

September 2001
Prepared by:
UNIX Software Division
Compaq Computer Corporation

Contents

High availability for all..... 3
 Tru64 UNIX—designed for
 the rigors of e-business..... 3
 Simplified built-in clustering..... 3
**UNIX file systems—the good,
bad, and indifferent** 4
**Characteristics of stand-
alone UNIX systems** 5
**Taking UNIX file systems
remote**..... 6
 Cluster File System goes
 remote seamlessly 7
Conclusion 13

Cluster File System in Compaq *TruCluster* Server

Extending the advantages of single-system file systems to high availability clusters

Abstract: High availability is traditionally considered a high-priced option, reserved only for an enterprise's most critical information systems. Today, however, nearly all information is critical, making high availability a necessity for enterprises of all sizes.

Clustering provides the necessary high availability, but is typically associated with an increased management burden. Compaq has eliminated this dilemma with the Cluster File System (CFS), an integral feature of Compaq *TruCluster* Server.

This white paper explores the capabilities of common UNIX file systems and presents how CFS overcomes many of their weaknesses by extending the performance, scalability, and management features of single systems across multi-system configurations.

Notice

Cluster File System in Compaq *TruCluster* Server
White Paper prepared by UNIX Software Division

Second Edition (September 2001)

©2001 Compaq Computer Corporation. Printed in the U.S.A.

Compaq, the Compaq logo, AlphaServer, and NonStop are Registered in the U.S. Patent and Trademark Office.

Tru64 and TruCluster are trademarks of Compaq Information Technologies Group, L.P.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation

UNIX is a trademark of The Open Group.

All other product names mentioned herein may be trademarks or registered trademarks of their respective companies.

Compaq Computer Corporation shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is subject to change without notice.

High availability for all

Not long ago, high availability was an expensive option, reserved only for an enterprise's most critical systems—for example, order entry or online transaction processing. But even in those areas, the complexity and management overhead associated with high availability forced some businesses to make risky trade-offs between accepting the complexity that comes with high availability or settling for just reliability.

Times have changed dramatically. Information today is the oxygen that enables an enterprise to thrive—in fact, acquiring, sharing, analyzing, processing, and protecting information is fundamental to an enterprise's very existence. That makes all information systems critical. For example, e-mail is now the backbone for communications among employees, partners, suppliers, and customers. Data warehouses—once a special project for a handful of analysts—now feed the day-to-day information needs of marketing and financial managers throughout an enterprise. And in the realm of e-business, where supply chains, procurement systems, and global e-commerce are pumping information day and night, the need truly rises to the level of mission critical.

The fact is, to succeed in today's new economy, high availability can no longer be an option. And thanks to the advanced capabilities of Compaq *Tru64 UNIX and TruCluster Server*, it doesn't have to be.

Tru64 UNIX—designed for the rigors of e-business

Compaq *Tru64 UNIX* is a new breed of UNIX, offering unprecedented scalability, superior performance, built-in clustering for nearly 100 percent availability, hassle-free management, and masterful innovations—like the fastest Java Virtual Machine in the industry and interoperability with Microsoft Windows NT/2000 that's continually ranked as the best in the business. This is truly an operating system designed for the rigors of e-business.

Tru64 UNIX running on the latest Compaq *AlphaServer* systems provides scalability to multi-terabyte configurations, including up to 16 TB for a single file system and 256 GB or more of memory. It has the performance to support millions of concurrent users, with awesome CPU power, partitioning, and dynamic tuning. Inherent redundancy and hardened reliability significantly reduce management overhead, and Web-based tools keep the job simple when administration is necessary.

Above all this, Compaq has made it very easy to take advantage of clustering by making *TruCluster Server* tightly integrated with the *Tru64 UNIX* operating system. This capability delivers critical high availability, as well as the ability to scale out by clustering multiple *AlphaServer* systems together. Built-in load balancing enables all the systems in the cluster to share the work, effectively increasing processing capacity by the number of servers in the cluster.

The result is, clustered configurations deliver the same high performance and easy management as a single system—with outstanding scalability and a level of availability previously thought unreachable outside a fault-tolerant system.

Simplified built-in clustering

The key to enabling robust, headache-free clustering is the notion of a single system image; that is, all the servers in a cluster are viewed as one system. This is one of the essential features of *TruCluster Server*, making it possible to automatically recognize additional servers as they are added to a cluster, and automatically balancing the load across the cluster.

