DISTRIBUTED FILE SYSTEMS FOR STORAGE AREA NETWORKS

Brad Kline
Advanced Digital Information Corporation
11431 Willows Road N.E.
Redmond, WA 98073-9757
425-881-8004
brad.kline@adic.com

ABSTRACT

This paper discusses some fundamental aspects of Distributed File Systems (DFS) for Storage Area Networks (SANs). Many desirable features make a SAN DFS attractive to a production environment that processes huge volumes of bulk data and requires high bandwidth. Throughout the discussion, the traditional server-attached storage model is compared against the distributed network-attached storage model. The remainder of the paper will focus on issues of DFS configuration, data placement strategies, cross platform data sharing, fault tolerance and performance. Finally, we will show that distributed file systems can provide superior performance over traditional server-attached storage shared file systems.

INTRODUCTION

Various shared file systems have been in use for more than fifteen years. They started as simple systems using file server protocols such as the Network File System (NFS) (Sandberg 1985). These products are still a very large part of today's shared storage solutions. With the advent of open standard technologies such as Fibre Channel and its ability to imbed SCSI commands (ANSI 1994), it is possible to connect mass storage directly into a network at relatively low cost. Using compatible Fibre Channel Host Based Adapters (HBAs) any workstation can address the storage and access data to and from it just as it would do to a directly connected SCSI storage device. The network, which includes HBAs, hubs, switches and network-attached storage, is called a Storage Area Network (SAN). Fibre Channel is a high speed (100Megabytes / Second) channel that is capable of connection distances measured in kilometers. The Fibre Channel protocol uses addressable nodes such that storage devices can be configured in a network fabric rather than point-to-point. This capability provides to us a storage network that is highly scalable, has high performance and allows complete sharing of any connected device.

The distinct difference between using the conventional server-attached storage shared file system and a SAN is that any workstation connected to the network fabric can directly access the network-attached storage devices. The new connectivity obviates the need for a server and promotes a more distributed approach to managing data. This concept has started a new surge of distributed storage product development and has been coined the SAN solution.

Many vendors have committed resources to developing hardware products that comprise a SAN by producing a variety of Fibre Channel compatible hardware components. In and by themselves they offer a high-speed access method to storage. What has been missing from the formula is the ability for software to leverage the distributed nature of the storage network. At the core of every operating system are file systems that abstract disk storage and allow multiple programs to reliably share this resource. Existing local file systems such as Microsoft's NTFS[1], SGI's XFS and Apple's HFS assume that any visible storage, whether in a network or locally connected, is owned by the local workstation. If the file system accesses the storage without coordination with other workstations in the network and two or more inadvertently share access to the network-attached storage devices, data corruption occurs. Because of this problem and due to no readily available software solution, the early SAN devices were used only in a server-attached shared

[1] All products described in this paper have registered trademarks owned by their respective companies.

storage environment. In the last three or four years however, a number of mostly third-party vendors have begun developing and offering distributed file system solutions that exploit the SAN distributed environment (O'Keefe 1998).

This paper will discuss how a SAN File System (SAN FS) operates. We will discuss the difference between a typical server-attached shared storage model using NFS and a distributed network-attached storage model using ADIC's CentraVision File System (CVFS) (Kline and Lawthers 1999). Then we will address important SAN FS features, such as configuration, data placement strategies, cross platform data sharing, fault tolerance and performance.

STORAGE MODELS

The Server-Attached Shared Storage Model

In a typical data processing center today, there may be many different methods of sharing storage among multiple workstations. The most prevalent storage model is server-attached. This model uses one or more large workstations or *file servers* to locally attach the shared disk storage. The storage is then shared over some network topology using protocols like NFS. The systems are inter-connected through network interfaces such as Ethernet, HiPPI or ATM. Extensive effort has been made by numerous vendors to make the file server model a high performance solution. A problem exists in that the model must copy from the data storage to the server's memory before it can transmit the information to the requestor. The performance therefore is gated by the speed of the server's memory and processing power. (See Figure 1.)

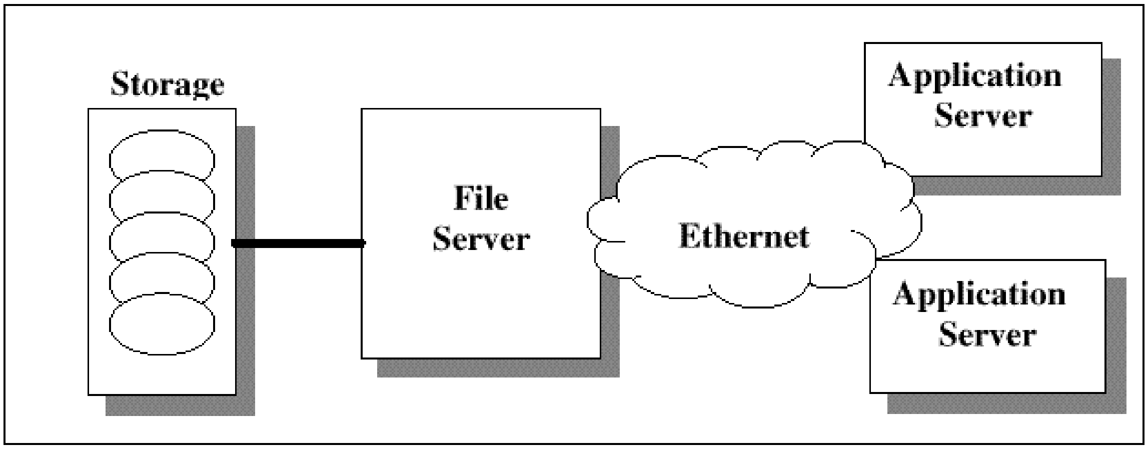


Figure 1. A server-attached shared storage model.

