Technology Insight Paper

Object Storage: Out of the Shadows and into the Spotlight

By John Webster

December 12, 2012

Evaluator Group

Enabling you to make the best technology decisions

Object Store Essentials

Why the Object-based Storage Device

The storage density of a single enterprise disk drive has grown several orders of magnitude since its birth in the late 1950's while the disk drive's block interface has remained essentially the same. Challenging this interface is an increasing demand among enterprise users for storage subsystems that can scale economically and efficiently into the multiple petabyte range. As that happens, the block storage interface is increasingly seen as an impediment to the development of petabyte-scale disk storage arrays. Traditional NAS filers also suffer from the same inability to scale for two reasons: First, capacity of non-clustered filers is limited to a single NAS array[1]; second, even "scale out" NAS that leverages clustered controllers to go beyond the single array limitation uses block storage behind these controllers.

A far more scalable and versatile interface based on stored objects, rather than blocks, can remove this impediment. Object-based storage is now coming of age since its inception in the late 1990's and is taking its place alongside block (DAS and SAN) and file-based (NAS) storage. The commonly used name for this technology is Object-based Storage Device (OSD).

Unlike blocks, storage objects are logical collections of bytes on a storage device[2] (a disk drive for example) and are of variable size. Therefore, they can be used to store and directly access a wide range of larger data structures such as entire files, database tables, digitized images, and even entire file systems and databases. Like a block, an object is a basic unit of storage. And like a file, a stored object is identified to an application and accessed using metadata. However, the metadata component of object storage is far more comprehensive in its content versus file-based storage. Unlike both block and file, the stored objects also have an ID that is unique to each object (see Figure 1 below). As such, they are versatile storage "containers" with an interface that is reminiscent of files, but is also much more expressive and independent of application and geographical boundaries.

[1] NAS is also block based at the level of the disks within a NAS array, but uses a file interface in between the application and the disks.
[2] "Object-Based Storage," Meisner, Ganger, Reidel, IEEE Communications Magazine, August 2003.

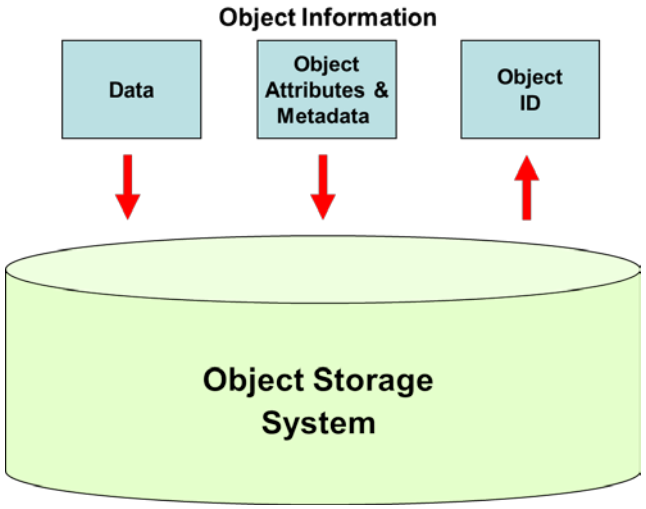

Figure 1: The OSD stored object consists of data, metadata, and an object ID; all of these attributes can be exposed to an application.

OSD and CAS

The first commercially successful implementation of OSD was as Content Addressable Storage (CAS) of which EMC's Centera was an example. Centera ran a polynomial algorithm against the content of a stored object (a medical image file for example) to generate a unique ID or "fingerprint" for that and only that object. The object ID then became the basis of a content address that Centera would use to find the object on disk when access was needed by an application.

Centera found widespread applicability in archival applications, particularly those that were sensitive to regulatory compliance and legal inquiry, because objects, once stored, were presumed to be immutable. If the object was accessed after the ID was created and the object's data was changed, the change would force a Centera to create a new ID and treat the resulting object as new and unique.

However, Centera required applications that accessed its stored objects to use an API that was specific to Centera. Therefore existing archival applications, for example, had to be modified to use Centera and new applications would have to support the API in order to store and access Centera's data. Additionally, it suffered from the same capacity and versatility limitations mentioned above.