By providing a single system image, operating system and application software need only be installed once for the entire cluster. Because the cluster operates as a single domain, there's no need to manually configure security privileges or other attributes that must be kept in synch across all servers. And because the cluster appears as a single IP address on the network, clients don't have to connect to a specific system for services. If the system providing a service goes down, client requests are automatically re-directed to another system in the cluster.

At the heart of the single-system image is a fully shared Cluster File System (CFS) that enables all systems in the cluster to share all file systems including common root, /usr, /var file systems, and application data. CFS is built on top of local file systems such as the Advanced File System (AdvFS) in Tru64 UNIX, as well as UFS, CDFS, and others. The result is pure simplicity.

Disks and files are equally accessible from all systems, at speed, concurrently, and without specific preparation—regardless of whether the file is stored on a locally attached disk or on a shared storage fabric. In the event of a failed storage or network controller, a system switches automatically to an alternate controller using multi-path I/O. And if all those paths fail, the system will continue to work by remotely using some other system's storage controller.

This ability to share and manage storage cluster-wide is a revolutionary development. It makes clustering easy. It affords the enterprise extensive flexibility to balance workload as needed without sacrificing performance. And it makes high availability affordable—in fact, Compaq's CFS design can help reduce overall storage costs up to 90%. That's on top of the reduced management overhead.

CFS is based on many well-established principals of UNIX file systems. But it goes further than many local file systems to overcome inherent weaknesses—and to deliver the advanced high-availability capabilities needed for today's e-businesses. Let's take a closer look at where CFS is coming from, and where it is headed.

UNIX file systems—the good, bad, and indifferent

UNIX operating systems generally support a number of local file systems, affording the system administrator a great deal of flexibility in meeting various computing requirements. *Tru64* UNIX, for instance, includes AdvFS as the default file system. AdvFS is a log-based file system that provides flexibility, compatibility, data availability, high performance, and simplified system management, along with the ability to handle files and file sets approaching 16 TB in length.

The configuration of AdvFS differs from the traditional UNIX file system in that the physical storage layer is managed independently of the directory layer. Therefore, system administrators can add and remove storage without unmounting the file system or halting the operating system. As a result, configuration planning is less complicated and more flexible. In addition, AdvFS supports online defragmentation, directory hashing, and other advanced storage techniques. From a user's perspective, AdvFS behaves like any other UNIX file system, while delivering higher performance and greater availability.

While AdvFS offers many unique and desirable advantages, *Tru64* UNIX also supports other common UNIX file systems, including the traditional UNIX File System (UFS), Network File System (NFS), Compact Disk File System (CDFS), DVD File System (DVDIFS), and Memory File System (MFS).

Figure 1 below illustrates the file system architecture of *Tru64* UNIX.

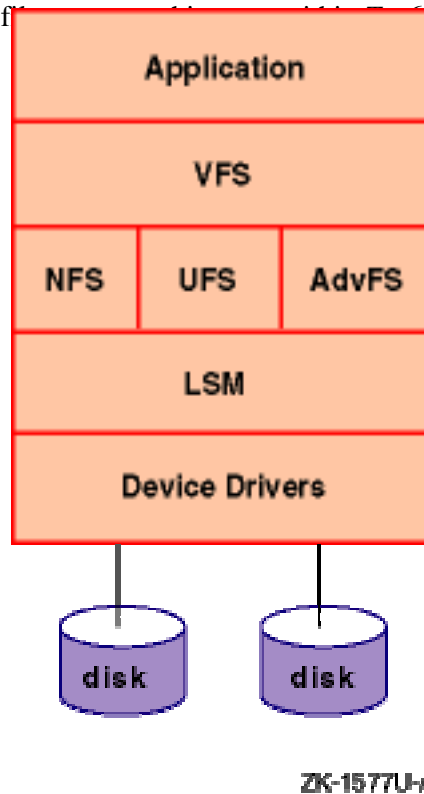


Figure 1

At one time, UFS was the primary file system used by UNIX. The *Tru64* UNIX implementation of UFS is compatible with the Berkeley 4.3 Tahoe release and supports file sizes that exceed 2 GB. UFS also supports file block clustering, thereby producing sequential read and write access that is equivalent to the raw device speed of the mass storage device.

NFS is a facility for sharing files in a heterogeneous environment of processors, operating systems, and networks. It enables this capability by mounting a remote file system or directory on a local system and then reading or writing the files as though they were local. *Tru64* UNIX supports NFS Version 3, as well as NFS Version 2, both fully tested for interoperability.

