues to escalate. With the volume of data being retained mounting as well, a variety of markets are recognizing the advantage of shared data in terms of both cost and performance. Traditionally, common access has been addressed with network-attached fileservers employing data sharing protocols such as the Network File System (NFS). A new approach, poised to deliver high bandwidth access by multiple, heterogeneous platforms to a common storage repository at reduced cost, is beginning to emerge. Storage Area Networking (SAN) is an open-storage architecture designed to eliminate many of the traditional bottlenecks associated with secondary and tertiary storage devices. Conventional high performance computing (HPC) sites and compute-intensive production sites can benefit from such architectures as the need to share computational input and output data sets expands and the mix of computational platforms continues to diversify.

Recognizing the potential value of SAN solutions in their overall data management roadmap, the Storage Technologies Knowledge Based Center of the Department of Defense commissioned a research project in mid-1999 to evaluate the functionality and performance of emerging SAN technologies. The initial focus has been on SAN file systems that offer management of disk-resident data. The desire, however, is to expand the effort to include other traditional data storage functions such as backup, hierarchical storage and archiving using tape technologies. The underlying goal is high bandwidth and reliable access to data with guaranteed long-term retention while presenting a seamless and transparent interface to the users regardless of data location. Operational stability and ease of administration are key requirements as is overall data integrity.

When complete solutions will be available and just how robust the family of products will be remains unclear. The magnitude of this challenge is realized when considering that production use of these technologies will entail serving numerous, likely heterogeneous clients managing a variety of file sizes (tens of kilobytes to multiple gigabytes) and dealing with a mix of applications and access patterns.

As a starting point for the testing, the Center established an environment that features a pair of SGI™ Origin™2000s, two SGI 320 Windows NT® platforms and a fibre channel switch fabric with shared connectivity to over one terabyte of RAID storage. This configuration is expected to grow in number and types of computers (operating systems) as well as with the addition of fabric-attached tape technologies. This preliminary report deals with using the environment to evaluate third-party SAN file systems and related infrastructure technologies. It is a snapshot in time with only initial testing completed. More comprehensive, on-going status and plans, observations and performance data are available on-line at

http://www.patuxent-tech.com/SANresearch

During this stage of the evaluation, each file system product is being exercised to determine its performance under load, its operability and scalability as a function of clients and traffic, and its overall functionality and usability. The motivation is to assess the readiness of SAN file systems to move into production and set realistic timeframe expectations for making such a transition. Although this initiative is conducted under the auspices of the Department of Defense, this research should prove relevant to any large data center operation.

1 Introduction

Several definitions of a Storage Area Network (SAN) exist as related to common, shared repositories of data. The implementation of interest is one that permits true data and/or file sharing among heterogeneous client computers. This differentiates them from SAN systems that permit merely physical device sharing with data partitioned (zoned) into separate file systems. Refer to Figure 1 for a depiction of a notional SAN system. The architecture is broken into three basic elements: SAN clients, a switch fabric and shared storage. The software orchestrating the architecture is what unites the components and determines exactly how these elements behave as a system. The optimum vision is a single file system managing and granting access to data in the shared storage with high bandwidth fibre channel links facilitating transfers to and from the storage.

