



VERITAS' Shared File Foundation

Research Note

David Freund
20 September 2004

The main attraction—and, frankly, the original promise—of storage networking is the ability to connect multiple systems to a common pool of storage. Despite many attempts, shared information access has remained frustratingly “just over the horizon.” But the wait now seems genuinely close to ending. A variety of shared file systems has emerged from major vendors, startups, and the open source community. This report—the second in a series—profiles one of these: the VERITAS Storage Foundation Cluster File System (SFCFS).

VERITAS' venerable volume manager and file system have been a major part of the commercial-Unix landscape seemingly forever. The company's clustering products have gained significant ground over the past couple of years—especially in mixed-architecture shops. While VERITAS products cannot mix architectures¹ within a single cluster, the fact that those products work the same way on servers from multiple vendors means that customers don't need to train their IT staffs on multiple clustering products—saving operational costs.

That combination has placed VERITAS' cluster file system as the leading time-tested, commercial-grade shared file system available on multiple OS environments. With the work already done in the company's “portable data containers,” a feature that allows volumes to be moved across architectures (that is, to be sequentially shared), the groundwork has largely been laid for VERITAS to turn SFCFS into a fully multi-platform, concurrent data sharing environment. Doing so would not only help VERITAS retain its position as the *de facto* commercial Unix file system, it would bring it one step closer to a position as the *de facto* shared enterprise file system.



Copyright © 2004
Illuminata, Inc. Licensed to
Veritas Software
Corporation for web
posting. Do not reproduce.

1. For example, AIX on pSeries servers and Solaris on SPARC servers cannot be in the same cluster. Additionally, 32-bit systems cannot be in the same cluster with 64-bit systems—even if they are running the same OS on the same processor type.

Cluster File Systems Go Mainstream, Almost

Over the years, storage networking has fulfilled much of its connectivity promise, but little else. Customers, meanwhile, have a growing interest in sharing *information*, not just capacity, among multiple systems. A significant advantage that Network Attached Storage (NAS) has along this line over Storage Area Networks (SAN) is its innate ability to offer access to file data simultaneously to multiple systems, even if they are running different operating system (OS) software.

Although it seems to surprise a great many, sharing direct access to SAN-based data has been possible since the first SAN was built. But the primary means for doing so is based on clustering. Many such products have been developed and sold over the years, with differing designs and target markets, and are heavily used in some environments. But their use falls mainly into two categories: scale-out compute clusters, using large numbers of small servers to solve huge mathematical problems; and high-availability clusters, using a small number (typically 2-4) of large servers with specialized software that automatically restarts applications if the server on which they're running fails. Both uses are specialized and require a high level of sophistication; clustering therefore has not become a seamless, simple, workaday reality for most systems.

A cluster file system (CFS) enables sharing of volumes, file systems, and even individual files among applications on multiple servers as though they were running on a single system. Because most CFS implementations have, however, been limited in their capabilities—while seemingly boundless in their management complexity—the shared file system aspect of clustering has been all but ignored. A few notable exceptions would include HP's OpenVMS, NonStop Kernel, and Tru64 UNIX;² and IBM's Parallel Sysplex; where concurrent, shared file access is a significant feature.

2. HP announced it was retiring Tru64 UNIX soon after acquiring Compaq. HP's promise to port TruCluster to HP-UX is a key reason Tru64 customers have remained loyal to the company.

A typical single-host file system coordinates access to data among multiple applications running concurrently, and maintains "metadata" to keep track of where directories and files are physically located; their ownership and access rights; when they were created, modified, and accessed; and other housekeeping information. A cluster file system extends that same functionality across multiple servers, while keeping application programming interfaces (APIs) largely unchanged.

As the name implies, most cluster file systems were developed as part of larger server-clustering products. The best performers allow direct I/O access to shared disks by each participating server. The easiest to use CFS products are those that present a seamless "single system image,"³ meaning that each server in the cluster "sees" the same file systems, and the same contents, at any point in time—and concurrently shares access to them. VERITAS' CFS has been available for years as part of the company's HA-cluster product set. In fact, it's held the crown as the only HA cluster with a single-system-image CFS that runs on more than one OS platform.