CDFS is another local file system supported by *Tru64* UNIX for reading ISO9660 CD-ROMs, and DVDFS is supported for reading DVD media.

MFS is basically a UNIX file system that resides in memory. No permanent file structures or data are written to disk, so the contents of a MFS are lost on system reboots, unmountings, or power failures. MFS, however, is a very fast file system, making it useful for storing temporary files or read-only files.

Characteristics of stand-alone UNIX systems

UNIX systems generally tie together these multiple file systems using a Virtual File System (VFS), which in *Tru64* UNIX, is based on the Berkeley 4.3 Reno virtual file system. VFS presents a uniform interface to users and applications, enabling common access to files regardless of the file system on which they reside. As a result, file access across different file systems is transparent to the user.

By keeping file access transparent to users and applications, the VFS allows modularity among multiple file systems, so new or different file systems can be incorporated as needed without affecting applications and users.

UNIX systems also rely heavily on caching to deliver high performance. The file systems (*i.e.*, AdvFS, UFS, NFS, CDFS, DVDFS) used by *Tru64* UNIX incorporate a Unified Buffer Cache (UBC), which interacts with the virtual memory system to dynamically adjust the amount of physical memory being used to cache file data. Caching, however, can be a weakness for certain applications like databases, where the cache becomes overhead and actually degrades performance rather than improving it. Compaq overcomes this problem with direct I/O. Direct I/O bypasses the UBC, allowing unbuffered I/O.

To enhance performance and flexibility further, *Tru64* UNIX also incorporates a Logical Storage Manager (LSM), which builds virtual disks, called volumes, on top of UNIX physical disks. LSM volumes take full advantage of data striping across multiple physical disks, while allowing the application to operate on the volume rather than on the physical disks. This capability supports higher performance or higher reliability depending on application requirements.

On stand-alone UNIX systems, data integrity is also upheld reliably. AdvFS is log based, enabling stable recovery and, thus, assured data integrity, and UFS uses a file system checker to guarantee data integrity. Protocols like NFS, however, provide lower data integrity due to inherent design characteristics—a significant issue when we move from single system configurations to multi-system configurations that must access remote file systems.

Taking UNIX file systems remote

In going from a single-system to multi-system configurations, file systems must be shared remotely, which challenges many of the design principles applied to local file systems. Several approaches have been developed to support remote file systems.

The most common method in UNIX to share remote file systems, however, is the Network File System (NFS). NFS extends a number of attributes of local file systems to a distributed environment, such as transparency, and it attempts to maintain the same high performance for accessing remote files as is possible in accessing local files. NFS Version 3, in particular, features a number of enhancements over previous versions, including improved client write throughput, reduced server load, improved support for systems using Access Control Lists (ACLs), and support for large (multi-gigabyte) files on NFS servers.

NFS also still contains weaknesses, however, that aren't acceptable in high-availability clustered environments. For example, NFS deviates from local file system semantics because it has to reallocate file attributes across multiple systems. On a local system, if one process writes data to a file and another process reads it back, the second process is guaranteed to read the most current data that had been written. NFS, however, loses cache coherency between systems—that is, caches on two or more systems are not kept in a consistent state. So, if a process on one node is writing data, and a different process on another node is reading data, there is an open window of time during which newly written data is still in cache without other systems being aware of it. As a result, users could be working with “stale” or inaccurate information.

This issue is commonly resolved by coding applications to use file locking. When NFS uses file locking, however, it suffers a significant performance degradation—as much as 50 – 90 percent—because it disables caching. In addition, NFS must write data to disk synchronously when the file is closed to ensure that the modified data will be seen when the file is opened on another system.

This thwarts some of the benefits of write behinds. Moreover, NFS does not even guarantee that write behinds will fit on available storage, resulting in potential data loss.

These issues of performance versus data accuracy simply do not exist on local file systems.

Cluster File System goes remote seamlessly

Due to the inherent weaknesses within NFS, Compaq *TruCluster Server* could not use this protocol for a cluster-wide file system. Therefore, Compaq developed a unique approach to sharing file systems in multi-system configurations that would retain all the characteristics of local file systems as they would behave on a single system—even avoiding the common pitfalls of remote file systems as described earlier. The result is the Cluster File System (CFS).

CFS extends all the desirable features of local file systems to a clustered environment, retaining transparency, performance, data integrity, and easy management, while enabling extensive scalability and the highest levels of availability.