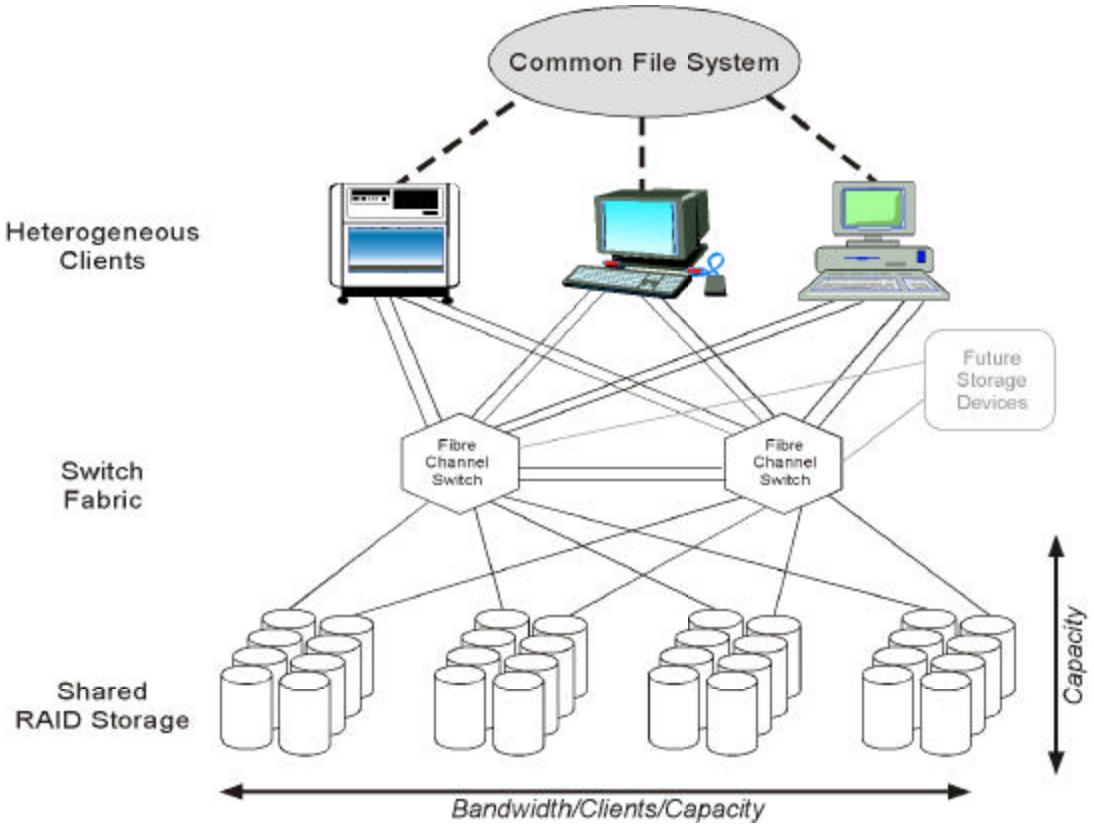

Figure 1. Notional Storage Area Network (SAN)

The advantages of the topology are readily apparent:

- File transfer performance as seen by the client compares with that of directly attached storage.
- The switch fabric can be expanded horizontally by adding switches (client and storage ports) to increase overall system bandwidth.
- Individual fibre channels can be added, combined and striped across to increase bandwidth between an individual client and storage.

- Multiple routes through the fabric between the clients and storage avoid single point failures and/or isolating data.
- Storage depth can be increased by adding or using higher density devices.
- The fabric topology can be expanded to include other storage technologies such as tape drives either directly or by using bridges.

The functioning of the common file system along with how files are opened, closed, read, written, etc. is fundamental to the operation of the SAN. File system control and metadata can co-exist with one of the application clients or be hosted on a dedicated computer. Metadata and locking information can be stored locally or on the SAN itself. A variety of implementations are technically feasible, each with its own functionality and performance implications.

2 Requirements Analysis and Test Planning

Recognizing the potential value of SAN solutions in their overall data management roadmap, the Storage Technologies Knowledge Based Center of the Department of Defense commissioned a research project in mid-1999 to evaluate the functionality and performance of emerging SAN technologies. The initial focus has been on SAN file systems that offer management of disk-resident data. The desire, however, is to expand the effort to include other traditional data storage functions such as backup, hierarchical storage and archiving using tape technologies. The underlying goal is high bandwidth and reliable access to data with guaranteed long-term retention while presenting a seamless and transparent interface to the users regardless of data location. Operational stability and ease of administration are key requirements as is overall data integrity. When complete solutions will be available and just how robust the family of products will be remains unclear. The magnitude of this challenge is realized when considering that production use of these technologies will entail serving numerous, likely heterogeneous clients managing a variety of file sizes (tens of kilobytes to multiple gigabytes) and dealing with a mix of applications and access patterns.

2.1 Requirements Drivers

A SAN file system, when deployed in the production environment, will be expected to maintain a very high level of performance, interoperability, maintainability and availability. Accordingly, the research effort is evaluating the attributes presented in Table 1 relative to the file system products under test. Note that this list reflects the current testing bias. Future activity will stress the interaction of the disk-based SAN technologies with a broad range of other storage functions such as Hierarchical Storage Management (HSM) software, backup software and magnetic tape devices.

2.2 Product Selection

The initial focus has been on researching and testing currently available third-party SAN file systems. Although on the surface the market appears rich with SAN file system offerings, only four products currently are ready for evaluation that meet the Center's criteria and configuration restrictions. They are listed in Table 2.

Table 1 - Requirements Drivers

Item	Parameters
1	Shared concurrent reading and writing of a single file
2	High performance throughput for a wide range of file sizes, with an emphasis on small files
3	Appropriate locking mechanisms at file and sub-file level
4	Sustainable client bandwidth ranging from 500 megabytes/sec to 1 gigabyte/sec
5	High aggregate bandwidth through entire fabric (effectively equal to the number of clients times the desired per-client bandwidth)
6	Low latency for data access
7	Scaling in terms of number of clients, amount of storage, metadata management and maximum number of files supported
8	Transparent integration of file system into existing systems, allowing ease of use
9	Existing user base with support for a variety of common applications
10	Heterogeneous mix of operating systems
11	Ability to serve clients not directly attached to the SAN fabric
12	Additional file system functionality such as executable support, ability to use file system to boot from, etc.
13	SAN volume management features
14	HSM support
15	Backup support
16	Comprehensive set of administrative tools for configuration, monitoring and troubleshooting, allowing ease of maintainability and operation
17	Full range of security features
18	Highly available and high-integrity overall operation

Table 2 - SAN File System Products

Product	Developer
CentraVision™ File System (CVFS)	MountainGate Imaging Corporation/ Advanced Digital Information Corporation (ADIC)
SANergy™	Mercury Computer Systems, Inc./ Tivoli Systems
DataPlow™ SAN File System (SFS)	DataPlow, Inc.
Global File System (GFS)	University of Minnesota with support from NASA, the Department of Defense and several corporations.

A separate initiative is evaluating SGI's Clustered SAN Filesystem (CXFS™). Note too that the market is already experiencing consolidation as evidenced by ADIC's acquisition of MountainGate, Tivoli's acquisition of the SANergy unit of Mercury, and Hewlett® Packard's acquisition of Transoft Networks, Inc.

Selection for this round of testing was based on a combination of factors. The primary criteria used were:

- Architectural diversity and technical approach.
- Support for heterogeneous clients running the most recent versions of target operating systems with emphasis on the latest versions of IRIX™.
- Existence of a product roadmap noting client operating support plans and addressing operational issues.