And yet clustering has not become a broad-based technology. Few customers have chosen to cluster solely for sharing access to files, because the rest of the cluster infrastructure—upon which most CFS products depend—tends to be complex, and thus expensive to set up and maintain. Moreover, the top clustering products, including VERITAS', don't provide shared access among servers running different operating systems—and multi-platform datacenters are a permanent reality in the vast majority of organizations.

Time to Drop Old, Complex Baggage

VERITAS' roots are in its Volume Manager (VxVM) and File System (VxFS), which it began OEMing to Unix system vendors back in 1989. As

3. A superset of a single, unified namespace. Not only are file and directory names kept unique, and their accesses managed globally, but the application views of their states and contents are also guaranteed consistent on all cluster nodes.

Unix grew into commercial use, customers desperately needed production-grade storage management—something common among proprietary and fault-tolerant systems at that time, but not a part of Unix' heritage. By providing these crucial facilities on a more timely and economical basis than the OEMs themselves could, VERITAS established its products as staples of the commercial Unix era.

The company continued working, through acquisitions as well as internal development, to enhance its position as the “go to” vendor for storing and protecting data on Unix systems. The company added backup/restore and replication functions to its portfolio, maintaining both a storage-centric and software-only focus. That paid off, enabling VERITAS to outpace competitors and partners in accommodating performance needs unique to relational databases such as Oracle and DB2 as they rose in prominence. The result was the company's “Database Edition” products, specialized versions of the file system that provided nearly raw-disk level performance for databases stored on file-structured volumes.

Because of the company's focus on storage management, VERITAS' approach to clustering began from that perspective: using redundant components to maintain data availability. However, VERITAS was not exactly an early mover in clustering;⁴ it was competing with clusters from server vendors, which had a largely application- and server-centric view of the world.⁵ So VERITAS produced a set of products as individual pieces, and eventually integrated suites, intended to compete with server vendors on their terms. VERITAS Cluster Server (VCS) was introduced in 1998 to watch for machine or application failures, and restart applications on another server when needed. Following common server-vendor design principles, its Volume Manager and

File System were enhanced to operate across a VCS-based cluster, giving birth to the Cluster Volume Manager (CVM) and Cluster File System (CFS). But these components took years to build—an experience common to companies that decide to implement a cluster file system.

Because of its storage focus, the rising popularity of Fibre Channel SANs quickly caught VERITAS' attention. Its cluster designs evolved away from the architectural assumptions common to a server world-view, and embraced the SAN as an *assumed* part of a cluster configuration, giving birth to a bundle of VCS, CVM, and CFS in January 2001 as SANpoint Foundation Suite HA. VERITAS was thus first to enable each clustered Unix server to both read and write files directly over the SAN. As database clustering started coming of age, it launched “Database Edition” products with the CVM, CFS, and tweaks specific to each database.⁶ As its portfolio grew, VERITAS started bundling its products to simplify its product line. However, the individual components remained separately available.

With its launches early this year of its Storage Foundation product suites, VERITAS has made a number of simultaneous shifts in product strategy. For one thing, the company is finally moving away from component-based thinking to one of layered capabilities—albeit very slowly. The basic “Foundation” is now a somewhat-inseparable blending of Volume Manager and File System, and forms the platform upon which additional products are based.⁷ For clustering, the expected product layering to come next would add VCS, forming Storage Foundation / HA (for High Availability).

Here's where VERITAS has made a surprising move: its cluster file system and volume manager

4. Digital Equipment Corporation was shipping commercial VAXclusters in 1984.
5. In 1997, VERITAS signed an agreement with Sun to develop a cluster volume manager for Sun Cluster 2.2 in a deal that would become typical of the simultaneously cooperative and competitive nature of VERITAS' relationships with its partners.