Single-system semantics

The foundation for preserving local file system characteristics is the single system image. CFS preserves full X/Open and POSIX semantics so file system access, file management interfaces, and utilities all operate in a cluster just as they would on a stand-alone system. Tokens and block reservations are used to preserve POSIX semantics cluster-wide, thus maintaining a stateful protocol on the cluster and ensuring data integrity just as on a single system.

By providing a single-system image regardless of how many members are in a cluster, files are visible to and accessible by all cluster members as though they were locally attached. Unlike NFS, CFS preserves cache coherency, so if a process on one system writes data, there is no window when subsequent reads might read old data. By maintaining cache coherency across cluster members, CFS guarantees that all members of the cluster have the same view of data in the cluster—at all times. What's more, any cluster member can serve file systems on devices anywhere in the cluster, as well as be a client to other cluster members. This client/server model enables the administrator to transition a member between client and server roles, which is especially useful for relocating file systems to balance the load across a cluster.

Transparency

With a single system image, CFS preserves the transparency of local file systems. That is, instead of “tacking on” another file system and designating it for cluster activity, Compaq has designed CFS as an access management layer that resides above local file systems. As such, CFS allows each cluster member to access and share files among multiple local file systems just as if they were on a single system. In cluster approaches using a separate file system specified for clustering, files can only be shared among that cluster-enabled file system. Figure 2 below illustrates the difference.

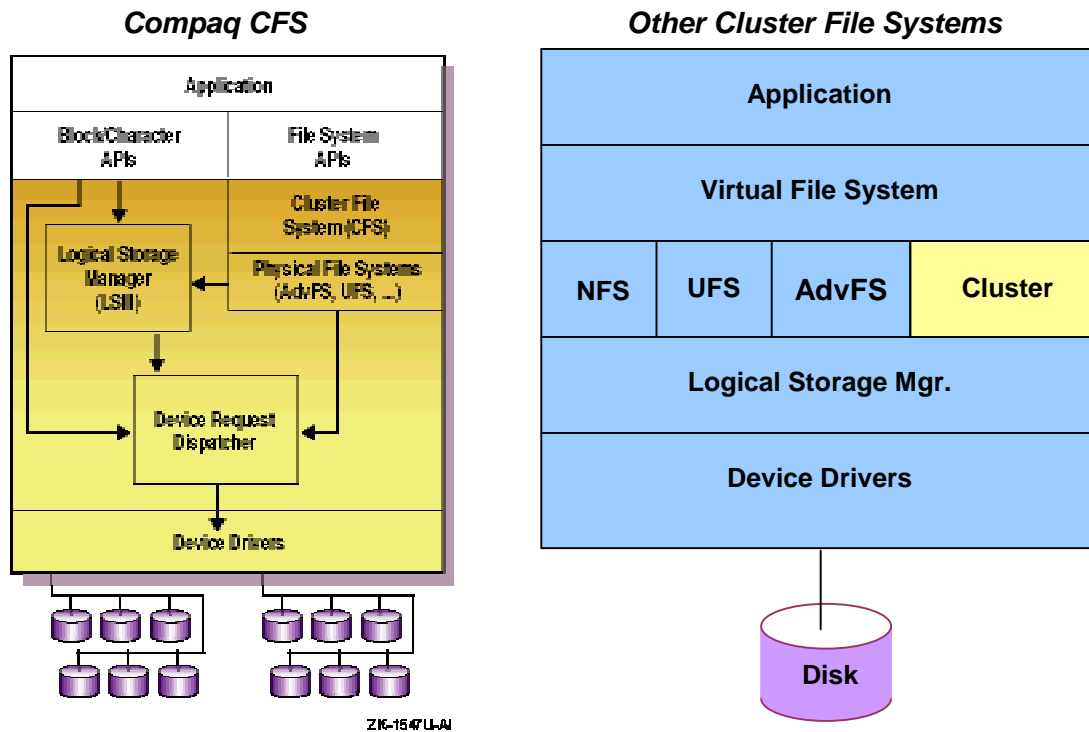


Figure 2

With CFS, applications and users gain full, transparent access to files on any supported local file systems, regardless of the physical system on which they're located. That means there's no need to convert existing file systems into a new format to participate in clustering. It also retains the modularity of single systems for mixing multiple local file systems and evolving local file system technology without affecting applications and users.