Given the overall excitement about SAN technologies and the projected growth of the market [1], other products will warrant evaluation as they mature. Candidates include the Concurrent Data Networking Architecture™ (CDNA)™ by DataDirect Networks, Inc. and FibreNet by Transoft Networks. Also under review are products from the VERITAS® Software Corporation and the EMC Corporation.

2.3 Testbed Configuration

As a starting point for the testing, the Center established an environment that includes two SGI Origin2000s and two dual controller SGI RAID systems (over 1 terabyte of raw storage) interconnected via two 16-port switches: one Storage Technology Corporation unit (Brocade Communication Systems, Inc., OEM) and the one Brocade unit (reference Figure 2).

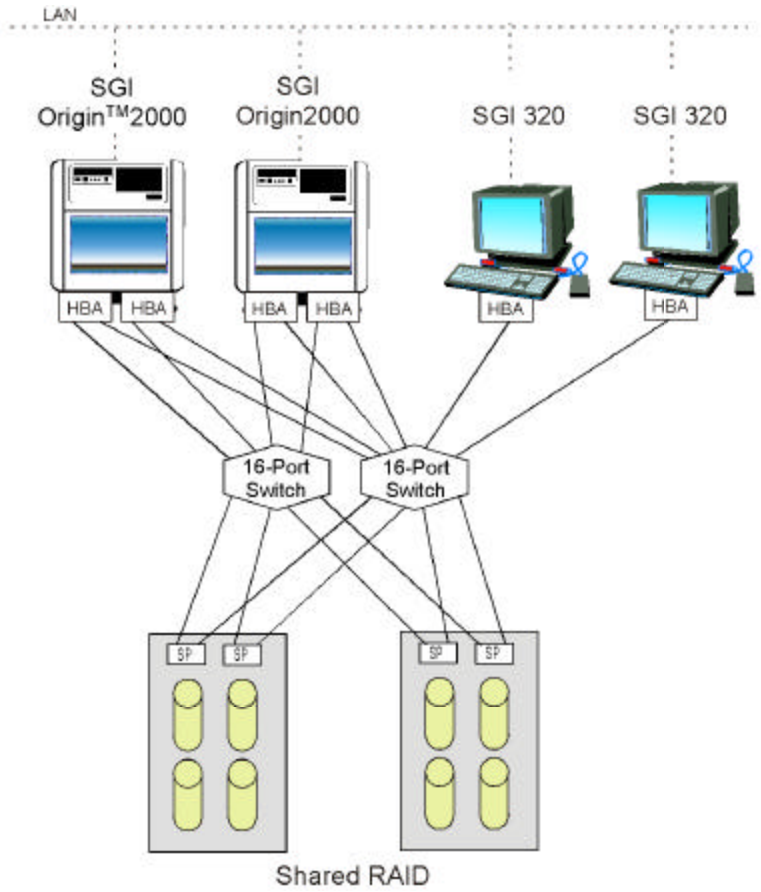

Figure 2. SAN Research Testbed Configuration

Each SGI Origin2000 has a pair of dual channel Prisa host bus adapters (HBA) for connectivity to the switch fabric. Two SGI 320 NT systems also are included for those file system products dependent upon a separate, NT-based metadata controller. They also facilitate heterogeneous SAN client testing. One of the SGI 320s uses an Emulex HBA; the other uses a Qlogic card. Both SGI 320s can, as an option, be booted under Linux. Low-bandwidth communication between the various computers is via traditional 100BASE-T LAN technology. Overall connectivity is flexible and changeable to support the testing requirements as they evolve.

Each RAID system (two total) is configured with four 8+1 RAID 3 logical units (LUN), with two LUNs assigned to each controller. Sustainable bandwidth peaks at 75 megabytes/sec per LUN. Configured usable storage is 576 gigabytes with some disks left unbound.

Table 3 provides a list of the key components with respective product numbers.

Table 3 – Research Testbed Hardware and Software Components

Vendor	Component
Origin™2000	IRIX Operating System
	Prisa NetFX-XIO64 HBA
SGI 320	Windows NT
	Red Hat™ Linux
	Emulex LP7000 HBA
	Qlogic 2200F
Storage Technology Corporation	Fibre Channel Switch 4000
Brocade Communication Systems, Inc.	SilkWorm® 2800
SGI RAID	SP THOR Disk Controller
	9GB Barracuda (ST1917FC)

2.4 Test Planning

The test planning is being shaped by the following objectives:

- Characterize the performance of the individual SAN file system products as a function of file access demands including the ability to stripe files across HBAs, switches and storage elements.
- Explore hot spots and scalability of the products as a function of load and file system fragmentation.
- Compare the performance of SAN file systems to the native file system and traditional file sharing techniques.
- Evaluate operational attributes of the different SAN configurations with respect to administration, availability and maintenance.
- Investigate mechanisms for serving SAN-based data to clients indirectly attached to the fabric via a server (such as NFS).

The projected outcome of the SAN testing is a qualitative and quantitative critique of the products under review measured against the requirements drivers outlined in Section 2.1.

The experiments are being conducted over a range of operating conditions. The test cases envisioned range from the simplest of constructs—single channel writes and reads from a single Origin2000—to multi-channel, multi-client mix load scenarios. In some cases the tests purposely overextend the capability of the system in order to assess the functionality and performance during saturation or when limited bandwidth is forced to be allocated across several active client channels.

