"KVDrive" Internet Protocol Drive for Object Storage Systems

● TANAKA Shingo ● GOTO Masataka ● Philip KUFELDT

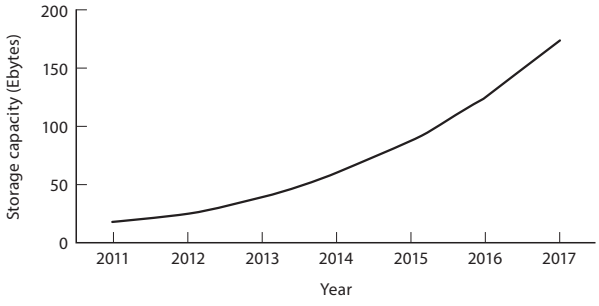
Object storage, a new storage system that enables large-scale storage systems to be constructed and managed at low cost, has recently been spreading in response to the explosive growth of digital data. Current object storage systems have several issues, however, including the need for a large number of servers when the system is enlarged due to its immature architecture, which relies on currently existing hardware and software.

Toshiba has developed the "KVDrive," an Internet Protocol (IP) drive storage system, in order to solve these issues. With the KVDrive, the need for a large number of servers is eliminated and the software architecture is simplified due to (1) the adoption of an architecture that allows direct access from clients, and (2) a key-value (KV) application programming interface (API). These features make it possible to realize a high-performance system with low total cost of ownership (TCO) using the KVDrive.

1. Introduction

With the spread of smartphones, social networking services (SNSs), cloud computing, and other digital innovations, the volume of digital data has been growing at a tremendous rate. Furthermore, since all the indications are that data processing technologies capable of handling a huge amount of data for big data analytics and other applications will become increasingly widespread, explosive growth of the world's total storage requirement in the enterprise and cloud computing sectors is forecast—a roughly fivefold increase over the four years from 2013 to 2017 (**Figure 1**)[1].

However, worldwide investment in information technology (IT) systems is expected to grow only about 3% annually[2], increasing slightly more than 10 percent over the same period, which is a drop in the ocean compared with the projected data volume growth. It is therefore critical to develop technology for building and operating storage systems at low cost in order to keep up with rising demand for storage capacity. To meet this demand, a new storage technology called "object storage" is entering widespread use.



E: exa (10^{18})

∗Based on Worldwide File- and Object-Based Storage 2013-2017 Forecast[1] from International Data Corporation (IDC)

Figure 1. Trend of global storage capacity for enterprise and cloud service use.

Explosive growth of the world's total storage requirement in the enterprise and cloud computing sectors is forecast: a roughly fivefold increase over the four years from 2013 to 2017.

This report describes the current situation and issues concerning object storage and presents the characteristics of the KVDrive, an IP drive storage system developed by Toshiba in order to resolve these issues.

2. Overview of object storage

Object storage has been designed primarily for cloud storage and other large-scale storage applications that are used at the back end of Web services.

Hard disk drives (HDDs) used in storage systems handle data in block units, and each block is addressed using a scheme called logical block addressing (LBA). Conventional storage systems generally use a redundant array of independent (inexpensive) disks (RAID) to set up HDD redundancy and provide application interfaces using filesystems to user applications (**Figure 2**(a)). Since RAID is controlled by one central controller, RAID-based systems can accommodate, at most, only several tens of HDDs. Filesystems are also designed to run on a single computer (i.e., in a scale-up architecture). This means, in order to scale up a system, a different technology is necessary to enable distributed control of multiple servers. Generally, users had no option but to rely on expensive dedicated storage systems from

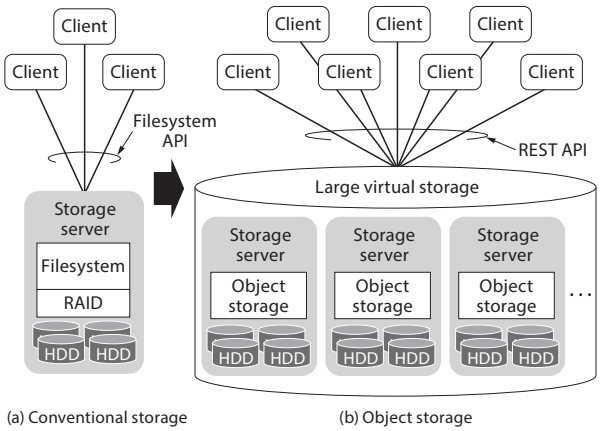


Figure 2. Configurations of conventional storage and object storage systems.
Whereas conventional storage systems use a scale-up architecture, object storage systems use a scale-out architecture.

storage vendors.

On the other hand, some advanced IT companies have been developing unique technologies on their own to realize large-scale virtual storage at low cost. With the gradual disclosure of these technologies to the public, their essence crystallized into object storage and became a de facto technology.

