

What I Did Instead of Buying a SAN

Adam Anderson
September 2001



The key concept of a SAN is that storage hardware (i.e., disk and tape devices) is abstracted away from the server and, more importantly, the network operating system (NOS). For instance, it is not unusual to have ample aggregate disk space in an enterprise, but have free blocks in all of the wrong partitions. A SAN solution could remedy this problem by aggregating all storage independently of server and partition conditions. Further complicating matters are the issues of sparing, testing, and purchasing additional disks and arrays for the typical mesh of different hardware platforms. The promise of a SAN is to allow a centrally managed, fault-tolerant, and standardized set of arrays to fulfill all the storage needs of an enterprise. So, instead of learning various RAID management interfaces, sparing many types of drives, and having all your free disk space attached to the wrong server/partition/NOS, SAN technology promises a fully independent and standardized deployment of storage.

Actual SAN solutions, at the price points that fit budgets in typical environments, have not delivered on these promises. Often requiring forklift upgrades and extensive re-engineering of the existing storage model, SAN technology has always been well out of reach of most environments. In situations where SAN technology is not cost-effective, I have used two alternative storage technologies that deliver much of the value of the "ultimate SAN" at reasonable cost and fit well into typical server-centric storage models.

These solutions fall into two categories, each of which I will examine via description of the actual design and deployment process of a representative product. The first type is a multi-port disk array that uses a SCSI-to-SCSI interface to achieve NOS independence and has excellent centralization and standardization benefits. The second type is a network-attached server appliance that uses a stripped-down NOS and speaks one or more standard file-service protocols. This article will focus on the design and deployment considerations of the type of device and some aspects of its operation, rather than an in-depth discussion of each product's merits or shortcomings.

Winchester Flashdisk: Design Considerations

One approach to storage is to consolidate several disks and SCSI interfaces in a single device. The Flashdisk from Winchester Systems integrates multiple SCSI interfaces into a single disk array, allowing several servers to attach simultaneously. Since the interface to the server is SCSI based, a vast array of NOS flavors and versions are supported without the need for specific RAID drivers. The configuration I have used is the Flashdisk OpenRAID Rackmount, configured with twelve 36.4-GB disks and two power supplies. The RAID partitions on the unit are managed independently from the SCSI interfaces, so the flexibility in configuration is very high. [Figure 1](#) shows a typical deployment of

the device.

Because the interface to the remainder of the environment is SCSI, the target NOS needs only to support an approved SCSI interface card (such as the Adaptec 2940U2W or 29160). The first environment in which I deployed this solution had Windows NT 4.0, Netware 4.2, and various UNIX flavors in active use. In almost every instance, the servers attached to each Flashdisk were of dissimilar NOS types. We installed the boot partition and data partitions of each server to the Flashdisk, so we could recover from a failed non-disk component by simply swapping out the server itself. The entire install and configuration of the server itself resided on the array. Additionally, for high-availability situations, it is simple to deploy the backup or cluster partner of a primary server to a second, separate Flashdisk unit.

Servers can be added and removed from the Flashdisk while it is operational if you're careful not to disturb the SCSI cabling. Essentially, the unit provides a centrally managed "virtual" disk to each server and its installed NOS. At boot time, the BIOS of the SCSI adapter identifies a disk of type "Flashdisk" as attached, and the geometry of this pseudo-disk is simulated by the array. The server, NOS, and SCSI interface card have no indication that an array, rather than a disk drive, is actually attached. The sparing and maintenance are significantly simplified, because there is only one RAID interface to learn, one type of disk drive and chassis to keep on hand, and no tight dependency of the set of servers on multiple hardware drive and controller types. It is not unusual to discard or fail to reuse proprietary disks and array components when a server's processor or memory expandability has been exhausted. Because the consolidated approach decouples the storage from the server itself, both the ease of reuse and the total usable lifetime of the array are significantly increased.

Winchester Flashdisk: Deployment Experience

The rack-mountable version of the array ships in a 5U enclosure with 8 full- or 12 half-height slots, 2 power supplies, and a variety of hardware cache size and controller redundancy options. Ours were configured for RAID 1+0 and yielded usable space of ~180 GB with two drives configured as in-chassis hot spares. In a RAID 5 configuration with no spares, the usable space would be ~400 GB. Our applications utilized both file-based and client-server RDBMS systems, where the file-based DB was our legacy application. The advantages of RAID 1+0 are several with only one drawback -- a 50% usable space to installed disk ratio. Performance on long reads is superior to RAID 5, and overall fault tolerance is improved because of the fully redundant mirror/stripe nature of RAID 1+0. Another advantage is that the replacement of a failed disk requires I/O only from its mirrored partner for rebuild under RAID 1+0, and not all of the other disks as in RAID 5.