2.4.1 Qualitative Testing

Qualitative review will consider the predictable list of product attributes. Of interest is:

- Quality of the documentation
- Ease of installation and configuration
- Ease of use
- Availability of administrative tools for monitoring and troubleshooting
- Transparency to user
- Fault tolerance
- Diagnostic capabilities
- Security features
- Volume management features
- File locking capabilities

2.4.2 Quantitative Testing

Quantitative testing on the other hand will be more performance oriented and is focused on calibrating two fundamental characteristics of the SAN file systems: metadata management and file system throughput as a function of load. The tests are being designed to present stressful yet operational-like conditions. Where possible, industry recognized benchmarks will be used. Several variables, many of which interact, will likely affect the performance of the different products. Most important perhaps are those that are administrator definable when building and instantiating a given file system. Given that the number and type of client access patterns will vary greatly by installation, it is critical to understand how and whether a file system can be tuned to optimally handle the expected workload. Adjustable parameters typically include the following:

- Record (block) size or the subdivision of file
- Stripe width or the size of the data block written to a given logical (or physical) disk in a group of disks that compose a file system
- Mapping of logical (or physical) disks to RAID controllers and HBAs.

2.4.2.1 Metadata Management

The metadata management tests are being designed to measure the number and type of metadata operations that can be accommodated in a given time for single and multiple-client scenarios. This is critical given the assumption that a single, common file system is responsible for data flow in a SAN with potentially a large number of users. The key issue is whether there are any hard scaling limitations in terms of number of clients or number of files. Also important is determining under what conditions latency becomes unacceptable from an access-to-first-byte perspective.

These tests are coming from two sources. One source is project-specific scripts run from single, isolated clients and/or from multiple clients concurrently. The scripts will initiate a large number of metadata-related operations without the associated data I/O while calculating the time per operation. Examples of metadata operations include:

- File open/close
- Get/set file attributes
- Create/delete file
- Rename file
- Make/delete directory

Third party benchmarks are also being considered as the second source. For instance, PostMark, a benchmark by Network Appliance, Inc., is a candidate. It is publicly available at

http://www.netapp.com/

2.4.2.2 File System Throughput

Throughput tests are being developed to measure sustainable transfer rates as a function of number of clients and access patterns, both directly to clients on the SAN, and also to clients not directly attached to the SAN fabric. A mix of test programs will be used, some publicly available, such as SGI's lmdd, while others will be simple C programs written specifically for this project. Also being considered is taskMaster, vxbench and lmbench. taskMaster is useful for simultaneously running variants of the same command on multiple computers.It is available on the GFS website:

http://www.globalfilesystem.org/

vxbench, developed by the VERITAS Software Corporation, provides for multi-threaded testing. Lmbench is a performance analysis tool distributed by BitMover, Inc., at:

http://www.bitmover.com/lmbench

Data will be gathered to measure the behavior of the file systems under normal conditions as well as stress in the midst of allocates, de-allocates, reads and writes, and fragmentation. The method for exercising a file system is multi-step:

1. Measure data transfer rates for a small subset of file sizes, transfer sizes, and access patterns using nominal file system build parameters. Repeat the test while adjusting the build parameters until an optimum performance point is determined.
2. Once the optimum build parameters are set, exercise the file system for individual and multiple clients by initiating:
    a. Single client, single process operations using different file and host block sizes for both reads and writes, sequential and random.
    b. Single client, multiple process operations to either the same or different files, for a predetermined subset of file and host block sizes for sequential versus random accesses, read contention and write contention, and the classic single writer, multiple readers.
    c. Multi-client operations running the same basic script against the same or different files for a predetermined subset of file and host block sizes for sequential versus random accesses.

3. Execute a final set of tests to determine the benefit of configuring multiple file systems with different build parameters as a method to increase total SAN throughput in mixed workloads.

3 SAN File Systems Overview

The SAN file system products being evaluated share certain fundamental characteristics that under optimal conditions tend to even out their performance. The objective of all the SAN file systems, at least from the Center's perspective, is to eliminate file servers between clients and storage with minimum or no impact to the controlling applications. Control information is typically separated from data traffic and in some architectures the two are isolated on completely separate networks. Clients have connectivity to storage via a switch fabric layer that provides the performance of directly attached disks. This allows data to be transferred at relatively high percentages of peak fibre channel bandwidth (100 megabytes/sec per link). All the approaches under test permit multiple HBAs per SAN client, increasing the potential bandwidth per client to a multiple of the base fibre channel rate. Also, the file systems are typically exportable, providing access to SAN resident data by clients that are not directly connected to the SAN switch fabric. Figure 3 depicts generic SAN data and control flow. The diagram shows the fundamental transactions that usually occur—exchange of metadata between requesting SAN client and a third-party metadata manager followed by the data transfer between the client and shared storage via the fibre channel fabric.