Figure 2(b) shows the outline of object storage. Object storage has three major characteristics:

(1) Simple API
Object storage generally handles Web content such as webpages, images, sound, and movies. Basically, it suffices to be able to read (GET), write (PUT), and delete (DELETE) these data of arbitrary length (objects). Object storage does not need any directory structures that are offered by conventional filesystems or complicated functions such as file opening and closing and access privilege control. The object storage API dispensed with these functions and uses a simple, representational state transfer (REST) interface, which is similar to a protocol for transporting Web content. That is, the interface is designed to get, put, and delete data objects of arbitrary length to/from non-hierarchical, flat storage space.

(2) Scale-out architecture
In object storage, an object identifier is typically input to a certain hash function, and the object data is stored in a storage server pointed to by its result. Clients make the same calculation in order to determine which server to access among the distributed storage servers. Thus, clients can directly access the target server to get, put, and delete objects without accessing any management server. Basically, this results in a scale-out architecture, which enables scaling of the system simply by adding new storage servers in parallel. In general, a separate management server is utilized for synchronization whenever storage servers are added or removed, thereby changing the system configuration. However, it is unnecessary to send queries to the management server each time a client accesses data in the storage servers. Thus, the management server does not create a performance bottleneck.

(3) Fault tolerance
Generally, an object storage system provides triple redundancy in which every data item is stored in three different storage servers. For example, even if an HDD in a given storage server fails, a target data is automatically returned to a client from another server, with the failure hidden from the client. Thus, object storage provides outstanding fault tolerance. Generally, each storage server has a certain level of intelligence to make it possible to efficiently transfer data between storage servers without any intervention of a management server.

Furthermore, open-source software called Ceph[3], which enables more sophisticated object storage, is becoming widespread. Ceph provides clients with not only object storage benefits but also conventional interfaces. **Figure 3** shows the configuration of Ceph.

While Ceph is based on object storage technology, it provides applications on clients with not only an object interface but also block and file interfaces to its common object storage cluster called reliable automatic distributed object store (RADOS). Consequently, Ceph makes it possible to provide a large-scale storage system for conventional applications simply by adding storage servers in parallel. A wide range of applications can benefit from the object storage advantages of scalability and fault tolerance. Ceph is highly compatible with an open-source cloud computing software platform called OpenStack[4], which can be deployed as infrastructure-

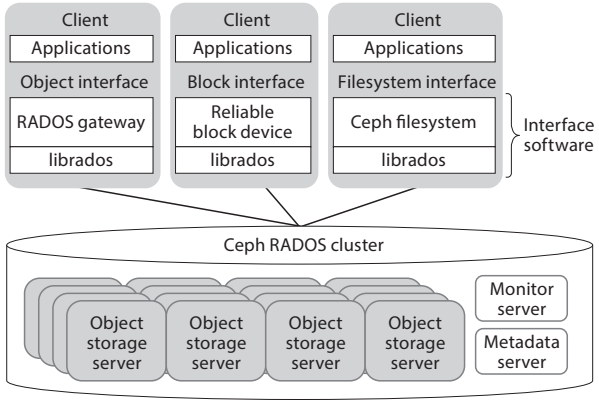


Figure 3. System configuration of Ceph distributed object storage.
Ceph allows three types of storage interface: object storage, block storage, and filesystem interfaces.

as-a-service (IaaS) Thus, Ceph and OpenStack are often used in combination.

3. Object storage issues to be resolved

As described so far, object storage simplifies the building of a large-scale storage system. At present, however, object storage has the following issues:

(1) Many servers are necessary.
Typically, general-purpose servers must be used to configure storage servers. Therefore, in order to increase storage capacity, it is necessary not only to add storage devices but also to add those servers to connect them to the network, which incurs extra costs.

(2) Object storage still depends on a filesystem.
Object storage software installed on storage servers is generally configured on a filesystem. Essentially, the only requirement for object storage is the ability to get, put, and delete arbitrary-length objects, and it is unnecessary to use complex data structures such as a filesystem. However, because a filesystem provides an easy means of handling arbitrary-length data, it is still used in object storage systems. The downside of this is that a filesystem degrades system performance.

(3) Users have yet to make full use of storage devices.
At present, object storage is primarily used for applications that do not require a very fast access speed (e.g., for archiving and cold storage). Thus, a common assumption is that an object storage system consists solely of HDDs. Although Ceph and other developments have been expanding the possible application of object storage to higher-performance areas, few systems are available yet that utilize solid-state drives (SSDs) and non-volatile memories such as battery-backed-up DRAM.

4. KVDrive

In order to resolve these issues, Toshiba has developed the KVDrive, an IP drive storage system (**Figure 4**). Major characteristics of the KVDrive are as follows:

(1) IP drive with a 3.5-inch form factor
Despite being the same size as a 3.5-inch HDD, the KVDrive serves as a microserver that incorporates both storage devices (SSDs and HDDs) and a system-on-a-chip (SoC). Its physical interface is Gigabit Ethernet(†). The KVDrive is an IP drive with network functionality based on the Transmission Control Protocol/Internet Protocol (TCP/IP) that allows direct access of GET, PUT, and DELETE from clients . Furthermore, software run on object storage servers can be installed on the KVDrive to make it function as a complete object storage server. Therefore, the KVDrive eliminates the need for general-purpose storage servers and thus contributes to a reduction in system TCO (**Figure 5**).

