Object Storage

A Fresh Approach to Long-Term File Storage

A Dell Technical White Paper

Dell Product Group

May 2010

Contents

Figures

Executive Summary

The world is increasingly awash in digital data – not only because of the Internet and Web 2.0, but also because data that used to be collected on paper or media such as film, DVDs and compact discs has moved online. Most of this data is *unstructured* and in diverse formats such as e-mail, instant messages, documents, spreadsheets, graphics, images, and videos. For storage managers, the growth in unstructured data is proving to be a challenge: Companies require the data be readily accessible for business, regulatory and compliance needs, but traditional file storage management systems such as NAS are proving to be both costly and inadequate. With unstructured data growth expected to continue unabated -- at a compound annual growth rate estimated to exceed 60 per cent[1] -- storage managers are looking at new ways to cope. An alternative that has emerged is Object Storage. This is an approach that is designed to solve many of the traditional NAS shortcomings, and is considered more cost effective. However, it is not a "one size fits all" solution and traditional NAS will continue to have a strong role to play in today's storage environment. In this white paper we explore Object Storage, compare it to traditional NAS, and demonstrate that an intelligent, policy based data management strategy is the best approach to determining when it is beneficial for organizations to use Object Storage, or continue to use NAS.

Introduction

We live in interesting digital times. It used to be that computers primarily stored *structured* data such as financial and supply chain information. This has changed. Today, more and more of the world's *unstructured* data – everything from videos, music files, blogs, images, instant messages and even the day-to-day paperwork generated by businesses is being created, distributed and stored digitally. This is a phenomenon that is pervading all aspects of human life: In the doctor's office, for example, x-rays that were once produced on films are now created and stored digitally. In banks, cashed checks that used to be stored in microfiche are now stored on computer hard drives. Legal contracts, too, which had been solely be paper based are now created and stored digitally, with "digital signatures" taking the place of handwritten ones. The end result is an explosion of predominantly unstructured data being stored on computer storage systems. It is estimated that the amount of digital information will double every 18 months, with 95% of this coming from unstructured data, and only the remaining 5% being driven by traditional structured data[2]. Unstructured data is expected to far outpace the growth of structured data well into the future.

For storage managers, this phenomenal growth in data, particularly in unstructured data, is creating new challenges. It means they must continue to find cost effective storage strategies while ensuring data is available as needed for business or compliance requirements. It means they must make sure the data is well protected according to back-up and retention policies. But it now also means that they must ask how they are to best accomplish these goals – as well as what they need to do differently - when most of the data they are managing is unstructured and inherently different from structured data.

[1] "Object storage gains steam as unstructured data grows," Beth Pariseau, Storage Magazine, November/December 2009

[2] IDC White Paper sponsored by EMC, As the Economy Contracts, the Digital Universe Expands, May 2009

The new challenges of unstructured data

Unstructured data has two characteristics that make it a greater challenge to manage than structured data. First, it is hard to maintain the *context* of the data. Second, it is difficult to know the *content* within an unstructured data file. In combination, these two characteristics make it difficult for storage managers to understand the value of an unstructured data file and how it needs to be protected in primary, secondary, or archival storage. The two characteristics also make it difficult to determine whether the data needs to be backed up and, if so, for how long it should be retained. And, importantly, the characteristics make it difficult for storage managers to help their organizations maximize the value of the information – e.g. for business intelligence purposes – within unstructured data. To understand this more fully, consider the difference between unstructured and structured data.

Structured data is most often generated to support a transaction and is then stored in a relational database. This makes it easy for storage managers to understand both the content and context of the data. For example, if a customer places a purchase order, it is easy to track the customer's name, address, item being ordered, and the required delivery date by querying the tables in a database. Similarly, if an investor places a stock trade, it is easy to track the investor's name, account number, the stock being purchased, the purchase price, and the date on which the transaction was made. Since structured data often supports transactions, it is necessary for this data to remain immediately accessible. For example, if an investor decides to sell a stock they've purchased, it is necessary for a stock broker to immediately recall the purchase price in order to properly credit the gain or loss to the investor's account. Taking this example further, it is also easy for storage managers to track the context of the transaction to understand when it is closed, will no longer be required, and therefore safe to move to archival storage.

