Seagate
We turn on ideas

TP-536 • From: Seagate Research • April 2005

Technology Paper

The Advantages of Object-Based Storage—Secure, Scalable, Dynamic Storage Devices

Introduction

This paper introduces the Object-Based Storage Device (OSD) interface, an extension to the SCSI protocol that was developed by the Storage Networking Industry Association (SNIA) over several years and ratified by ANSI in September 2004. This new command set can be carried over the same physical interfaces as SCSI is today[1], but it represents a major shift of functionality and intelligence into the storage device. Instead of being spread throughout the many layers of a modern storage system, metadata is associated and stored directly with individual data objects, allowing it to be automatically and transparently carried between layers and across devices.

We will describe the benefits of OSD, the environments where OSD can be deployed and a brief history of the work that led us to Object-Based Storage.

This new interface allows storage devices (whether individual disc drives, disc arrays or other storage controllers) to explicitly manage space allocation, a function traditionally performed in the host file system or database software. The interface also provides for strong security protection enforced by storage devices—each request must be explicitly authorized. Finally, the interface allows attributes to be associated with data objects. These attributes can give the storage device detailed information on the characteristics of a particular data object: how it is to be handled and how it will be used.

An end user that deploys a storage system containing OSD-enabled devices will see several benefits, among which are improved scalability, native security, enhanced reliability and dynamic reconfiguration.

Benefits—Improved Scalability

The Association of Storage Networking Professionals (ASNP) estimates that there were over 1 million full-time or part-time storage administrators in 2004. A survey of over one thousand ASNP members indicates that twenty percent of them manage over 100 terabytes of data. Assuming 100 gigabytes per disc drive, this is over 1,000 individual disc drives. Many Fortune 500 companies are known to be approaching 1 petabyte of data (10,000 individual drives). A clustered system at a U.S. national research lab is bringing 2 petabyte under a single file system. As currently being built at Lawrence Livermore National Laboratory (LLNL), this would be nearly 20,000 drives. Industrial research labs are approaching similar data sizes, for example British Petroleum (BP) uses 600 terabytes or 6,000 drives for petroleum discovery. Building systems of this size requires multiple levels of abstraction and management, each of which must be configured and operated separately.

[1]The OSD command set does require long CDB (command descriptor block) support. The OSD-1 command set uses 200-byte CDBs, where the traditional SCSI Block Commands (SBC) command set uses a maximum of 16 bytes.

Most modern storage systems, large and small, contain many layers of metadata. Individual disc drives map logical block addresses (LBAs) into variable-density drive zones, remapping around failures and managing error-correcting codes. Disc arrays organize blocks from multiple discs into stripe units and logical units (LUNs), mapping blocks for reliability and performance. Various types of virtualization appliances contain metadata to remap blocks for replication or device heterogeneity. File servers, or NAS appliances, map blocks into files within a single system and manage access control lists. When organized as a NAS head or a distributed file system, this metadata is shared among separate metadata servers remote from the hosts. In addition to the space management metadata at various storage system layers, individual applications add policy and descriptive metadata of their own.

Since all this metadata is spread throughout the system, it can become difficult to manage. With an OSD interface, much of metadata can be associated and stored directly with each data object and is automatically carried between layers and across devices. Space management metadata—which blocks belong to which objects—becomes the responsibility of the storage device.

The use of OSD-enabled storage devices in a system allows metadata layers to be collapsed, simplifying the storage system and improving scalability. With reduced processing requirements and overhead, a single server can manage a much larger number of devices than with traditional block-based interfaces. This allows users to build storage clusters for very large installations or to use more economical systems than is possible with block-based storage.

Benefits—Native Security

The improved security provided by OSD-enabled devices is most valuable when the OSD interface is implemented on individual disc drives, providing defense-in-depth at the lowest system levels. The granularity of object-level security provides a natural mapping to how storage end users think about protecting their data.