After we physically racked the array, and configured and initialized it, we subdivided the usable, raw space into partitions and assigned them to the SCSI interfaces for server connections. Each partition can be assigned to one or more interfaces. Thus, this unit can support shared-disk clustering schemes, although I have not used it this way. As noted previously, this assignment can occur dynamically, so, in use, it is possible to assign some of the raw disk space to a partition and interface and leave the remainder unassigned. As servers are added to the environment, each can be added and space apportioned without disturbing the others. It is not, however, possible to extend an existing partition without dropping it at the server/NOS level and reformatting or otherwise remaking the file system.

In one case, we did add a new partition to an existing and in-use SCSI interface. After a reboot and quick format, the space was available. Since we needed it only temporarily, the partition was returned to the unused pool a few weeks later. If we had purchased a new drive cage, RAID card, and drives and then had to configure and deploy them, more staff time would have been consumed, and later there would have been excess capacity on a platform that could not easily be leveraged. This solution is essentially a SAN with limited expandability and simple configuration options. The lack of exotic Fibre

Channel or SCSI switching and absence of a high-priced software management license are a welcome change from "high-end" SAN solutions. All of the technologies used should be very familiar to administrators and engineers, since the interconnection is SCSI, and the RAID controller presents each partition to the BIOS and NOS as a single SCSI disk.

Winchester Flashdisk: Performance

Along with data growth considerations, disk I/O performance is an acutely pressing issue for administrators. As data stores grow and applications become more complex, solutions to I/O-bound applications become more critical. By pooling the disk spindles and creating partitions across them, the consolidated approach of the Flashdisk and similar devices seeks to improve performance/cost tradeoffs. If budgets were not a constraint, each server could have a dedicated array of many, many disks and a massive read-ahead cache to itself.

Before the deployment of the Flashdisk arrays, most servers had three or four disks each and very little cache on the RAID controller (due to cost considerations). The belief was that the consolidated array would better leverage a large cache and 12 disks at RAID 1+0 than 4 separate smaller arrays at RAID 5. We carefully balanced the expected load from each server and application against the projected deployment to the arrays, to prevent overloading any one of them. We also found that our I/O load patterns were bursty enough to allow very high-performance sharing of the array between our servers. The performance we observed after consolidating 12 large disks in the array and sharing it across servers was a dramatic improvement over having a few disks attached to each individual server. A strong advantage of the Flashdisk is that the cost of 12 disks and 1 high-end chassis and controller is roughly comparable to the cost of 12 disks and 4 low-end controllers and drive cages for individual servers. In many cases, the cost/GB was actually lower for the Flashdisk than for the array option for the servers we were using.

In actual testing, via the simple tactic of copying a 4.5-GB directory from the server to five other servers simultaneously, steady state performance was just above 12 MB/s and saturated a 100-Mb/s Switched Fast Ethernet link. The files were copied repeatedly from the server to drive the data out of the 256-MB cache and fully exercise the array itself. More basic testing yielded 15-16 MB/s copying data from one partition to another on the same unit, between two servers with gigabit network interfaces. The largest improvement was in backup speed, however, and we cut the total time nearly in half. The most significant improvement was a server that went from an average of 50 MB/minute to more than 200 MB/minute after the upgrade. By no means are these numbers based on any disciplined methodology or testing suite, but they do give a sense of the improvement in applications and especially backup times that can be gained by going to a dedicated high-performance storage platform.

Network Appliance Filer: Design Considerations

Another approach to consolidated storage is to eliminate the server altogether. An alternative to SCSI-based disk arrays, the set of appliances, network-attached storage (NAS) or filer devices uses a stripped down NOS to provide the protocol over which file service occurs. Many of these devices also support mounting of a partition or directory on the device for use as host to the database devices of an RDBMS platform. The key consideration here is that the appliance you choose must support all of the file-service protocols you need, such as CIFS using NETBIOS over TCP, NFS, and HTTP. The Network Appliance Filer is an example of this approach and speaks all of the protocols above. CIFS is used for integration into a Windows NT/2000 domain and allows the Filer to appear to the other clients and servers as if it were a server itself. NFS support is, of course, for UNIX environments, and the HTTP functionality allows integration of the device into large-scale Web hosting and e-commerce environments.

Since the Filer integrates as a sort of doppelganger into the existing NOS environment, the deployment is not as simple as the SCSI interface approach with consolidated disk storage. In contrast as well, the Filer offers SnapMirror, a real-time replication product for use between two filers and Snapshots, a backup technique that produces several read-only copies of the data online each day for rapid file recovery. Also, the available space can be grown easily via the addition of disks and is available without any system interruption. Because the customized NOS of the Filer provides a layer of abstraction between the CIFS, NFS, and HTTP interfaces, many operations that require reformatting of a partition or reboot of the server can be performed transparently.