Unstructured information is different. It is often generated at the time of a particular event and then stored outside of a database. It also may not be touched or needed again after the particular event. Take x-rays, for example. These are most often created to help a physician diagnose a patient. Once the diagnosis is complete and if the patient is cured, the x-rays are no longer needed and are stored away. On the other hand, if the patient continues treatment, the x-rays may need to be recalled. This example shows the challenges in managing unstructured data and the importance of understanding the context around the data -- x-rays which are no longer needed should be sent to archival storage, while those still needed should be kept on near-line storage. The difficulty with unstructured data is that there isn't a mechanism analogous to a database that allows this context to be maintained. Instead, the context is often lost or separated from the data, and storage managers must make decisions based purely on the data type – e.g. the x-ray image - itself.

Similarly, it is difficult to know the *content* in unstructured data and use this information to help guide storage decisions. Consider, for example, a company that stores its product blueprint drawings as JPEG files. Without knowing the content in the company's JPEG files, storage managers can incorrectly give the blueprint files the same importance and storage priority as the JPEG picture files sent around to announce the arrival of an employee's new born baby. Similarly, for example, storing HR (e.g. employee offer letter or performance review) information for an employee clearly has a different priority than the minutes for a staff meeting even though the data for both may be stored within the same Microsoft® Word format.

Given the characteristics of unstructured data, the difficult question facing storage managers is how they can effectively and efficiently store this data and be mindful of both the context around, and the content, within the data to make the right storage decisions. Part of the challenge in solving this problem is that storage managers are working with a limited set of storage tools. Today's dominant approach is to store unstructured data on file systems such as Network Attached Storage (NAS). However, NAS was designed in a different age and time, when the world was much less digitized and unstructured data was not as prevalent as it is today.

A Need for New File Storage Solutions

File systems were invented approximately 30 years ago to provide an interface for end users and applications to store file (non database) data. They were designed originally to allow concurrent access to smaller groups of data files shared among a few users. As such, they were built to enable both read and write operations; hence file systems included overhead to manage permissions, and operations such as file locking. Today, however, much of the unstructured data that is generated does not require concurrent access. This means that the file systems' overhead is unnecessary and simply adds cost and complexity. This is particularly an issue if the context or content of the unstructured data requires that it should be stored in secondary or archival storage. In this case, using NAS storage for unstructured data means that too high a price is being paid for capacity to store aged or rarely accessed data.

In addition, file systems store data in hierarchical structures ("trees") consisting of directories, folders, subfolders and files. The objective of a file system, then, is to manage the *location* of data according to a logical sequence and via an easily understood hierarchy of nested folders. As a consequence, the content and data contained within a file is not important and each file will have only basic metadata attached to it. For example, viewing a file directory will tell you the file name, when it was created, when it was last modified, the file type, and potentially the person who created the file.

This limited amount of meta-data associated with each file means that IT teams do not have the context and content information they need to efficiently manage and use the unstructured data they have in their organization. They cannot, for example, know where to automatically place an individual file in a storage tier since they do not know the content of the files or its importance. They cannot tell, for example, whether a particular file is important to backup, or maintain for compliance reasons. This lack of knowledge at the individual file level means instead they must rely on blanket policies that apply across file types (e.g. all JPEG files must be stored indefinitely).

Compounding the problem is that as the amount of data grows, so do the number of nested folders. The result is a set of large tree structures that makes it cumbersome and challenging to find any particular file, especially if the specific name, date created, or file type is not known. In addition, as the tree structure grows, the performance of the file system starts to degrade and backup becomes more difficult.

There is a mismatch between the cost, design and capabilities of file systems such as traditional NAS and the new requirements for storing the unstructured or file-based data being generated today. It is clear that a fresh approach is required.

A fresh approach: Object Storage

Object storage is an approach to storage where data is combined with rich metadata in order to preserve information about both the *context* and the *content* of the data. To see the difference between Object Storage and traditional File Storage, consider the example of storing an MRI scan as a file versus as an object, as shown in Figure 1.