With OSD, request authorization is performed directly by the storage device. In order to gain access to data on an OSD-enabled device, a logical capability check is performed for each request. This can prevent both accidental access for undesirable requests (for example, misconfigured machines) and malicious access for unauthorized requests (for example, hackers, whether internal or external). The value this level of protection would provide has been illustrated recently in multiple public incidents.

In a well-publicized case, IBM lost a disc drive with valuable customer data. That incident cost IBM over $500,000 in hard costs alone, before considering any pending lawsuits. More recently, Wells Fargo Bank offered $100,000 for the return of a lost laptop, no questions asked. The concern in both cases was the exposure of lost customer data. It should be no surprise that customer data is much more valuable than the disc drives it resides on. With today's block-based devices, there is no way to prevent unauthorized access to data. If physical access is possible, any data can be retrieved from an individual disc drive.

A group of MIT researchers put this to the test by acquiring over one hundred used disc drives from various sources. They were able to find a trove of valuable data that should never have been left accessible. The researchers were able to take SCSI and ATA drives that had been used in various sensitive applications, attach them to a PC and retrieve data—in some cases even after the drive had been formatted. With an OSD interface, the individual disc drive is able to validate each read and write request. A computer system accessing an OSD-enabled device must supply a credential (called a capability) that the device can validate before allowing access. If a disc drive is removed from a system, attaching it to a PC will not enable access. Since the PC system cannot supply the necessary authorization, the drive might as well be empty. No access to data is possible.

Benefits—Dynamic Reconfiguration

With an OSD interface, metadata is associated directly with each individual data object and can be carried between layers and across storage devices. Each object has provisions for millions of individual attributes which are divided by vendor or system layer. When objects pass through a certain system layer or device, that layer can act on the values in the attributes that it understands. All other attributes are passed along un-modified and not acted upon. This means that, for example, objects marked as *High-reliability* can be treated differently than objects marked *Temporary*.

Attribute	System Layer	Possible Action
High-reliability	Drive AND Array AND Host	Protect this object with higher reliability
Temporary	Drive OR Array OR Host	Do not protect with redundancy; do not migrate
Read-only	Drive	Do not allow writes to this object
Sequential	Drive OR Array OR Host	Optimize this object for sequential streaming
Random	Drive AND Array AND Host	Optimize this object for random access

Figure 1. Possible object attributes and their usage

The ability to associate semantic metadata directly with objects makes it possible for storage devices to flexibly manage the data in the object for performance, reliability or other options.

In addition, since data objects are known to the storage device, and since management attributes can be associated with data objects, a system can more automatically move data objects among storage devices or provide data-specific functionality. Objects can be migrated between devices in their entirety, with metadata intact.

Benefits—Enhanced Reliability

One of the problems with storage being managed at the block level is that if there is an error in a block, it is impossible to determine what part of the file system is affected. It may be that the particular block in error does not even contain any active data. This could happen, for instance, when the array controller is organizing data for a particular RAID level or when performing a physical backup. In addition, some file system metadata structures are internally duplicated by the file system itself. Should the array or disc have an error in one of these areas, it does not have the ability to locate the redundant copy and recover the data.

One of the most difficult reliability problems that can surface in a user site is a second drive failure during a RAID rebuild. When a failed drive has been replaced and its contents are being reconstructed on the replacement drive, any error while reading the other drives in the RAID set will result in the entire file system being lost. At this point, all the user can do is restore from backup tapes—even though the actual data block in error might well be an unused block or in the redundant metadata mentioned above.

OSD-enabled disc drives know what space is unused and will not attempt to rebuild it. If there is an error in the metadata structures that the device maintains, it can recover these by maintaining its own internal copies. Even if the worst happens and an unrecoverable failure occurs in an active file, the file in error is readily identified and just that file can be restored from a backup. In fact, the OSD-enabled device can identify the specific byte range lost, and in some cases the recovery might be limited to restoring just a few bytes. This represents a huge savings in time and money over doing a complete file system restore for a single file or small number of files.