The Distributed Network-Attached Storage Model

By making storage a part of the network rather than a part of the local server, a DFS can use a different access method. The storage model now changes from server-based to peer-to-peer between the workstation and the storage. The advantage to this model is that with proper system software the data can be transmitted directly from the storage to any application without using a file server as a middleman (See Figure 2.)

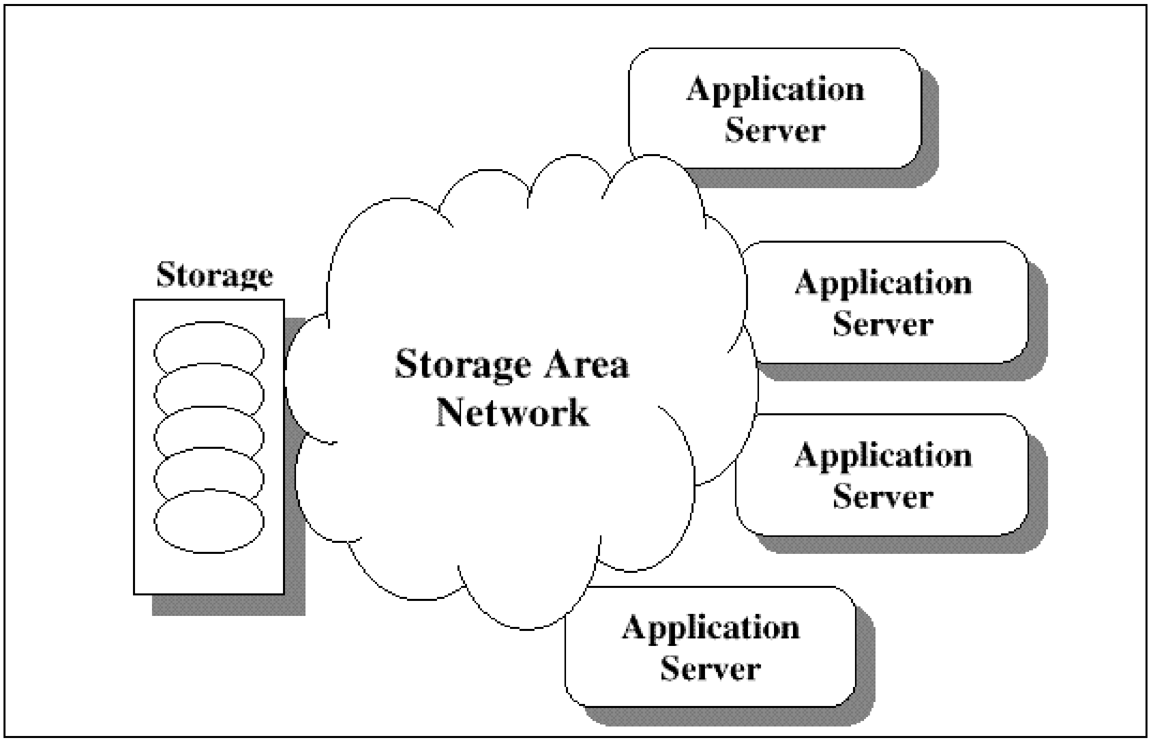


Figure 2. A distributed network-attached storage model.

When storage is placed on the network it is, in fact, a distributed object. Many workstations can access the storage equally, which adds a new level of complexity. The file system software on each workstation must now somehow coordinate critical operations to the *meta-data.* Meta-data operations are related to the management of the allocation and name space of the data on the storage. For example, a user creates a file called ***myfile*** and writes 2,000 bytes to it. In this situation, two operations must be globally unique to the file system. First, the name space must be guaranteed to be unique. There cannot be two files called ***myfile*** in the same directory. Second, the allocation of data blocks must not collide with other space allocation. If two workstations simultaneously request a block of storage then they must not receive the same block. These operations require some sort of locking, such that *serialization* occurs. A DFS must therefore have an object that manages, at a minimum, the locking of meta-data components. These objects may be in themselves highly distributed components but there must be some agreement of mutual exclusion.

The Hybrid Network-Attached Storage Model

There are other ways of managing critical meta-data operations. For example, a DFS can use a distributed lock manager and meta-data server using a communications network (Mohindra and Devarakonda 1994). This method requires both a SAN and a communications network to successfully operate the file system. It is possible to operate a communications network over Fibre Channel, but in essence, it must be considered a separate logical network from the SCSI interfaces to the network-attached storage. This model is referred to as a *hybrid network-attached storage model.* A hybrid model consists of a control path that uses a *meta-data server* protocol and a data path (like Fibre Channel) that directly accesses the storage. (See Figure 3.)

