OSD: The Future of Storage

Object-based storage device (OSD) technology promises to change forever the enterprise architecture world by making upgrades easier and more affordable

Joe Straub and Kenneth Samarra

Enterprise architectures is the hot new term in information technology (IT) circles. It refers to a comprehensive system overview that includes the entire IT structure for an organization – including mainframes and permanent data storage, along with all support software, and the servers and work stations connected via WAN/LAN (wide area network/local area network).

It is appropriate that the entire enterprise is now getting attention because, in the past, ad-hoc improvements to specific pieces of computer system equipment or upgrades to system software often had detrimental effects on the overall operation. In fact, most IT professionals have horror stories to tell about upgrades costing more in time, frayed nerves, and value than if the original system had remained in place without modification.

By focusing on the entire enterprise and adjusting the components with consideration for system bottlenecks, a few well-chosen decisions can often significantly reduce the time required to install or upgrade software versions. The enterprise architecture approach also prevents unintended consequences associated with narrowly focused computer system modifications.

Object-based Storage Devices

One of the most important changes in years is now taking place within the enterprise architecture community in the form of object-based storage device (OSD) technology. OSD will likely make enterprise upgrades much easier and more affordable for a variety of users.

Just as object-based programming forever changed the software development world more than a decade ago, OSD promises similar achievements for enterprise architectures.

OSD provides a straightforward approach to connecting all enterprise data storage peripherals. OSD simply assigns more control of data movement directly to the storage devices. This means that user requests are handled at a high level (as objects) and each OSD peripheral device can respond as needed to optimize local performance. For example, in the client-server world, a request for data goes from the client workstation to the server, which then passes it to a storage device, retrieves the needed data set, and passes it down to the client workstation for viewing and use. In the OSD scenario, the request goes from the client workstation directly to the storage device, which has enough intelligence to do the work previously performed by the separate server.

The implication is this: In client-server technology, each client workstation must have software installed on it that is compatible with the server. When software is upgraded, each server and each client workstation must be upgraded, too. With OSD, the upgrade occurs at

At the Core

This article
- discusses object-based storage device (OSD) technology
- explains how OSD works within an enterprise architecture system
- explores the future of OSD and storage

the storage device.

In addition, there are currently myriad proprietary technologies to manage storage. With OSD, each of these individual technologies or operating systems can coexist within the same network.

OSD has been made possible by the availability of powerful computer capability in small chips. Much of the intelligence that was contained in a central source (the mainframe) in the past can now be shifted to each peripheral device. Although this capability has been available for many years, computer system architectures are slower to adopt it because they are built on a variety of standards that cannot be easily changed.

IT researchers at Carnegie Mellon University's Parallel Data Lab (*www.pdl.cmu.edu*), under the direction of Garth Gibson, have been working on this new technology that radically overhauls the structure of storage as we know it. Many have called OSD the "Holy Grail of Storage," mainly because it allows for self-managed storage devices.

Figure 1: A Generalized Computer System

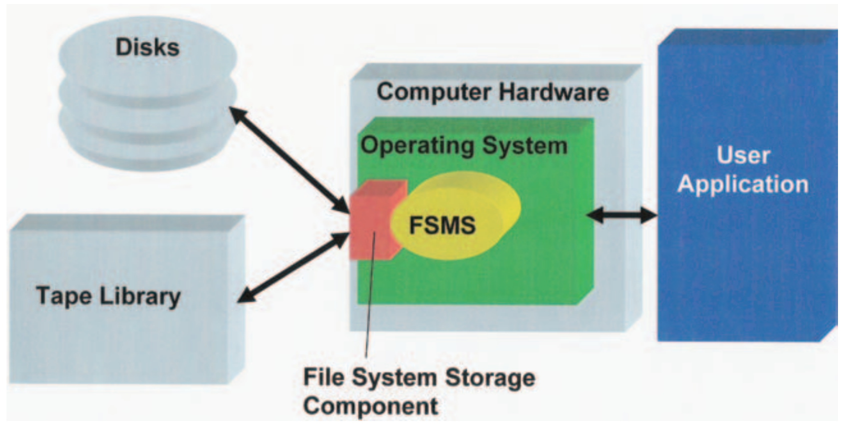
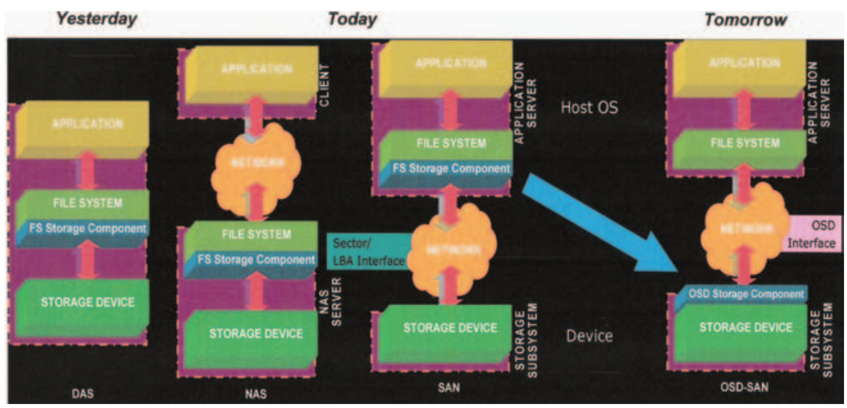