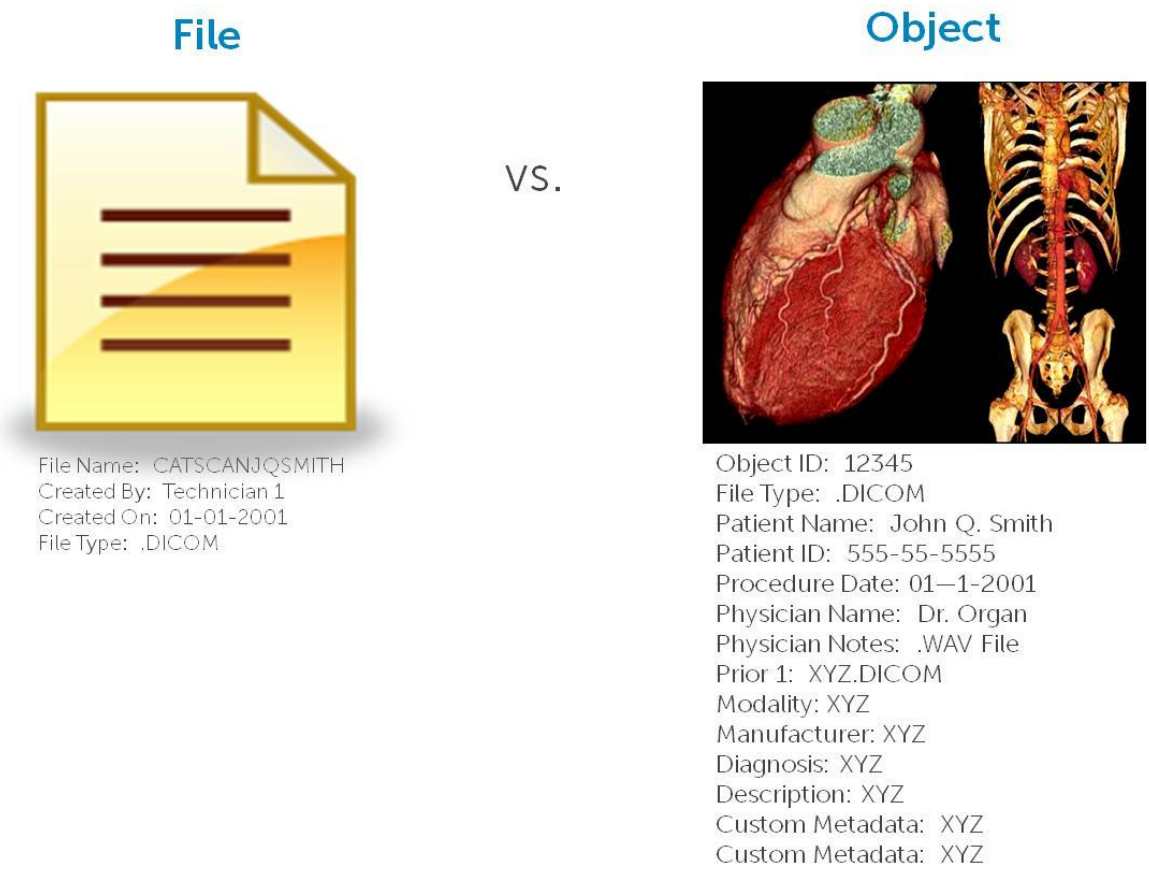

Figure 1. Example contrasting the amount of metadata associated with an Object vs. a File

When an MRI scan is stored as a file the typical metadata attached to it is basic and may include only information such as file name, creation data, creator, and file type. When the MRI scan is stored as an Object, on the other hand, the generating application can include all the file metadata plus additional metadata information that might summarize the content contained within the file and include the patient name, the patient's ID, the procedure date, the attending physician's name, the physician's notes, as well as any other metadata that can help add context to the MRI scan.

The metadata present in Object Storage gives users the context and content information they need to properly manage and access unstructured data. They can easily search for data without knowing specific filenames, dates or traditional file designations. They can also use the metadata to apply policies for routing, retention and deletion as well as automate storage management. For example, with the MRI scan, a storage policy can be set to look at the metadata associated with it, track the patient's name and then determine if the patient is currently admitted to the hospital. If the patient is not, the MRI scan can be sent to archival storage. On the other hand, if the MRI object is for a current patient who is admitted for ongoing treatment, the object can be routed to near-line storage so that it can be immediately retrieved the next time the patient visits the hospital. As another example, if a storage manager comes across an MP3 file and the metadata indicates that the data contained within it is an employee's personal music file, the storage policy will know to manage it differently than if the MP3 file was the recording of something important to the company or institution (e.g. a physician's recorded notes).

A richer set of metadata can also make it easier to apply eDiscovery and business intelligence tools to help an organization uncover data assets and gain new insights. For example, the metadata can make it possible for a hospital to find all stored MRIs for a particular disease and then collect statistics on, for example, the number of MRI scans done per stage of the disease condition to help allocate resources. In this way, object storage helps to ensure that the value of information contained within unstructured data is maximized and preserved for future use.

Objects are also useful in directly keeping related information together by enabling multiple file types to be grouped together. For example, it is possible to group the MRI image with the physician's recorded notes (in an MP3 file) along with the text file that has the patient's history. This is similar to the way that information is managed in the paper based world where a patient's file can contain different file types. And similar to the way that a traditional hospital file may be used, the object with its grouped files can be used by any subsequent physician treating the patient to easily access the previous physician's notes attached to the scan and obtain additional context about the patient's condition.