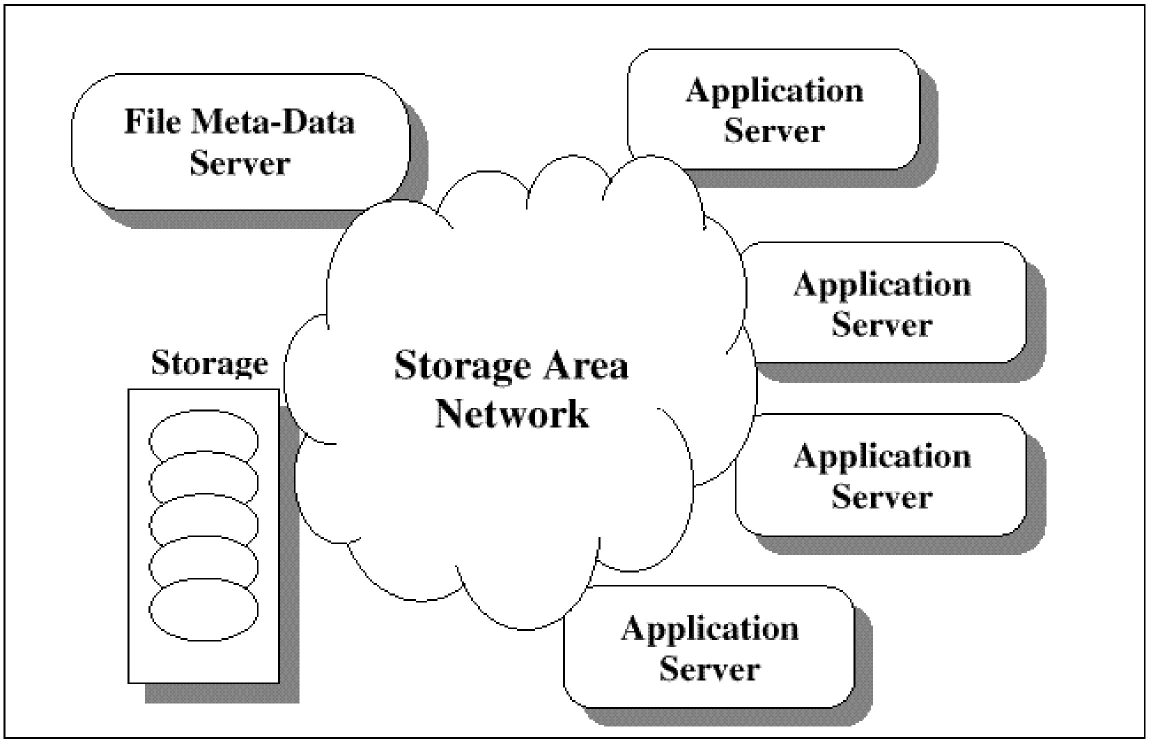


Figure 3. A hybrid network-attached storage model.

An example of a hybrid file system is Advanced Digital Information Corporation’s CVFS. All meta-data requests are managed by a meta-data server using a TCP/IP network. All data operations are managed directly by the requesting application server using a Fibre Channel SCSI interface. Thus, the server-based model is used for controlling meta-data operations and the distributed model is used for data. A minor drawback to this model is that meta-data operations are gated by the speed of the TCP/IP network and the capability of the meta-data server. However, the volume of data transmitted between the application server and the meta-data server are just a fraction of the total bandwidth available. When contrasted to the hundreds of megabytes per second transmitted through the data path versus kilobytes of control data transmitted over the TCP/IP network, the model has been shown to scale very well.

CONFIGURATION

In server-attached storage models using protocols such as NFS, configuration is typically simple. A top-level directory is exported from the server and any application that has permission can access the file system. Details of the file system are managed on the server since NFS is really a network file system on top of a local file system. For a DFS, configuration tends to be more complex. It contains the same complexities that are found in managing a TCP/IP network. There are issues of storage identification (Storage Network Address versus IP address), connectivity (HBAs versus NICs) and topology (switches and hubs.)

In CVFS, there is a concept of *Network* S*torage Pools* first described by Soltis’ Global File System (GFS) (Soltis et al. 1996), termed *Stripe Groups*. Stripe groups are sets of storage devices grouped into a logical set. The complete set of stripe groups is then portrayed to the system and end-user as a single file system. A stripe group might be a stripe of JBOD (Just a Bunch Of Disks) units, a RAID-5 disk array or any other storage set that makes sense. (See Figure 4.)

These groupings of storage can be tuned to optimize each pool so that it makes the best use of their bandwidth in the SAN. For example, eight Seagate Barracudas in a Fibre Channel chassis configured as a JBOD using Raid-0 can just about saturate a single Fibre Channel. By placing this component on one channel and connecting it to a Fibre Channel Fabric Switch, it makes the best use of its potential bandwidth. It also allows the file system and associated applications the flexibility of allocating storage blocks to specific stripe groups based on their described characteristics.

Stripe groups enable the CVFS file system to support *concurrent maintenance.* Concurrent maintenance is the concept where a stripe group can be taken off-line for maintenance without severely impacting access to the rest of the file system. The importance of this feature grows as a function of the size of the file system. Downing a five terabyte file system that is attached to many users can be quite disruptive. It is more efficient to down only the failed component and keep the rest of the file system running. CVFS supports the ability to up and down its stripe groups individually. It can also switch on or off specific read or write capabilities to a stripe group. This feature is used when a component of a group is failing and data must be drained from the failing group to another. The stripe group is marked *write disabled* so no further allocations or writes will be made to it. Then the group is maintenance copied to other fully functional pools in the SAN.