CFS supports a full range of local file systems, including AdvFS, NFS (server and client), UFS, CDFS, DVDFFS, and PC-NFS server. It's interesting to note that with CFS, *Tru64* UNIX is able to support NFS with clustered configurations, just as it did with single systems. Since the cluster acts like a single system, clusters can be part of a larger distributed environment to support specific high-availability requirements wherever they are needed in the enterprise.

The single shared root and global name space of CFS also enable cluster-wide, transparent access to all physical storage, including SCSI, magnetic tape, DVD, and CD-ROM. A Device Request Dispatcher controls all I/O to physical devices to enforce single-system open semantics. Because device names are consistent throughout the cluster, the Device Request Dispatcher can make physical disk and tape storage available to all cluster members, regardless of where the storage is located in the cluster. This allows great flexibility when configuring hardware, since a cluster member doesn't need to be directly attached to the bus on which a disk resides in order to access storage on that disk. This also provides for higher availability, since the Device Request Dispatcher can handle device path failures transparently. If all paths to a device are lost from a cluster member, then the Device Request Dispatcher will simply issue the I/O through another cluster member's adapter.

Although the single name space ensures transparency and simplifies management, there are some configuration files and directories that should not be shared by all cluster members. To account for these files and directories, TruCluster Server uses a special form of symbolic link called Context Dependent Symbolic Link (CDSL). CDSL allows a file or directory to be accessed by a single name, regardless of whether the name represents a cluster-wide file or directory, or a member-specific file or directory. Using CDSLs, TruCluster Server keeps traditional naming conventions, while providing the underlying capabilities to ensure that each cluster member reads and writes its own copy of member-specific system configuration files. Any application can take advantage of CDSLs for additional benefits. For example, single-system applications may run across multiple clustered systems simultaneously to gain scalability.

Performance & Scalability

CFS enables *Tru64* UNIX clusters to maintain the high performance enjoyed by single systems, while delivering outstanding scalability.

CFS is stateful, but lightweight, which enables *Tru64* UNIX clusters to cache aggressively to maintain high performance without sacrificing data integrity. With NFS, however, file attributes must be revalidated and data must be synchronized when written to the server, resulting in a significant loss of performance. Locking negatively affects performance since it disables caching. (see Figure 3 below).

