Compliments of IBM

IBM Limited Edition

File & Object Storage

FOR DUMMIES®

A Wiley Brand

Learn:

- About file and object storage
- The new IBM Spectrum Scale™
- IBM Elastic Storage™ Server
- Ten Advantages of IBM Spectrum Scale with Object Storage

Neal Ekker

File & Object Storage

FOR DUMMIES®
A Wiley Brand

IBM Limited Edition

by Neal Ekker

FOR DUMMIES®
A Wiley Brand

File & Object Storage For Dummies®, IBM Limited Edition

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

For general information on our other products and services, or how to create a custom *For Dummies* book for your business or organization, please contact our Business Development Department in the U.S. at 877-409-4177, contact info@dummies.biz, or visit www.wiley.com/go/custompub. For information about licensing the *For Dummies* brand for products or services, contact BrandedRights&Licenses@Wiley.com.

ISBN: 978-1-119-14974-3 (pbk); ISBN: 978-1-119-14975-0 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Publisher's Acknowledgments

Some of the people who helped bring this book to market include the following:

Project Editor: Carrie A. Johnson

Editorial Manager: Rev Mengle

Acquisitions Editor: Steve Hayes

Business Development Representative: Christiane Cormier

Special Help: Derek Gascon, Dean Hildebrand, Chandra Mukhyala, Douglas O'Flaherty

Table of Contents

Introduction

The rapid, accelerating growth of data, transactions, and digitally aware devices is straining today's information technology (IT) infrastructure and operations. At the same time, storage costs are increasing and user expectations and cost pressures are rising. Though the increase in structured data — the type most easily handled by databases — has been dramatic, the growth of unstructured data generated by the majority of commercial, scientific, and personal activities — text, images, even sensor telemetry — has been simply breathtaking, and certainly challenging. Over the years, enterprises of all types have turned to file systems to store and manage their unstructured data. Recently, the explosion of unstructured data and the accelerating need to manage and search it more efficiently has led to the development of a new storage paradigm — object storage.

This staggering growth of data, especially the unstructured variety, is driving the need for high-performance file and object storage solutions. *File & Object Storage For Dummies,* IBM Limited Edition, introduces file and object storage solutions all based on one powerful tool — IBM Spectrum Scale.

Spectrum Scale is a high-performance and proven product used to store and manage mission-critical data by thousands of commercial, scientific, academic, and governmental organizations worldwide. *File & Object Storage For Dummies,* IBM Limited Edition, introduces this exciting storage platform and then explores its benefits. We also examine some additional technologies such as OpenStack Swift and Elastic Storage Server to see how they can be deployed to help your enterprise lower IT costs, improve decision-making, and accelerate the insights and value derived from one of your most valuable assets — information.

About This Book

File & Object Storage For Dummies, IBM Limited Edition, examines data storage and management challenges and describes an innovative solution for high-performance, cost-effective file and object storage using Spectrum Scale.

Foolish Assumptions

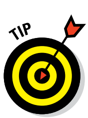
This book is written primarily for technical readers and decision makers such as storage administrators and IT managers. Therefore, I make a few basic assumptions about the reader's understanding of basic enterprise data storage technologies and your interest in solving storage management challenges.

Icons Used in This Book

Throughout this book, you occasionally see special icons to call attention to important information. No smiley faces winking at you or any other cute little emoticons, but you definitely want to take note. Here's what you can expect.

This icon points out information that may well be worth committing to your own nonvolatile memory — along with anniversaries and birthdays.

You won't find a map of the human genome or the blueprints for IBM's Watson here (or maybe you will, hmm), but if you seek to attain the seventh level of NERD-vana, perk up. This icon explains the jargon beneath the jargon and is the stuff legends — well, nerds — are made of.



Thank you for reading; hope you enjoy the book. Please take care of your writers. Seriously, this icon points out helpful suggestions and useful nuggets of information.

Proceed at your own risk . . . well, okay — it's actually nothing *that* hazardous. These helpful alerts offer practical advice to help you avoid making potentially costly mistakes.

Chapter 1

Storage 101

In This Chapter

- Recognizing data access and management challenges
- Exploring the basics of what storage does
- Understanding different types of storage
- Distinguishing between different storage technologies
- Looking at cluster file systems

Enterprise users and data consumers are churning out vast amounts of documents and rich media (such as images, audio, and video). Managing the volume and complexity of this information is a significant challenge in organizations of all types and sizes as more and more applications feed on and increase this deluge of unstructured content and as individuals and businesses collaborate for intelligence, analytics, and information sharing.

Before delving deeply into the specific topics of file and object storage, this chapter provides some general background on digital storage technologies and the storage industry. I start by quickly summing up storage industry challenges and then you take a look at what function storage serves in information technology (IT) infrastructure. This includes the different types of storage available and the advantages of each, depending on the application use case. Next, you look at the hardware nuts and bolts of storing digital data and common software approaches to organizing data and finish up with an introduction to scalable file system storage.

Data Access and Management Challenges

Digital data — both structured and unstructured — is everywhere and continues to grow at a stunning pace. Every day, approximately 15 petabytes of new information is generated worldwide, and the total amount of digital data doubles approximately every two years. At the same time, storage budgets are increasing only one to five percent annually; therefore, the gap between data growth and storage spending is widening (see Figure 1-1). The data growth explosion, as well as the nature and increasing uses of data, are creating tremendous data storage challenges for enterprises and IT departments everywhere. Simply put, storage needs to be less expensive in order to keep up with demand.

Structured data refers to data that's organized, for example, in a database. *Unstructured data* refers to data that doesn't have a defined model or framework — for example, multimedia files.

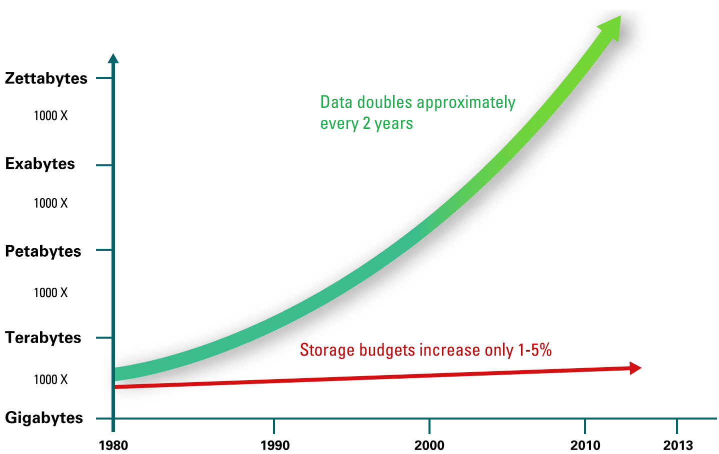


Figure 1-1: Storage requirements are devouring CAPEX and OPEX resources.

Some of this growth can be addressed with larger hard disk drives and networking components getting faster and faster. But as these technologies advance, making the data useful becomes more difficult. Larger hard disk drives enable you to store more data, but in many cases the hardware appliances that utilize these drives aren't able to keep up. Data gets trapped in them, safely stored but not fully utilized to drive business growth or enhance decision-making.

To address the challenge of unstructured data growth and the proliferation of file-based information, many IT managers today are adding more and more network-attached storage (NAS) devices. NAS devices can be relatively cost effective to purchase, but growing your file storage by adding more devices causes data administration and management (such as migration, backups, archiving) costs to skyrocket. And while this approach may solve a short-term storage capacity problem, it doesn't necessarily improve application input/output (I/O) performance, and it does nothing to reduce management costs or improve application workflow.

Many data centers have become victims of "filer-sprawl" — IT departments deploying numerous NAS appliances or "filers" in a futile effort to keep up with out-of-control storage demands. But filers are actually quite expensive when you view then in terms of cost for capacity, and they aren't very adaptable to many of the new data types and application workloads. Finally, beyond simply trying to keep up with the need for ever more storage, "stop-gap" or temporary fixes often lead to other data access and management challenges, including

- Rising administrative costs to manage file-based data
- Data accessibility that's limited in remote locations
- Continuous data availability and protection that becomes increasingly difficult to maintain
- Backup and archival operations that can't keep pace with growing data
- Decreased performance as the amount of data grows due to inability to adapt to application performance requirements over time

One of the consequences of filer sprawl can be the impact on your business. Each NAS filer is a data silo limiting applications and employee access to information. Global teams, big data analysis, logistics, customer service, and collaboration all depend upon a common view of your business information.

REMEMBER

Effective data access and management is critical to an efficient computing infrastructure. An efficient infrastructure must be balanced properly between three key components: compute, network, and storage. The network and the data storage are normally the most difficult challenges for enterprise IT departments.

Three Important Functions of Storage

At its most basic level, enterprise storage performs four important functions:

- Store data (intermediate and final)
- Manage that data while it's stored
- Protect data from being lost
- Feed data to computer processors (so they can keep doing work)

Storage administrators have always recognized the need for storage capacity, management, and data protection, and traditional storage vendors do a good job of providing solutions that satisfy these functions.

However, storage administrators are now increasingly focused on getting data from the storage to the processor because this has become a performance bottleneck, and most storage vendors have done a poor job of addressing this storage issue. Consider that over the past decade

- **CPU** performance has increased 8 to 10 times.
- **DRAM** performance has increased 7 to 9 times.
- **Network** performance has increased nearly 100 times.
- **Bus** speeds have increased 20 times.
- **Hard disk drive (HDD)** performance has increased only 1.2 times.

One result of this storage bottleneck is that your applications may be running slow, which negatively impacts productivity and wastes the capacity of other expensive infrastructure in your data center.

Defining Types of Storage

Not all storage is created equal. Storage systems are commonly divided into block, file, and object storage systems and include direct-attached storage (DAS), network-attached storage (NAS), and storage area networks (SAN).

TIP

Scaling-up means adding more storage to a single array or system, such as replacing 2 terabyte (TB) drives within a storage system or array with 6TB drives. Scaling-up does not add performance to the storage. Scaling-out means adding more storage components/nodes or entire arrays. Because each scale-out unit also includes more compute and networking, scale-out adds performance. However, scale-out requires more coordination and management. Sophisticated scale-out solutions, such as the IBM storage software called *Spectrum Scale,* can achieve near linear performance gains for many workloads.

Block storage

Block-based storage stores data on a hard drive as a sequence of bits or bytes of a fixed size or length (a block). In a block-based storage system, a computer's (server) operating system (OS) connects to the hard drives. Block storage is accessible via a number of interfaces, including

- Fibre Channel (or Fibre Channel Protocol, FCP)
- SCSI (Small Computer System Interface) and iSCSI (Internet Protocol SCSI)
- SAS (Serial Attached SCSI)
- ATA (Advanced Technology Attachment) and SATA (Serial ATA)

TECHNICAL STUFF

Fibre Channel and iSCSI interfaces, as well as SAS, are commonly used for storage housed outside of the computer, most commonly network-connected. SAS and ATA are typically used in DAS. Block-based storage systems can be implemented as either DAS or network based.

Direct-attached storage

DAS is the simplest and cheapest type of block storage for computer systems. As the name implies, DAS is directly attached to the computer or server via a bus interface (see Figure 1-2).

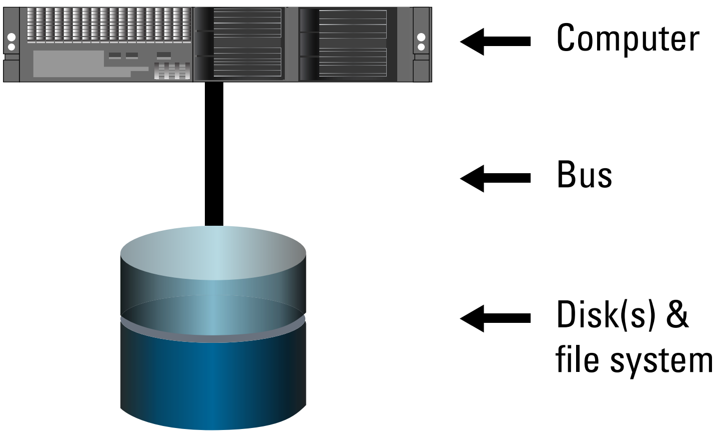


Figure 1-2: DAS connects hard drives directly to the computer or server via a bus interface.