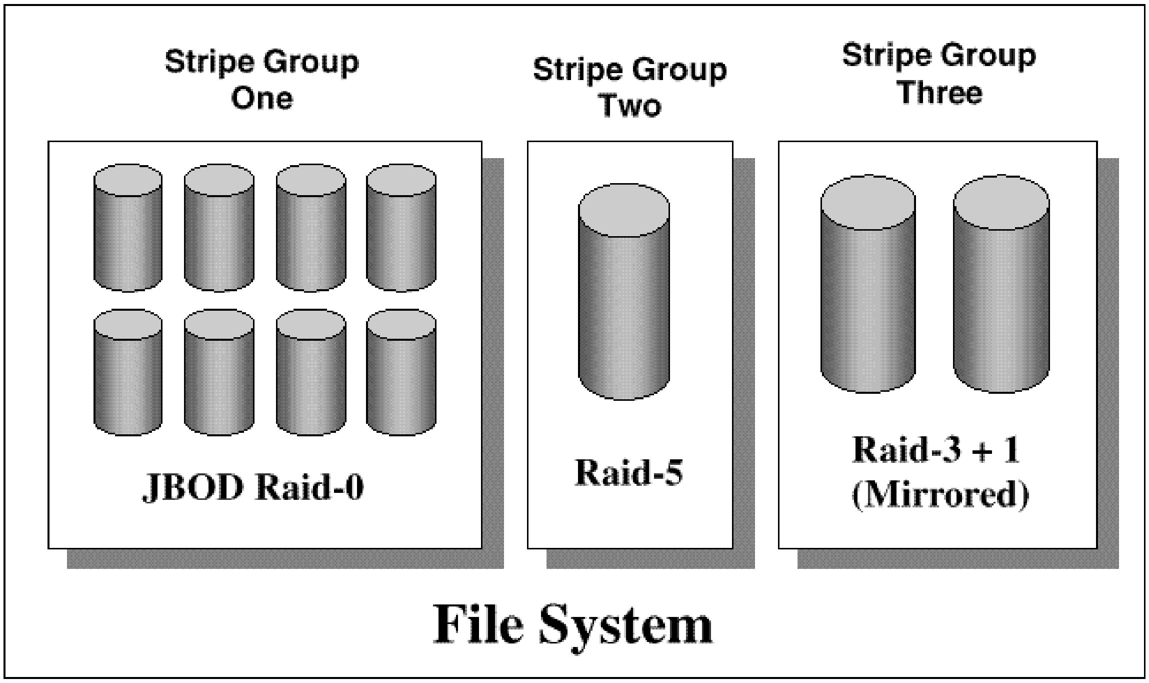


Figure 4. Stripe groups

DATA PLACEMENT

Using NFS there are no mechanisms for an application to describe how the data is to be used. Allocation of data is controlled by the local file system on the file server. This problem is typically solved by creating separate mount points for each unique file system group on the server and placing the onus of decision upon the application to use the proper storage path. While this solution is workable, it lacks flexibility. If something in the server-attached storage changes, then all the applications that share the storage must manually change their access method. For example, they would be required to change the pathname of the mounted NFS storage. It would be more efficient if issues of storage placement were handled transparently in the distributed file system. If a storage requirement changes then administration could be applied to the file system without notifying all of the applications to change their access method.

The CentraVision File System has a concept of *data placement*. Data placement is the ability to allocate storage onto different stripe groups using specific constraints. For example, a file system has a stripe group optimized for High Definition (HD) Video streaming. Another group is optimized for regular bookkeeping or text files. Data placement is achieved either by using a special command or an Application Program Interface (API) call to the file system. The application can request its high bandwidth file on the HD stripe group and let all the other files default to the regular stripe group. This separation of performance data from bookkeeping data allows optimal usage of the two stripe groups. It also can prevent excessive interleaving of data blocks that can reduce performance due to channel contention and storage head motion (disk chatter). This feature is transparent to most applications. It does not require interaction if some thing on the file system changes. Essentially changes only have to be administered to the file system and not to all of the participating share storage applications.

CROSS PLATFORM DATA SHARING

It is unusual that a data center has all of its machines running the same operating system. A typical film and video post-production house has SGI, NT and Apple systems. These systems are typically used for different specific functions in the workflow but they all are working on some aspect of the same project. When data resides on server-attached shared storage, it is very common for projects to transfer important pieces of data to the local workstation's storage simply to save time. This data replication is very expensive and the time it takes to transfer the data increases the project's time to complete. It also adds a new class of problems if the shared storage version of data is updated but the local workstation's copy is not.

It is unusual that a data center has all of its machines running the same operating system. A typical film and video post-production house has SGI, NT and Apple systems. These systems are typically used for different specific functions in the workflow but they all are working on some aspect of the same project. When data resides on server-attached shared storage, it is very common for projects to transfer important pieces of data to the local workstation's storage simply to save time. This data replication is very expensive and the time it takes to transfer the data increases the project's time to complete. It also adds a new class of problems if the shared storage version of data is updated but the local workstation's copy is not.

One of the more complex and expensive issues in the development of a DFS is allowing the file system to be used simultaneously by these different operating system platforms. Each platform has its unique features and idiosyncrasies. For example, on most UNIX platforms there is the concept of a User Identifier (UID), a Group Identifier (GID) and permissions, or modes for owner, group and all others. On Windows NT, there are access lists and privileges. It is very difficult to map these disparate models together. CVFS accomplishes this by having the ability to map NT and Apple users to a common UNIX user / group model. As SAN products become more popular, consumers will expect a more complete solution to address security issues.

