Quantum®

WHITE PAPER

QUANTUM LATTUS:

Next-Generation Object Storage for Big Data Archives

CONTENTS

Executive Summary. 3
The Limits of Traditional Storage Solutions 3
Object Storage for Increased Scale and Flexibility 5
Limits of Pure Object Storage 7
Quantum Lattus: The Future of Object Storage 8
Summary 9

EXECUTIVE SUMMARY

The biggest storage challenge facing most organizations today is managing the balance between cost and accessibility in increasingly large repositories of data.

Technology is driving increasingly sophisticated data capture tools, which in turn is driving an increased appetite for high-quality, rich data sources. At the same time, increased computing capabilities and analysis techniques are driving additional value out of historical data. Together this means that organizations seeking to drive more value out of their data need to collect and store more data for longer periods. As a result, many traditional data management solutions, including data protection and archive platforms, need to be revisited to ensure the increased accessibility and capacity demands of modern data growth. Some new technologies—including object and cloud storage technologies—can provide scalable data management solutions exceeding the limits of traditional storage, but they can also introduce new operational and functional constraints.

To respond to these challenges, Quantum introduced Lattus™—a new generation of storage that incorporates the strengths of object storage while maintaining operational and functional flexibility. This next generation of storage provides organizations the certainty that their data is safe while future-proofing infrastructure for long-term data growth and the next generation of big data applications.

THE LIMITS OF TRADITIONAL STORAGE SOLUTIONS

Struggles to Scale Efficiently

The foundation of traditional storage systems is RAID. RAID provides increased performance and capacity by spreading data across multiple disks, extending the size of volumes, and thereby data sets, with additional disks. RAID can also provide increased data protection by mirroring data or adding data checksums. The development of disk drives capable of storing hundreds of gigabytes or terabytes has effectively allowed management of data sets on a single, consistent logical unit sized up to 30 terabytes.

The growth of data, though, has outpaced the technology of disk drives. RAID has proven effective at managing the integrity of data in a single RAID group of 4 to 12 disks, but petabyte-sized data sets either require use of disk groups larger than 12 disks or they require dividing data across multiple RAID groups. The former option significantly increases data risk through hardware failure, and the latter increases the cost and complexity of managing data consistency and integrity across the multiple units.

Larger disk sizes also lengthen the rebuild time when failures occur. The failure of a 3TB or larger drive can result in increased risk and degraded performance for 24 hours or more while rebuilding on a replacement drive. While the introduction of dual- and triple-parity RAID can mitigate this risk of data corruption, adding more parity checks is not a scalable solution, as it undermines the cost and performance benefits of RAID.

The Problem of Bit Errors

As data sizes increase, they also make the bit error rates of typical disk drives a practical concern. Normal drives have an expected bit error rate of roughly 1 in 100 trillion. While that makes a random bit error highly improbable for megabyte- or gigabyte-sized data files, for larger data sets it is a very practical concern. For example, when reading data from a full 3TB disk RAID array with 10 disks, there is roughly a one in three chance of running into undetectable data corruption caused by random bit errors. RAID has no mechanism to proactively detect bit errors, and bit errors occurring during a RAID rebuild will lead to data loss.

The Complexity of RAID Upgrades

Finally, RAID relies on disks in a group having a consistent size and layout. Storage upgrades to denser disks typically require building a new RAID array and moving data from old to new RAID groups. These upgrades can require significant coordination and downtime. RAID also requires that all the disks be local—normally on the same attached controller. This means that RAID offers limited protection against node-level failures, and no protection against site-level disasters.

Increased Protection but Decreased Efficiency with Replication

Replication is deployed to address some of these shortcomings of RAID solutions. Replication can enable better data integrity, recoverability, and accessibility, but it reduces the ratio of usable storage space and introduces new complexities which limit the overall cost effectiveness of a storage environment.

Typical replication architectures will copy data from one site to another. The files can be recovered from the secondary location in the event of a failure at the primary location. Depending on the completeness of a replication solution, organizations may be able to do a complete business recovery from a replica at their secondary site.

Unfortunately, maintaining replicas can be extremely complex and expensive. Replicas need to be maintained far enough away from the primary data so that they can provide adequate protection in the event of regional disasters, but near enough to maintain the synchronization of files necessary for recovery point objectives. This requires expensive high-bandwidth networking dedicated to storage replication in addition to the expense of the disk storage systems at the replication site. Additionally, replicated files need to be synchronized in a state that guarantees data integrity. This requires flushing and quiescing data writes on the primary storage during the synchronization. These operations are complex to coordinate, administer, and maintain.

Overall, replication can provide additional recovery and integrity protection, but it can more than double the cost of the storage infrastructure.

OBJECT STORAGE FOR INCREASED SCALE AND FLEXIBILITY

Traditional storage systems store data in a hierarchical directory of folders and files mapped to blocks on disk. This mapping often comes with constraints on the number of files or the size of files that can reside in a specific directory. It can also lead to "hot spots" of data where highly utilized groups of files map to specific disks or RAID volumes. Those high-use RAID sets have increased risk of failure and bit rate errors, but the high utilization also makes it difficult to maintain integrity through replication or backup.