DAS has very limited capacity because you can only install as many drives as the number of physical slots available on the server.

Storage area networks

A storage area network (SAN) is a separate storage system connected to a server or servers through a dedicated storage network (see Figure 1-3).

A SAN can be used by multiple servers. Each server has one or more fast, dedicated storage connections to one or more storage arrays. A SAN allows multiple computers to share access to a set of storage controllers. This provides great flexibility for maintaining enterprise IT infrastructures. In large organizations, SANs enable a division of labor where the system administrators manage the computers and the storage administrators manage the SAN. Having multiple computers sharing access to the same data is important to many applications and business processes.

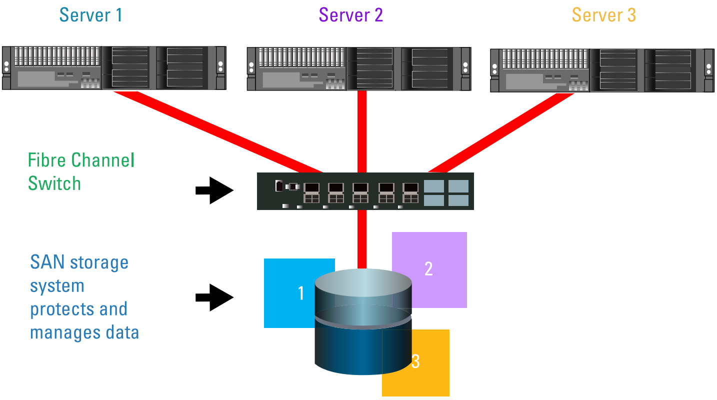


Figure 1-3: A SAN connects a storage array to servers via a dedicated storage network.

A LUN (Logical Unit Number) is an identifier assigned to a collection of disks within the storage system, defined in a storage controller and partitioned so that host servers can access it. A computer can then use the LUN to store data. For example, you can create a file system within a LUN as a place to store files. A volume is part of a LUN created within volume management software.

SANs are commonly used in mission-critical or high-transaction/IOPS (I/Os Per Second) environments — for example, online transaction processing (OLTP) databases, ERP (Enterprise Resource Planning), and virtualized systems.

Compared to other storage systems, SANs can be relatively expensive because they're engineered for maximum reliability and performance. Plus, managing a SAN may require a dedicated administrator and plenty of attention, especially if attempting to scale a SAN to meet today's data requirements.

Networked attached storage

File-based storage systems, such as network attached storage (NAS) appliances, are often referred to as "filers" and most often store unstructured data on a hard drive as files in a directory structure. These devices have their own processors and OS and are accessed by using a standard protocol over your local area network (LAN), rather than a dedicated network such as a SAN. Common NAS protocols include

- **SMB (Server Message Block) or CIFS (Common Internet File System):** SMB (or CIFS) is commonly used in Windows-based networks.
- **NFS (Network File System):** NFS is common in Unix- and Linux-based networks.

NAS appliances are relatively easy to deploy, and client access is straightforward using the common protocols. Computers and the NAS appliances are all connected over your LAN or a shared Internet (TCP/IP) network, and the data stored on NAS appliances can be accessed by virtually any computer, regardless of the computer's OS.

NAS appliances are fairly common in data centers today. However, NAS appliances have several significant disadvantages. They're typically slower than DAS or a SAN and can be storage performance bottlenecks because all data has to go through the NAS' own processors. NAS appliances also have limited scalability. When a NAS appliance fills up, you add another, and another, and so on. This creates "islands of storage" that are very inefficient to manage (see Figure 1-4). NAS most often uses custom hardware and proprietary technologies that increase costs. Also, they don't integrate easily with other products and/or systems.

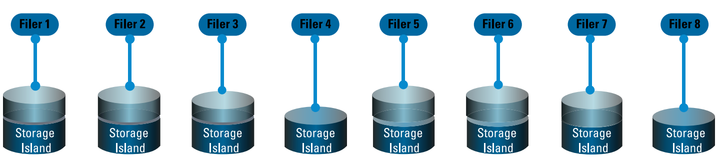


Figure 1-4: Filers often lead to "islands of storage" in the data center.

Object storage

Object-based storage systems use logical constructs to store data known as *objects* in a flat address space instead of the hierarchical, directory-based file systems that are common in file-based storage systems (see Figure 1-5).

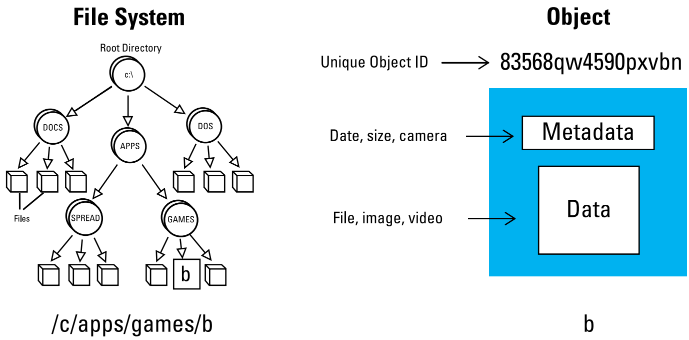


Figure 1-5: Comparing the file system to the object-based storage system.

An object stores the actual data (for example, an image or video), the metadata (for example, date, size, camera type), and a unique name assigned when the object is created and used to reference and locate the object when needed. The data in an object-based storage system is typically accessed using the Internet protocol (HTTP) through a web browser or directly through an application interface (API) like REST (representational state transfer). The flat address space in an object-based storage system enables simplicity and massive scalability.

Object-based storage is commonly used for cloud storage services by providers such as IBM SoftLayer, Amazon S3, and Google.

The three most common protocols are

- **Accounts:** At the top level, accounts store one or more containers, which then contain objects.
- **OpenStack Swift:** A part of the OpenStack Cloud project, Swift is an Open Source project with widespread support from corporations, various public organizations, and technology vendors. In Swift, the objects are stored in *containers*.
- **S3:** The object specification used by Amazon Web Services. As one of the first fully defined object protocols, S3 is used by many applications even if they aren't deployed in AWS. Under S3, objects are stored in *buckets*.

OpenStack Swift and S3 APIs are similar and many object stores support both.

TIP

What makes object storage appealing for scaling also makes it easy for users to locate the data. Without a file system hierarchy, an application only needs to know the name of the object to locate and access it. However, without a file system, there's no hierarchy for a user to navigate to search for the object. Adding more storage can be done within the same array, or across large distributed networks such as the Internet.

Distributed File Systems

To address the limitations of conventional NAS systems, more and more enterprises are deploying distributed file systems often called *cluster file systems.* These serve data from two or more storage devices or nodes. With CIFS and NFS, data is typically served from a single NAS appliance or node to which the storage is directly attached or connected via a SAN. But with a cluster file system, multiple storage devices can be accessed from many computers at the same time over a network. A cluster file system achieves high I/O performance by spreading data across multiple storage devices, all sharing the same global namespace, which also increases scalability. Cluster file systems are especially well suited for storage environments that must scale or grow quickly and extensively.

Cluster file systems are similar to network protocols, such as CIFS or NFS, but there are a couple of differences:

- Direct access from the application host computer provides a cluster file system with several performance advantages over standard network protocols.
- Cluster file servers are tightly connected to each other and communicate at a more sophisticated level to enable an application, for example, to have multiple nodes reading and writing to a single file.

Most cluster file systems extend the same functionality over TCP/IP or Infiniband. This is similar to NFS and CIFS, because both approaches use the network to access data. But in the case of a cluster file system, the network protocol used to transfer data is integral to the cluster file system solution itself. This tight integration allows the cluster file system to provide high performance and advanced access patterns over the network. These protocols can leverage technologies such as Remote Direct Memory Access (RDMA) for faster processing of data.

Their especially tight integration means that cluster file systems can be fast and provide advanced functionality, but they aren't particularly well suited for workstation access. You wouldn't run a cluster file system on your tablet to access your music, for example. Instead, cluster file systems are designed to provide enhanced file data access for enterprise IT infrastructures, like the systems from which you download your music.

Cluster file systems can read and write data in parallel across multiple storage nodes, providing extremely high performance, scalability, and data protection. You find out more about cluster file systems in Chapter 2.

Chapter 2

Focusing on File and Object Storage

In This Chapter

- Learning why there is more than one type of storage
- Examining some file and object use cases
- Introducing software-defined storage

As the processing and networking parts of information systems have evolved, so has storage evolved to better serve data for different applications. When all computing was done on a single system, such as a mainframe, storage could be addressed via local, proprietary connections optimized for that environment. As different and distributed compute systems became widely available, common storage systems to serve them based upon understood principles of device, folder, sub-folder, and name were developed. Although the underlying technology has changed, this paradigm has served the industry well for decades.

File systems evolved because they made order from chaos; they took information without structure and gave it an overlying structure that's human readable and easily navigated. However, file systems have their limitations.

When collections of information called *data sets* grew larger than a single storage device could handle, the concept of multiple storage devices working closely together evolved. Among these are the clustered or distributed file systems I mention in Chapter 1. They essentially disconnect applications from the physical location of their data so that there can be many storage devices, but to the application they all just appear as one. An advantage this design or architecture offers is that it makes adding or scaling-out more storage capacity and performance much easier.

As compute servers became capable of working together in clusters, parallel access to storage across multiple network links became important. Thus cluster file systems were born. Both of these developments didn't change the basic construction of file access. Whether clustered or parallel, to the application and end-user the information still appeared as a single device with a well known hierarchy of device, folder, sub-folder, and file.

As the number of computers working together scales up, it has precipitated a new data storage type. Object stores eliminate the well-known hierarchical structure in favor of unique naming of each unit of data.

File and Object Storage Use Cases

Before you turn your attention to the storage technologies that are the focus for the rest of this book, take a step back and consider a few basic questions: Why are file and object storage used? What are they used for? And who uses them?

Structured data brings with it some inherent internal organization or structure. This is most often hierarchical in nature and lends itself well to being managed and manipulated by applications called databases. For decades, databases have provided the structured data that powers most of the applications that drive business, government, and science.

Nonetheless, information without inherent internal structure or organization has always existed. The content that constitutes this *For Dummies* book is a good example. To make it more efficient and effective to store, find, manage, manipulate, search, and analyze this unstructured type of information, file systems were developed. A file system adds a layer of structure over the basic unstructured data — directories, folders, sub-folders, and files — which enables users and applications to more efficiently store and catalogue unstructured data, find specific data, retrieve it, use it in whatever ways needed, and clearly track any changes to it. Almost every business generates basic documents, presentations, spreadsheets, and images that are stored in file systems accessible over their Local Area Network (LAN).

File storage has properties suitable to the way people work. To protect accuracy, files can be locked so only one person at a time can edit the data. And files can be made larger or smaller, without renaming them, which is very useful for the way you access, use, and update this type of data. These attributes make files a suitable solution for databases and applications that want to control and contrast their data. Common use cases for file systems include the following:

- **Collaboration:** Thanks to file system technology, multiple users can access the same file at the same time without causing version issues.
- **Structured data applications:** File systems can enable unstructured data to be used by applications designed for structured data.
- **Rapidly growing files:** File systems easily handle files that change and grow quickly, such as device logs, media recordings, and scientific information.
- **Big data analysis:** Processing large collections of information can be accelerated by splitting it into multiple parts, each of which can be access over a network and processed in parallel. File systems are excellent for these environments.

Because the volume of unstructured data is growing even faster than structured data, file systems are today more important and more pervasive than ever. Much scientific research relies on massive file systems. When human and animal genomes are sequenced, file systems are involved. Of course businesses and governments around the globe are generating ever-greater quantities of documents and information. Then consider the new applications — social media, mobile apps, and perhaps the biggest unstructured data reservoir of all, the Internet. The use cases today and well into the future for file systems make a list much longer than I have room to include here.