Another constraining issue is that most operating systems differ in their support of 64-bit sizes in file systems. In large bulk-data processing applications, file sizes regularly can exceed four gigabytes. A majority of applications use the default POSIX *stat ()* system call interface (IEEE 1988) but vendors describe 32-bit containers for its file size. CVFS maintains all of its file sizes in 64-bit containers and sometimes has to modify the system interface to support only 32-bit sizes. This problem must be addressed and solved by operating system vendors and the standards community. In the interim, some constraints will be placed on certain incompatible cross platform operations.

Heterogeneous support is one of the big advantages to the NFS server-attached storage model. The NFS protocol is consistent and is easily implemented on a number of operating system platforms. However, poor performance is the result if data and meta-data must be abstracted into Remote Procedure Call (RPC) protocols and small buffered data blocks. The cost is measured in high latency and low bandwidth. CVFS appears to each platform as a local file system. It does not use bridges or adapters to convert a native format into the format of the target operating system. It also uses the Direct Memory Access (DMA) capabilities of each hardware component on each platform. This capability delivers high performance to all the different system platforms without compromise.

FAULT TOLERANCE

In a file server environment, if the server goes down all the application servers accessing the shared storage are essentially down. If their critical data is on the shared storage there is nothing that can be done until the file server is back on-line. In other words, the model has a single point of failure that is difficult to solve. Many vendors have produced a number of solutions to this problem but most are expensive to implement. The predominant NFS solution is to use fail-over file servers and switched storage units.

Using a DFS, the single point of failure problem can be solved in many different ways. Each workstation in the DFS still has access to the storage even if other workstations have failed or are off-line. Storage networks by their nature can be highly fault tolerant. Of course, there are hardware planning issues to consider. For example, using dual channels to the SAN Fabric Switch or even using dual switches will reduce point of failure problems. By using mirrored storage arrays, even the data can be redundantly stored.

CVFS and other hybrid network-attached storage models inherit the file server's single point of failure problem because the meta-data is usually served by one machine at a time. The salvation is that the meta-data can be easily placed and shared in the SAN. If the meta-data server fails on one machine, another machine can quickly become the meta-data server and take over the file system's management. This solution can be achieved with no additional hardware costs since each workstation already has access to the storage.

An additional consideration to make a hybrid file system more fault tolerant is to have redundant communication networks. If the communication network is somehow partitioned due to a hardware failure, some portion of the SAN users may be excluded from meta-data server access. If a meta-data server fail-over occurs because a workstation cannot reach the active server due to a network partition we experience the symptom coined the *split brain*. This problem can potentially be severe if two servers attempt to manage a single file system. By making the control network redundant, this problem is greatly reduced. CVFS also has additional measures in place to prevent this problem by verifying fail-over situations through the Fibre Channel storage network itself.

PERFORMANCE

Performance of a distributed file system in a SAN must first be measured by its ability to reduce the time spent to complete a project. Not having to move data from one platform to another saves time. Using optimal placement strategies to best exploit the capabilities of the storage saves time. Not having fileserver bottlenecks and having direct, high-speed connections to each workstation saves time. These reasons by themselves can justify the adoption of SAN DFS technology. When vendors address NFS performance, they measure the number of transactions or operations a second a NFS file server can produce. While this is a very useful measure of performance, it ignores the issue of data bandwidth to the application. An NFS fileserver may be able to execute thousands of transactions per second. However, to an individual application it may only be able to generate data transfer rates of a few megabytes per second. A SAN DFS can attain a high transaction rate by its very nature. Soltis treats this subject in (Soltis et al. 1996), comparing their SANDFS over NFS. In addition to a high transaction rate, a SAN DFS can also sustain high data transfer rates to an individual application. The net effect is superior application performance and thus less time to complete a project.

The next measurement we must make of a distributed file system is that of single workstation performance. This type of performance is typically measured as the number of megabytes per second of real data payload that can be delivered to the application. Using a single 100 MB / Second Fibre Channel connection and SCSI protocol, a DFS can deliver about 94 MB / Second data-payload performance capability *on that single channel*. This assumes that the storage is capable of delivering that speed. Therefore a single high-performance workstation, given the right IO conditions and storage speeds, should be able to sustain at least the channel bandwidth. With more than one workstation competing for the same100 MB / Second channel, performance bottlenecks can occur due to storage disk head motion and channel contention. These artifacts are hard to manage. What a DFS needs to manage is the overhead of the file system and to some extent the access distribution. The measurement for multiple workstations accessing a DFS is called the *aggregate performance.* The goal is to make the single channel aggregate performance as close to the best single workstation performance as possible.

As mentioned earlier, ganging storage devices together as a striped storage pool can greatly enhance performance. A Seagate Barracuda disk can achieve a read performance of around 10 MB /Second. By placing eight Barracudas on the same Fibre Channel and striping the IO across all the units, close to channel speed can be achieved. Conversely, a single CIPRICO RAID-3 Array might best be placed on a channel by itself since it can saturate most of the Fibre Channel's bandwidth.

Single Workstation Performance

The following graph (Figure 5) shows individual performance of a Fibre Channel loop with eight Seagate Barracuda 9 GB drives striped RAID-0. The stripe breadth is 128 blocks and each block is 4096bytes. The host is an SGI Challenge using the 6.2 IRIX operating system running with a Prisa Fibre Channel HIO host adapter.