Object storage offers a fundamentally different approach to data storage. Object storage presents a namespace of simple key and value pairs. By leveraging the scale-out capabilities of IP networks, this addressing allows data administrators to scale digital data sets to nearly boundless size. By using a flat namespace and abstracting the problem of addressing from the problem of physical storage, object storage systems also have a lot of flexibility in how and where data is stored and preserved.

Load Balancing and Distribution

In object storage systems, applications read and write data using simple network-based protocols. The most common is HTTP, which allows clients to address their data through a key utilizing simple PUT and GET semantics. This means that much of the addressing and routing of requests can be offloaded to high-performance network switches and routers, making it easy to distribute data over multiple storage nodes with virtually no overhead. Some object storage mechanisms add data distribution features like data load balancing and rebalancing across nodes, to minimize risk and increase performance. Other more sophisticated systems allow heterogeneous disks and nodes. Object storage system capacity can typically be expanded without downtime, performance degradation, data migrations, or rebuilds.

Another benefit of the network-friendly protocols and the simple addressing and distribution logic of object systems is that data can easily be distributed over multiple data centers or sites across the globe. While accessing data over long distances always introduces latency, the network protocols used by object storage systems can be designed to minimize the impact of latency.

Simple Object Storage Data Protection

Most first-generation object storage algorithms also provide a simple form of data protection, where the data is copied to three or more nodes simultaneously. Retrieval can be made from any of the three copies. This does result in less effective usable space when compared to RAID5, but it is comparable to the usable space available to a RAID5 with a RAID5 remote mirror. Finally, whereas replicated RAID storage systems are only available as read-only mirrors and need to be managed to maintain synchronicity of replicas, object storage nodes distributed at multiple sites can all be active and maintain synchronicity with very little complexity. Consequently, many object storage systems have inherent cross-site access and recovery capabilities built in with fewer administrative challenges.

Erasure Code Data Protection

Recent object storage implementations are designed with more sophisticated data distribution and protection algorithms. These algorithms—known as erasure codes—allow greater levels of data protection with greater efficiency. Erasure codes have been used for decades in space communications to preserve communication transmission integrity of streaming data. RAID works by slicing up data into a fixed number of data blocks and checksums and writing each chunk or checksum onto an independent disk in a defined set. Erasure code algorithms transform data objects into a series of codes. These codes are much like parity blocks, only there are no corresponding data blocks. These codes are then dispersed across a large pool of storage devices, which can be independent disks, independent network-attached storage nodes, or any other storage medium. The unique nature of erasure coding algorithms is that while each of the codes is unique, a random subset of the codes can be used to retrieve the data. Using erasure codes liberates the data storage from the constraints of fixed-size RAID groups.

The nature of the coding algorithm also allows a wider range of protection policies. Data protection using erasure codes is expressed as the ratio of two numbers. The first is the minimum number of codes over which the data is dispersed. The second represents the maximum number of codes that can be lost without losing data integrity. This ratio is referred to as the durability policy. As an example, with a durability policy of 20/4, the object storage system will encode each data object into 20 unique codes, and then it will distribute those codes over 20 separate storage nodes, often from a pool of a hundred or more independent nodes. When retrieving data, the object storage system will read the codes from the storage nodes and decode them to recover the original object. Since the system only requires 16 codes to decode the original object, data is still accessible even in the event of the loss of 4 of the 20 independent nodes.

These algorithms offer a lot more protection flexibility versus traditional RAID. They allow a much wider array of policies that can protect from disk, node, or even site failures all on the same scalable syste—with much less waste lost to redundancy than RAID and replication solutions. The algorithms apply data integrity through individual codes rather than through whole disk sets, allowing levels of data protection in the same object storage system. Organizations can tailor their durability policies to different data protection requirements without hardware changes and without copying data out of the system.

Object storage using erasure coding can also proactively manage against bit errors. Nodes may automatically test the data integrity of the individual codes and can automatically generate replacement codes for those that are found to have errors. Erasure code systems also use low-level hardware monitoring—like the SMART system built into every modern disk drive—to detect pending disk and node failures, and will proactively generate replacement codes on different disks or nodes.

LIMITS OF PURE OBJECT STORAGE

Addressing Restrictions

Object storage relies on unique IDs to address data. This means that in order to retrieve the data, the client needs to already know how to address the data. Without that key, the client has no context to retrieve it. A good metaphor is that object storage is like a valet parking system. You can conveniently store your car with the valet, and the valet can efficiently stack cars to achieve maximum use of the parking lot. If you lose your ticket, though, it can be very difficult to retrieve your car without some additional context.

Object storage relies heavily on the applications to maintain their own object ID mapping, which means any alternative addressing—common names, paths, or searchable index—needs to be stored and managed outside of the object storage by an application. This makes it very difficult to share data across applications unless they have been specifically written to share the same object ID map. The key mechanism also makes ad-hoc use of the data by users very difficult. Users can't directly access data by navigating through a familiar file and folder structure. They can only access data through applications that know and can read the object index. Most significantly, this means that object storage is not suited to handle organizations' largest growing segment of data: unstructured data.