When specific object attributes are used, such as the *High-reliability* attribute mentioned earlier, an OSD-enabled device might selectively make multiple internal copies of the user data.

In addition, an OSD-enabled device can use its knowledge of unused space and object layout to scan for errors and correct them without a danger of losing data.

Applicability—High-Performance Clustered Storage

The scalability advantages of OSD-enabled devices are demonstrated in large-scale clustered storage systems. These systems provide a single, shared storage system for a large number of networked servers. Such systems are increasingly popular in high-performance computing applications. A cluster of commodity workstations or servers replaces a special-purpose supercomputer. These systems often use the Linux kernel and open-source operating system and middleware components.

The Lustre file system from Hewlett-Packard and Cluster File Systems, Inc. is such a clustered file system that uses a variant of object storage technology. Lustre Object Storage Targets (OSTs) share a common heritage with the OSD interface and provide similar benefits. Lustre is used in numerous high-performance computing sites today, scaling to hundreds of storage nodes in prototype installations with plans for thousands of storage nodes in future systems.

The Panasas ActiveScale Storage Cluster also contains object storage devices that use an OSD interface. This system achieves impressive scaling and performance by eliminating the server bottleneck between hosts and storage devices. By changing the ratio between storage devices and metadata devices in the system, workloads with different requirements can be flexibly supported.

Applicability—Archival Storage

Vendors of content-addressable storage (CAS) systems also use abstractions similar to the data objects standardized in the OSD interface. Using proprietary and open APIs, data objects are placed into these systems and retrieved in their entirety. In several of these systems, the object identifier is a hash value derived from the object contents—hence the term content-addressable. The name is derived from the contents. These systems also associate attributes with data objects and are most often used in a Compliance or Information Lifecycle Management (ILM) application. Attributes may specify the lifetime of a data object (when it is to be deleted) or the provenance (where and when it was created, and by whom). Both of these functions can benefit from the ability to associate attributes directly with data objects as data moves from device to device and between systems. In addition, the strong security of OSD could allow such systems to make verifiable claims about data provenance.

Applicability—Heterogeneous Networked Storage

In addition to the scalability benefits for large-scale clustered systems, OSD-enabled devices can also benefit smaller systems through improved device interoperability. Since space management metadata is handled by the storage devices, it becomes easier to move devices between different file systems and hosts. It also becomes easier for hosts with different operating systems to share the same storage devices. In this way, an OSD-enabled device is similar to network-attached storage (NAS). However, an OSD-enabled system still allows direct access between hosts and storage devices—as is common in storage area networks (SAN). This eliminates much of the server bottleneck often experienced in NAS systems, which impacts performance and flexibility. The same storage devices can be shared by different hosts and file systems, and safely depend on the security model of OSD to decouple data and access control.

Reseachers at IBM have experimented with the benefits of object storage as part of their SAN file system and distributed file system efforts. The use of OSD would allow these systems to more flexibly optimize, configure and secure storage to ensure separate resources for distributed hosts.

History—from Research Project to ANSI Standard in Ten Short Years

The first SCSI disc drive was introduced in 1983, and the standard was ratified by ANSI in 1986. In the 20 years since then, this basic protocol has not changed significantly. There have been massive changes in the physical interface between storage devices and host computers—from wide SCSI to fast SCSI to Fibre Channel (FC) to Serial Attached SCSI (SAS). The initial interface speed in 1983 was 5 megabytes per second, today's state of the art is 400 megabytes per second. The first SCSI drive had a capacity of 5 megabytes, while today, the SCSI logical interface is used on 400 gigabyte disc drives, as well as on 40 terabyte disc arrays from EMC, Hitachi Data Systems and IBM. However, the logical interface—command set—has only seen minor additions during this time.