All of these features come at roughly twice or more \$/GB above the server-attached solution described above. Other critical considerations are the existence and quality of support for the NOS file-service protocols you require. If Netware is your platform, this type of storage may well be useless, because I am not aware of any vendor that provides network-attached storage via the NCP protocol. The deployment of a very large data store on a Filer also all but mandates the use of a very high-speed (gigabit or ATM) interface to allow the other clients and server to make use of its I/O prowess. An example deployment is shown in [Figure 2](#).

Network Appliance Filer: Deployment Experience

The Filer is simple to deploy and configure, and the management tools and design of the device are geared heavily toward this goal. Setup is accomplished via the command line or a Web-based setup wizard. The experience is very similar to turning up a server, although care must be taken during the initial IP setup since the Filer will use DHCP to get an address. The wrinkle is that the Filer will not renew the DHCP lease, so it is necessary to either assign another IP address statically to the Filer, or use the DHCP management tools to dedicate the initial IP from the DHCP assignment to the Filer permanently.

For CIFS/Windows environments, the Filer receives a domain account to allow integration and management of security via the domain tools. Under the CIFS/SMB file-service model, the server authenticates users and enforces permissions, so the domain account is necessary to allow access to the domain-level authentication and authorization information. For NFS environments, the Filer supports NIS as a client to allow centralized administration of the files that control access permissions. With an NFS environment, unlike CIFS, the Filer exports NFS mounts and relies on the client to perform authentication of users accessing the exported directories from that machine. Additionally, if multiple protocols are in use, the differences in the protocol's treatment of file system dates, case-sensitivity, file access, and other issues must be dealt with.

In our environment, only the CIFS protocol is in use, and the predominant function of the Filer is to provide the storage platform for the applications hosted for our clients. Each Web server uses directories on the Filer for its Web root, allowing synchronization of content and applications across the load-balanced Web servers. Additionally, the SnapMirror data replication feature is used to provide a backup of all Web content and data at a second data center. [Figure 3](#) shows this configuration.

Network Appliance Filer: Performance

Both the command-line and SNMP performance management functions are excellent. The Filer command-line tool `sysstat` provides CPU, disk I/O, network interface I/O, and cache aging information via the console interface at a user-configurable interval. All of this information at a greater level of detail and granularity is also available via custom SNMP MIBs for use with a network management framework.

Our Filer provides Web content and data storage for our online banking applications in a hosted data center environment. At the time of this writing, there were 186 customers hosted in the primary data center with their data on the Filer, all of which are mirrored via SnapMirror to the Filer in the redundant data center. An analysis of the performance data from the 1 p.m.-5 p.m. CST period on a typical weekday showed 30-50% CPU utilization to provide an average of 2526 CIFS operations per second. The average disk channel and network interface utilization were 1.13 MB/s and .62 MB/s. This puts the average CIFS operation at roughly 256 bytes, which is indicative of the nature of our application. The CPU utilization is high given the relatively low total disk throughput, and assessment of the performance data shows an extremely strong correlation between CPU% and CIFS Ops/s.

As the peak load period passed, CPU load fell with CIFS Ops/s and the average cache sitting time increased dramatically, nearly doubling as the overall load on the Filer decreased roughly 33%. One last interesting observation is that while network utilization and disk read/write counters are almost perfectly correlated (not a surprise), the throughput across the network interface is almost twice that of the disk I/O. I expected to see some encapsulation and protocol inefficiency with CIFS, but a 2:1 ratio of network traffic to disk I/O was much more than I expected. Again, the small average read/write size is almost certainly the culprit here. As always, since each environment is unique, this performance discussion is geared more toward demonstrating the quality of the Filer's toolset than providing any useful benchmarking information to a prospective user. Both Network Appliance and Winchester Systems have excellent Web sites with extensive information about the performance of their products, which I encourage anyone who is interested to read and draw their own conclusions as a result.

Conclusion

Storage infrastructure choices are complicated by both the large fixed costs associated with high-availability/performance offerings and the seemingly boundless appetite of a typical enterprise for disk space and throughput. A bewildering array of non-compatible solutions further challenges IT planners and engineers alike. The goal of this discussion has been to present two alternatives as representative of typical vendor approaches to the set of storage problems enterprises face. I hope the information presented here will assist those facing this difficult set of decisions.

Adam Anderson is an IT Manager who knows the pain of slow backups and shrinking free partition space. He can be contacted at: adam_d_anderson@hotmail.com.

Copyright © 2001 Sys Admin, [Sys Admin's Privacy Policy](#). Comments about the Web site:
webmaster@sysadminmag.com