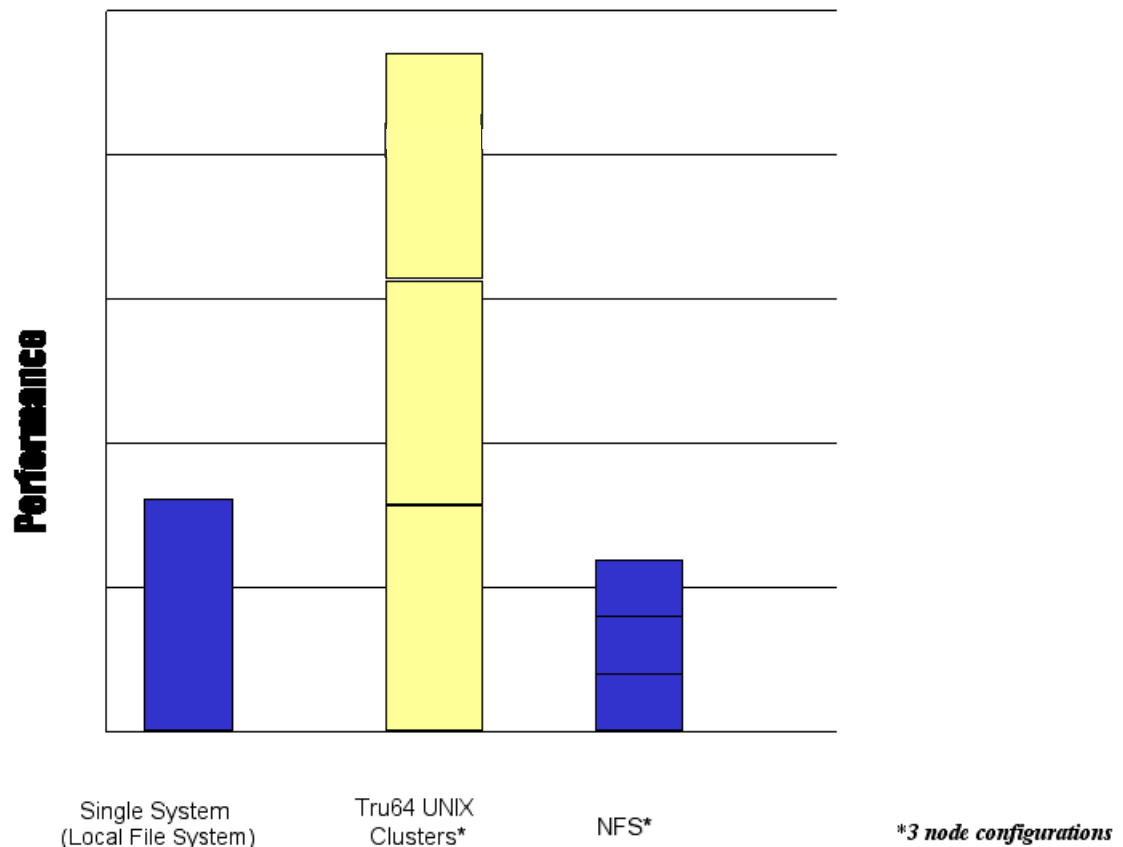


Figure 3

By using a lightweight token mechanism for cache coherency, CFS can maintain high performance across the cluster. The token is simply “piggy-backed” onto other operations and retained after the file is closed for quick reuse. That means that, unlike NFS, CFS can maintain caching even when file locking is being used. For applications, such as databases, where caching degrades performance, the direct I/O feature of *Tru64* UNIX operates with CFS just as it does in single-system configurations, allowing unbuffered, synchronous I/O for databases across the cluster and, thus, ensuring peak performance. Figure 4 below illustrates how *Tru64* UNIX maintains cache coherency.

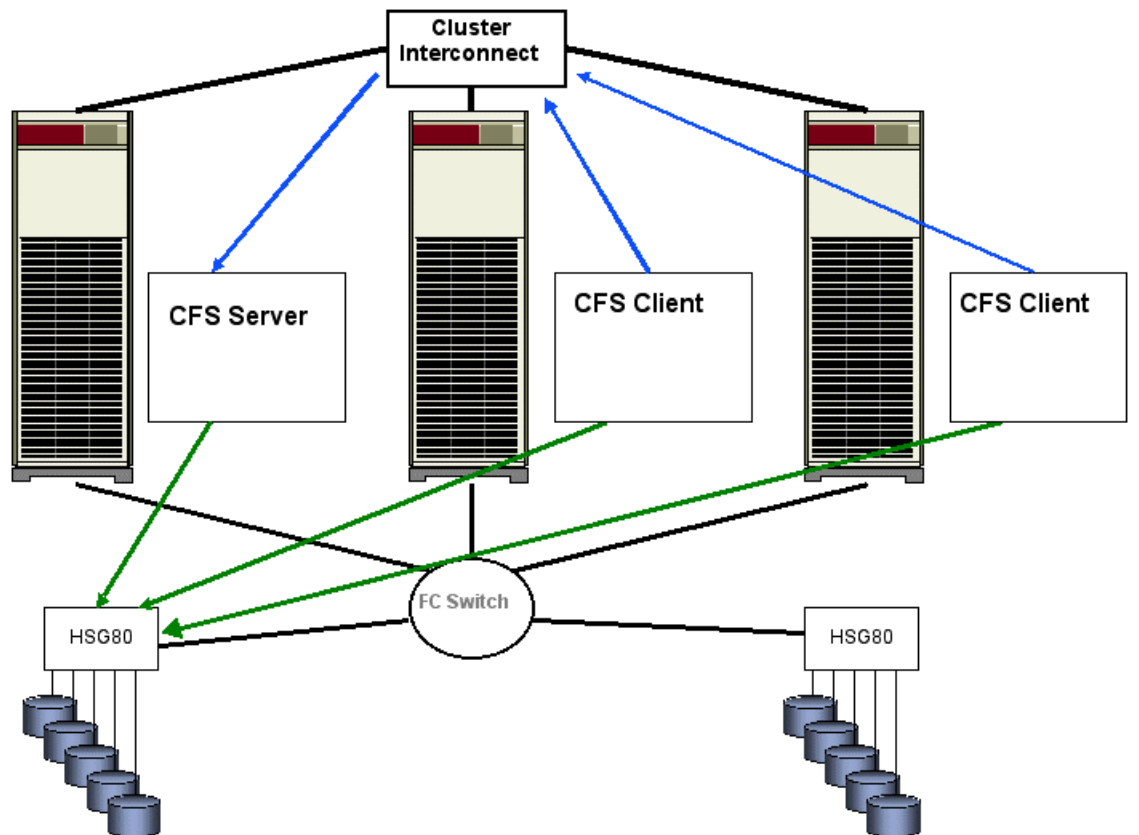


Figure 4

A block reservation subsystem is used to allow writes to occur asynchronously in the background while ensuring that the storage has adequate space. In addition, the approach used to track asynchronous write behinds enables *Tru64* UNIX to consolidate I/O activity efficiently, and enables a cluster to recover file systems quickly, since there is no need to replay “lost” data during recovery. Therefore, just as on a single system, CFS delivers both performance and data integrity in a clustered environment.