The interface that is today standardized as OSD-1 originated in research work done by Carnegie Mellon University in a government-funded research project called Network-Attached Secure Disks (NASD) that began in 1994. Over the next few years, this effort was supported and advised by a group of industry collaborators, including Seagate®, organized by the National Storage Industry Consortium (NSIC). One result of this work was a draft interface specification that was submitted for standardization to the committee responsible for the SCSI interface, ANSI T10 as project T10/1355-D. Over the next several years, this interface was modified and extended by the OSD Technical Work Group of the SNIA with varied industry and academic contributors, culminating in a draft standard to T10 in mid-2004. This standard was ratified in September 2004 and became the OSD-1 command set. Technology never stands still; the SNIA group continues to work on further extensions to the interface.

Objects, a New Addition to the Storage Network

The diagrams below illustrate how OSD fits into the Shared Storage Model created by the SNIA. An OSD-enabled storage device straddles the block and file/record layer, providing basic space management (part of the file/record layer) as well as reliable storage (the block layer), but relying on an external server for file and security metadata (as NAS does).

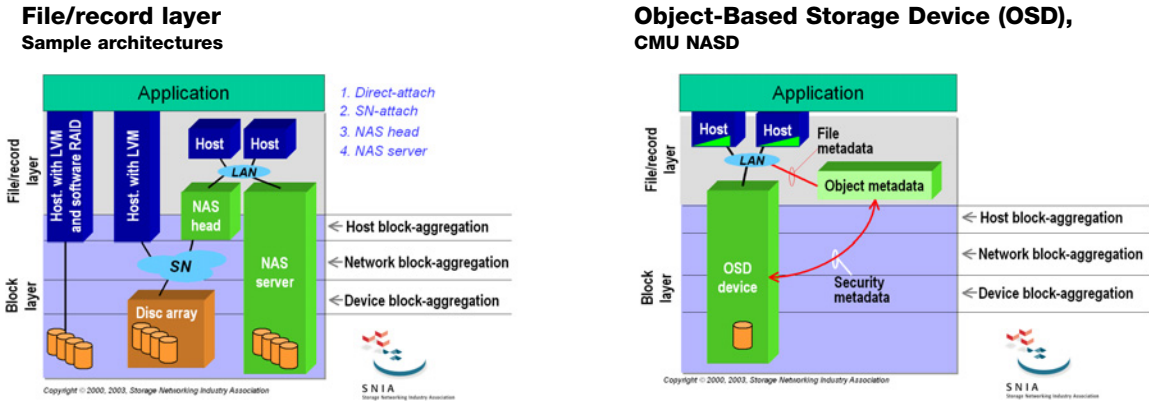

Figure 2. SNIA Shared Storage Model—OSD architecture

The second diagram above shows only one possible configuration of OSD-enabled devices. Like block-based SCSI, the OSD protocol can be deployed as an interface to individual disc drives, disc arrays, aggregation appliances and NAS devices, whether on the local area network (LAN) or storage network (SN). The benefits of scalability, security, reliability and dynamic reconfiguration can be realized in all of these scenarios.

Objects, a Logical Change in System Software

The diagram at right illustrates how OSD-enabled devices fit into the operating system software of a host. In a traditional system, the file system is separated into a storage component (space allocation and data access) and a user component (namespace, access control, concurrency management).

In an OSD-enabled device, the functionality of the storage component is moved to the storage device, while the user component remains in the host.

In a system with a single storage device, this allows the device to better optimize data layout, optimize data access and protect user data. It also allows the device to be more easily moved between hosts.

In a distributed storage scenario with many storage devices, the file system user component may be located on multiple machines to enable shared access. Two variants of such a system are shown in the diagrams below from the SNIA Shared Storage Model.