But a funny thing happened on the way to the future of information — velocity. Add it to proliferating data volumes and varieties and you create new challenges for IT infrastructure. Imagine the technical challenges involved in streaming a television episode over the Internet. That's small fish compared to making the results from experiments conducted at the Large Hadron Collider in Cern, Switzerland, available to scientists worldwide. Actually, just making the photo of your grandparents on your favorite social app page available for download to anyone around the world is quite a technical challenge, especially if you want that photo to never be lost and download as quickly in Tokyo as in Toledo.

These types of challenges have driven the development of object storage systems. An object is a collection of data that includes information about itself that is normally furnished in other data storage technologies by databases or file systems. Because objects supply this information themselves, they can exist somewhat independently of these storage technologies, giving IT solution providers more flexibility in how object storage is implemented and utilized. Next, object storage is not necessarily organized in a hierarchical structure. Instead, objects use some form of simple name or key stored in a "flat" (non-hierarchical) address space.

These unique attributes of object storage give it some advantages in the modern world of streaming video and cloud computing. Object storage can lower the complexity and overhead of its management software and systems, giving it great flexibility of utilization, especially in Internet-based applications.

This means you can search for episodes of *Star Trek,* or *I Love Lucy,* or *Bonanza,* or the *X Files,* download them, and play them on your laptop quickly and easily — amazingly so considering all the underlying technical challenges. Or your doctor can discuss your PET Scan results with an expert in India while they are both viewing the same image. And multiple copies of the photo of your grandparents can "live" in different storage systems in different locations around the world, ensuring that a disaster in one data center won't change the availability over the Internet of that precious photo to anyone, anywhere, who wants to view it.

Object storage can be used by many computers in different locations. These environments include clustered applications like big data analytics, or different applications that need the same data, such as using the same picture on different websites. Object storage's Internet (RESTful) protocols and application interfaces (API) can be load balanced between different systems. This capability to live in many places at once makes it especially good at surviving unexpected system calamities in any particular physical location without storage failures or even slowdowns.

Object stores have eliminated the need to lock files while one user is accessing and updating the information, to avoid version issues. In fact, once created, an object can't be made larger or smaller. Whether large or small, when the object is written, its storage system enforces a unique name. To handle updates to the data, a new object is written. Common use cases for object storage include the following:

- Web services that leverage loosely coupled storage interfaces because the compute can be added
- Applications that reference the same data for different services
- Write-infrequently, which reads many times types of data, such as a virtual machine image
- Scale-out of infrequently accessed data, which leverages an object store's capability to scale out the storage without protocol overhead

Software-Defined Storage

For a number of reasons, both file systems and object storage lend themselves well to storage solution designs or architectures that employ essentially commodity hardware underneath a layer of very intelligent and functional software. This approach to a storage solution is called *software-defined storage.*

Defining software-defined storage

At its most basic level, *software-defined storage* is data storage that uses standard hardware with all the important storage and management functions performed by intelligent software. It might be understood best by thinking of it as storage virtualization — the hardware is "uncoupled" or separated in a logical sense from the actual functioning of the storage system, which is controlled by the software. This is essentially similar to the way that virtualization works in application servers.

The best software-defined storage delivers automated, policy-driven, application-aware storage services through orchestration of the underlining storage infrastructure in support of an overall software-defined environment.

Additional characteristics of software-defined storage can include

- Automated policy-driven administration for storage management functions, such as information life cycle management (ILM) and capacity provisioning
- Storage virtualization, where the application isn't aware of the physical location of the data
- Separate control and data planes to manage the storage infrastructure and data in the storage infrastructure, respectively

These characteristics are in contrast to traditional storage systems that depend heavily on custom hardware-based controllers to perform storage functions.

Key benefits of software-defined storage

Enterprises today are recognizing many significant benefits of software-defined storage in their data centers. These include

- Increased flexibility
- Automated storage management
- Cost efficiency
- Limitless scalability
- The option to employ the hardware of their choice
- Increasing capacity by adding systems (scale-out)

Increased flexibility and agility

Traditional enterprise storage platforms such as SAN and NAS (check out Chapter 1) are typically based on proprietary systems and come with a high total cost of ownership (TCO). SAN solutions typically require the use of costly and complex SAN switches, storage arrays, and other proprietary components.

NAS devices may have acceptable cost, but they offer limited scalability. When you run out of space on a NAS, you simply add more NAS devices. However, this isn't a true scale-out capability because each individual NAS device is presented as separate, stand-alone storage that's separately managed.

A software-defined storage solution increases flexibility by enabling organizations to use non-proprietary standard hardware and, in many cases, leverage existing storage infrastructure as part of their enterprise storage solution. Additionally, organizations can achieve massive scale with a software-defined storage solution by adding individual, heterogeneous hardware components as needed to increase capacity and improve performance in the solution. The scale-out capabilities facilitated by software-defined storage are especially beneficial in file system and object storage environments with large and rapidly growing volumes of unstructured data.

Intelligent resource utilization and automated management

Automated, policy-driven management of software-defined storage solutions helps drive cost and operational efficiencies. As an example, software-defined storage manages important storage functions such as disk caching, snapshots, replication, striping, and clustering. In a nutshell, these software-defined storage capabilities enable you to put the right data in the right place, at the right time, with the right performance, and at the right cost — automatically.

Cost efficiency

Rather than using expensive proprietary hardware, software-defined storage uses standard hardware to dramatically lower both acquisition costs and TCO for an enterprise-class storage solution. The software in a software-defined storage solution is standards based and manages the storage infrastructure as well as the data in the storage system.

In many cases, organizations can leverage their existing investments in storage, networking, and server infrastructure to implement extremely cost-effective file and/or object software-defined storage solutions.

In a July 2012 report, Gartner, Inc., found that the average acquisition cost per gigabyte of traditional multi-tiered storage systems ranged from $0.90/GB to $5/GB. By comparison, software-defined storage solutions averaged $0.40/GB in 2012 and have continued since then to drop in cost.

Limitless elastic data scaling

Unlike traditional storage systems, such as SAN and NAS, software-defined storage enables you to scale out with relatively inexpensive standard hardware, while continuing to manage storage as a single enterprise-class storage system. As you scale out your file and/or object storage infrastructure, performance and reliability continue to improve. As an example, IBM Spectrum Scale, the file system management member of IBM's Spectrum Storage family of software-defined storage products, delivers orders of magnitude more in I/O performance improvement as hardware is added, compared to conventional NAS (see Figure 2-1). IBM Spectrum Scale is introduced and described in detail in Chapter 3.

Software-defined storage provides massive, virtually limitless scalability. For example, IBM Spectrum Scale supports

- A maximum file system size of one million yottabytes
- 2^{63} (or approximately 9 quintillion) files per file system
- IPv6
- 1 to 16,384 nodes in a cluster

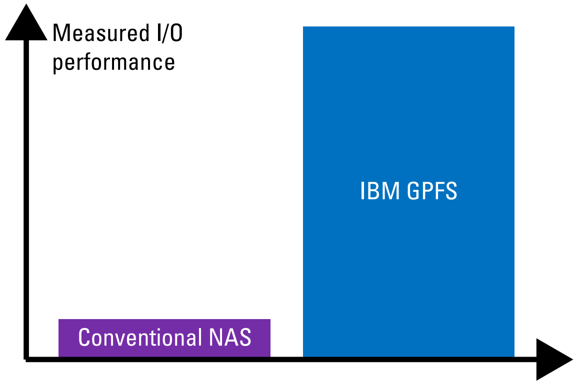


Figure 2-1: IBM Spectrum Scale delivers extreme I/O performance.

A yottabyte is equal to one trillion terabytes.

Enabling Unified File and Object Storage Systems

Software-defined storage has a very important role to play in the story about file and object storage. As file-based data continues to grow very quickly and as object storage becomes more prevalent — both of these trends thanks to the explosion of unstructured data — organizations naturally tend to simply "tack on" systems to accommodate them and their enormous growth. But just as adding rooms and wings to an originally orderly and rather cozy home can transform it into a chaotic, ramshackle mansion, adding storage systems compounds complexity and multiplies costs in many different ways.

IT budgets aren't growing much, even though the volume and velocity of data is. This situation drives enterprises and their IT departments to squeeze every bit of efficiency and cost savings they can from their information systems, including their file and object storage.

And this is exactly what software-defined storage solutions are designed to accomplish. By logically or "virtually" unifying many different, disparate, and often siloed storage systems into one solution that includes the capabilities to provide both file and object storage, software-defined storage dramatically increases simplicity and efficiency, leading to significant cost reductions. When your applications look in the mirror of software-defined storage, they see your ramshackle storage mansion as a tidy, orderly, easy to navigate family home. This is where the story about file and object storage take you next.

Chapter 3

Introducing IBM Spectrum Scale

In This Chapter

- Introducing IBM Spectrum Scale
- Learning about Spectrum Scale's unified file and object architecture
- Seeing the benefits of using Spectrum Scale with OpenStack
- Exploring the benefits of Spectrum Scale with object storage

IBM is a market leader in software-defined storage. The Spectrum Storage family of software-defined storage products includes IBM Spectrum Scale, a mature, highly sophisticated, and yet extraordinarily simple and powerful solution for deriving maximum value from structured and unstructured data. In this chapter, you discover much more about the many ways Spectrum Scale can help your enterprise lower IT costs, increase business agility, and gain competitive advantage. Spectrum Scale provides unified access to both file and object storage. The layered architecture of Spectrum Scale enables users to implement the most advantageous file and object storage configurations and protocols for their particular application needs.

Introducing IBM Spectrum Scale

IBM Spectrum Scale started out as a clustered file system and has evolved into so much more. Today it's a full-featured set of data management tools, including advanced storage virtualization, integrated high availability, automated tiered storage management, and high performance configurations to effectively manage very large quantities of data. Spectrum Scale is designed to support a wide variety of application workloads using a variety of access protocols and has been proven extremely effective in very large, demanding environments.

Unlike other storage solutions that must implement separate, add-on systems to handle files or objects, all data, regardless of the way it is accessed, is stored in the Spectrum Scale file system. After data is stored in the core file system, regardless of whether it's block, file, or object-based data, it can be accessed and managed in essentially the same ways. And a impressively wide range of data storage services and features can be applied across all the data, as appropriate, including snapshots, information life cycle management (ILM) tasks, storage tiering, asynchronous or synchronous data replication, and some unique data protection strategies.

And unlike other software-defined storage products, especially those in the object storage market, IBM Spectrum Scale offers native, high-performance and scalable access to block, file, and object data via almost all the standard storage protocols, including OpenStack Swift, Amazon S3, CIFS, NFSv3, NFSv4, and POSIX. Other solutions tie the access protocol to how the data is stored and need to use gateways to translate between the various data access protocols to fetch data.

Spectrum Scale isn't simply an object store, nor is it a pure NAS box. It is a very stable, exceptionally scalable clustered file system optimized for I/O performance with a breadth of support via different protocols built upon it. This is very different, and much more powerful and flexible than any other system.

A layered architecture

Spectrum Scale's software defined storage can be thought of as a core layer of storage services on which multiple data protocols may be deployed. Storage services can span virtually any type of storage, from tape to disk and even flash. As software defined storage, Spectrum Scale can span practically any storage pool, including a server's local disk, as seen in Figure 3-1.

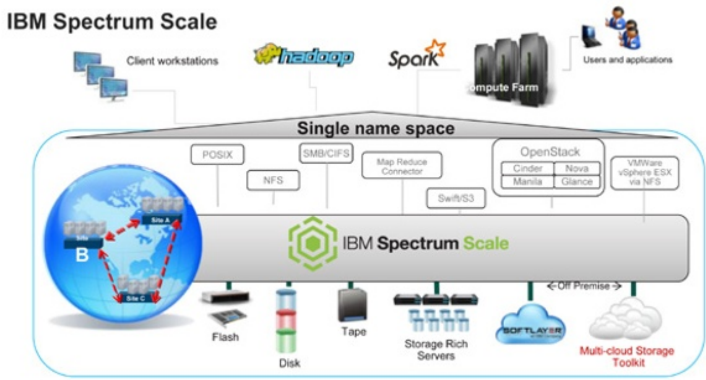


Figure 3-1: IBM Spectrum Scale provides a common storage plane.

