Object Drive
in enterprise storage


WITH RISING DEMANDS for scale-out storage systems, we rush into an era of booming object storage solutions such as Hadoop, OpenStack Swift, and Ceph among others. However the imagination and the spirit of innovation do not stop there, instead it finds a new home in an area where traditional approaches were deemed invincible.

Innovators of object storage approaches correctly observed a rising tide starting with cold storage where the combination of low cost devices and good-enough performance are receiving more and more attention.

In order to meet these challenges, the traditional SCSI interface has been replaced with TCP/IP interfaces over standard Ethernet and the typical storage stack of drivers, volume managers and file systems has been replaced with direct object access to the drive.

Object Drives will be the better choice for storing heterogeneous unstructured and semi-unstructured data. As the growth of data accompanies the rapid development of information technologies, Object Drives are expected to thrive ubiquitously in storage tiers, not only in cold storage. With Object Drives, the traditional intelligent functions, which use to reside on storage systems likely, will be moved to the drives.

So who would buy object drives? System vendors and integrators are expected to purchase object drives because of their capability to radically simplify the software stack, reducing complexity and costs to create storage systems.

Hyper-scale datacenter customers will use object drives to reduce acquisition costs using commodity hardware and open source software. Finally, Enterprise IT will follow the hyper-scale customers to achieve the same results on a smaller scale.

So how do object drives address the issues that these customers are experiencing? CPU resources are being disaggregated and moved to the drives themselves where they can be optimized to the task (I/O) without the need to be general purpose (a growing trend in many other areas of technology as well e.g. cell phones, smart wearables, etc.).

The combination of CPU, Memory, Network and Storage is one component and is managed as such. Volume shipments will drive the cost to the customer down over time like other commodities. Since a variety of media (persistent memory, solid state drives, traditional and Shingled Magnetic Recording rotating platters) is part of the drive, the non-storage resources can be matched and sized appropriately for the I/O requirements of that media.

However, object drives are not a panacea. With disaggregation comes an order of magnitude increase in the number of nodes, which require some form of management (even if management software were provided with the drives, similar to storage systems of today, the burden on administration is still increased). Other storage system features, such as multi-tenancy, end-to-end data security, etc. are currently left unaddressed for object drives.

The Storage Networking Industry Association (SNIA) plans to address these issues and promote system interoperability with the formation of the new Object Drive Technical Working Group (TWG). Currently, SNIA defines two major types of object drives:
Key Value Protocol Drives require minimal additional CPU and Memory over that in a typical storage drive.

Data Node Drives provide sufficient CPU and Memory resources that object node software (such as CEPH) can be hosted directly on the drive.

In both cases, the interface abstracts the actual media technology, allowing for performance and capacity tradeoffs.

Key Value object drives are an emerging drive type that is equipped with Key Value interface software, typically over IP. This type of object drive allows storage system software to access data in the form of keys with corresponding values of arbitrary length.

These types of drives are mainly used in green field applications where new software is being written to take advantage of this new paradigm. Hyper-scale customers are already doing this, the Key Value organization of data is growing in popularity as demonstrated by the popularity of non-relational databases such as Cassandra NoSQL.

In the traditional architecture for this type of data organization, the file system/database software translates the key/value to traditional Logical Block Addresses (LBAs) as shown in Figure 1. This means that operations need to configure and manage the whole Key Value structure and convert a Key Value pair to Logical Block Addresses in the SCSI/SATA protocol, at last storing them on the traditional disks.

On the other hand, object drives bring some changes to the traditional object storage architecture; the new architecture is shown in Figure 2.

A major benefit that this new architecture can bring to the storage industry is that there is no complex protocol translation. Besides eliminating the typical storage stack, applications have direct access to the desired interface without the need for Cassandra or NoSQL middleware making the overall storage solution simpler and more efficient.

SNIA is referencing SFF-8639 (which is a high speed multifunction plug and receptacle connector designed for use as a common connector system supporting SAS, SATA, Ethernet and PCIe based devices) in a
developing specification. The Small Form Factor (SFF) Committee (a committee separate from SNIA) has added two new pinout definitions in the corresponding SFF-9639 standard. One is the Open Compute Project (OCP) Kinetic pinout for the Kinetic open source project based drives and the other is titled SNIA Ethernet Drive.

Some differences between Open Compute Project (OCP) Kinetic and SNIA Ethernet Drive pinouts are described below:

- **Presence Detect Pin**
  The OCP Kinetic pinout uses one hardware pin to detect whether the object drive is connected. SNIA Ethernet Drives use an inband Ethernet handshake protocol instead to make sure the object drive is connected and operating.

- **I2C**
  OCP Kinetic pinout adds an I2C interface. Storage systems can do out-of-band management through this I2C interface. For the SNIA Ethernet pinout it is expected that in-band management will be used for this purpose, so no I2C interface is needed.

- **Power Disable**
  OCP Kinetic does not define a Power_Disable pin, as OCP Kinetic can use an I2C command to shutdown an object drive. SNIA Ethernet Drive pinout reserves a Power_Disable pin. In this way, a storage system can shutdown an object drive (and bring it back up) through this hardware pin.

A Key Value (KV) framework has been proposed in order to bring the benefits of object storage as well as setup a more lightweight, extensible interface for object drives.

The three main components in the framework as shown in Figure 3 are:

- **Key Value Library (KVL).** Storage Operation System can access the object storage provided by Object Drive after connecting to KVL successfully.
- **Pool** is a collection of objects, which is used to manage objects as a flat hierarchy.
- **Objects** are the Key Value pair stored in the system.

Based on the Key Value framework, applications provide detailed callback functions to operate each component (Object, Pool and KVL). Currently there is also an open source effort to help developers to implement the Key Value (KV) Framework design. Please refer to [http://huaweistorage.github.io/ananas/](http://huaweistorage.github.io/ananas/) for details.

The imagination and power of innovation won’t stop here, and we may see the possibilities of KV Framework being adapted to even more layers. With Object Drives, the traditional intelligent functions, which used to reside on storage systems, will likely be moved to the drives.

For more information about the work that SNIA is doing on Object Drives, please visit [https://members.snia.org/apps/org/workgroup/objecttwg](https://members.snia.org/apps/org/workgroup/objecttwg)