Figure 2: The Evolution of Storage Systems



Past Imperfect

Over the past 30 years, the basic components of data storage architectures have largely remained the same. There is the software application that users "order" to access certain data for a computational routine. Applications have input/output (I/O) requests of varying complexity. Requests are submitted to a file storage management system (FSMS), as shown in figure 1, that organizes and prioritizes all requests. The FSMS's file system storage component then directs the movement of data to and from physical storage devices. Finally, there is the storage device itself (typically disk or tape drives) where the data physically resides on some media for action now or in the future.

In the mainframe era of the past, storage devices were attached directly to the computers they served via a method known as direct attached storage (DAS). DAS featured various storage layers but there was no network – the storage device is attached to the host computer. The one-to-one relationship between host computer and storage device made this type of attachment simple to install and manage, with the file system storage component talking directly to the storage device (see Figure 2). Both tape and disk could be configured in this way. But because of the fundamental limitations of the architecture, DAS devices could not talk to more than one computer at a time. DAS used block-based storage, a method in which files are chopped up into chunks of data that are written as blocks.

Once networks arrived on the scene in the late 1980s, a new type of storage emerged: network attached storage (NAS). Here, a network existed between the host computer and the file system (see Figure 2). NAS hid the block-based nature of storage communication and allowed access on a file level. With NAS, all access was to individual files instead of to blocks within a file as was the case in the older DAS design. Unfortunately, because NAS tended to decentralize storage, device and request management became a nightmare. Frequent requests to the same device by different users presented fundamental problems of device request conflicts and slowness.

Out of this predicament emerged SAN (storage area networks) in the mid-1990s. SAN re-centralized storage using state-of-the-art fiber optics. SAN places the network between the file system storage component and the storage device itself (see Figure 2). This placement mitigates most device request conflicts and all information is sent via disk sectors. But the SAN approach uses block-based methods to store the data and logical block addressing (LBA) to fulfill requests.

The storage virtualization of SAN

The ABCs of Storage

Cache – an area of memory or storage used for rapid access to frequently used data

DAS – direct attached storage

FSMS – file storage management system that organizes and prioritizes all requests

File system storage component – part of the FSMS that directs the movement of data to and from physical storage devices

I/O – input/output requests

LBA – logical block addressing

NAS – network attached storage

Pre-fetching – automated retrieval of data or images based on pre-defined workflows

RAID – redundant array of independent drives

RAIT – redundant array of independent tapes

SAN – storage area network

SCSI – small computer standard interface

WAN/LAN – wide area network/local area network

Organizations

ANSI – American National Standards Institute

NASD – National Storage Industry Consortium's Network Attached Storage Devices

SNIA – Storage Networking Industry Association

was intended to fix the diversity of problems that occur when disparate devices are simultaneously accessed by multiple users. It was a prime example of how marketing hype can surpass the real benefits of a product. Virtualization failed because it was based on an outdated storage design that has existed since the dawn of the computer revolution: block-based protocols. To date, users have not been able to construct a truly heterogeneous SAN because there is no standard file system format across computer platforms. For example, MS Windows file system format is different than that of Sun Solaris.

Future Tense

The best place to insert the network for modern storage systems is between the file system and the OSD storage component (see Figure 2) (a version of the file system storage component). The interface here is thin, meaning that it does not have a large command set. The significance is that the file system storage component has been moved out of the host computer's operating system and into the storage device itself where it can be most effective. File systems are currently implemented differently across various platforms (e.g., Microsoft Windows vs. Sun Solaris). OSD enables cross-platform compatibility, a sort of universal file system. With low-level storage management routines moved into the storage device itself, greater interoperability is achieved and a true heterogeneous architecture is possible.