6. At the same time, product names became almost humorously awkward, such as “VERITAS Database Edition / Advanced Cluster for Oracle9i RAC,” which came to be known as DBE/AC for 9iRAC. *Catchy!*
7. VERITAS hasn't quite left its component heritage behind yet. Volume Manager, File System, and Cluster Server remain available separately.

no longer require VCS failover—making it possible for VERITAS to offer a Storage Foundation Cluster File System (SFCFS). Despite its name, the product is not merely the CFS; it's the entire Foundation *plus* the CFS and CVM components. File sharing is, at long last, available from VERITAS without the need to deal with the usual complexity of configuring and managing an application-failover cluster.⁸

VERITAS SFCFS Architecture

True to VERITAS' heritage as a storage-software company, it assumes a SAN is at the heart of modern shared-file infrastructures. All cluster members connect directly to SAN-shared storage devices. SFCFS then provides a single, coherent view of the files and directories to applications regardless of the cluster node upon which each is running.⁹ File I/O—reads and writes—are performed directly to the storage device, and are cached locally. SFCFS includes VERITAS Cluster Server (VCS), Cluster File System (CFS), and the Cluster Volume Manager (CVM). VCS is the first component installed and configured, and provides basic cluster communication, configuration, and membership services. The VCS on each server communicates with others in the cluster via a private Ethernet link.¹⁰

The Cluster Volume Manager (CVM), just like its single-system cousin, creates logical volumes, and makes them available for concurrent access by up to 16 cluster nodes. CVM presents each node with the same logical view of shared device configurations, including changes. VCS informs CVM of any changes in cluster membership; when a node joins a cluster it gains access to shared disks. When a node gracefully leaves a cluster, it loses its access. If a

node fails, other nodes can still access shared volumes without disruption. One node acts as the "master node," coordinating activities such as creating or reconfiguring volumes. Other nodes are called "slave nodes." Any node can be a master node; initially, it's the first node to join the cluster. If the master node leaves the cluster for any reason, one of the slave nodes is chosen as the new master.

CFS uses a "primary/secondary" design to manage file system metadata, which is stored on the same SAN-attached disks as the actual file data. The first node that mounts a file system is designated as its "primary" node; the remaining nodes are "secondaries." As applications create, delete, extend, or otherwise modify file attributes, secondaries send requests to the primary to perform the requisite metadata updates. If a primary's server fails, the remaining cluster nodes elect a new primary.¹¹ To avoid potential bottlenecks, the cluster admin can designate which nodes in a cluster will act as CFS primary nodes, as well as which nodes can take over for a failed primary, to balance the metadata-management (and related I/O) workload.

CFS uses the Group Lock Manager (GLM) to coordinate its actions across the cluster, maintaining single-host Unix file system semantics and behavior. This is crucial for applications—and the survival of an organization's data. For example, when an application writes a stream of data to a file in a non-clustered environment, a subsequent application reading from the same area of the file expects to see the new data, even if it has been cached by the file system and not yet written to physical disk. This behavior must be preserved in a cluster; otherwise, the second application in can end up with stale or partially-updated (meaning corrupt!) data. To maintain single-host write semantics, file-system caches on cluster nodes must be kept "coherent"; each must instantly reflect any updates to cached data, regardless of the node from which they originate. The GLM is used to coordi-

8. VCS is still installed to provide inter-node communications, but system and application service monitoring is no longer configured as part of product setup. A "Storage Foundation Cluster File System / HA" is available that includes VCS failover functions.
9. A.k.a. a single global namespace.
10. For availability and performance, redundant Ethernet links are required. They must also be separate from the "public" LAN, which is used by applications to communicate with entities outside the cluster.