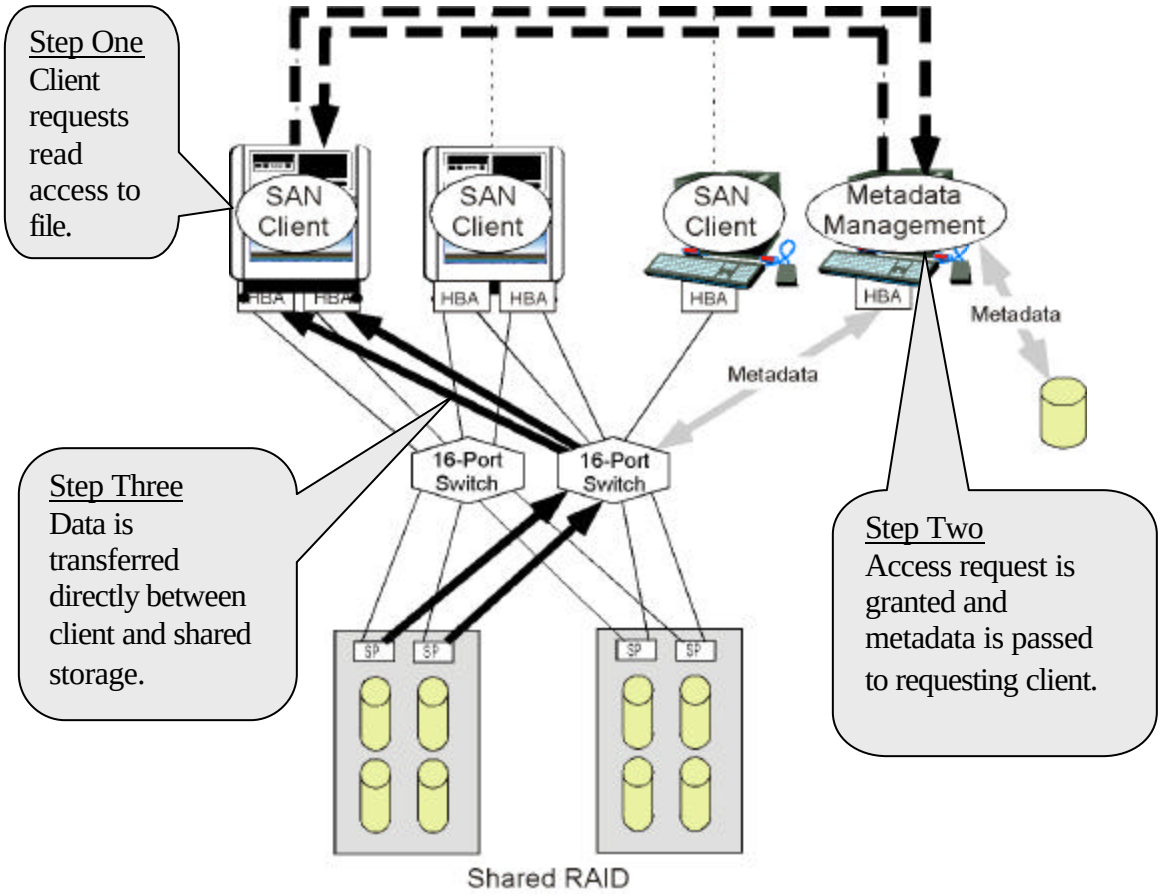

Figure 3. Generic File System Data and Control Flow

Differences in the products show up in two primary aspects of the designs. The first aspect is the approach taken to deal with the file system metadata both in terms of where it is stored (locally or on the SAN) and whether it is centralized or distributed. The metadata design has direct effects on performance, scaling and availability. The second aspect is the relation of the SAN client software to the host operating system. How client software is positioned in the software stack impacts performance and also ties directly to the ease of porting it to new revisions and/or to different operating systems. Table 4 summarizes the key attributes of the products being tested. Subsequent sections elaborate on the overall design approach of each

Table 4 - Product Summary of Key Attributes

Product	SAN File System Design	Metadata Management	Supported Operating Systems
CentraVision File System	Proprietary	Centralized	IRIX 6.2 to 6.5 NT 4.0
SANergy	Proprietary	Centralized	IRIX (all current releases) Solaris (all current releases) Mac 8.0+ NT 4.0 AIX (all current releases) Compaq Tru64 UNIX™ (all current releases)
DataPlow SFS	Proprietary	Centralized/ Distributed	IRIX 6.2, 6.3, 6.5 Solaris 7 and 8
GFS	Open Source	Distributed	Linux

GFS is notably not heterogeneous but inclusion is warranted given the current popularity of the open source model of software development. To date, CVFS and SANergy have been installed and initial testing has started.

3.1 CVFS (Version 1.3.8)

CVFS is a distributed file system designed specifically [2] for fibre channel and SAN technology. CVFS provides sharing of common network storage across multiple heterogeneous systems. The CVFS file system is a hybrid implementation transferring data directly between fabric-attached storage and the SAN client's application, while using TCP/IP transports under a client/server model for control and metadata. CVFS is designed for sequential bulk-data file transfers (megabyte or greater) that are typically streamed into an application. This exploits the read-ahead capabilities and serial nature of the I/O schema. Performance equals or surpasses that of the local file system for well-formed I/O.

The key element of the CVFS is the File System Services (FSS). The FSS is a user-level application that acts as a server for the file system clients. It is responsible for the file system's name space, file allocation, bandwidth management, virtual file management and configuration. The FSS is a POSIX compliant (IEEE Std 1003.1-1990), multi-threaded application that runs on either an IRIX or NT-based host. SAN clients

communicate with the FSS for allocates, reads, writes, etc. over a typical LAN to obtain access to SAN-resident data in a fashion similar to interchanges with the local operating system. Once acknowledged, file extents are passed from the FSS to the requesting client via the LAN, then data is transferred directly between the client and the shared storage via the fibre channel fabric. All communication packets between the FSS and its clients conform to network endian with 64-bit extensions. It does not need to run on a workstation that is physically connected to Fibre Channel fabric because it communicates with the clients via TCP/IP sockets. Metadata is stored using the FSS host's native file system and local system disk. Note also that the FSS host also can be a SAN client.