Cache coherency also comes into play for tracking metadata. With CFS, a single system within a cluster is responsible for physical metadata updates to a given file system. So, rather than caching metadata across all members of a cluster, one system can support the entire cluster. This eliminates the wildly fluctuating I/O patterns that can occur in some cluster configurations, thereby reducing the overhead of constantly re-caching metadata. This is particularly important as a cluster grows.

For example, if metadata had to be cached across each member of a cluster, the I/O that would be required cluster-wide would greatly degrade performance as the number of cluster members increased. With the single-system caching approach used by CFS, clusters can scale extensively without diminishing performance. That is, *Tru64* UNIX will maintain high performance across the full range of cluster sizes, optimizing performance for a cluster regardless of the number of systems in the cluster.

Figure 5 below illustrates performance degradation in typical cluster approaches and compares it to performance levels with *TruCluster* Server.

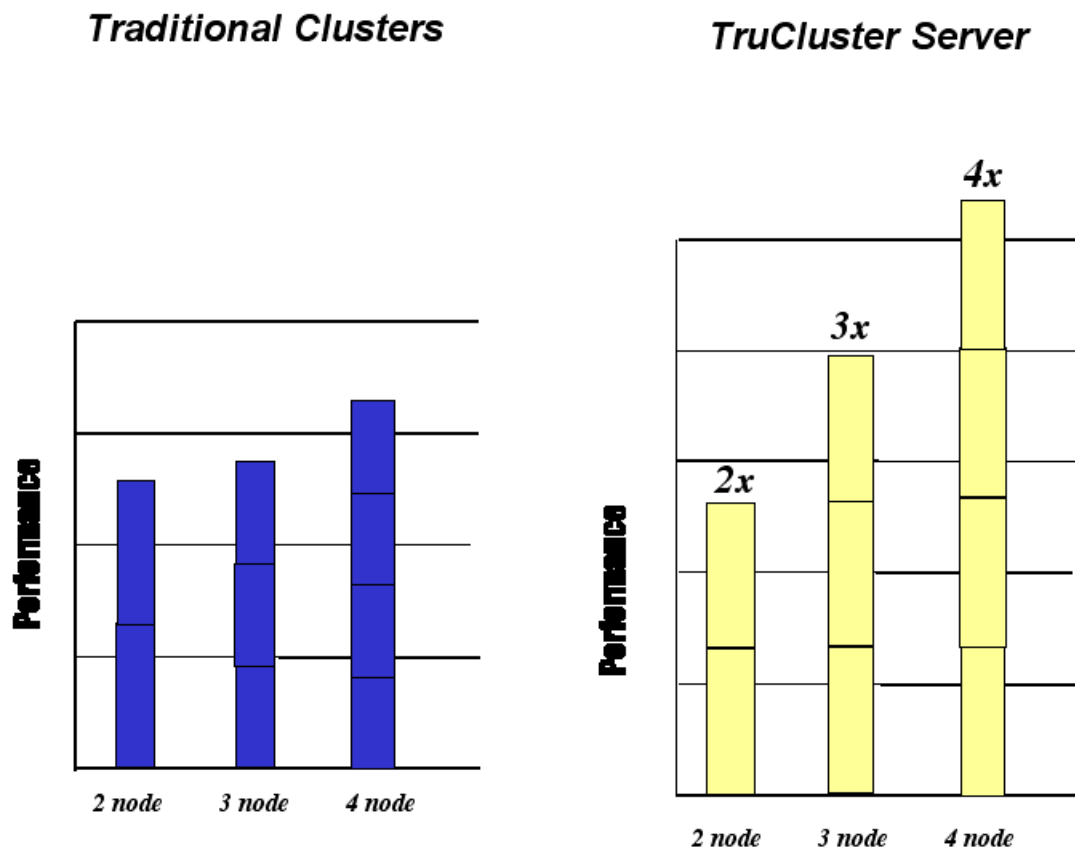


Figure 5

Tru64 UNIX clusters also scale in their ability to serve multiple file systems to multiple clients in parallel, just as if a single system were serving multiple clients. Consider a cluster serving NFS clients and PCs.