11. Any "in-flight" metadata updates are recorded in the file system intent log. If a new primary election takes place, the new primary reads the intent log, and completes any unfinished transactions.

nate these activities, as well as which applications can access which files, or byte ranges within those files.¹²

Like other SFCFS components, the GLM has a lock “master.” Each file system starts running with one lock master for all objects in that file system. However, if mastering all of those locks were to remain on that node, it might become a performance bottleneck if applications on several cluster nodes were simultaneously creating, deleting, and extending files at a rapid pace. VERITAS has had several years to perfect its lock-management algorithms, and has come up with ways to spread the lock-management load, and even to master locks where they’re most needed; to do this it uses a mechanism similar to “locality of data” caching algorithms. Once a node is interested in an object, its GLM component becomes the “proxy” lock master for that object.¹³

The bane of any cluster engineer’s existence is preventing a pathological condition in clustering known as “Split Brain.” A split brain, or “partitioned” cluster, occurs when a communication failure occurs between cluster nodes, or one or more nodes become too busy to respond to messages from other cluster members. This causes some nodes to have divergent views about which nodes are current, active members of the cluster. If more than one cluster partition continues running in a

split brain scenario, access to data on the shared storage is no longer coordinated, and data becomes at risk of being corrupted. *Not good!*

Just this year, VERITAS improved on its previous “VCS seeding” mechanism to avoid split-brain, replacing it with a technique called “I/O fencing.” Even if all network links fail, if multiple nodes can still access the same disk device (and the device supports SCSI-3 Persistent Reservation), each machine will attempt to “reserve” the device for exclusive write access. If it can, then it can assume a master role for its part of the cluster, and continue operation. If not, it leaves the cluster and stops accessing cluster-shared volumes. Since the storage device itself is enforcing exclusive write access from a single server, others are prevented from altering any data. In the absence of I/O fencing, if a node has no way of determining whether another machine has failed (as opposed to simply not communicating but still running), it automatically stops accessing whatever disks it had been sharing with the allegedly “failed” machine.

VERITAS has, in its current release, brought its CFS and CVM close to complete feature parity with their single-system cousins. A few features, like software RAID-5, can’t be used on cluster-shared volumes. Beyond such minor restrictions, managing SFCFS data can be done with the same techniques an administrator would use on a single server. SFCFS features for protecting application and users’ data include file-system checkpoints and snapshots; volume-level snapshots, mirroring, and replication; and even traditional file backup via any node in the cluster.

SFCFS Pros

VERITAS Storage Foundation Cluster File System provides a platform for users and applications to share and organize data. There are many such options in the market, from NAS filers¹⁴ to “out-of-band” file virtualization software like IBM’s SAN File System. Some of SFCFS’s high points include:

12. Some applications read and write to the same file concurrently. One application, for example, could be appending while others are concurrently reading from various regions. Many CFS implementations lock an entire file, even to perform I/O to a small region. For better performance, VERITAS locks only ranges of bytes that correspond to each I/O request. That way, I/O requests can only conflict with each other—causing one or more to be blocked—if their I/O ranges overlap. SFCFS also adds the notion of an “append lock,” allowing one application to extend a file while others remain free to lock, read, and write other previously-existing parts.
13. Recent internal tests on a 4-node cluster of 1- and 2-processor, midrange RISC/Unix servers had the reconstructing of 3 million locks—considerably more than a typical commercial workload would ever have simultaneously outstanding on a single file system. Electing a new master took about 8 seconds.

Improved Storage Consolidation. As a capable shared file system, SFCFS allows multiple applications on more than one server to access the same physical copy of data. Using multiple servers to solve a given problem or to perform different tasks on a common set of data is often much less expensive than using a single, much-larger server. Without a shared file system, the data must be replicated, copied, or moved (either physically, or over a network). Those alternatives waste storage capacity—but even more important, introduce complexity, management, and coordination costs. Using a shared file system such as SFCFS therefore increases the benefits of the physical storage consolidation that Fibre Channel SAN technology enables. It can also save time, as applications on each clustered server immediately “see” any updates made by other cluster members instead of waiting for the next scheduled asynchronous replication update or snapshot clone.