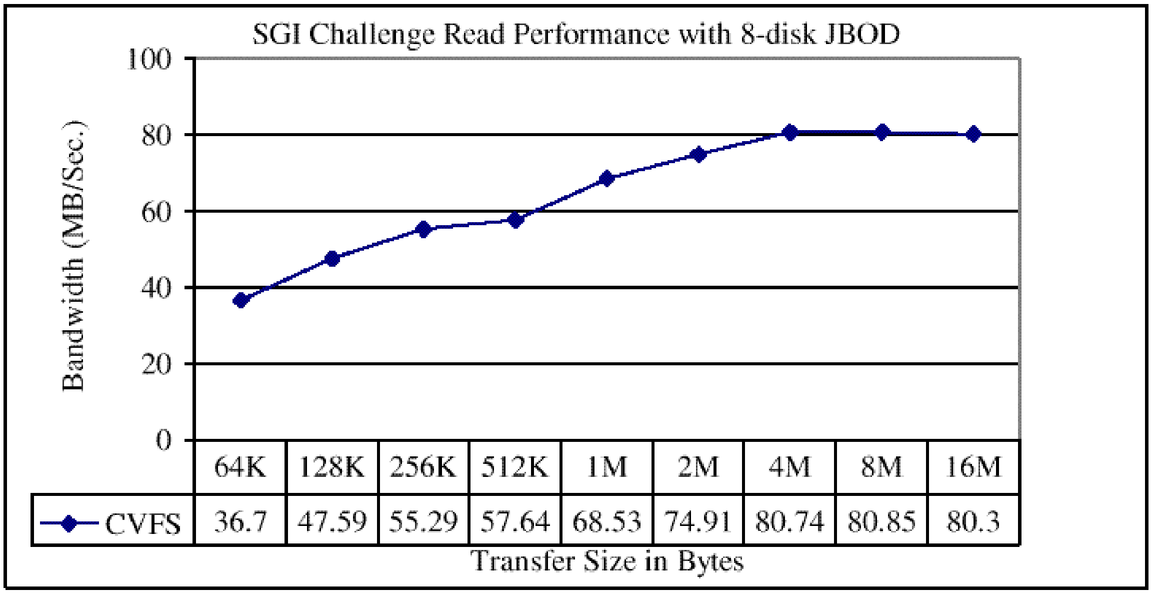


Figure 5. SGI Challenge Read Performance using eight-drive JBOD.

A server-attached storage comparison is not shown here since it is difficult to create a comparative environment. We defer to other papers written about NFS performance. For example, in a white paper written by SUN Microsystems, Inc. (SUN NFS 1995), they demonstrate NFS write through performances measured at around seven MB / Second. This is due to the file system's inherent necessity to run through a server and in this case its requirement to communicate to the workstation over 100 Base-TX Ethernet. Wood (Wood 1995) measured 12.5 MB / Second sustained NFS bandwidth using a high performance OC3 (115 Mbit) ATM channel.

In Figure 6 below, we show a test where we ran single CVFS performance runs on a PC NT. This machine consisted of a 200 MHz Pentium with 64-MB memory and an Emulex 7000 Fibre Channel host adapter connected to eight Seagate Cheetah 9 GB disks through a 16-port Brocade Silkworm Fibre Channel switch. The operating system was Windows NT 4.0, Service Pack 3.

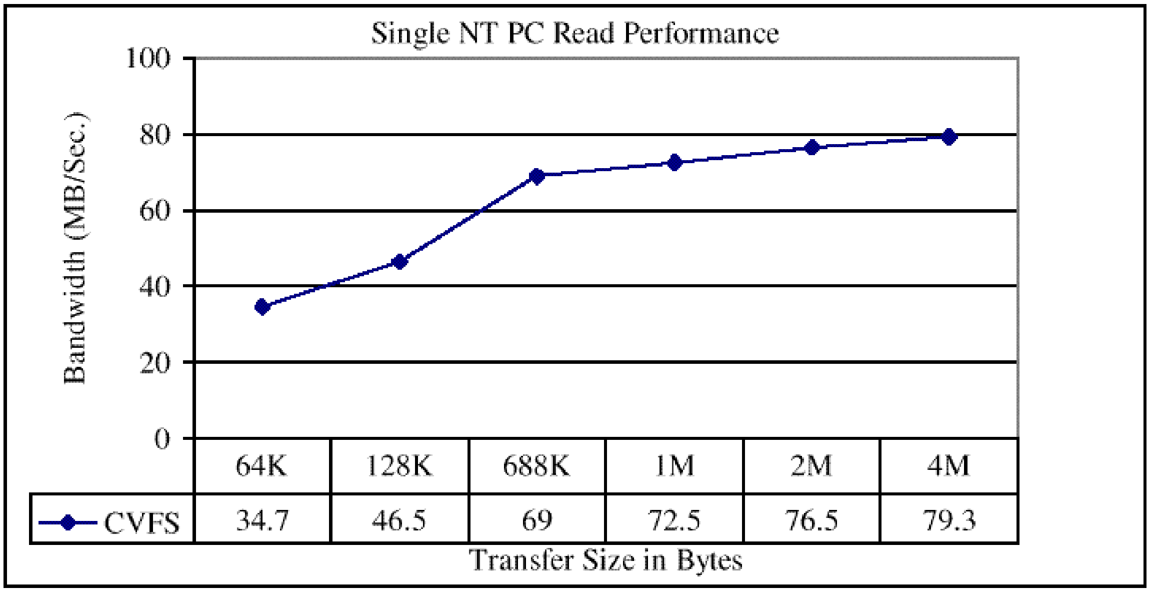


Figure 6. Single Workstation Read Performance – NT PC and Emulex FC-HBA

In the following single performance graph, we show a machine with more advanced IO capabilities. Figure 7 shows the single workstation read performance of an SGI Octane. This machine has dual 195 MHz CPUs and 256 MB of memory. The adapter is an Adaptec XIO Fibre Channel host adapter connected directly to a JBOD of eight 9 GB Seagate Cheetah Drives. The file system was striped RAID-0using 128, 4096 byte blocks per disk.