Accessing the storage is enabled by the data services. These can be high-speed POSIX industry standard file access, NFS, CIFS/SMB, HDFS, or object store with OpenStack Swift or S3 interfaces. Regardless of the data access protocol, Spectrum Scale's intelligent storage services such as replication, tiering, and compression are available to the storage administrator. POSIX (Portable Operating System Interface) is a file-oriented IEEE (Institute of Electrical and Electronics Engineers) family of standards for maintaining compatibility between different variations of Unix and other operating systems. This is the standard file interface used to access data on most operating systems.

Spectrum Scale can scale performance and throughput using multiple file or object servers with fail-over capabilities to provide resiliency. It allows a group of servers to have concurrent access to a common set of file data over a common storage infrastructure, a network, or a mix of connection types. The servers can run any mix of AIX, Linux, or Windows Server operating systems. Spectrum Scale provides storage management, ILM tools, and centralized administration, plus it allows for shared access to file systems from remote Spectrum Scale clusters, providing a global namespace or related pool of storage all under the same management.

A Spectrum Scale cluster can be a single server, two servers providing a high-availability platform supporting a database application, for example, or thousands of servers used for applications such as the modeling of weather patterns. Spectrum Scale was designed to support high-performance workloads and has since been proven very effective for a variety of applications. Today, Spectrum Scale is installed in clusters supporting applications such as big data analytics, gene sequencing, digital media, and scalable file serving. These applications are used across many industries, including finance, retail, digital media, biotechnology, science, and government.

Spectrum Scale provides a unique set of extended interfaces and commands that can be used to provide advanced application functionality. Using these extended interfaces, an application can perform operations such as determining the storage pool placement of a file, creating a file clone, and managing quotas.

Removing data-related bottlenecks

Over the past decade, processors, memory, network, and bus performance have all increased exponentially, but disk speed performance has only increased 1.2 times. This performance gap slows data-heavy applications, delays schedules, and wastes expensive infrastructure. Spectrum Scale accelerates time to results and maximizes utilization by providing parallel access to data (see Figure 3-2).

Spectrum Scale achieves high performance I/O by

- Striping data across multiple disks attached to multiple servers
- Providing efficient client-side caching
- Executing high-performance metadata (inode) scans
- Supporting a wide range of file system block sizes to match I/O requirements
- Utilizing advanced algorithms that improve I/O operations

- Using locking based on a very sophisticated token management system to provide data consistency, while allowing multiple application servers concurrent access to the files

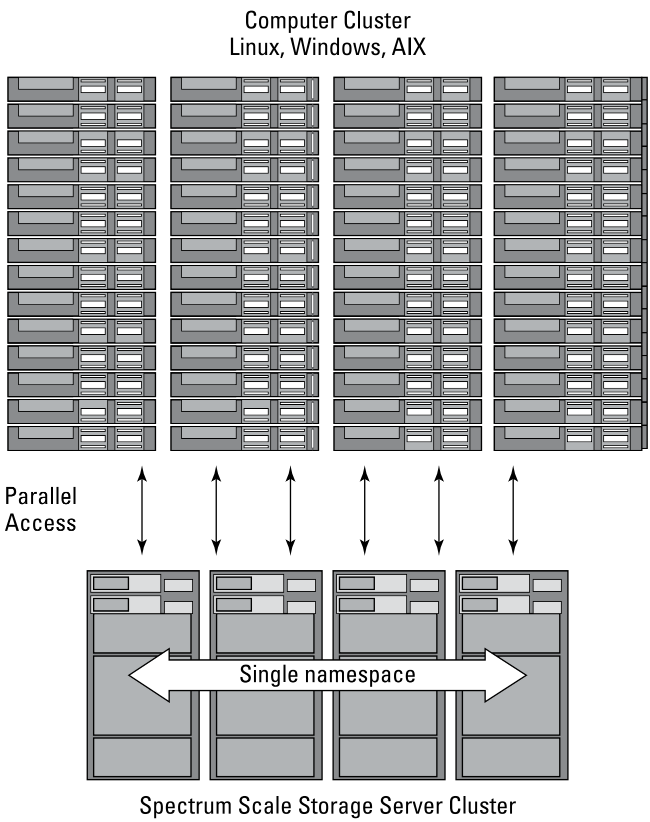


Figure 3-2: IBM Spectrum Scale eliminates storage bottlenecks.

When many servers need to use the same set of files at the same time, the file system must ensure that all the files are protected so one server can't change a file without the other servers knowing about the change. Keeping thousands of servers "in the loop" on file status while maintaining high performance is difficult, but it's a key-scaling feature of Spectrum Scale.

Spectrum Scale provides file integrity protection through a token process that keeps file data consistent by always ensuring there is only one owner for any given file. This method scales well in Spectrum Scale because any server in the cluster can be assigned file protection duty. There are two parts to managing tokens and file consistency: handing out the tokens and keeping file metadata up to date.

The server(s) that initially has the token for all files that aren't in use is called the *token manager.* You can assign one or more servers to be a token manager. Multiple token managers help each other by sharing the workload and by taking over when a fellow token manager fails. When a file is opened, the token manager hands off the token for that file to the server that's opening the file. The server using the file is now responsible for all metadata changes to that file. If a server wants to open a file that is already open on another server, the token manager redirects the request to the server that already has the file open and lets the two servers work out the details among themselves. This sharing of metadata maintenance across the entire cluster is what enables Spectrum Scale to grow very effectively. Many other file storage technologies rely on a single metadata server or centralized database. Such a single/centralized approach quickly limits how much data can be stored. Spectrum Scale addresses this limitation by distributing the workload across the cluster.

Simplifying data management

Creating a billion files isn't that difficult, but maintaining a storage solution containing a billion files or more takes industrial class tools. Spectrum Scale has the tools needed to manage petabytes of data and billions of files. The global namespace is easy to administer and can be scaled quickly, as desired, by simply adding more scale-out resources — eliminating "filer sprawl" and its associated issues.

Spectrum Scale has an administration model that's easy to use and consistent with standard file system administration practices. These functions support cluster management and other standard file system administration functions such as user quotas and snapshots.

A single Spectrum Scale command can perform a function across the entire cluster, and most can be issued from any server in the cluster. Optionally, you can designate a group of administration servers that can be used to perform all cluster administration tasks, or only authorize a single login session to perform admin commands cluster-wide. This allows for higher security by reducing the scope of server-to-server administrative access.

You can set quotas by user, group, or at the directory level to monitor and control file system usage across the cluster. When using quotas, you can easily see usage reports that include user, group, directory, inode, and data block usage. You can use snapshots to protect data from human error. You can create a snapshot of an entire file system or a subdirectory (called a *fileset,* which may contain a single dataset such as an object container). A snapshot is used to preserve the file system's contents at a single point in time. It contains a copy of only the file system data that has changed since the last snapshot was created and stores that data in the same pool as the original file, which keeps space usage at a minimum. Using the same pool simplifies storage administration because you don't have to set aside additional space for snapshot data. Snapshots provide an online backup capability that allows you (or an end-user) to easily recover from an accidental file deletion, or the ability to compare a file to an older version.

Most application clusters need a method to get data into and out of the cluster. Because Spectrum Scale runs directly on a standard server, you can use a variety of tools to get file data in and out of a Spectrum Scale file system. To better enable end-user access to a Spectrum Scale file system, the file system can be exported to clients outside the cluster through multiple protocols, including the capability of exporting the same data from multiple servers.

Basic cluster configuration

When it comes to cluster configuration options, Spectrum Scale is a multi-function tool. It can be deployed on practically any server or in the cloud. The same Spectrum Scale software is installed on all the servers in a cluster. What a server does, and how it participates in the cluster, is based on the hardware it has available and what you need it to do. Cluster configuration is independent of which file system features you require.

Network shared disk configuration

Spectrum Scale clusters often use a cluster configuration with the network shared disk (NSD) protocol to provide high-speed data access to applications running on LAN-attached application servers. Data is served to these client servers from an NSD server. In this configuration, storage can be provided through the NSD servers in several ways, including direct attached and storage area network options. Each NSD server is attached to all or a portion of the storage.

Spectrum Scale uses a network to transfer control information and data to the application clients. The network doesn't need to be dedicated to Spectrum Scale, but it should provide sufficient bandwidth to meet the needs of Spectrum Scale and your other applications sharing the bandwidth.

In an NSD server configuration, a subset of the total server population is defined as NSD servers. The NSD servers are responsible for the abstraction of data blocks across an IP-based network. The fact that I/O is remote is transparent to the application. Figure 3-3 shows an example of a configuration where a set of application servers are connected to a set of NSD servers via a high-speed interconnect or an IP-based network (such as Ethernet). In this example, data to the NSD servers flows over the shared storage network, and data and control information flows to the application clients across the LAN.

The choice of how many servers to configure as NSD servers is based on individual performance requirements and the capabilities of the storage subsystem. High bandwidth LAN connections should be used for clusters requiring significant data transfer. To enable high-speed communication, Spectrum Scale supports TCP/IP using whatever hardware you have available (1Gbit and 10Gbit Ethernet, link aggregation, Internet Protocol over InfiniBand (IPoIB), and Remote Direct Memory Access (RDMA) on InfiniBand for control and data communications.

InfiniBand is a switched fabric communications link commonly used in High Performance Computing (HPC) installations. With speeds up to 100Gbit/sec and very low latency, it is well suited to workloads with high I/O demands.

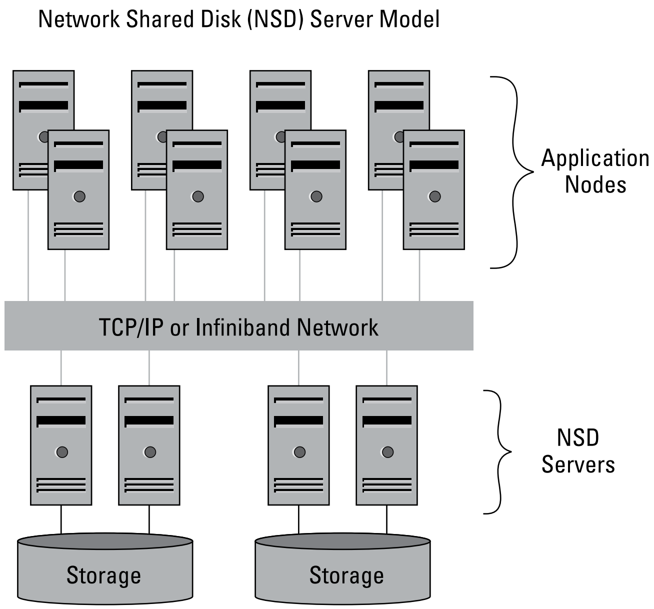


Figure 3-3: Network I/O.

TIP

Spectrum Scale provides the capability to designate separate IP interfaces for intra-cluster communication and the public network. This provides a more clearly defined separation of communication traffic. An NSD server architecture is well suited to clusters with sufficient network bandwidth between the NSD servers and application clients — for example, statistical applications like financial fraud detection, supply chain management, or data mining.

Sharing data across Spectrum Scale clusters

You can share file system data across Spectrum Scale clusters via two methods: Spectrum Scale multi-cluster and Active File Management.

Spectrum Scale multi-cluster

Spectrum Scale multi-cluster allows you to utilize the Spectrum Scale NSD protocol to share data across clusters. With this feature, you let other clusters access one or more of your file systems, and you mount file systems that belong to other Spectrum Scale clusters for which you've been authorized. A multi-cluster environment permits the administrator access to specific file systems from another Spectrum Scale cluster. This feature permits clusters to share data at higher performance levels than file sharing technologies like NFS or CIFS.