An object is also different from a file in that a unique ID is assigned and associated with each object. This ID is generated using a 128-bit random number generator and guarantees that every object is uniquely identified. It allows objects to be stored in an infinitely vast flat address space containing billions of objects without the complexity file systems impose. Similar to the function of URLs in the Internet, an Object ID serves as the unique pointer to the object; hence there is no directory hierarchy (or "tree") and the object's location does not have to be specified in the same way that a file's directory path has to known in order to retrieve it. The unique identifier also allows objects to be easily migrated from one storage node or system to another without interrupting application or user access if the underlying hardware is being upgraded.

In addition to the unique Object ID a hash signature also has a strong role to play in managing storage for unstructured data, particularly in the removal of duplicates and in helping to address compliance mandates. Since the hash signature associated with each object is generated according to the data contained within the object, if the same signature is recognized as already being in the hash table, it is immediately known that duplicate data exists in the storage system. With this knowledge, storage managers can decide how they want to treat it.

Similarly, demonstrating that data has not been tampered with is one of the important requirements where compliance is important. One way to prove this, if required, is to show that the hash signature has not changed. With Object Storage, if the data is changed or tampered with, the hash computed to verify authenticity will change and not be allowed. Conversely, if the data has not been modified, the hash signature remains the same, which can provide proof, if and when required, that the data is correct and authentic. This property makes Object Storage useful for archiving data while protecting it, and in helping to meet regulatory requirements, particularly for data that has high legal or compliance risk.

In addition to its storage management benefits, Object Storage, deployed effectively, can also be more cost effective than NAS. This is for two reasons. First, Object Storage does not require much of the overhead present with NAS to manage inodes, concurrent read/writes, file locks and permissions that improves performance and enables massive scaling in terms of object count and capacity. As a result, companies can simply and affordably scale object storage to petabytes of data – a key advantage as they look to manage the rapidly growing amount of unstructured data. Secondly, storage managers can use the metadata contained within objects to appropriately route to the right storage tier and free up primary storage capacity. This allows organizations to reduce costs in comparison to file storage where, in contrast, data may be needlessly stored on more expensive tiers because there is not enough information about the data to make sure it is stored at the most cost effective storage tier.

Object Storage and Traditional NAS Coexist

Object Storage works best when large numbers (millions or billions) of unstructured data files need to be stored. In this case, the storage interfaces that are present with file systems, e.g. separate LUNs, folders and permissions are inefficient overhead. Object Storage is ideal for archiving when the data is relatively static and not frequently accessed. This isn't a restrictive criterion for Object Storage use cases. By some estimates, 70% of data that is generated is never accessed after its initial creation and remains static, while 20% is semi-active[3].

However, 10% of all data is actively used, and it is for this data that traditional file systems, such as NAS, are best suited. For example, a company developing marketing collateral will often have a team working on the content. This team will collaborate and in doing so may be simultaneously reading and writing to the data (contained in a Microsoft Word document). In these I/O and performance prioritized cases, traditional file storage systems like NAS can, and will, continue to play a role. Consequently, as shown in Figure 2, the frequency of data usage is a driver for multiple storage system types within a storage environment and demonstrates that both Object Storage and traditional file storage systems have strong roles to play within an organization.

[3] Measurement & Analysis of Large-Scale Network File System Workloads, University of California Santa Cruz, 2008