Multi-platform support. VERITAS has been playing its multi-platform, independent card with increasing success. However, VERITAS has yet to provide simultaneous sharing of data among different OS platforms (every node in the cluster currently needs to be the same make and run the same OS). The closest VERITAS can come doesn’t involve CFS at all. Storage Foundation’s new “Portable Data Containers” (PDC) allow a volume that was created on a Unix system, for example, to be later mounted by a Windows or Linux system. If necessary, Storage Foundation converts file-system metadata for the new platform, an operation that takes a number of minutes (depending on how many files are stored on the volume). Since PDC requires common VERITAS components on any systems sequentially sharing volumes in this

14. An SFS client system can also act as a NAS filer, serving files via CIFS, NFS, NCP, etc. the same way it would serve files stored locally. In fact, multiple systems can be configured as NAS filers for the same files—forming a NAS cluster. SFS is already coordinating access among its clients, so data integrity is assured—so long as no caching is performed by the NAS software itself. Such caching is unnecessary, anyway, because SFS already provides a local cache on each client.

manner, VERITAS has an opportunity to create a truly-unified metadata storage format. Such a format would eliminate the need for conversions—and enable CFS-based sharing of file data among different OS platforms. VERITAS says that, for its part, it sees little demand for heterogeneous file sharing from its customers. But interest in, and use of, SFCFS has been surging. As customer experience and sophistication increase,¹⁵ and especially if other multi-platform products begin to pose a competitive threat, it would be reasonable to assume VERITAS will eventually adopt heterogeneous sharing.

Simple Data Administration and Protection. A shared file system eliminates the need to perform backup operations on every server. A single, separate server can perform that task¹⁶ without consuming any of the other servers’ CPU and I/O resources, keeping application performance unperturbed—something traditional network backup techniques simply cannot accomplish. Because files can be shared, file backups are no longer needed on each individual host. The same is true for other file-maintenance operations such as virus scanning, cleaning up temporary files and directories, etc. Applications running on multiple systems can share a single set of configuration and data files, making multiple instances of an application almost as easy to administer as a single instance.

High performance. Performance has been second only to reliability as priorities for VERITAS’ file system efforts. That focus led to the company’s decision to rely upon a SAN infrastructure for clustered volume connections. That way, CFS engineers could optimize performance for a SAN configuration, as well as bypass the complexities of handling nodes “serving” direct-attached volumes over a network. Data blocks travel directly between the storage device and the requesting-application’s server. This has a distinct performance advantage over NAS file serving in latency-sensitive applica-

15. See Illuminata report “IBM SAN File System” (July 2004).

16. Using traditional backup software, snapshots, and other technologies.

tions such as heavily-loaded relational databases. SFCFS also maintains a local cache on each cluster node for file data, keeping recently-accessed blocks of data¹⁷ in memory. Most applications are not designed to be particularly efficient in handling file I/O, and re-read and rewrite the same part of a file quite frequently. Because each node is sharing direct access to cluster volumes, changes made by any node are immediately reflected in the other nodes' caches.

Database Savvy. In order to give applications the performance boost of caching file data in memory, file systems are effectively forced to guess which parts of an application's files are needed next, based on coarse estimations of overall application behavior. Unlike most applications, however, databases such as Oracle, DB2, and SQL Server maintain their own in-memory caches, based on intimate knowledge of what parts of the database files are needed and when. Any caching done by a file system would be redundant, wasting memory and processor resources—and even I/O bandwidth, since the file system could write modified cache blocks just before the database needs them again, and pre-fetch data the database doesn't need. This overhead is a major reason database vendors have, for several years, recommended using "raw" disks for storing relational data. Eliminating this overhead was the impetus for VERITAS' creation of its "Database Edition" products, now known as Storage Foundation for Databases. Storage Foundation for Databases bypasses as much of the file-system code as possible to give databases such as Oracle or DB2 direct access to the blocks within their files. VERITAS has added these same optimizations to its Cluster File System bundle to form additional products, such as Storage Foundation for Oracle RAC.¹⁸

Maturity. Some may see the term as denigrating, but maturity is a *highly* desirable quality in any

17. And, in some circumstances, via pre-caching, blocks that have not been read by an application yet, but which may be needed in the near future.