Figure 4. KVDrive.
Despite being the same size as a 3.5-inch HDD, the KVDrive serves as a microserver that incorporates both storage devices and an SoC.

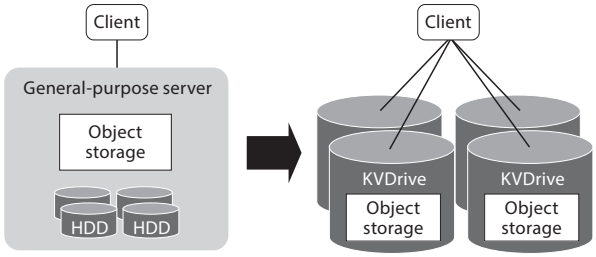


Figure 5. Configurations of conventional and KVDrive-based object storage systems.
The KVDrive eliminates the need for general-purpose storage servers that are required by conventional object storage.

(2) KV API
The KVDrive provides a KV data interface. KV is a data representation that identifies data with a key. Both keys and data (called "values" in this parlance) can be of arbitrary length. It is possible to get, put, and delete values by specifying keys. The KV API is highly compatible with object storage that is specifically designed to handle arbitrary-length data using GET, PUT, and DELETE. The use of the KV API eliminates the need for a filesystem, making it possible to simplify the software hierarchy and thus improve system performance.

(3) Optimal control of SSDs and HDDs
The KVI API also provides a storage control function for the optimal control of SSDs and HDDs. Previously, it was up to users to develop technology to make full use of different types of storage. Each time a new type of storage appeared, users had to work on optimizing it. The KVDrive, with its API that is abstracted one level more than conventional LBA, provides users with the technology to utilize those storage devices, leveraging Toshiba's strength as a vendor of both SSDs and HDDs. Thus, users no longer need to develop technology to utilize each new type of storage medium. Now,

new types of storage media including shingled magnetic recording (SMR) HDDs and storage class memories (SCMs) are expected. The KV API helps make the best use of their advantages to deliver high-performance systems in a timely manner. The KVDrive provides the maximum data throughput of approximately 110 MBytes/s, which is equivalent to the bandwidth limit of Gigabit Ethernet(†). This performance is derived from the use of SSDs and is approximately twice as fast as conventional IP drives available now.

The specifications described above in (1) and (2) are compliant with the Kinetic Open Storage Platform(5).

5. Conclusion

We have developed the KVDrive, an IP drive storage system for object storage systems. Although the KVDrive has a drive form factor, it is designed to be directly accessible from clients as object storage. Consequently, the KVDrive helps reduce the number of servers that have conventionally been required to accommodate storage drives and thus reduces system TOC. Furthermore, the KV API eliminates the need for the conventional filesystem and incorporates provisions for using various types of storage media such as SSDs and HDDs. Since the KV API facilitates the use of new storage media, it helps users enhance system performance.

Our next step is to work toward practical use of the KVDrive. To this end, we will standardize the API and hardware interface specifications and proceed with the development work, primarily targeting Ceph, which is expected to become widespread in the near future.

References

(1) IDC (International Data Corporation). 2013. *Worldwide File- and Object-Based Storage 2013-2017 Forecast*. USA. IDC.

(2) Gartner. "Gartner Worldwide IT Spending Forecast." Gartner website. Accessed July 19, 2015. http://www.gartner.com/technology/research/it-spending-forecast/.

(3) Inktank Storage. "ceph." Ceph website. Accessed July 19, 2015. http://ceph.com/.

(4) OpenStack Foundation. "OpenStack: The Open Source Cloud Operating System." OpenStack website. Accessed July 19, 2015. https://www.openstack.org/software/.

(5) Seagate Technology. "Seagate Kinetic Open Storage Platform." Seagate Technology website. Accessed July 19, 2015. http://www.seagate.com/jp/ja/solutions/cloud/data-center-cloud/platforms/.

• *Ethernet is a trademark of Fuji Xerox Co., Ltd.*

TANAKA Shingo

Chief Specialist. Storage Solution Promotion Department, Storage Products Division, Semiconductor & Storage Products Company. He is engaged in the research and development of storage application products.

GOTO Masataka

Group Manager. Storage Solution Promotion Department, Storage Products Division, Semiconductor & Storage Products Company. He is engaged in the research and development of storage application products.

Philip KUFELDT

Senior Manager. Storage Product Business Unit, Toshiba America Electronic Components, Inc. He is engaged in the research and development of storage application products.

Notes:

"-inch" means the form factor of HDDs or SSDs. It does not indicate drive's physical size.

This publication may include Toshiba's original estimation, and/or data based on Toshiba's evaluation.