Spectrum Scale multi-cluster isn't intended to replace file sharing technologies that are optimized for desktop access or for access across unreliable network links. Multi-cluster capability is useful for sharing across multiple clusters within a single physical location or across multiple locations.

In Figure 3-4, Cluster A owns the storage and manages the file system. It may grant access to file systems that it manages to remote clusters, such as Cluster B. In this example, Cluster B doesn't have any storage but that isn't a requirement. Commonly, in the case where a cluster doesn't own storage, the servers are grouped into clusters for ease of management. When the remote clusters need access to the data, they mount the file system by contacting the owning cluster and passing required security checks. In Figure 3-4, Cluster B accesses the data through the NSD protocol.

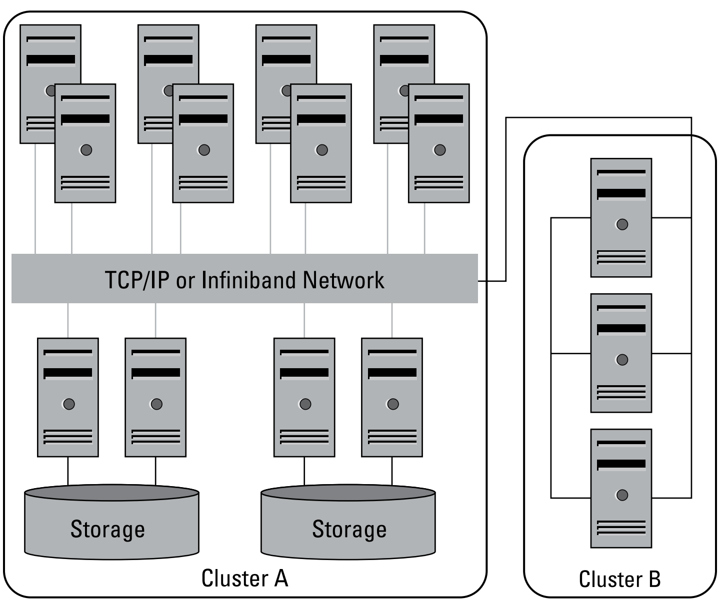


Figure 3-4: Multi-cluster configuration.

Multi-cluster environments are well suited to sharing data across clusters belonging to different organizations for collaborative computing, grouping of clients for administrative purposes, or implementing a global namespace across separate locations. A multi-cluster configuration allows you to connect Spectrum Scale clusters within a data center, across campus, or across reliable WAN links. ***Note:*** Strong consistency of the data is maintained between all clusters accessing a file system. This means all updates are immediately synchronized across all clusters, which is highly desirable for groups collaborating on a single dataset.

Empowering global collaboration

So what can you do if your network link is really long or not so reliable? Spectrum Scale provides low latency access to data from anywhere in the world with Active File Management (AFM) distributed disk caching technology. AFM expands the Spectrum Scale global namespace across geographical distances, providing fast read-and-write performance with automated namespace management from anywhere in the world. As data is written or modified at one location, all other locations get the same data with minimal delays. These game-changing capabilities accelerate project schedules and improve productivity for globally distributed teams.

At first glance, AFM may seem like any other cache. But when you start looking at what you can do with these basic behaviors, the options start multiplying. To understand some of the possible options, take a look at how AFM can be used.

A cache relationship is defined when you create a fileset. You can have up to 1,000 independent cache relationships in each Spectrum Scale file system. Each cache fileset can have its own mode and configuration parameters and still share a common set of storage. A cache file system can be an entire file system in size, or it can be limited by quotas. A read-only cache contains data that can't be modified — it's read-only. Data is copied into the cache fileset by Spectrum Scale, on demand, by opening a file or running a command to prefetch (or pre-cache) a list of files. A single target can have as many read-only cache relationships as it has bandwidth available, because the cache does all the work. A target doesn't know there's a cache, and a cache only knows about its own target.

Two filesets in the same file system can be caching data from the same target, and they wouldn't know about each other.

The isolation between the cache and target is what makes this model scale so well. You can have as many as 1,000 caches because each cache only has to track one relationship. And then it gets interesting! Beyond the many-to-one cache relationships, you can also cascade caches. A target can have a dual personality; it can be a cache and a target at the same time. For example, say the data originates in London and is cached in New York. The office in Tokyo needs a copy of the same data, so it could use either New York or London as its target depending on which site has the better connection.

In read-only mode, data consistency is simple: The target is the only place where the data can be created and modified. But there are other caching modes that provide additional functionality. For example, in an *independent writer* cache, data can be created and modified in the cache and at the target. When a file is created or modified in the cache, the write operation completes locally and the changes (and only the changes) are automatically and asynchronously copied to the target. You can also have multiple independent writers updating data for the same target. When a file is created or modified at the target, the caches automatically detect the change and update themselves.

By using these basic caching techniques, you can allow all your sites to see the same data all the time. Figure 3-5 shows one example of using AFM to create a single view of all the data from multiple sites.

In this example, each site is the target for one-third of the data. The other two sites have cache relationships with the other sites. This means that no matter which site you log into, you see exactly the same file system structure and you have access to all the data — it may just take a little longer to read a file if it has not yet been copied to your site.

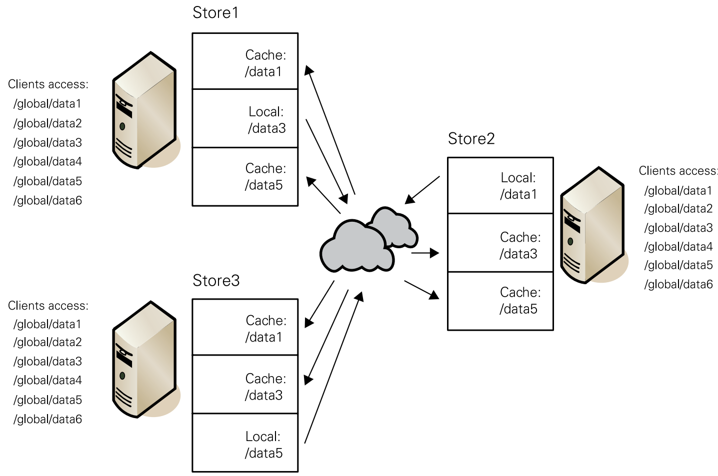


Figure 3-5: Global namespace using AFM.

Managing the full data life cycle at lower costs

Spectrum Scale enhances information life cycle management and lowers your data management costs significantly by using multiple tiers of storage, including tape. With powerful policy-driven automation and tiered storage management, you can create optimized tiered storage pools by grouping storage resources based on performance, locality, or cost characteristics. Data migrated to tape remains visible in the file system and is directly accessible by end-users. Migration policies transparently move data from one storage pool to another without changing the file's location in the directory structure.

For example, you can create a rule for thresholds that moves files out of the high performance pool if it's more than 80 percent full, thereby mitigating potential bottlenecks in the high performance pool. Also, Spectrum Scale monitors the activity levels or "heat" of each file. You can establish policies that move data to higher performance storage (such as flash) automatically when it gets hot enough and back to more capacity-optimized hard disk drives when the data cools. Spectrum Scale information life cycle management capabilities and benefits include

- Policy-driven automation and tiered storage management
- Flexibility to match the cost of storage to the value of data
- Ability to create tiers of storage that include high-performance flash, high-speed disk drives, or high-capacity disk drives
- Full integration with other IBM Spectrum Storage family members such as Spectrum Control and Spectrum Archive that provide the following functionality:
    - Spectrum Scale handles all metadata processing, then hands the data to Spectrum Archive for storage on tape.
    - Data is retrieved from the external storage pool on demand when an application opens a file, for example.
    - Policies move data from one pool to another without changing the file's location in the directory structure.
    - Thresholds can move files out of the high-performance pool if more than 80 percent full, for example.

TIP

Tape migration provides tiered data storage for approximately one-fifth the cost per terabyte of disk.

Spectrum Scale with Object Storage

If you have large-scale file system management needs, IBM Spectrum Scale can fill the bill. It has been deployed over the past decade by thousands of commercial, scientific, academic, and governmental enterprises around the globe and has effectively tackled some of the largest file system challenges on the planet.

Object storage now brings the promise of secure and ubiquitous data access, Spectrum Scale can also be used in a tightly integrated solution with other software defined storage tools to tackle the unique challenges of object storage.

Spectrum Scale and OpenStack

Now, turn your attention from addressing file system management needs to solving the unique challenges presented specifically by object and cloud storage. For this task, OpenStack is added to the mix. One key OpenStack project is Swift — the open source object storage software platform that's widely deployed in public and private clouds. Together, Spectrum Scale and key elements of OpenStack Swift provide an enterprise-class object storage solution that efficiently stores, distributes, and retains critical data.

OpenStack technology consists of a series of interrelated projects that control pools of processing, storage, and networking resources throughout a data center. OpenStack has a modular architecture with various components:

- **OpenStack Compute (Nova):** A cloud computing fabric controller
- **Block Storage (Cinder):** Provides persistent block-level storage devices
- **Object Storage (Swift):** Scalable redundant storage system
- **Image Service (Glance):** Provides discovery, registration, and delivery services for disk and server images/backups
- **File Share Service (Manila):** Presents the management of file shares (for example, NFS ***and*** CIFS) as a core service to OpenStack
- **Identity Service (Keystone):** Provides Identity, Token, Catalog, and Policy services for all OpenStack projects

With OpenStack, you can control pools of processing, storage, and networking resources throughout a data center. And while OpenStack provides open source versions of block and object storage, many OpenStack developers have identified a need for more robust storage to support cloud-scale applications. While many OpenStack developers feel they can architect around limitations in OpenStack compute capabilities and robustness, storage has a much "higher bar" as far as resiliency and reliability go.

IBM Spectrum Scale unifies OpenStack virtual machine (VM) images, block devices, objects, and files with support for all the OpenStack projects. The ability to authenticate administrators and users (Keystone), who can provision file shares (Manila), that can then be used to provision and store volumes and images (Cinder and Glance), and then use file clones to efficiently and quickly share data within and between components is a big advantage for cloud-scale OpenStack application developers.

The robustness and features of Spectrum Scale combined with OpenStack Swift object extensions provide an enterprise-grade object store with high-performance, high storage efficiency through industry leading erasure coding algorithms and compression, tape integration, wide-area replication, transparent tiering, encryption, end-to-end checksums, and snapshots — capabilities most object-based storage offerings can't match today. OpenStack on Spectrum Scale delivers compelling efficiencies in a single unified storage solution that can support object and file access to the same data with robust and efficient Spectrum Scale data protection. Spectrum Scale can also reduce the amount of raw storage you need to use compared to object storage systems that rely strictly on replication.

Spectrum Scale for Object Storage

Spectrum Scale provides a robust object storage solution through inclusion of the most widely used open source object store today, OpenStack Swift. The result is a multi-tenant storage system that supports the two most popular object APIs: Swift and Amazon S3, along with all of Spectrum Scale's enterprise features. Further, Spectrum Scale for Object Storage can be deployed across multiple sites, building a robust Active-Active cloud storage solution that can support the failure of one or more sites and continue to make all data available without interruption. For example, with three sites, Spectrum Scale will place objects at two of the sites to ensure availability of the objects upon a single site failure.

OpenStack Swift provides a robust object layer with an active developer community that is continuously adding innovative new features, such as versioning, rate limiting, and expiration to name just a few. Further, standard APIs and SDKs allow Spectrum Scale users to customize the solution to their environment and requirements. To ensure compatibility with the Swift packages over time, no code changes are required to either Spectrum Scale or Swift to build the solution. Further, Spectrum Scale removes the worry of managing or deploying OpenStack Swift by providing a robust set of deployment and management tools, including CLIs as well as a management and monitoring GUI.

