Do you Always Need to Backup?

by Curtis Preston, Senior Analyst

Typically backup is the copying of data from one type of storage system (e.g. primary storage) into another type of storage system (e.g. backup system) for the purposes of recovery in case the first copy becomes unavailable. This definition works with a lot of different types of backup in use today. It, of course, includes traditional backup, whether that backup is sent to disk or tape. It also includes backup methodologies such as continuous data protection (CDP) and near continuous data protection (near-CDP), both of which involve replicating data out of the protected storage system into a different storage system. All of these different types of systems are based on one premise: that the original storage system is not able to protect itself. The question is: what if the storage system could protect itself? Would you still need to deploy a traditional backup regimen?

Why is it important to backup data?

Human error is the most common reason one might need to recover data from the backup system. Humans corrupt data by modifying it in a way they did not intend. They also accidentally delete files, they drop tables in a database – or even delete entire VMs. All of these scenarios count on the backup system. Sometimes humans corrupt or delete data because they are trying to harm the company in question. The most recent type of example would be *ransomware*, where malware encrypts a significant number of files and then sends a ransom demand in order to unencrypt the files.

The other reason we need backups include failures of the hardware for various reasons. Disk drives and even flash drives occasionally fail. RAID helps mitigate this risk but it can never remove it. In addition, RAID does not scale well and with the advent of larger capacity drives, rebuild times literally have moved from hours into days. During rebuild time, an additional read error could prove catastrophic and place data at risk of loss. Yet, for primary / real-time data, this approach does deliver high performance albeit with a high infrastructure expense. Fire, lightning or other disasters can take out entire sites. This is why we have backups and why they need to be off site.

To recover from each of these issues, the backup system stores historical versions of files, databases, and servers, so IT can use those historical versions to restore the data if it was damaged or deleted by a human or disaster. Many organizations go so far as to have multiple copies of backed up data, "just in case". IT can also use the backup system to separate this copy from its original so that if some sort of disaster took out the original site, the data would be available in another location. Perhaps, however, with a different kind of storage system that can protect itself, these high cost and redundant activities could become a thing of the past.

Do I have to backup all data to ensure its availability?

Tier 1 data typically requires data protection and generally includes some type of backup implementation. Not only does the data need to be in another system and available for immediate restore, it might also need to be on a system that can immediately take the place of the primary system in every way, including the same performance and availability of the original system, in other words a failover scenario. But what about data that does not require the same level of service as primary data? This is data that was created and used for a brief period of time but is not in real-time usage at present and probably would not be looked at it again for a long time – if ever. Reference data is a perfect example of such data. Log files, previous versions of files that are no longer needed for production, previous runs of tests that are being kept for reference purposes are all good examples. For various reasons, it's not possible to delete this data, but it doesn't warrant being placed on a tier 1 system which will then cause backing it up to a tier 1 backup system. Instead, such data can move into a self-healing object-based storage system that will still protect data and ensure accessibility just as a traditional backup system would do. The difference is the reduced complexity and cost of on an object storage system, consolidation of the number of copies of data being created, and its ability to grow to cloud scale. If the object store is the backup target for primary data, the need for multiple cold copies likely could be collapsed to one or none.

Some object-based storage solutions support versioning, so that if an object becomes corrupt or deleted, the previous version of the object is a few mouse clicks or commands away. Many systems can also guarantee immutability of the object to protect against malfeasance.

In addition, certain object-based storage implementations also support distribution of the data across multiple drives, shelves, nodes, or geographies so that the loss of one or in some cases more of any of these does not affect data availability. This is typically done by one of two methods: replicating each object to multiple sites, or the use of erasure coding techniques to accomplish the same goal. Erasure coding requires a slight additional parity overhead. A given object is divided into *N* shards or chunks. This N is expressed as X/Y erasure coding, where X shards are created for each object, and Y shards are the number that can be unavailable before data becomes inaccessible. For example, in 18/5 erasure coding, an object is split up into 18 shards that are distributed across multiple disk drives; thus a user can access the object even if up to five of the shards are not available. In other words, the first 13 shards retrieved are sufficient to deliver the object. In a well-designed solution that distributes these 18 shards across 18 drives spread amongst multiple shelves, nodes, and locations, the loss of a disk shelf, controller node, or even an entire site would not compromise data accessibility.

Achieving the same level of data protection with full object replication in a triple mirror scenario would require three separate copies and an overhead of 300 percent, and that is just the capital expenditures. There will also be a significant uptick in operational expenditures. An 18/5 erasure coding implementation requires only one full copy with a parity overhead of about 60 percent. There is a performance penalty for this type of implementation, but for most reference or archive data the system performance acceptably meets the business need.

So if the storage system can automatically recover from logical corruption, human error and malfeasance, media failure and site failure, what do you need backups for? Some would argue that you still need backups to protect against some type of rolling code failure; however, there does not appear to have been any such incidents with object or tier 1 storage systems in the history of computing. The truly paranoid will still have interest in some type of cold backup, just in case. But the type of backup system that even the truly paranoid would ask for in this scenario would only require a single backup of data as it migrates into the system, with no further backups required. Thus, even if the organization prefers a cold offsite copy even if the data is stored on object storage, they could still perform significant consolidation of their backup capacity. This means it would be a very inexpensive system to own and operate. And, of course, there are some who would argue that even a one-time cold backup of object storage is unnecessary.

Be selective

A significant portion of the data in any given enterprise is data that really doesn't warrant tier-1 storage, or even tier-2 storage. Each of those tiers would need to be regularly backed up to the backup system, at significant cost. Companies that can identify the data that really doesn't need this type of storage can achieve significant savings by moving it into self-protecting object storage that would not need to be regularly backed up to the backup system. Companies can then be very selective when deciding which data goes on primary storage, saving significant amounts of money.

Summary

An object-based storage system that has data immutability, high data durability, and geographic distribution of data can supply both the storage needs and backup needs of data in a single solution; therefore, data stored in such systems probably does not require the traditional multiple layers of backup copies. Even those who are not convinced of that would be satisfied with a single initial backup of new data to some type of backup system, which would cost a fraction of traditional backup approach. If companies could identify data that would be appropriate for such a system and move that data off of primary storage and into object storage, they could save a significant amount of money.

Sponsored by HGST, a Western Digital brand

Storage Switzerland is the leading storage analyst firm focused on the emerging storage categories of memory based storage (Flash), big data, virtualization, cloud computing and data protection. The firm is widely recognized for its blogs, white papers, and videos on such current technologies like all-flash arrays, deduplication, software-defined storage, backup appliances, and storage networking. The "Switzerland" in the firm's name indicates our pledge to provide neutral analysis of the storage marketplace, rather than focusing on a single vendor or approach.