18. VERITAS plans support for additional clustered databases in the near future.

commercial file system. After all, code bugs can directly cause customers' data to be damaged or lost! But maturity can also be reflected in the experience of a product's developers, and thus translated back into the product. The SFCFS Group Lock Manager's ability to distribute metadata lock handling across cluster nodes to improve performance is one example. Maintaining reliably-consistent caches among multiple nodes—each of which is accessing the same files and volumes over its own SAN connection—is another.

High-end availability and scalability. SFCFS, and its high-availability SFCFS HA variant (which includes the full VCS), enables a melding of advantages typically considered to be the exclusive domain of either scale-out clustering or high-availability clustering. Customers can still increase application performance by aggregating the power of multiple, smaller machines—and since they can all share access to a common physical set of data, each has the ability to seamlessly take on the workload of another should one of a cluster's servers fail. Few shared file systems are integrated with application failover capabilities as tightly as SFCFS and VCS—if at all.

Additional Storage Automation. VERITAS has built FlashSnap into its SFCFS bundle, enabling administrators to create and use point-in-time copies of a cluster file system. These copies, called "snapshots" or "checkpoints,"¹⁹ can be used for both read-only purposes such as backup, or for read-write operations, such as testing new application code against a copy of the production database—without jeopardizing the actual production database itself, and without taking either the time or space required to make such a copy. The bundle also includes "Intelligent Storage Provisioning," and "Quality of Storage Service" (QoSS) to help admins dynami-

19. VERITAS' snapshots are maintained in the snapping server's memory, and do not survive when the node is shut down or the file system is unmounted.

VERITAS' checkpoints, by contrast, are maintained on-disk, making them available to multiple nodes, and for an unlimited amount of time.

cally grow and allocate their storage capacity among applications.

SFCFS Cons

On the other hand, as is frequently the case with enterprise-class software, SFCFS is not as simple to install and manage as it seems at first glance.

Cluster Config Requirements Remain. Using SFCFS still requires setting up a cluster based on VCS technology. While numerous steps in planning and configuration of traditional VCS clusters, such as setting up failover groups, scripts, etc., have been eliminated, quite a bit remains. For example, admins must connect the hardware to use a public IP-based network, two private networks,²⁰ and a Fibre Channel SAN. They must decide on a name and a unique ID (from 0 to 255) for the cluster, names for the cluster nodes, and get the device names for the NICs connecting each node to the private networks. The I/O Fencing feature, used by SFCFS to avoid “split brain” problems, must be configured manually.²¹ The system clocks on all nodes must be synchronized using some external component, such as a Network Time Protocol daemon.

Metadata Load Balancing Is Manual. System administrators must determine how to balance metadata updates across the cluster, by designating which node is considered “primary” for each file system being mounted. The first node that mounts a file system becomes the primary for that file system; it can be changed afterwards via the command-line.²²

20. If I/O Fencing is used, a public network *can* be used for cluster communications, although VERITAS recommends against doing so. If non-cluster traffic becomes heavy, SFCFS performance would suffer.

21. Avoiding split-brain through I/O fencing requires that at least 3 disks or disk-array LUNs be set aside and dedicated to function as “coordinator disks.”

22. Another subtlety relates to a file-system’s tunable parameters, stored in a file “tunefstab” on each system’s directly-attached boot disk. Administrators must ensure the files are identical on every node, as the entire cluster uses the parameters stored on each file-system’s primary node.