On the client side, CVFS is written as a file system driver operating at the kernel level in order to transparently attach CVFS managed storage to the client operating system. In IRIX, this is the Virtual File System (VFS) layer; in Windows NT, it is the File System Driver (FSD) layer. Each port provides a completely native interface and is written specifically for the candidate platform. The remainder of client software, however, provides for significant code re-use. Each client operates as if it is directly attached to local storage. The data resides on the managed storage in CentraVision file format. In general, the stored data format can be considered raw data. CVFS uses 64-bit "containers" and accommodates both "big-endian" and "small-endian" file structures. CVFS looks like a local file system with utilities such as cvfsck to check the file system for consistency. Currently, CVFS mounts the NT file system as a network drive. However, in a forthcoming release, the NT version will have a local drive implementation. On IRIX, it currently appears as a local-drive. The final result is that all clients (no matter what platform) perceive the data as native.

Several administrative decisions that directly impact performance must be made when building a CVFS file system:

- Disks (LUNs) are specifically labeled as CVFS entities.
- Disks (LUNs) are assigned to Stripe Groups. This assignment allows for increasing both the bandwidth and storage depth of a given file system.
- Block size and Stripe Group Breadth are adjustable, permitting tuning of the file system versus the application/user access patterns.
- Affinities can be established so that specific files can be stored in the most performance favorable fashion.

Another important operational consideration is CVFS behavior in the event of failures. When a client fails, transactions by the client in transit are accepted into the FSS and are committed to the metadata files. All connections are then cleaned-up with the failed client. When the client re-establishes contact, the client's picture of the SAN is re-established through normal system recovery operations. To the user and to the file system there are no apparent seams to the FSS picture other than the possible transactions lost on the client (that didn't make it to the server) during the failure.

Currently, FSS switchover to a redundant server is a manual operation. However, the release of a more resilient version is imminent. The new FSS design requires that the metadata be placed on a shared storage device, either the SAN itself or any device

accessible by at least two servers. Also in the new version, the FSS becomes a journaled file system. This feature provides for hard-crash integrity and very rapid recovery time. Any platform that supports the FSS can be a participant in the fault tolerant configuration. NT and IRIX servers can freely exchange server responsibilities. When a primary and one or more secondary FSSs are configured, the secondary FSSs are poised to take over the service. They are fully operational and have complete access to file system metadata including in-process I/O transactions. If the primary fails, a vote is executed to determine which secondary can take over. There are two ways the vote is stimulated:

- Lack of response from the primary server—if a client or administrator tries to access the FSS and it is does not respond.
- No update to the Arbitration Control Block on the shared metadata Stripe Group – a running FSS must update its respective "heart-beat" block on the metadata Stripe Group.

For additional information regarding CVFS refer to
http://www.centravision.com/

3.2 SANergy (Version 1.6)

SANergy is a hybrid of conventional networking and direct attached storage [3]. Now patented, it is an operating system extension built on standard system interfaces. SANergy fully supports the user interface, management, access control, and security features of the native host file systems, providing all the file system management, access control and security expected in a network. SANergy clients can be heterogeneous with data being freely shared by all clients attached to shared storage.

SANergy operations center around the Metadata Controller (MDC) that provides centralized metadata management. The Version 1.6 SANergy MDC is based on a Windows NT environment and the NT File System (NTFS). NTFS inherently provides key features such as security, transaction logging and journaling. SANergy intercepts data transactions, then separates and accelerates them using high-bandwidth transports typically fibre channel. Metadata is intertwined with the real data on the shared storage system. Hence, metadata traffic is mixed with data transfers through the switch fabric. The metadata is exchanged between the MDC and SAN clients using standard LAN technologies. NFS is a UNIX client requirement necessitating the NT-based MDC to run an NFS server application. CIFS is used to communicate with NT clients. When a file operation is requested by a SAN client, extent information is retrieved from the appropriate NTFS volume and is passed back to the requester via the MDC. SANergy supports locking primitives down to the byte level with coordination provided by the MDC.

On the client side, SANergy acts as a layered filter driver. It sits on top of the file system(s) either handling an I/O request directly, or passing it on to its natural path, or both. The code is kernel/driver code and is loaded like any other device driver. Since it is wrapped around the primary drivers supplied by the operating systems, SANergy's exposure to any major systems internal change is minimized. Clients have no

prerequisite knowledge of NTFS. Rather, all they need is the block location and order, information that is provided by the MDC. Ultimately data is delivered in a format acceptable to and usable by any application built for cross platform environments.

When building a SANergy file system several operational considerations are worthy of note:

- Disks (LUNs/volumes) are labeled, partitioned and formatted as NTFS file systems using the NT Disk Administrator, a process that writes over any disk resident file and/or configuration information. The MDC must be connected to the switch fabric regardless of whether it is also participating as a SAN client.
- Disks (LUNs) can be assigned to Stripe Sets that allows for both increasing the bandwidth and storage depth of a particular file system. Stripe size is fixed at 64KB.
- NTFS supports multiple partitions (file systems) per volume.
- File record size is adjustable, permitting tuning of the file system versus the application/user access patterns.