Figure 3-6 shows the basic Spectrum Scale architecture with the various OpenStack components, including Swift, which enables object storage capabilities. At the top are shown various types of applications that might access Spectrum Scale data, such as a user application, mobile apps, or the common analytics software known as Hadoop.

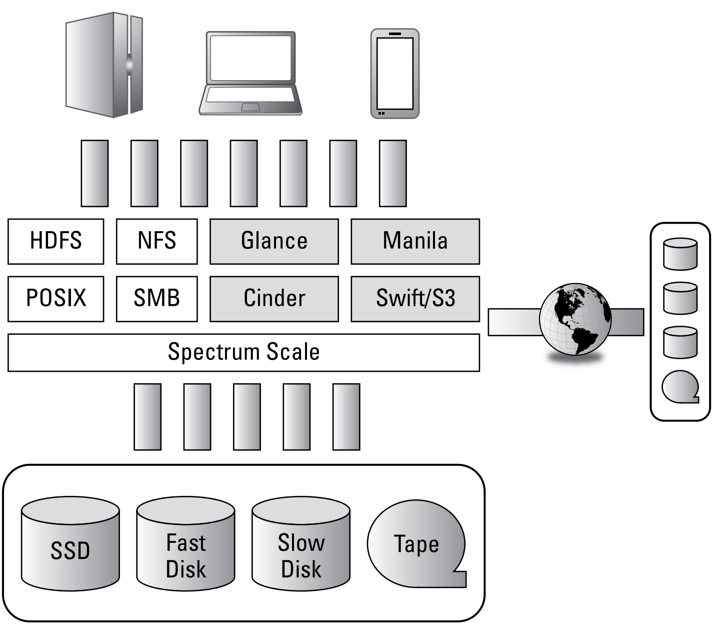


Figure 3-6: Spectrum Scale with OpenStack architecture.

The basic Spectrum Scale with object storage architecture is simple. Users/client applications access Spectrum Scale through high-performance Ethernet networks. The configuration of Spectrum Scale is mostly independent of its use in Spectrum Scale with OpenStack. For existing deployments of Spectrum Scale, OpenStack components such as Swift can be layered on top, with its data stored in a new Spectrum Scale independent fileset.

Because Spectrum Scale can store data for many different applications, it is wise to place the object store data in a separate Spectrum Scale management entity that is called an *independent fileset*. By doing so, it allows Spectrum Scale information life cycle management, including snapshots and backup, to uniquely identify and manage the object store data.

The physical storage architecture is also largely independent, but it will affect the level of efficiency, fault tolerance, and performance. Because the storage also constitutes the bulk of storage costs, the choice of storage hardware also determines the cost/performance of the system. The example physical storage configuration used in the architecture diagram shows that SSDs (flash drives), fast (SAS) and slow (SATA) disks, and tape can all be deployed to cover a wide range of workload requirements.

In the example Spectrum Scale for Object Storage architecture, consider the following key points:

- Data protection is handled by the Spectrum Scale, with Swift writing a single instance of each object into the file system.
- Additional Spectrum Scale nodes provide both additional object access bandwidth and an additional level of fault tolerance in case of node failure.
- Each Spectrum Scale server can independently access all objects.
- Additional capacity can be quickly deployed, avoiding the need to increase costs by over-provisioning capacity.
- Most (if not all) existing Spectrum Scale enterprise features can be used with the object storage solution.

File and Object in a single storage system

Both file and object storage have benefits depending on the application requirements. It's important to remember that applications that require files shouldn't be forced to suffer with object semantics, and the same holds true for the reverse.

TIP

Spectrum Scale supports native file and object access, avoiding the need for costly application re-writes. Native protocol support eliminates the complexity and overhead of gateways that copy and translate between protocols. With Spectrum Scale, all the benefits of both files and objects are realized in a single system without suffering from the headache of layering protocols.

There are many use cases requiring both file and object access in a single storage system because most applications today are written for file. Typical use cases are

- Data created as a file and shared globally using objects. Content repositories of images, videos, records, or backups are often built this way.
- Collecting data from many sources via object and then analyzing it using SAS, Hadoop, Spark, or other analytics framework. Running analytics natively on the file interface is critical to realize good performance.

Summary

Spectrum Scale for object storage is ideal for multiple environments:

- Sites that already use Spectrum Scale and are seeking to support Object Storage along with their existing data
- Building OpenStack private clouds
- New analytics workloads and web services that scale out using REST interfaces, such as SWIFT and S3
- Deploying capacity storage with the ease of scaling and lower costs than traditional storage systems
- Sites that have both file and object requirements, either to the same or separate datasets

- Sites that are simply seeking an enterprise-ready, cost-efficient, and high-performing object solution across one or more sites
- For this use case, Spectrum Scale initially provides data management for the object store only, but gives users the option to extend to further types of applications as their requirements grow.

What differentiates Spectrum Scale with OpenStack Swift object storage from other object storage solutions is its capability to offer an industry-leading solution for workloads that demand the following capabilities:

- Enterprise-ready features, such as snapshots and backups to prevent data loss
- High-throughput access to the objects
- High-capacity, dense storage
- Levels of data scaling that can enable your IT infrastructure for years into the future
- Unified file and object storage with federated identity controls

Spectrum Scale for object storage transforms object storage from a major hurdle within your enterprise into an opportunity for innovation and competitive advantage. If you have already deployed IBM Spectrum Scale, then adding OpenStack Swift to create this leading-edge solution is a breeze. If you're starting relatively from scratch, then the commitment and support that IBM has provided to the OpenStack initiative and community helps your organization when you implement this full-featured object storage solution with proven interoperability.

Chapter 4

Getting to Know the IBM Elastic Storage Server

In This Chapter

- Introducing the Elastic Storage Server (ESS) solution for IBM Spectrum Scale
- Highlighting ESS fault tolerant features

IBM Spectrum Scale is a software solution installed on clouds or server and storage hardware to provide capacity. It's possible to deploy a Spectrum Scale on any standard server and storage hardware. Many prefer an integrated solution that combines the relevant hardware and Spectrum Scale software. Such an integrated solution is quick to deploy, ensures performance, and eliminates risk of building it yourself.

This chapter introduces the IBM Elastic Storage Server. ESS is the optimal building block for IBM Spectrum Scale. It combines the performance of IBM Power servers with Spectrum Scale software to offer a high-performance, scalable building-block approach to modern file and object storage needs. ESS allows you to start with a configuration that meets your organization's current needs and expand capacity and bandwidth with each additional ESS to meet your future needs.

Disk drive performance hasn't changed significantly in many years, but in the last ten years disk drive capacity has increased from a maximum of 100GB per disk to more than 6,000GB — a 60x increase! In fact, 8TB disks are already coming into service and 10TB drives loom on the near horizon. To put this in context, assume that you can read a single disk drive at a speed of 60MB per second. In 2003, backing up that drive took 28 minutes. Today, because disk drives haven't gotten much faster, backing up a single disk full of data can take nearly a full day. This same performance issue also impacts disk maintenance operations, such as RAID rebuilds. When a 8TB disk fails, how long does it take to read and write enough data to put a new disk in its place? These types of performance issues are one of the key driving forces behind the invention of the IBM Elastic Storage Server (ESS).

Delivering an End-to-End ESS Solution

An ESS implementation of IBM Spectrum Scale is constructed by deploying one or more Elastic Storage Servers, each of which runs Spectrum Scale software and shares both storage management and capacity duties. This building block or grid architecture approach offers great resiliency to the overall Spectrum Scale implementation as well as many other advantages, such as linear scaling of both capacity and performance, as additional ESS building blocks are added.

An individual Elastic Storage Server is made up of IBM Power servers and storage media, such as flash or disk drives, and includes all the software and networking components in a fully integrated solution. IBM thoroughly tests and optimizes each ESS for reliability, interoperability, and performance so enterprises can quickly deploy the system and get to work achieving their business goals.

Features of the Spectrum Scale ESS solution include

- Scalable building-block approach to storage
- Multiple configurations to support various capacities of flash and disk drives
- Up to 348 drives in the GL6 model
- FDR InfiniBand, 10 and 40Gbit Ethernet interconnects, or both
- Spectrum Scale RAID technology using erasure codes for sustained, predictable performance and rapid rebuilds
- No hardware controllers; disk pooling and management are performed by the Spectrum Scale RAID software

The ESS approach to a Spectrum Scale implementation offers a number of benefits:

- **Data redundancy:** Spectrum Scale RAID supports highly reliable 2-fault-tolerant and 3-fault-tolerant Reed-Solomon-based parity codes (Erasure Coding) as well as 3-way and 4-way replication.
- **Large cache:** Using a combination of internal and external flash devices along with the large memory cache in the Power servers, ESS is better able to mask the inefficiencies and long latency times of Nearline SAS drives while still leveraging the high-density of the drives themselves.
- **Graphical user interface:** The intuitive ESS GUI allows management and monitoring of the system, both locally and remotely.
- **Superior streaming performance:** The system can deliver over 25GB per second of sustained performance.
- **Scalability:** As server configurations are added to an installed configuration, the capacity, bandwidth, performance, and the single namespace all grow. This means installations can start small and grow as data needs expand.

ESS is an optimum storage building block for Spectrum Scale because

- Each ESS model has the optimum combination of compute power to go with the back-end storage.
- It utilizes IBM Power servers built for processing power and high I/O bandwidth for moving data.
- It includes Spectrum Scale RAID software.

Using IBM Power Systems servers for ESS brings some real advantages:

- Massively parallel with 96 hardware threads per 12 core processor and 8 threads per core
- Up to 410GB per second (peak) memory bandwidth to CPU

- Up to five times the I/O bandwidth, three times the cache, and as much as four to six times the memory bandwidth compared to typical x86 systems
- Up to two times better per-core performance over Intel for Java
- Compression offload for simplification

Figure 4-1 shows the hardware that's used in the ESS and the different shipping configurations.

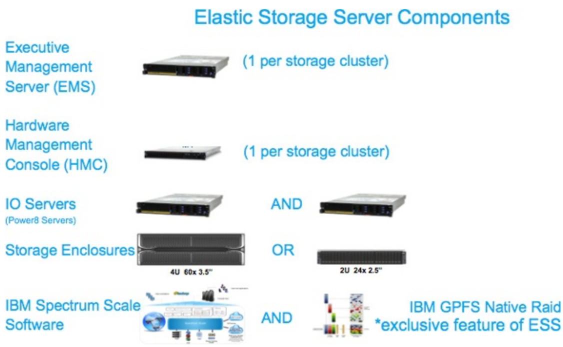


Figure 4-1: The IBM Elastic Storage Server.

ESS always includes a pair of servers for high availability matched with disk enclosures (JBOD — just a bunch of disks) in configurations of either two rack units (U) high with 24 2.5" disks or 4U high with 60 3.5" disks. ESS comes in two different models: GS or GL using either the 2U or 4U JBODs. Each model allows choosing the number of JBODs or the type and capacity of the disk drives.

TECHNICAL STUFF

RAID (Redundant Array of Independent Disks, originally Redundant Array of Inexpensive Disks) is a data storage technology that distributes data across multiple drives in one of several ways (called RAID levels), depending on the level of performance and protection required.

How does ESS defy physics?

Well, it doesn't. As physics professors will be quick to point out, you can't defy physics. But you can take a completely new approach to solving a problem. ESS incorporates a new approach called *Spectrum Scale RAID* to solving the issue of long RAID rebuild times with really large disks.

In a traditional RAID 6 array, for example, a set of disks are grouped together and data is stored on all the disks in the set along with chunks of information called *parity bits* that can be used to "rebuild" a failed drive. When a drive fails in a RAID 6 array, parity bits on the other disks are read to reconstruct the data from the failed drive, which is then written to a new disk. The entire contents of the failed disk must be written to the new disk before the array is again completely protected from further disk failures.