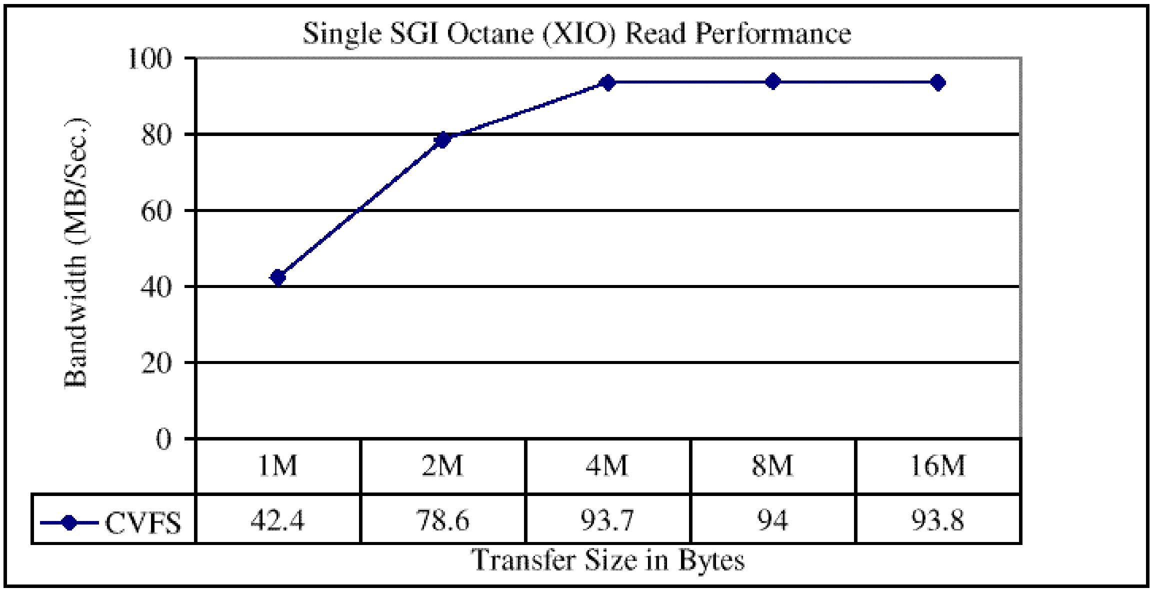
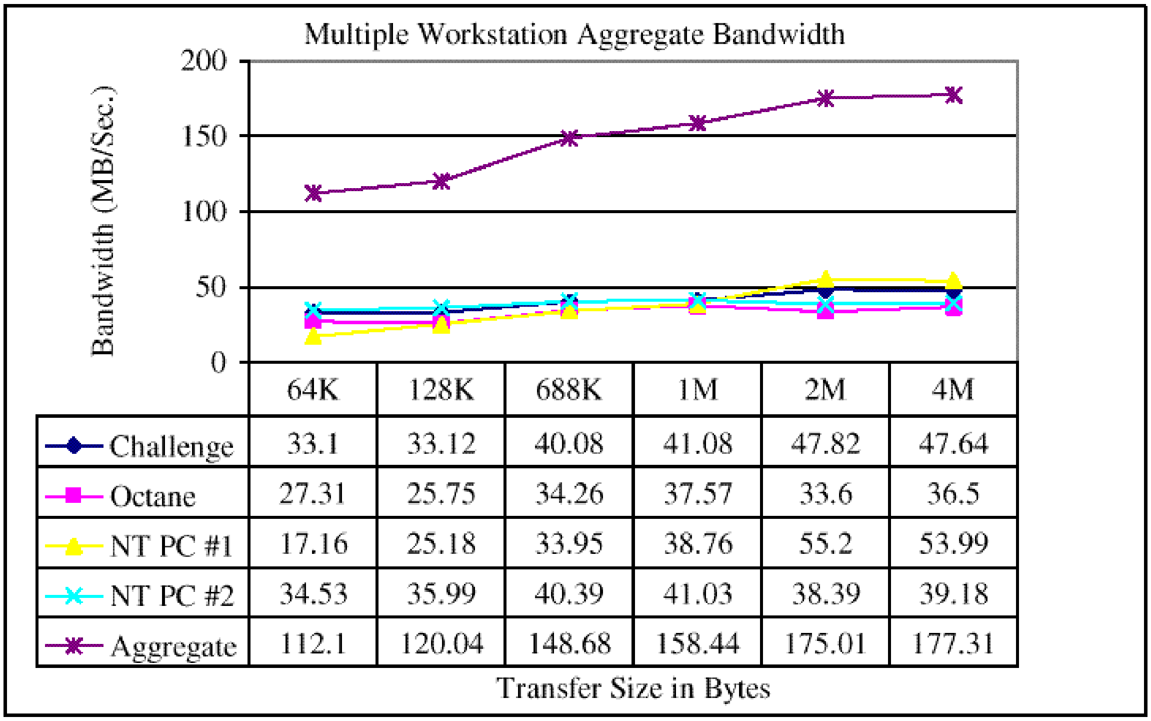


Figure 7. Single Workstation Read Performance – SGI Octane and XIO HBA

As can be seen, the Octane and Cheetah combination is capable of saturating the Fibre Channel at near the 4-MB transfer size.