The SANergy architecture is flexible in that the MDC can also be an active SAN client. Perhaps the biggest differentiator for SANergy however is the range of supported SAN client operating systems as noted in Table 4. Also, a new version of SANergy (2.0) recently has been released. It supports failover, a critical requirement in operational environments, and also a Sun UFS-based version of the MDC. Failover is handled by an additional product called XA. Any machine running SANergy software also can run the XA software with any XA machine watching any number of MDCs. Should one fail, it will become the MDC for whatever volumes that were owned by the failed machine. Plus, it will send "remap" messages to other SANergy clients (with or without XA software) to remap any mapped shares to the new MDC. The new Sun MDC reportedly provides the key features of the NT version while improving greatly on the striping options allowed when establishing the SAN file system. Although SANergy is most powerful in large file applications, a version is being developed that will be more amenable to small file applications.

For more information, refer to the SANergy web site at

http://www.sanergy.com/

3.3 DataPlow (Version 1.2)

The DataPlow SAN File System (SFS) is a distributed file system with full operating system integration. A key design feature of DataPlow SFS is the separation of metadata into two fundamental components – the higher level namespace-oriented information managed by a metadata server and the more detailed, extent-level data stored directly on the shared disks. File operations require a SAN client to communicate with the metadata server to obtain the location of the more fine-grained information that the client reads directly from the shared storage. In order to facilitate heterogeneous environments, SFS software stores metadata on the server and shared disks in a format that is operating system independent.

The metadata server can be hosted by any one of the SAN clients or it can be free standing. In either case, all SAN clients must have TCP/IP connectivity with the metadata server. SFS clients are able to share SAN file data with LAN and WAN-based clients of any platform through use of traditional protocols such as NFS, CIFS, and HTTP.

If configured for high-availability, metadata server functionality can failover to a secondary server should the primary fail. Just as critical, the failure of an individual SFS client should not harmfully affect the entire SAN. The metadata server simply disconnects the client and releases locks held by the client. Traditional techniques (journaling, file system utilities, etc.) help ensure overall data integrity.

Several administrative options are available when building an SFS file system:

- SFS is able to utilize various commercial volume managers. This flexibility permits numerous striping and mirroring configurations that accommodate a wide range of bandwidth, scalability, cost, and availability requirements. Volume managers that support multiple operating system platforms can be used in conjunction with SFS software to enable heterogeneous file sharing.
- File system block size is adjustable. The block size parameter is used when tuning for small files and reduced fragmentation.
- File systems may be partitioned into several segments in order to exploit parallelism during block allocation and de-allocation. Depending upon the physical device configuration, segmentation further enhances parallelism during data transfers. Segmentation is hidden from users and applications.

DataPlow SFS supports common operations such as synchronous and asynchronous buffered I/O. Additionally, SFS provides support for direct I/O, a caching policy that bypasses the system buffer cache in order to achieve near raw performance. SFS invokes direct I/O either after an explicit system call request by the user application or automatically once file request sizes reach a predetermined size.

Currently, SFS operates in IRIX and Solaris environments. Additional client implementations are in development. Also in development are HSM interfaces such as DMAPI to improve backups, restores, etc.

For additional information refer to

http://www.dataplow.com/

3.4 GFS (Antimatter Anteater)

GFS is a distributed file system based on shared, network-attached storage [4]. GFS is built on the premise that a shared disk file system must exist within the context of a cluster infrastructure of some kind for proper error handling and recovery and for the best performance. SAN clients service only local file system requests and act as file managers for their own requests; storage devices serve data directly to clients. GFS uses callbacks from clients requesting data held exclusively by another client, so that the client holding the data exclusively releases it some time after the request. This implies direct client-to-

client communication. Overall the design permits aggressive metadata and data caching resulting in GFS performance being on a par with local Linux file systems like ext2fs.

GFS provides transparent parallel access to storage devices while maintaining standard UNIX file system semantics—user applications still see only a single logical device via the standard *open, close, read, write* and *fcntl*. This transparency is important for ease of use and portability. However, GFS allows some user control of file placement on physical storage devices based on the appropriate attributes required such as bandwidth, capacity, or redundancy.

The GFS structure and internal algorithms differ from traditional file systems, emphasizing sharing and connectivity in addition to caching. Unlike local file systems, GFS distributes file system resources, including metadata, across the entire storage subsystem, allowing simultaneous access from multiple machines. *Device Locks* are mechanisms used by GFS to facilitate mutual exclusion of file system metadata [5]. They also are used to help maintain the coherence of the metadata when it is cached by several clients. The locks are implemented on the storage devices (disks) and accessed with the SCSI device lock command, *Dloc*k. The Dlock command is independent of all other SCSI commands, so devices supporting the locks have no awareness of the nature of the resource that is locked. The file system provides a mapping between files and Dlocks.

To allow recovery from failures, each GFS machine writes to its own journal. When a GFS machine modifies metadata, this is recorded as a single transaction in that machine's journal. If it fails, other machines notice that its locks have timed out, and one of the other machines replay the failed machine's logs and re-boots the failed machine. Other machines in the GFS cluster can keep accessing the file system as long as they do not need any metadata in the failed client's journal.