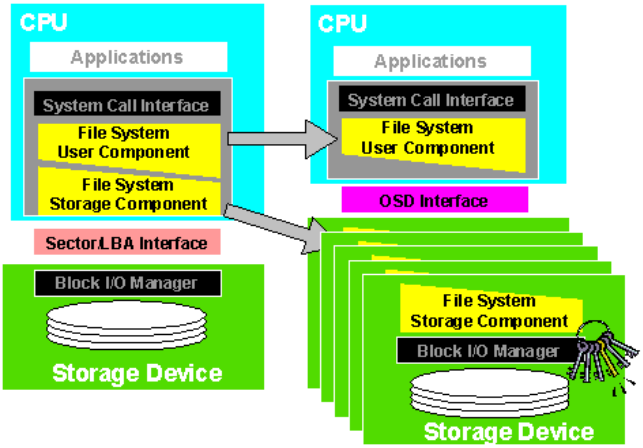

Figure 3. OSD in a typical software architecture

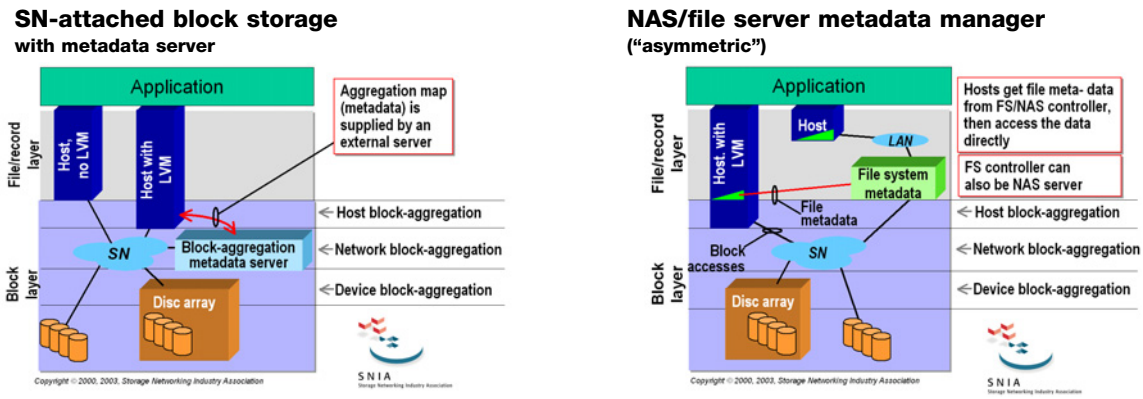

Figure 4. SNIA Shared Storage Model—Metadata managers with OSD

Shared metadata might be managed by an aggregation server within the storage network (block layer) or by a file system metadata server (file/record layer). OSD-enabled devices are able to store metadata directly with object data, and metadata servers are only required to mediate shared access (concurrency management and access control policy). This allows data objects and storage devices to be more easily moved between metadata servers, and reduces the total work that must be done by the metadata servers—lowering system and management costs.

Objects, a Logical Progression of Device Interfaces

OSD is a logical extension to the SCSI interface which first introduced logical blocks as a basic storage abstraction.

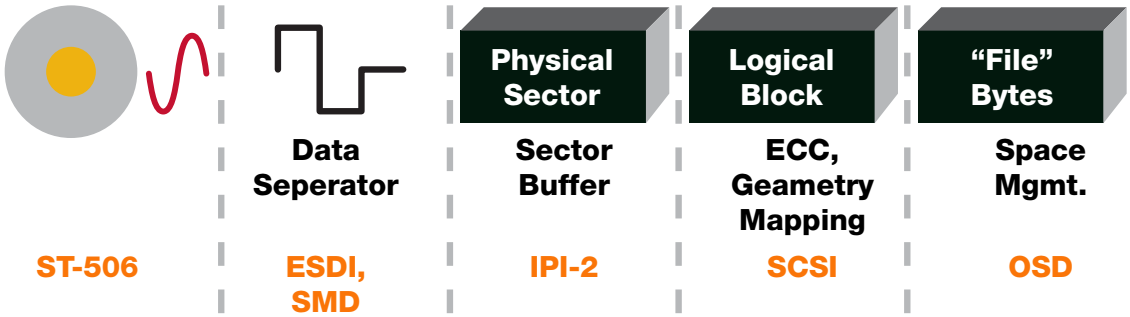

Figure 5. Progression of interface intelligence

Logical, byte-addressed objects with explicitly associated metadata represent a more powerful abstraction for end users and for those who build storage systems, and more powerful freedom of action for storage device designers. They also represent a logical progression of interface technology from the days of ST-506, when hosts were responsible for directly managing the physical aspects or disc drives, through sectors and logical blocks to objects today.