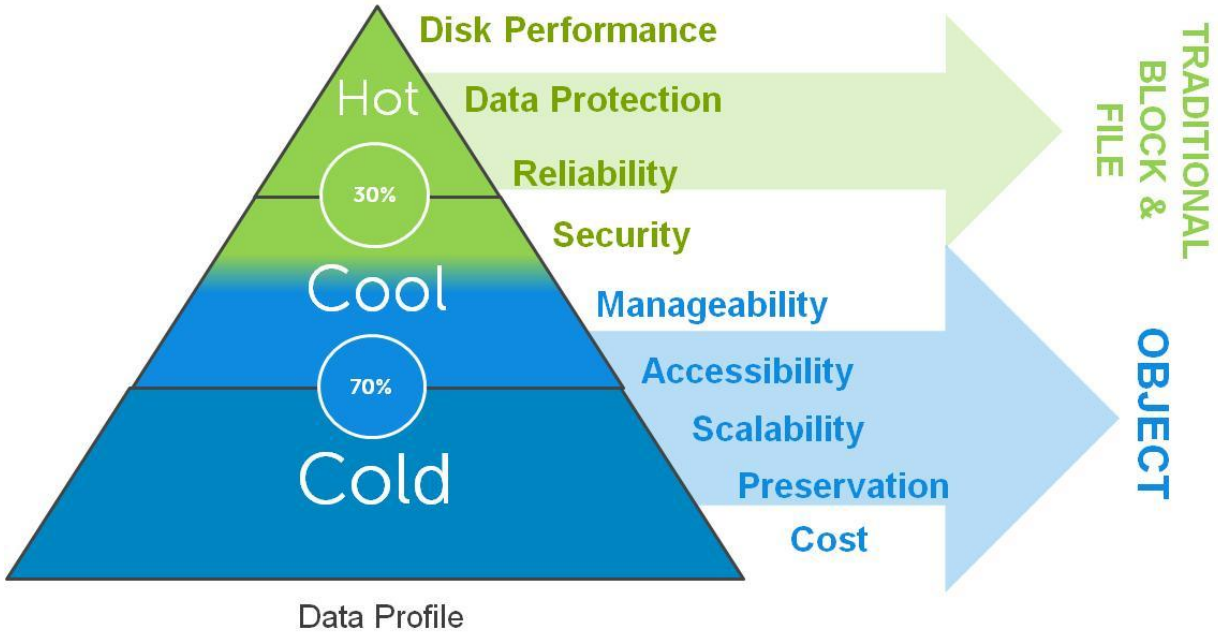

Figure 2. The frequency of data usage is a factor in using Object vs. traditional file storage

Object Storage in Intelligent Data Management

One of the primary benefits of Object Storage is the role that it can play in intelligent data management. For storage managers, the mantra they are increasingly working with is to "store everything" and "store it forever." The traditional brute force approach to this problem is to continuously throw storage resources at the data. This unfortunately only provides short term relief – with ever growing volumes of data, storage managers find that their budgets are strained trying to keep up. Dell is focused on developing approaches to help organizations intelligently manage their data. The idea is to automatically route data to the right storage systems and the right tier and protection levels within those systems according to its value and stage in the data lifecycle. Through this approach, the most expensive high performance systems are reserved for frequently accessed primary data. Infrequently accessed secondary data is routed to comparatively less expensive storage arrays, ensuring that costs are minimized. Object Storage, with its rich metadata, can play a significant role in automating Intelligent Data Management for unstructured data by enabling users to apply policies based on metadata values and automatically route data to the right storage systems. In this way, Object Storage provides organizations with new capabilities to increase the efficiency by which they can manage and optimize storage.

Summary

As an increasing amount of the World's information is born and lives digital, IT organizations will need to simultaneously manage two challenges. The first is that most of the digitized information will be unstructured, which means that it is inherently highly variable and not easily managed without understanding the content and context of the data. The second will be the huge growth in data both currently and in the years ahead. This growth in data will exceed the capabilities of the file systems upon which storage managers have traditionally relied to store unstructured (non database) data. As the volume of unstructured data continues to grow, organizations will find it increasingly difficult to cope.

Object Storage is a promising solution for managing the complexities of unstructured data and ensuring long-term retention and access that flexibly scales to meet high growth in terms of the number of objects and storage capacity. It uses rich metadata attached to the data to carry "information about the information." This gives users the information they need to understand the content and context of the data. Object Storage also provides a unique identifier for each distinct data object helping to ensure it can be specifically located and retrieved. The separation of the identifier from the hash signature is a powerful tool for enabling decisions on how to deal with duplicate data in storage, and ensures the hash can be upgraded since it is not used as the address for objects. The hash signature is also a powerful mechanism for helping to meet compliance mandates – if the data is tampered with, the signature will change and if the hash is compromised it can be upgraded to a more secure algorithm.

Object Storage is an ideal solution for efficiently managing large unstructured data sets and for archiving data with high compliance and legal risk such as medical and legal records, e-mail, invoices and financial records. It is also useful in helping to unlock the value of stored content through business intelligence applied to the object's metadata, and it does not require much of the management and backup overhead present with file systems. As such, it is designed to lower the cost of storage, and let organizations affordably scale to petabytes of data. In addition, by enabling users to correctly differentiate data instead of treating it all equally, Object Storage provides the benefit of freeing up capacity on primary storage – a particular concern in tough budgetary times. It also opens the door to exciting new possibilities for data management by providing the information needed to define policies for intelligently and automatically routing data to the right storage systems and the right tiers in those systems according to its value and stage in the data's lifecycle. In sum, Object Storage is a very powerful approach to long-term file storage.