Aggregate Performance

Now that we have investigated single workstation performance, focus returns to the more important aggregate performance issues. In the next experiment, we put the SGI Challenge, the SGI Octane, the NT PC and one other NT PC onto a 16-port Brocade Silkworm Fabric Switch. The new PC had the same configuration as the other except it was from a different manufacturer (Dell versus HP). We then connected two 8-drive JBOD chassis consisting of Seagate 9 GB Cheetah disk drives. Since the two JBOD chassis each have a Fibre Channel connected to them, the maximum theoretical bandwidth possible is 200MB / Second. In the Octane test, we saturated the channel at 94 MB / Second, therefore we can surmise that the potential aggregate bandwidth of the storage network is around 188 MB / Second. We created a single file system of 16 disks on the two chassis. The file system was divided into four stripe groups (storage pools) of equal size. The four-drive stripes were set at 128, 4096 byte blocks per disk. Using the CVFS data placement feature, we directed each workstation to a different stripe group. This helped eliminate the artifacts of head movement and disk cache interaction. The test was accomplished by reading four two-gigabyte files simultaneously and measuring the individual performance of each workstation. All applications were synchronized to begin their tests at exactly the same time. The aggregate bandwidth was then summed up from the individual performance numbers. (See Figure 8.)Bandwidth (MB/ Sec.)



	64K	128K	688K	1M	2M	4M
◆ Challenge	33.1	33.12	40.08	41.08	47.82	47.64
■ Octane	27.31	25.75	34.26	37.57	33.6	36.5
▲ NT PC #1	17.16	25.18	33.95	38.76	55.2	53.99
× NT PC #2	34.53	35.99	40.39	41.03	38.39	39.18
✱ Aggregate	112.1	120.04	148.68	158.44	175.01	177.31

Figure 8. Multiple workstation aggregate bandwidth

The individual performance numbers were observed to be highly volatile and this graph is an average representation of a number of passes. However, the aggregate performance value remained very consistent through all of the iterations. Using four-megabyte transfer sizes, CVFS and the Fibre Channel hardware sustained 88.5% of theoretical and 94.1% of available bandwidth through the storage network. What is demonstrated in these numbers is that a very large portion of the potential bandwidth of the storage network can be exploited even when there is contention among multiple workstations.

CONCLUSIONS

We have described the basic differences between the conventional server-attached shared storage model, a distributed network-attached storage model and a hybrid network-attached storage model. We presented our opinion of important SAN DFS features that make the model valuable to a production environment. We have shown that a distributed network-attached file system can be superior over the more conventional server-attached shared storage model. Due the bottlenecks of a server-attached storage model we can imply that without distributed file systems, the benefit of a SAN is not fully realized. It is clear that by allowing a direct connection from each application server to the network-attached storage, we can see dramatic performance gains. These gains are realized in productivity to the end-user and therefore, savings to the customer. It is also apparent that more development and cooperation is required by various operating system vendors to support cross-platform distributed file system products. It is possible that a new standard must emerge to help vendors drive towards DFS compatibility and inter-operability for the same reasons that POSIX standards were made for end-user applications.

REFERENCES

ANSI 1994, “X.3230-1994-Fibre Channel Physical and Signaling Standard (FC-PH)”

IEEE 1988, “IEEE Standard 1003.1-1988 Portable Operating System Interface for Computer Environments”, 1988.

Kline, B. and P. Lawthers, “CVFS: A Distributed File System Architecture for the Film and Video Industry”, White Paper, *http://www.centravision.com/cvloindex.html*, June 1999.

Mohindra, A. and M. Devarakonda, “Distributed Token Management in Calypso File System,*”Proceedings of the Sixth IEEE Symposium on Parallel and Distributed Processing,* April 1994.

O’Keefe, M., “Shared File Systems and Fibre Channel”, *Fifteenth IEEE Symposium on Mass Storage Systems*, March 1998.

Sandberg, R and D. Goldberg, S. Kleiman, D. Walsh, B. Lyon, “Design and Implementation of the Sun Network File System”, *Proceedings of the Summer USENIX Conference*, pp. 119-130 1985.

Soltis, S. and T. Ruwart, M. O’Keefe, “The Global File System”, *Fifth NASA Goddard Space Flight Center Conference on Mass Storage Systems and Technologies*, College Park, MD. September 1996.

Sun NFS 1995, “The NFS Distributed File Service”, White Paper, *http://www.sun.com/software/white-paper/wp-nfs*, March 1995.

Wood, C., “Client/Server Data Serving for High-Performance Computing”, *Proceedings of the Fourteenth IEEE Symposium on Mass Storage Systems*, Monterey CA, 1995

BIOGRAPHY

Brad Kline is the Principal Software Architect at Advanced Digital Information Corporation and was the architect of the CentraVision File System. Before that, Brad was a Core Design Engineer for Cray Research, Inc. and then SGI, where he was a member of the design team for the Cray T3D and T3EMassively Parallel Processor super-computer systems.