Lifecycle Management and Data Security

Object storage presents significant challenges for information lifecycle management. If an application loses or erases object IDs without notifying the object storage system that the storage space used by the objects should be freed, the object storage system will effectively lose capacity to inaccessible data. The distributed nature of the data layout makes it difficult to monitor access times and patterns, which also factor heavily in data lifecycle management decisions. Object storage systems rely entirely on the applications to manage the lifecycle of the objects in the system.

Additionally, object storage systems that work off a simple key system allow anyone with the key to retrieve the data, making it very difficult to manage security access. Some object storage systems have implemented authentication or network access controls to limit access to named users or specified hosts, but these mechanisms are a step backward from the mature directory-based access controls and hardened mandatory access control systems used with filesystems.

QUANTUM LATTUS: THE FUTURE OF OBJECT STORAGE

Building a Better NAS

The key to an organization's successful use of object storage is the ability to manage unstructured data in the object store. The most common way organizations manage unstructured data is through network attached storage systems (NAS). NAS systems have simplified and centralized unstructured files for most organizations, making shared filesystems available to a wide range of network clients using protocols like CIFS and NFS. Extending the benefits of traditional NAS storage to object storage has many crucial advantages.

First, by providing a traditional filesystem namespace, organizations can begin migration of their largest-growing segment of data—unstructured files—into object storage, where they can achieve greater durability at lower costs. Second, providing CIFS and NFS access to object storage guarantees broader compatibility with operating systems. Allowing filesystem access to object storage also allows organizations to expose their data to direct, ad-hoc usage by end users. By doing this, they can maximize the return on their critical data assets by enabling use by the widest range of users and applications. Finally, mapping object storage to a filesystem also allows administrators to leverage many of the traditional operational best practices for data management and security, and it provides a framework to transition to more sophisticated strategies.

Increased Storage Policy Management

Another critical feature for organizations with rapidly growing data sets is the ability to actively manage data lifecycle through the use of mature storage management policies. While object storage can increase the overall efficiency and manageability of storage, organizations should look to adopt data lifecycle management tools that help manage durability policies. One of the key features of erasure code-based object storage is that durability policies can be different for different data objects within the same object storage environment. This makes it possible to migrate from one durability policy to another without having to relocate objects. Organizations can set durability policies to organizational objectives and let the object storage system handle the details. For example, organizations can potentially set policies to migrate data to more cost-efficient but lower-protection durability policies as data ages, but set rules to maintain high durability for critical regulated data. Because different durability policies can be managed in the same object storage system, these durability changes can happen with complete transparency to the end users.

Object Storage as Active Archive

Object storage can also serve as an attractive long-term archive. Object storage can offer data protection levels equivalent to or exceeding tape, but at much lower latencies. This makes it an attractive option as an online archive.

Organizations that use object storage, or who plan on using object storage as part of an integrated storage management environment, will also be able to interface with object storage cloud providers. This can open a wide range of off-site durability options with lower latency compared to vaulted tape archives.

Multi-Site Recovery

Object storage architectures can be designed to inherently provide multi-site recovery. Because data is spread over nodes via standard networks, the nodes can be a mix of local and remote, and over two or even more sites. Data durability policies can be configured to guarantee that data can be recovered at a remote site even in the event of a whole data center failure. This provides automatic multi-site protection without the need for installing, configuring, and coordinating dedicated replication capabilities.

Shared Application Object and File Access

Quantum Lattus enables sharing of the object storage between both the filesystem-based clients and applications that are engineered specifically to use object storage. This not only enables sharing of the object storage pool across architectures, but it allows applications that are limited to traditional OS filesystem access to share data with applications that are written specifically for HTTP-based object storage access. This guarantees the widest accessibility of data within an organization.

SUMMARY

Object storage addresses many of the scalability and reliability problems that are introduced with modern data management, including the limitless scalability of namespaces. Erasure coding provides a new mechanism to achieve much higher data integrity and durability with far greater efficiency than traditional storage solutions.

Quantum Lattus extends the capabilities of object storage, blending the manageability and accessibility of traditional filesystems with the scalability and durability of object storage. With the best of both worlds, Quantum Lattus storage opens up new use cases for organizations, enabling broader use of big data, while maintaining certainty of data integrity and longevity despite the pressing problems that big data introduces.

BE CERTAIN

ABOUT QUANTUM

Quantum is a proven global expert in Data Protection and Big Data management, providing specialized storage solutions for physical, virtual and cloud environments. From small businesses to major enterprises, more than 100,000 customers have trusted Quantum to help maximize the value of their data by protecting and preserving it over its entire lifecycle. With Quantum, customers can Be Certain they're able to adapt in a changing world – keeping more data longer, bridging from today to tomorrow, and reducing costs. See how at **www.quantum.com.**

www.quantum.com • 800-677-6268

Quantum®

www.quantum.com

WP00185A-v03 Sept 2013