Conclusion

Today, too many IT resources are tied up in low-level storage details and functions because the storage environment exposes too much complexity. This not only results in a lot of wasteful activity, it also represents a big opportunity cost. Some of this complexity has been offloaded to various storage management software. Reducing this complexity further depends on the ability to delegate low-level functions deeper, to reduce complexity at the source.

This article has introduced Object-Based Storage Devices (OSD), a fundamental change in technology that enables previously higher-level infrastructure activities to be delegated to lower-level storage devices. This decreases management traffic above the storage device, enabling greater scalability and performance.

This delegation also improves manageability and enables functionality that is not well-achieved in today's systems—including dynamic reconfiguration, interoperability, native security and enhanced reliability that fits the end-user context. Both IT users and system builders benefit with more competitive solutions that allow them to focus more on application and data characteristics, and less on low-level storage management.

Further details of Seagate technology offerings, the interface standard, and the past and future work of the SNIA OSD Technical Work Group can be found by following the references listed at the end of this article.

We encourage you to find out more about this exciting new technology.

REFERENCES

- Seagate OSD Technology Leadership, specials.seagate.com/osd
- SNIA OSD Technical Work Group, www.snia.org/osd
- ANSI web site for INCITS 400-2004, www.t10.org/pubs.htm

[i] David Joachim, "Storage Pipeline: A Look at the Storage Professional," Network Computing, 16 September 2004.

[ii] Bill Boas, "NNSA Advanced Simulation and Computing Program (ASC)," Storage on the Lunatic Fringe, SC'03 Panel, 19 November 2003.

[iii] Keith Gray, "HPC at BP: Storage Strategy," Storage on the Lunatic Fringe, SC'03 Panel, 19 November 2003.

[iv] Mel Duvall, "Case 062: Memory Loss," *Baseline*, March 2003.

[v] Mark Rasch, "The Wells Fargo example," SecurityFocus online, December 2003.

[vi] Simson L. Garfinkel and Abhi Shelat, "Remembrance of Data Passed: A Study of Disk Sanitization Practices," IEEE Security & Privacy, January/February 2003.

[vii] Lustre Project, www.lustre.org

[viii] Panasas, Inc., www.panasas.com

[ix] Deni Conner, "Fixed content storage grabs users' attention," Network World, 26 May 2003.

[x] Pushan Rinnen and Nick Allen, "Object-Based Archive Products Emerge in Information Life Cycle Management," Gartner Dataquest, 30 August 2004.

[xi] Jai Menon, David A. Pease, Robert Rees, Linda Duyanovich, and Bruce Hillsberg, "IBM Storage Tank—A heterogeneous scalable SAN file system," IBM Systems Journal, July 2003.

[xii] Ohad Rodeh and Avi Teperman, "zFS - A Scalable Distributed File System Using Object Disks," IEEE Mass Storage Systems & Technologies, April 2003.

AMERICAS **Seagate Technology LLC** 920 Disc Drive, Scotts Valley, California 95066, United States, 831-438-6550
ASIA/PACIFIC **Seagate Technology International Ltd.** 7000 Ang Mo Kio Avenue 5, Singapore 569877, 65-6485-3888
EUROPE, MIDDLE EAST AND AFRICA **Seagate Technology SAS** 130-136, rue de Silly, 92773 Boulogne-Billancourt Cedex, France, 33 1-41 86 10 00

Publication Number: TP-536, April 2005