No Windows Support. VERITAS’ relationship with Microsoft has not been nearly as close as with its Unix partners, making it significantly more difficult to build for the Windows platform. As a result, VERITAS’ Windows products have historically lagged behind their Unix brethren. Unfortunately, this remains true today. VERITAS provides volume-management and high-availability failover clustering with Windows versions of CVM and VCS, which are bundled together in Storage Foundation for Windows, but VERITAS’ efforts to field a cluster file system for Windows remain frustrated.

Limited Multi-platform support. VERITAS supports multiple Unix platforms with its cluster file system products, but the firm requirement for all members of a cluster to run the same OS remains. That means that file sharing is limited to a specific platform. That limits the amount of consolidation and efficiency gains available to enterprise data centers, the majority of which continue to depend on multiple platforms. Other shared file systems, such as IBM’s SAN File System, or any of a vast array of NAS products, have an advantage in providing shared access simultaneously to a wide variety of OS platforms. Their tradeoff is that, while good enough for many applications, NAS products can’t match VERITAS’ cluster file system performance in latency-sensitive applications.

Partner Nervousness Over Control Points. IBM, Sun, HP, and even EMC are increasingly VERITAS’ competitors. As each seeks to spread its influence over more of its customers’ IT stack, each is likely to become progressively wary of VERITAS, which is seeking to grow in similar ways. Each company is now articulating its vision of a dynamic, flexible architecture for customers’ information infrastructures. While those visions all sound similar, and actually share the majority of their goals, they tend to conflict significantly in their approaches (each approach naturally favoring the products built or acquired by the presenting vendor). With VERITAS talking about enabling “Utility Now,” and making provisioning, virtualization, and document-archiving acquisitions of its own, the amount of the company’s strategy and product set that competes

with its partners is growing. Those partners, therefore, will tend to minimize their public support for VERITAS' ambitions, keeping it positioned within a file-and-volume management niche as much as possible in the customers' minds. They are, for example, reluctant to sell Storage Foundation for Networks, a version of VxVM embedded in Fibre Channel switches,²³ because it would place VERITAS at the center of a customer's storage fabric, and in control of the very foundation upon which all information architectures are built—storage volumes. Unix-server vendors were willing to cede ownership of volume-management functions to VERITAS on servers, especially since doing so helped push Unix as a viable alternative to mainframes and proprietary minicomputers. The thought of doing so again—for an entire datacenter fabric—is not nearly so attractive.

Conclusion

VERITAS Storage Foundation Cluster File System (SFCFS) is a truly enterprise-class shared file system, providing shared, direct access to SAN-attached file data. The company enjoys a well-deserved position as the volume-management and file-system incumbent in commercial Unix shops. The company has also gained stature for its hardware-agnostic stance, making its products available with the same features and administration tools on multiple platforms. It's a theme that resonates well with commercial customers, most of whom use equipment and operating systems from multiple suppliers. VERITAS has given customers a kind of

23. Currently available for Cisco MDS 9000 switches.

storage-software consolidation. And it has yielded tangible results—money that otherwise would have been dispersed on licenses, personnel and training for products from multiple vendors could instead be concentrated on just one.

VERITAS is building on this momentum, responding to a natural evolution of customer needs as they embrace networked storage: the need to share data as well as capacity among multiple systems. The company has streamlined and simplified its clustering bundles, providing the data-sharing benefits of a cluster file system while removing some of the complexity that has kept Unix clusters out of the commercial-computing mainstream. SFCFS is the result.

VERITAS enjoys a strong multi-platform reputation, and the company will continue playing that card with relish. SFCFS is designed to run on multiple platforms. It cannot, however, handle different platforms within the same cluster—meaning it cannot provide multi-platform shared file access. NAS has so far fit that bill in most cases. However, as customers' networked-information sophistication builds, so will the sophistication of their information-handling product requirements.

As the reigning volume- and file-management champion, VERITAS has the technical prowess necessary to make its Storage Foundation Cluster File System a fully multi-platform, *concurrent* data sharing environment. Like the holder of a heavyweight boxing title, competitors are preparing to fight VERITAS for the championship.