Object Storage 2.0

A standard SCSI command set for OSD was developed by SNIA and subsequently approved by the International Committee for Information Technology Standards (INCITS) in 2004. OSD Version 2 now includes support for snapshots, collections of objects, and improved error handling. We believe that this and follow-on versions will drive further development and user adoption of OSD.

We note that while there are many different commercially available implementations of OSD, vendors have been reluctant to tag these products as OSD-based, perhaps because users will automatically associate OSD with CAS and assume the two are one and the same. As noted above, CAS is a particular implementation of OSD, developed specifically for immutable archival storage applications. The OSD implementations now appearing are far more versatile.

A common attribute of the OSD product implementations is that they present a "flat" namespace to the accessing application. File systems and relational databases structure data to make it easier to find from the point of view of the application. However, the organizational structure in between the application and the data can limit storage scalability. Object storage frees data from limitations imposed by organizational structure, creating a "flat "and potentially infinite namespace, but one that uses a metadata repository in place of the file system or RDBMS structure. Therefore, an important requirement of these subsystems is the ability to preserve the OSD array's performance as it scales upward in capacity.

Another common attribute is the replacement of RAID-based data protection with a collection of techniques known variously as Forward Error Correction, Erasure Coding, Fountain Coding, and Reed Solomon codes. Because OSD enables significantly greater disk array capacities, high density (1TB+) disks are typically used to build the array. However, a RAID-based storage subsystem can take days to rebuild a single failed, high density drive. Erasure Coding generates algorithms that encode data objects into redundant check blocks that are spread widely over the entire storage pool addressed by the OSD array controller. Erasure Coding requires only a subset of the check blocks to restore the original data object, as determined by the user-specified policy. In this way, multiple devices (disk drives and/or system nodes) can fail simultaneously without data loss or affecting data availability, making this type of array-based data protection far more scalable than RAID because it allows data to be dispersed across independent hardware elements ('nodes').

Throw the File System Away?

It is worth noting that while an OSD array does not necessarily require something like a file system structure to function, that doesn't mean that it can't benefit from the addition of some form of organizational or content management model. File systems have undergone decades of evolutionary

development. As a result, catalogue integrity has been optimized making them desirable owners of metadata. Therefore, continuing to use this catalog while leveraging the scalability and independence of object storage can be beneficial.

In addition, file systems that scale to billions of objects can be used in conjunction with an object-based disk array with beneficial results. Some applications need more single stream performance than highly parallelized OSD array can provide without assistance. The file system can assist the array controller in finding the data the application needs to access among millions and potentially billions of objects, organizing the storage workflow, and speeding access time.

It is also the case that existing applications that don't "speak" OSD natively will need some kind of interface. Traditional file systems (like NFS and CIFS) can have a critical role to play for IT administrators wanting to leverage object storage without trying to modify existing applications that are built around CIFS and NFS access.

Here we look more closely at applications that benefit from the use of OSD as well as additional structural models implemented "on top" of OSD.

OSD for Streaming Content

Meeting the growing demands for availability of digitized video content typically requires the ability to find and deliver it quickly. This is true for a growing list of applications including video surveillance and social media. Media and Entertainment (M&E) companies, for example, are converting existing vaults to "live archives" allowing them to monetize previously dormant digital assets by repackaging them or enabling online delivery of this content to consumers. A petabyte-scale, OSD disk-based live archive could support electronic content distribution in real time to multiple sites, and could be scaled upward in capacity as requirements dictate. With a file system or database structure on top of OSD-based disk storage, access to video objects is significantly faster because the structure organizing the data can pre-sort the objects rather than search across a flat namespace that could contain millions of objects. Additionally, such a system can be a shared storage resource for existing applications, breaking-down video storage siloes.