To address these long disk rebuild times and the lowered data protection that occurs during that period, Spectrum Scale RAID uses erasure coding technology where the number of disks in the Spectrum Scale RAID group is dynamic and the data and parity are striped across all available disks as the data arrives. Using Erasure coding, each block of file data is cut into what is called a stripe. So if you use an 8+2 configuration, you get 8 data stripes and 2 parity stripes; the difference is that this striping is independent of the number of available disks. This means that if, for example, you have 58 drives and you are using 8+2 parity, the first block of data is spread over 10 of the drives, the second block of data is spread over 10 other drives, and so on.

So, how does this improve disk rebuild speeds? Instead of having to rebuild a entire failed disk as in RAID 6, Spectrum Scale RAID just needs to ensure that all the data that was on the failed drive is copied or computed (parity is computed) on other drives. So ESS doesn't defy physics, it just spreads out the work. Instead of writing 4TB to a single spare drive, the surviving 57 drives just need to swap some data and move on.

A rebuild on a RAID 6 array with 4TB drives can negatively impact the performance of your entire storage array for 17 to 24 hours. But with Spectrum Scale RAID using erasure coding, the impact is less than 15 minutes.

Bringing Parallel Performance to Fault Tolerance

The speed of recovery, along with other advanced data protection features, makes ESS much more reliable, even as the number of drives increases. At the heart of this capability is an important feature of ESS called Spectrum Scale RAID technology, which is not tied to a set number of drives. In ESS, there are standard configurations, but its data protection/RAID technology can work with as few as 11 drives to as many as several hundred drives. This keeps ESS relevant, even with new and different drive packages and types.

Advanced data protection

With advanced data protection, you have two categories: data integrity and the disk hospital.

Data integrity

For a long time, Spectrum Scale has been able to rely on the RAID controller to take care of things like making sure what you wrote to the disk last week is correct when you read it this week. Managing the data all the way down to the raw device comes with additional responsibility and some great functional benefits. One of these responsibilities is to make sure the data hasn't changed since it was written to disk. In ESS, this is accomplished in software using a data checksum. A *checksum* is a number that's computed by looking at a chunk of raw data. This number is stored in a separate area and used when the data is read to ensure it hasn't changed since it was written. Because Spectrum Scale now has that checksum information, it can be used to ensure data integrity all the way from the disk to the Spectrum Scale application client, thereby protecting data from storage, system, or network errors.

Hard disks don't report some read faults and occasionally fail to write data while actually claiming to have written the data. These errors are referred to as silent errors, phantom-writes, dropped-writes, or off-track writes. Spectrum Scale RAID implements an end-to-end checksum calculated and appended to the data to detect silent data corruption.

If the checksum or version numbers are invalid on read, Spectrum Scale RAID reconstructs the data using parity or replication and returns the reconstructed data and a newly generated checksum to the client. Thus, both silent disk read errors and lost or missing disk writes are detected and corrected.

Disk hospital

Another responsibility Spectrum Scale RAID now shoulders is to make sure the disks are healthy. Now, all storage components are monitored and maintained regularly through Spectrum Scale's "disk hospital" functionality. If there are media errors, path issues, or simply disks behaving badly, Spectrum Scale responds to these events with a series of actions, including rerouting requests and power-cycling disks.

More efficient use of disk performance

Compared to conventional RAID, Spectrum Scale RAID implements a sophisticated data layout scheme that uniformly spreads user data and redundancy information across all the disks of an ESS storage array.

With Spectrum Scale RAID, you don't have idle spare disks sitting and waiting to be called into service. Spreading data over all available drives, including spares, transforms a hot spare into hot spare space. So, instead of assigning a drive to be a spare, the Spectrum Scale RAID software just keeps space free for failure events on all drives in the disk group.

Spectrum Scale RAID can significantly shorten the time to recover from a disk failure, which reduces the rebuild overhead for client applications. When a disk fails, data is rebuilt using all operational disks in the array, thus providing bandwidth that is much greater than in a conventional RAID 6 group.

The disk hospital functionality is a key feature of Spectrum Scale RAID that asynchronously diagnoses errors and faults in the storage subsystem. Spectrum Scale RAID times out an individual disk I/O operation after about ten seconds, thereby limiting the impact from a faulty disk on a client I/O operation.

The suspect disk is immediately admitted into the "disk hospital" where it's determined whether the error was caused by the disk itself or by the paths to it. While the hospital diagnoses the error, Spectrum Scale RAID uses its redundancy codes to reconstruct lost or erased data stripes for I/O operations that would otherwise have used the suspect disk.

Bringing It All Together

Elastic Storage Server is a storage appliance that follows in the Spectrum Scale tradition of being adaptable. All the Spectrum Scale RAID software exists in addition to the existing Spectrum Scale functionality. This means that in addition to the new capabilities of Spectrum Scale RAID, you can continue to use all the Spectrum Scale features, including the ability to mix any storage into the cluster. You can use network shared disk protocols on an ESS in an existing cluster and even in an existing file system.

Already proven in the field, Spectrum Scale RAID has demonstrated the capability to succeed under very demanding file and object storage I/O workloads. ESS is a key foundational component of the overall Spectrum Scale solution architecture.

Chapter 5

Ten Advantages of IBM Spectrum Scale with Object Storage

In This Chapter

- Highlighting several advantages that IBM Spectrum Scale with object storage offers
- Looking at ways that IBM Spectrum Scale itself can help you solve storage challenges

There are many different file and object storage challenges, each with a unique set of requirements related to scale, performance, retention, access, availability, and capacity. This chapter describes ten (okay, 11 — it's like a bonus!) advantages that Spectrum Scale with object storage offers to help you solve your file and object storage challenges.

Unify File and Object Storage

Applications can store a variety of data types by using almost any protocol in a single Spectrum Scale file system, including file-based information, objects, and even block-based data. The native support for multiple protocols enables you to choose the protocol(s) that work best for each of your applications.

TIP

Unifying object and file storage with Spectrum Scale offers significant cost and efficiency benefits for your overall storage infrastructure. You can implement one storage architecture to address all your storage requirements. This single solution can bring your entire enterprise information store under one namespace; replicate it to any geographical location according to policies you set; support your entire range of applications — from databases to rich media file serving; and then make it all faster and lower cost.

TIP

Digging a little deeper into the advantages of integrating file and object storage into one system, you see that consolidating all data types into a common management domain allows you to access information of differing types. For example, you may create data via the Spectrum Scale file interface but then share it globally using the object interface. Your data analytics can be significantly enhanced by moving in the other direction — you ingest data as objects but then make it available to analytics engines such as Hadoop using the file interface.

Leverage Advanced Data Protection

Using Spectrum Scale to provide data protection services increases both the efficiency and performance of your object storage system in the following ways:

- With Spectrum Scale RAID, storage efficiency rises from 33 percent to up to 80 percent:
  - You can leverage inexpensive storage such as "slow, fat" disk drives or even tape.
  - You get much faster disk rebuild times than with conventional RAID schemes.
- Spectrum Scale offers policy-driven snapshots for both files and objects.
- Disk failure recovery doesn't cause data to flow over the storage network, lowering network traffic. Recovery is handled transparently and with minimal impact to applications.
- Applications now realize the full bandwidth of the storage network because Spectrum Scale writes only a single copy of each object to the storage servers.

- With Spectrum Scale data striping, the maximum object size is increased up to 5TB. Large objects do not cause capacity imbalances or server hot spots, and they don't inefficiently use available network bandwidth.
- No separate replication network is required to replicate data within a single cluster.
- Spectrum Scale node failure does not require any movement of data between nodes or the underlying storage media.

Improve Application Performance

Data-intensive applications are defined by the fact that they need to read or write a large amount of data to get the job done. Speeding up data-intensive applications is easy — use a faster storage infrastructure. In reality, that may not be the case because many storage solutions don't scale efficiently. The only way to improve I/O performance when the storage device doesn't scale is by using many small data storage pools or devices and modifying your applications to utilize them concurrently.

If data for a project isn't spread across multiple pools/devices, data hot spots can be created due to evolving application. This occurs when you have a high level of I/O required by applications/clients for a set of data stored on a single device or within a single storage pool/resource. Spectrum Scale eliminates hot spots by spreading data across all the available storage resources. It's also able to move data to faster storage such as flash, and it offers advanced data caching features to speed access even further.

Spectrum Scale is designed to support I/O workloads run by a large number of processes and servers concurrently accessing a common set of file data. It stores and accesses data in blocks striped across many disks or other storage resources such as the modules in a flash array. Striping data for the set of files being used allows the underlying storage to deliver at full performance.

For many storage solutions, it's necessary to partition data across separate storage pools or devices to achieve better overall throughput. However, this approach greatly complicates management tasks and makes it difficult to keep all your storage resources busy. The opposite is true for Spectrum Scale, which allows you to fully leverage the performance of all the underlying storage resources. It spreads the data across all the available storage so there are no idle storage resources, and investment isn't being wasted.

For example, in the entertainment industry, when a celebrity makes the news, it creates a burst in interest and activity as fans flock to download pictures and movie clips of that person. It causes a set of data associated with the celebrity to become "hot," which can degrade performance. Some storage solutions use techniques such as file replication or DRAM caching to support these burst types of workloads.

The challenge with replication-based solutions is that they require extra storage space for file copies and additional bandwidth to create copies. In the case of streaming media, the load against each file varies with time, and replication must be done continuously. DRAM caching is really only cost-effective for very large servers and thousands of concurrent media streams.

Spectrum Scale enables all the servers in a cluster equal access to all system data and metadata in parallel. This improves performance for metadata-intensive applications by accelerating application I/O operations. Spectrum Scale allows any server in a cluster to read from or write to any of the storage, which enables applications performing concurrent I/O to achieve very high data access rates.

Leverage Spectrum Scale Features

Spectrum Scale allows object storage to benefit from its many robust storage management features, such as

- Simplified GUI
- Global namespace

- Encryption
- Backup
- Disaster recovery
- Dynamic tiering
- Tape integration
- Remote caching

Compared to conventional systems, Spectrum Scale's data management features offer much greater range and power. Traditional storage systems "know" less than half a dozen attributes about file data, including I/O size, type of storage, latency, and location. Spectrum Scale is far more intelligent and knows nearly 50 attributes of a file. These include

- File name and type
- Latency and location
- Time of file creation
- Time of last access and last change
- Access permissions
- File "heat"
- The group and user who own the file
- The space efficiency of the file
- The time of the last metadata change

These attributes allow data to be managed with much greater granularity, control, and efficiency. One or more of these data attributes can be used to set storage system policies to be applied to specific files or data sets. For example, policies may be established to improve the storage performance at one of your field offices or increase security by further restricting access to especially sensitive information. These attributes enhance an object storage environment in which object metadata is a key ingredient in storage management by providing more use and performance related attributes.

Proven Product Reliability and Resiliency

Spectrum Scale is a proven storage platform successfully deployed in thousands of customer environments and in production for many years.

Downtime — whether planned or unplanned — is costly, both in terms of budget and in additional hours spent by IT staff. Ensuring applications are always able to access data requires a robust, high availability storage solution. Applications must be able to continue to run in the event of hardware failure and during system changes such as capacity expansion or hardware and software upgrades.

The core Spectrum Scale software is fault tolerant and will continue to function should a server or even a storage system fail. Other nodes continuously monitor the health of the cluster and file system components. When a failure is detected, the appropriate recovery action is taken automatically. Extensive logging and recovery capabilities maintain metadata consistency when application servers holding locks or performing cluster services fail.

For additional protection, Spectrum Scale can be configured to create two or three synchronous copies of metadata and/or file data. When data is replicated, either copy of the data can be accessed for reads in an active-active mode. Spectrum Scale also allows the administrator to determine which copy to read from so read access remains local if you are replicating across sites. This benefits read-intensive workloads and keeps certain traffic off the WAN. It should be noted that when both copies are located in the same data center reads come from both copies, thereby doubling your read bandwidth.

REMEMBER

Spectrum Scale introduces new ways to manage data, such as file-based replication in an active-active configuration. This provides the flexibility to keep data safe and the ability to use it at the same time, across various types of hardware.