Tru64 UNIX includes Advanced Server for UNIX (ASU) software, which is a layered application that implements Windows NT server services and functionality on the same system that's running *Tru64* UNIX. ASU functions identically whether running on a single system or a cluster. As discussed earlier, *Tru64* UNIX also supports NFS in both stand-alone and cluster configurations. In supporting both NFS and PCs, each member of the *TruCluster* Server cluster can serve both NFS clients and PCs in tandem, spreading the workload across the systems in the cluster.

As illustrated in Figure 6 below, a four-member cluster effectively supports four times as much workload as a single system, enabling extensive scalability, while maintaining high performance along with the inherent high availability of clusters. This same approach can be used to configure the cluster as a highly scalable and available Web server, ftp server, or to support mail and messaging clients such as SMTP, IMAP, and POP.

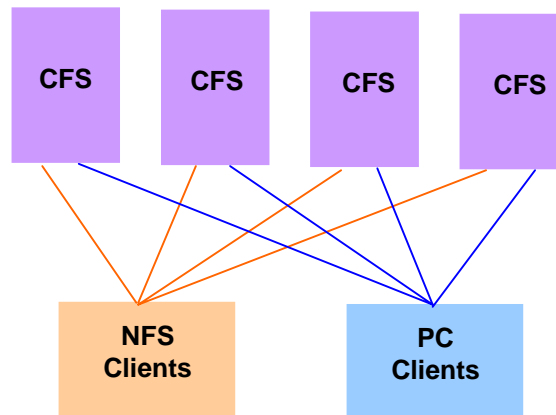


Figure 6

Easy cluster management

With CFS, the ease of management enjoyed on single systems is also preserved in clusters. One of the key means to ensuring easy management is the shared root supported by CFS.

In typical cluster approaches, each system in the cluster has its own private root and `/usr` file systems. Since most system software and configuration files are located in the root or `/usr` file systems, having separate root and `/usr` files on each system means that every configuration change or software installation must be performed individually on each system. This approach levels a heavy burden on administrators.

CFS eliminates this problem by supporting a shared root and `/usr` files, the operating system need only be loaded once and all the system software and configuration files will be automatically shared with other cluster members. Adding additional systems to the cluster is a very quick operation. And if a configuration change is made, it can be made just once.

In addition to the shared root, CFS enables clusters to take full advantage of the AdvFS capabilities enjoyed by single systems. For example, by managing the physical storage layer independently of the directory layer, system administrators can add and remove storage without unmounting the file system or halting the operating system on any member of the cluster. This not only saves time, but provides administrators with extensive flexibility to address issues without disrupting system operations that might affect users.

Built-in, cluster-wide load balancing also minimizes the burden on administrators, providing the flexibility and cost-effectiveness of building clusters out of groups of smaller systems without incurring undue management overhead.

Conclusion

Single system configurations can take advantage of many valuable attributes of UNIX file systems. In particular, systems running *Tru64* UNIX gain the advantages offered by Compaq's AdvFS file system, which provides extensive flexibility, huge capacity, data availability, high performance, and simplified system management.

Unlike typical cluster approaches, which sacrifice many of the advantages offered in single systems, Compaq *TruCluster Server* with its Cluster File System preserves all the best features of single systems—even improving upon the weakness of some file systems. As a result, enterprises can adopt the high-availability capabilities of clustering without risking data integrity or taking on a heavy management burden.

As with single systems, CFS preserves transparency, high performance, and predictable behavior across the cluster. CFS also supports extensive scalability with the ability to serve multiple file systems and, thus, multiple clients in parallel.

Compaq has been awarded a number of patents for its cluster technology, providing strong validation that the *TruCluster Server* architecture is truly a breakthrough in high availability solutions. As a result, enterprises can capitalize on the advantages of *Tru64* UNIX for their competitive advantage. Moreover, because Compaq has developed its own approach to clustering, enhancements and further innovations will continue to give enterprises a leading edge.

With *Tru64* UNIX and *TruCluster Server*, the advantages of high availability no longer need to be weighed against the burden of typical cluster approaches. Compaq is truly delivering clustering for all, enabling enterprises of all sizes to protect their critical information assets and grow seamlessly while maintaining the ease of management of a single system.