OSD for NFS and CIFS

There are many existing NFS- and CIFS-based mission critical applications that now create fast-growing volumes of unstructured content. Microsoft Exchange and SharePoint are two that IT administrators often point to as constantly needing additional storage capacity. Using OSD instead of block or file-based storage could remove scale limitations imposed by traditional RAID arrays as discussed earlier. However, these applications will need a standard storage access interface. Neither NFS nor CIFS access

OSD natively. Consequently, the OSD array needs to support these file systems in order to work for these applications.

OSD and Enterprise Hadoop

Hadoop is now under review within enterprise IT as a more flexible and lower cost alternative to traditional data warehouses. It is an open source software framework with an embedded file system (Hadoop Distributed File System – HDFS). The challenge for IT administrators as Hadoop makes it charge into the data center is this: HDFS assumes that storage is distributed across Hadoop cluster nodes as DAS to provide data locality and reduce latency. Networked/shared storage (SAN and NAS), while scalable, resilient and very familiar to enterprise IT, is typically not used because it adds another network to the Hadoop cluster thereby increasing latency. Therefore, the familiar data protection, management and preservation features common to shared storage are not available to users of the standard Hadoop distributions. The good news for enterprise users is that it doesn't *have* to be that way.

Emerging is the use of networked storage as a "secondary" storage tier for Hadoop clusters—storage that essentially functions as a data protection and active archival storage layer in conjunction with Hadoop's DAS-based primary storage layer. While doing so requires either replacement of HDFS with a file system that is Hadoop-compatible or a modification to HDFS, we believe there is significant value available to enterprise Hadoop users that can result from pursuing this option, particularly when OSD-based storage is considered.

OSD can serve as the basis for a highly scalable, performance oriented, secondary storage tier for Hadoop. What value does this tier add to the cluster?

1. Hadoop administrators typically add Data Nodes to the cluster in order to add storage capacity. However, Hadoop Data Nodes are processing nodes that also have their own DAS. A more rational strategy would be to create a secondary storage tier that contains all data that is older than 90 days. Because of its potential for large scale, OSD is ideal for this secondary storage tier. This way, the need for more storage capacity can be mostly absorbed by an OSD-based array.
2. Data residing on each of the Data Nodes could be periodically copied to an OSD array that is shared by all data nodes. This procedure would result in a set of point-in-time data copies that could be used to recover the cluster from an outage or a data corruption event. To date, Hadoop has no easy way to recover from human or machine-generated data corruption.
3. An OSD array that supports existing applications could be presented to the Hadoop cluster as data source. Consider the Microsoft applications from the above discussion. Users wishing to run Hadoop analytics processes against Exchange data, for example, could simply present a copy of this data to a Hadoop cluster that is also attached to the OSD array, eliminating the need to move data

from Exchange storage into Hadoop. Hadoop is also being used to analyze video content in surveillance applications that require quick response. The time required to move large volumes of video data from these systems into the Hadoop cluster would add unacceptable latency. The time required to simply present data to Hadoop is a significant consideration when running analytics processes against large volumes of data.

Conclusion

As we have noted, many storage vendors have been reluctant to expose the fact that their large-scale disk storage subsystems are built on OSD implementations, perhaps because few applications can access OSD natively resulting in a perceived mismatch among users. So in order to present this technology to existing applications, vendors bridge this gap with an interface which is often a standard file system protocol. This masks the underlying benefits of OSD. However, doing so enables archival storage for example with a native ability to share data across disparate applications, in effect creating a private archival storage cloud.

Many situations continue to emerge where a petabytes-scale disk storage subsystem can be of value. These include archival applications where continued access to data is needed even though it is in an archival or static state, and a growing list of analytics applications now commonly referred to as Big Data. We believe that a better understanding of OSD will expose its wide applicability to IT administrators.

About Evaluator Group

*Evaluator Group Inc. is dedicated to helping **IT professionals** and vendors create and implement strategies that make the most of the value of their storage and digital information. Evaluator Group services deliver **in-depth, unbiased analysis** on storage architectures, infrastructures and management for IT professionals. Since 1997 Evaluator Group has provided services for thousands of end users and vendor professionals through product and market evaluations, competitive analysis and **education**. **www.evaluatorgroup.com** Follow us on Twitter @evaluator_group*