OSD enables a high-performance solution, as there are no potential bottlenecks in the system between the hosts and the storage devices. For IT storage expansion, OSD gets around the problem of having to back up all the data, add more space, and then restore the data – a process that can easily take hours for terabytes of data.

OSD can accomplish all this with intelligent drives. Storage devices have had on-board microprocessors for years. However, this power has been significantly underutilized. By using that latent computing power to provide intelligence to the drives themselves, the OSD model separates the file system storage component from the host computer's operating system by moving it onto the storage device (see Figure 2).

Sector and LBA will be accomplished by the device itself, leaving the operating system to communicate via a new OSD interface. With this mechanism, the host operating system need only send simple commands (e.g., "create," "open," "read," "write") to access the storage device. The operating system no longer accesses disk drives on a logical block level.

OSD is not limited to just disk drives. It will work with devices such as tape libraries as well, so that an operating system such as Windows will not have exclusive control of its associated tape devices. Essentially, every storage device that is now a proprietary file storage format on the SAN will become an object, capable of communicating with all other similar objects.

Traditionally, storage administration can cost significantly more than the storage hardware itself. This is because the storage management system's multiple tentacles must be treated with care when new updates occur anywhere in the system. In the OSD model, all metadata resides on the storage device. This allows for self-management and a more effective use of cache and pre-fetching. Not only will storage take care of growing itself, it will also perform off-line backups and manage redundant array of independent drives (RAID) functions or redundant array of independent tapes (RAIT).

OSD: Coming Soon?

It will take a few years for OSD technology to really arrive. A team of experts from Hewlett-Packard, IBM, Quantum, Seagate, StorageTek, and Carnegie Mellon University are working not only to define the technical specifications, but also to evangelize OSD's merits. But it is not enough for the hardware vendors merely to collaborate. If OSD is to succeed, the operating system vendors must also cooperate. After all, they are going to be giving up some control over these new devices. Middleware suppliers, database developers, and others will have to modify their software to accommodate the new structure.

To encourage this transition, a set of well-defined standards is necessary. Most of the OSD standards work is being coordinated among the National Storage Industry Consortium's Network Attached Storage Devices (NASD), American National Standards Institute (ANSI), and the Storage Networking Industry Association (SNIA) committees. ANSI has taken this new storage protocol into the T10 committee, which deals mainly with the small computer standard interface (SCSI) – the most widely used interface for connecting storage devices – and the OSD specification is now in its eighth revision.

OSD will do for hardware what object-oriented programming did a decade ago for software. Since its inception, the software world has been transformed into a highly efficient enterprise. It is now possible to write code just by dragging and dropping. Just like its software counterpart, OSD will have methods, properties (or attributes), and events as the core object model. Such an object model will enable flexible policies that are not possible with current technology.

Only time will tell if OSD will be adopted by the industry, but its future looks bright.

Joe Straub *is a member of high-performance computing and data storage industries. He is an independent consultant who works with companies developing advanced storage systems for safe and reliable access to permanently retained data. He may be contacted at* joseph.straub@verizon.net.

Kenneth Samarra *has been an active member of the Storage Networking Industry Association (SNIA), Fibre Channel Industry Association, and InterNational Commitee for Information Technology Standards' T11 Technical Commitee that produces interface standards for high-performance and mass storage applications. He is also CEO of Kairos Information Technology. He may be contacted at* ken@kairos.com.

Read More About It

Samarra, Kenneth. "DAS, NAS, SAN and Beyond …" Available at *www.ncne.nlanr.net/training/techs/2002/0728/presentations/200207-smarra1.htm* (accessed 10 November 2004).

———. "The Holy Grail of Storage." Available at *http://romulus.gsfc.nasa.gov/msst/conf2002/PPT-PDF/c16-sama.pdf* (accessed 10 November 2004).

———. "LaserTAPE: The Future of Storage." Presented at THIC Meeting at the National Center for Atmospheric Research, Boulder, Colorado, 11-12 June, 2002. Available at *www.thic.org/pdf/Jun02/lots.ksamarra.020611.pdf* (accessed 10 November 2004).

———. "OSD: The Holy Grail of Storage?" *InfoStor*. September 2000. Available at *http://is.pennnet.com/Articles/Article_Display.cfm?Section=Archives&Subsection=Display&ARTICLE_ID=81800&KEYWORD=samarra* (accessed 10 November 2004)

Storage Networking Industry Association. OSD Technical Workgroup. Available at *www.snia.org/tech_activities/workgroups/osd* (accessed 10 November 2004).