Increase Data Security

Data needs to be protected and Spectrum Scale provides the tools to secure data. Government regulations, such as HIPAA and Sarbanes-Oxley in the United States, define requirements for controlling data access, encryption, and proper deletion of information.

Security begins with controlling data access. Spectrum Scale provides a RESTful API to access object storage using HTTPS, which is secure and encrypted. Authentication of users for accessing data using either file or object protocols can be centrally managed with either Lightweight Directory Access Protocol (LDAP) or Active Directory (AD). This makes it simpler for IT to deploy Spectrum Scale into an existing LDAP/AD environment.

Data in the storage system not currently being accessed by an application or client is referred to as data at rest. By encrypting this data, Spectrum Scale protects it from unauthorized reading or writing by another application or administrator. It also protects data from security breaches, unauthorized access, and from being exposed in the event data is lost, stolen, or improperly discarded.

Spectrum Scale follows the best practices recommended by the U.S. Federal Government National Institute of Standards and Technology (NIST SP 800-131A), HIPAA, EU, and Sarbanes-Oxley, as well as Federal Information Processing Standard (FIPS 140-2) to encrypt data before it is stored on disk, flash, or tape. Files are encrypted before they are stored on disk, while the encryption Master Keys are never written to file-system disks.

Deleting or destroying data such that it can't be recovered is also required by government regulations in many countries. Rather than merely marking storage as over-writable, but leaving the data intact, Spectrum Scale can securely delete data in its entirety. There is no "digital shredding" and no multi-pass overwriting; secure deletion is a cryptographic operation. Spectrum Scale's secure deletion is performed using cryptographic algorithms for speed, completeness, and reliability offering the ability to destroy arbitrarily large subsets of a file system.

Intelligent Storage Efficiency

Spectrum Scale RAID drives many efficiency benefits. It dramatically lowers disk rebuild times and improves performance by spreading all the storage work across all the available resources. It utilizes all storage resources more fully, ensuring that disks aren't standing idle.

Spectrum Scale's "intelligence" also increases storage efficiency. Remember that Spectrum Scale collects over 50 different data and file attributes, allowing it to more granularly and efficiently manage your data. Data compression provides an informative example. Spectrum Scale can leverage the power of policy and compression to use resources efficiently. You can use the data attributes collected by Spectrum Scale to set up a policy that's applied to only the data volumes you want compressed. It allows for compressing data where it's most beneficial and not where it's less effective.

All the data attributes collected by Spectrum Scale enable the performance of very effective and efficient Information Lifecycle Management (ILM). This can significantly lower costs by placing data on the appropriate tier of storage, including tape and even cloud.

It's been estimated that as much as 90 percent of all file data is never needed again after it's initially created. Unlike fine wine, most data grows less valuable as it ages. With petabytes of data and billions of files having accumulated over the years, it just isn't practical to ask each application group to "clean up its stuff." Additionally, government regulations often require certain data to be retained for many years. Solving this problem requires automation, and Spectrum Scale provides a set of ILM tools designed to dramatically increase the efficiency of storage. It can be tightly integrated with other products in the IBM software-defined portfolio, including Spectrum Protect and Spectrum Archive.

Spectrum Scale helps simplify storage administration through its policy-based automation framework and ILM feature set. These tools enable Spectrum Scale to automatically determine where to physically store user data, regardless of its placement in the logical directory structure. Defined storage pools, filesets, and policies provide the capability to match the cost of storage resources to the value of the data to determine where that data is stored.

You can create tiers of storage pools by grouping performance, locality, or reliability characteristics. For example, one pool could be flash arrays and another could be more economical disk storage.

When a file or object is created, Spectrum Scale needs to know where to put it. This is done by using placement policies. You can create placement policies based on anything Spectrum Scale knows about a file or object when it's created, including name and the user who's creating the file.

After a file has been created, Spectrum Scale knows much more about it. In addition to the attributes available when a file is created, it extracts additional information, including the size of the file, how long it's been since someone accessed the file, and whether it's been changed. Policies that operate on existing files are called *file management policies* and allow you to move, replicate, or delete files. File management policies can be used to move data from one pool to another without changing the file location in the directory structure. One popular use for file management policies doesn't involve moving data at all — it's used for reporting. The policy syntax is very powerful, allowing you to generate custom reports, for example, on the type of files using the most space.

REMEMBER

ILM tools need to have rich features, be automated, and be capable of operating on very large data sets to be useful when working with petabytes of stored data. The Spectrum Scale ILM toolset is designed for large scale environments managing multi-petabyte capacity and billions of objects.

Scale without Limits

Spectrum Scale enables you to start small and grow based on your organization's demand to near limitless capacity — build your own data ocean if you like. You can scale in many directions: seamlessly add additional storage capacity, increase storage performance, and extend to include another storage protocol, all without disrupting any application.

These benefits are largely due to Spectrum Scale's modular architecture. Elastic Storage Servers (ESS) function as building blocks; their CPU and storage resources are shared across the entire Spectrum Scale implementation. When another node is added, storage capacity, performance, and overall system resiliency increase linearly. Growth is seamless and transparent because storage capacity can be increased without requiring other solution components such as OpenStack Swift to rebalance the files or objects. With Spectrum Scale, you can scale in multiple dimensions: capacity, performance, and number of objects. Growth is nearly limitless, with a maximum file system size of 2^{63} (approximately 9 quintillion) files per file system and 16,384 nodes in a single cluster.

Globally Share and Collaborate

A single Spectrum Scale cluster can support a mix of workloads using the same or even different types of storage for two files in the same directory. Often data sets are generated at multiple sites, such as an organization investigating alternative production strategies for optimizing oil reservoir management. This requires large numbers of simulations using detailed geologic descriptions. In order to better predict reservoir properties and drive new simulations, the output from reservoir simulations is usually combined with seismic data sets, which are often stored on different systems and in different locations.

Another example can be seen in the automotive and manufacturing industries where product design and innovation often happen among distributed teams located across the globe. CAD (computer aided design) diagrams and other engineering data from multiple teams located in different geographic sites may need to collaborate and integrate various aspects of a design. The data is often fragmented into small storage islands using different technologies and managed by independent teams with their own local use policies. Cross-site projects are hindered by this inconsistent approach and teams struggle to expand their IT environments to meet the growing needs of collaboration.

With an enterprise-wide Spectrum Scale environment, you can consolidate data storage on a unified platform to achieve cost-effective and efficient collaboration with features such as a common file system and massive global namespace across computing platforms. Users can seamlessly access data from any node without having to first transfer the data from another location, thus streamlining the collaboration process. Improvement in cost effectiveness and energy efficiency also result because enterprises don't have to purchase additional disk space to store duplicate files. In addition, the data is available in a highly parallel manner, making access to massive amounts of data much faster.

Add New Workloads Like Hadoop Seamlessly

There is an industry trend toward taking a MapReduce approach to analyzing large amounts of data, sometimes referred to as Big Data. The expected cost benefit of using MapReduce is realized by achieving high I/O throughput on very inexpensive hardware. The approach is to leverage the internal I/O performance of a server, while being able to task a bunch of these servers to solve big problems. To run a workload of this type requires storage software that can support this hardware architecture and provide the right interface for MapReduce applications to find the right data. Both the Hadoop Distributed File System (HDFS) and IBM Spectrum Scale are designed to provide a storage platform for data supporting MapReduce workloads on large-scale standard hardware consisting of thousands of servers.

TECHNICAL STUFF

MapReduce is a programming and data organization model for processing large data sets in parallel on a distributed computational cluster.

HDFS and Spectrum Scale both provide the storage tools needed to run MapReduce workloads, but that is where the similarities end. HDFS is a basic storage solution for MapReduce, whereas Spectrum Scale is an enterprise storage solution that supports MapReduce. Some limitations of HDFS include

- Centralized master-slave architecture
- No file locking

- File data striped into uniformly sized blocks that are distributed across cluster servers
- Simple coherency with a write-once, read-many model that restricts what users can do with data

Spectrum Scale features include

- High-performance, shared-disk cluster architecture to speed data access
- Distributed metadata, space allocation, and lock management that improves data ingest
- File data blocks striped across multiple servers and storage resources for highest performance and reliability
- Ability to open, read, and append to any section of a file

Spectrum Scale includes a set of features that support MapReduce workloads called File Placement Optimizer (FPO), which is a distributed computing architecture where each server is self-sufficient and utilizes local storage. Compute tasks are divided between these independent systems and no single server waits on another. Spectrum Scale FPO provides higher availability through advanced clustering technologies, dynamic file system management, and advanced data replication techniques.

In addition, Spectrum Scale supports a whole range of enterprise data storage features, such as snapshots, backup, archiving, tiered storage, data caching, WAN data replication, and management policies. Spectrum Scale can be used by a wide range of applications running Hadoop MapReduce workloads and accessing other unstructured file data.

TECHNICAL STUFF

In laboratory tests, benchmarks demonstrate that a Spectrum Scale FPO-based system scales linearly so that a file system with 40 servers would have 12GB/sec throughput, and a system with 400 servers could achieve 120GB/sec throughput.

TIP

An advantage that Spectrum Scale provides over simpler file systems such as HDFS is that Spectrum Scale can handle both file and object workloads. It's becoming more and more common for raw data to be ingested as objects. But MapReduce-based analytics use a file system approach. Using Spectrum Scale, this difference can be easily overcome. You can ingest object data and make it available to MapReduce analytics by using the Spectrum Scale file interface.

This capability eliminates the need for separate storage systems to handle object ingest and then to analytics applications. You can enable real-time analytics and maintain data consistency, all while reducing data duplication and extra maintenance overhead. This means you can run MapReduce directly without having to wait for data transfers between systems.

Incorporate Cloud Storage

Cloud storage provides a scalable, virtualized infrastructure as a service, hiding the complexity of fine-grained resource management from the end-user. According to IDC (International Data Corporation), the amount of information in the world is set to grow 44-fold in the next decade, with much of that increase coming from the rise in cloud computing. As the volume of data continues to increase, more organizations are looking to leverage cloud storage as a means of expanding capacity while minimizing cost.

IBM Spectrum Scale is well suited to address cloud storage given its extreme scalability, reliability, and cost efficiency requirements. Spectrum Scale-based solutions can scale to thousands of servers while supporting hundreds of GB/sec of sequential throughput.

Not only does Spectrum Scale provide complete support for all the OpenStack storage protocols, but also it supports traditional applications that rely on POSIX file APIs and provides a rich set of management tools. A cloud storage built on a POSIX-based cluster file system makes it easy to support applications by providing a scalable infrastructure that doesn't require them to be modified.

Cloud storage is shared across different classes of applications, so standard file semantics in addition to object protocols are important. Support for new workloads including MapReduce eliminates separate point solutions for these workloads. For mixed workload environments that require access to large amounts of data in a cloud environment, Spectrum Scale is a good choice.

Notes

Notes

Introducing the new storage paradigm — object storage

The rapid growth of data, transactions, and digitally aware devices is straining today's information technology (IT) infrastructure and operations. At the same time, storage costs are increasing and user expectations and cost pressures are rising. This staggering growth of data, especially the unstructured variety, is driving the need for high-performance file and object storage solutions. *File & Object Storage For Dummies*, IBM Limited Edition, introduces file and object storage solutions all based on one powerful tool — IBM Spectrum Scale.

- ***Start smart* — add capacity, performance, and agility**
- ***Explore the benefits of Spectrum Scale* — redefine unified storage**
- ***Examine technologies* — OpenStack® Swift and Elastic Storage Server**
- ***Accelerate the insights and value derived from one of your most valuable assets* — information**

Open the book and find:

- **A primer on modern storage**
- **The lowdown on IBM Elastic Storage Server**
- **Ten advantages of IBM Spectrum Scale with Object Storage**

Go to Dummies.com®
for videos, step-by-step examples, how-to articles, or to shop!

FOR DUMMIES®
A Wiley Brand

Also available as an e-book

ISBN: 978-1-119-14974-3
Part #: DCM03004-USEN-02
Not for resale