

Reducing File Backups with Object Storage

WebTech Q&A Session – May 12, 2010

1. What about physical disaster recovery that is, offsite?

Nearly every object store includes replication in its core capability set. This, in effect, lets the customer meet their offsite requirements that drive tape backup while permitting real-time access to this data in the face of disaster or a more localized recovery need. In addition, these systems will usually apply their ability to validate the content during the replication procedure, ensuring that the offsite data was not corrupted along the way.

2. How does an object store protect you from accidental deletes, which we could recover from tape?

This will depend upon the feature set of the object store. Nearly all have retention management capabilities for objects stored as Write Once Read Many (WORM). The retention setting allows the object store to reject any delete request until the specified period of time has passed.

Some systems, like the Hitachi Content Platform (HCP), can also store objects that change. For this type of data, HCP has an object versioning feature where old versions of the object are kept on the system as new ones arrive. These older versions are kept available for restore operations. A delete request on a versioned object is permitted, but if it was a mistake, the version prior to the delete will still be available for restore. HCP versioning feature includes automatic deletion of old versions based on customer configurable time periods.

3. So, would I backup to an object store instead of tape, create content on an object store or tier data to an object store from primary storage? Or, how do I get data into an object store in the first place?

Data going to an object store will usually get there from either an enterprise application or from a customer homegrown application.


Nearly every enterprise application, be it email, document management, NAS systems, digital imaging, etc., will have integrations with object stores and provide ways to target content to move to the object store, remove it from the backup stream, and keep it available online. For example, NAS systems often have a tiering feature where file stubs are kept, but the file content is migrated to an object store.

For homegrown applications, a well designed object store will have a simple, web compatible (usually REST) API that a customer can use in the application.

On top of these API supported mechanisms, some DOS support common file data transport and storage standards such as CIFS, NFS, WebDav, NDMP and SMTP.

4. You mentioned the system “evolving” with technology, can you elaborate?

To be effective, an object store must stand the test of time. These systems need to scale to very large sizes in order to deal with the volume of unstructured data being created now and in the future. In addition, this data will often carry a retention requirement of many years. Because of these needs it is simply unacceptable to require expensive and time consuming data migration projects and “forklift”



upgrades every 2 to 3 years as components of the solution reach their end of life. Equally important, if a system will be used for years, even decades, it must have a way to keep up with ever improving computing and storage technologies. Can you imagine if your object today had to rely on 40MB disk drives from 1990?

The Hitachi Content Platform's design in this regard is a great illustration of the concept. It is able to support adding new, more advanced storage types to an already installed system without disruption, running right alongside the older storage technology. For example, when an old array in the system is beyond its end of service life, HCP can be told that it should mark the array for retirement. HCP will stop writes to that device and, quietly behind the scenes, objects on the system will move themselves off the old array on to newer storage in the system. Applications continue to have full access to the objects and there are no changes to their references to the objects. When all objects have migrated themselves, HCP will alert the customer that the old storage is ready to be removed physically. The process has completed with no service engagement required and no downtime.

5. Why does a distributed object store have less management overhead?

A distributed object store is really a type of application. They treat much like appliances the storage and other hardware components they use in order to present a massively scalable manager of data objects. As such, they take on the burden of configuring and monitoring these components. For example, they will typically virtualize the storage that they leverage, removing an administrator from planning provisioning, raid groupings and multipathing. As we described in this WebTech, they require no backup removing the management headaches associated with that activity. DOS continually self-validate their data and automatically perform repairs when necessary. And finally, by being the holding store for unstructured fixed content which is a very large percentage of an organizations data, they allow other higher management cost storage infrastructure components to be smaller.