As an alternative to disk-based locks, GFS also can use a lock daemon running on any machine accessible to the GFS cluster over IP. Hence, special SCSI disks with DLOCK firmware are not required to run GFS. GFS can also be run without locks as a local file system. Lastly, lock handling has been modularized so that GFS can use almost any globally accessible lock table. This positions GFS to exploit the coming developments in Linux clustering, where highly scalable clusters will be available (to thousands of nodes) with fully recoverable, distributed lock manager technology.

Currently GFS is only operational in a Linux environment. An open source operating system, such as Linux, is ideal for developing the new kernel code required to implement the GFS constructs [6], [7]. However, development of other UNIX variants is likely in the future, including FreeBSD and IRIX.

For additional information on GFS refer to

http://www.globalfilesystem.org/

4 Initial Observations

Testing to date has dealt largely with establishing the basic functionality of the SAN environment and understanding the nuances introduced by the switch fabric environment. Some key activities have included:

- Learning the capabilities and restrictions of the "plug and play" functionality of fibre chanel switches, HBAs and storage devices.
- Establishing the most advantageous RAID configuration with the objective being to maximize the disk throughput available to the various file systems.
- Determining proper procedures for sequencing equipment on-line to ensure that the fabric is operational.
- Using the information available from the fibre channel switches to manage and monitor the fabric activity and status.

Time also has been spent investigating the benchmarking products commonly available for the various areas of quantitative testing to be carried out. By using standard benchmarking products, results can be presented in a way allowing comparison with other industry-sanctioned testing and evaluation efforts.

The CentraVision File System and SANergy have been installed on the testbed and preliminary experiments have been conducted. CVFS has been exercised hosting the FSS both on the SGI IRIX and Windows NT computers. SANergy has been tested exclusively with a Windows NT-based MDC. Performance testing of simple read/write operations has yielded similar results with both CVFS and SANergy delivering a relatively high percentage of peak bandwidth for large sequential file operations. Additionally both seem to operate as advertised and data sharing across heterogeneous platforms works as evidenced by a rather simple test of exchanging a PDF file. More extensive testing is required and planned, as detailed earlier.

5 Future Testing

Testing beyond the initial configuration and file system products is already being planned. A greater emphasis on archiving and backup technologies is envisioned. Items currently being considered are:

- Additional/different SAN file systems. Notably absent from the discussion are offerings from some of the more prominent companies in the storage and networking industry, specifically the VERITAS Software Corporation and the EMC Corporation. Developments by these and other companies are being monitored for possible inclusion in future testing.
- Additional/different client hardware and operating systems.
- Additional/different disk storage devices.
- Additional fibre channel switch devices.
- Data flow to/from tape systems attached to the switch fabric.

Future activities will rely in part on an expanded test environment. Several technologies – hardware and software – are under consideration.

6 Test Results

Given the continuing and evolving nature of this research effort, a web site has been established to deliver a variety of timely information on-line at

http://www.patuxent-tech.com/SANresearch

It will provide operational reviews of each of the products under test including a pro/con style evaluation as well as any future evaluations that are planned. Also available will be relevant vendor comments regarding the evaluations in addition to public domain plans for future product feature sets especially as they pertain to any noted shortcomings. Market impressions and links to relevant websites also will be provided.

7 Acknowledgments

The authors would like to thank several individuals who provided technical content for the write-ups describing the SAN file systems currently in evaluation. They are:

- MountainGate Imaging Corporation/Advanced Digital Information Corporation
  - Brad Kline, Software Architect
- Mercury Computer Systems, Inc./Tivoli Systems
  - Chris Stakutis, VP-Engineering/ CTO
- DataPlow, Inc.
  - Steve Soltis, CEO

Appreciation is also extended to Matt O'Keefe, University of Minnesota, for his GFS insight and overall support in the editorial process.

References

[1] Robert C. Gray. The 1999 Outlook for Storage Area Networks. Presented at Getting Connected '99 hosted by the Storage Network Industry Association Board. May 19 - 20, 1999.

[2] Brad Kline, Pete Lawthers. CVFS: A Distributed File System Architecture for the Film and Video Industry, a MountainGate White Paper, June, 1999. MountainGate Imaging Systems, Inc. and Prestant Technology, Inc., http://www.centravision.com/

[3] SANergy Users Guide. GS-06-12 7/16/99. Mercury Computer Systems, Inc. Chelmsford, MA 01824-2820.

[4] Kenneth Preslan *et al*., Implementing Journaling in a Linux Shared Disk File System. In *The Eighth Goddard Conference on Mass Storage Systems and Technologies in cooperation with the Seventeenth IEEE Symposium on Mass Storage System*s, College Park, Maryland, March 2000, (these proceedings) http://www.globalfilesystem.org/Pages/gfspapers.html

[5] Kenneth Preslan *et al*. SCSI Device Locks Version 0.9.5. In *The Eighth Goddard Conference on Mass Storage Systems and Technologies in cooperation with the Seventeenth IEEE Symposium on Mass Storage System*s, College Park, Maryland, March 2000, (these proceedings)

http://www.globalfilesystem.org/Pages/dlock.html

[6] M. Beck, H. Bohme, M. Dziadzka, U. Kunitz, R. Magnus, and D. Verworner. *Linux Kernel Internal*s Addison-Wesley, second edition, 1998.

[7] Alessandro Rubini. *Linux Device Driver*s. O'Reilly & Associates, 1998.