

Enabling Hybrid Cloud Storage for IBM Spectrum Scale Using Transparent Cloud Tiering

Nikhil Khandelwal

Rob Basham

Sandeep R. Patil

Anbazhagan Mani

Amey Gokhale

Jinesh Shah

Kedar Karmarkar

Donald Mathisen

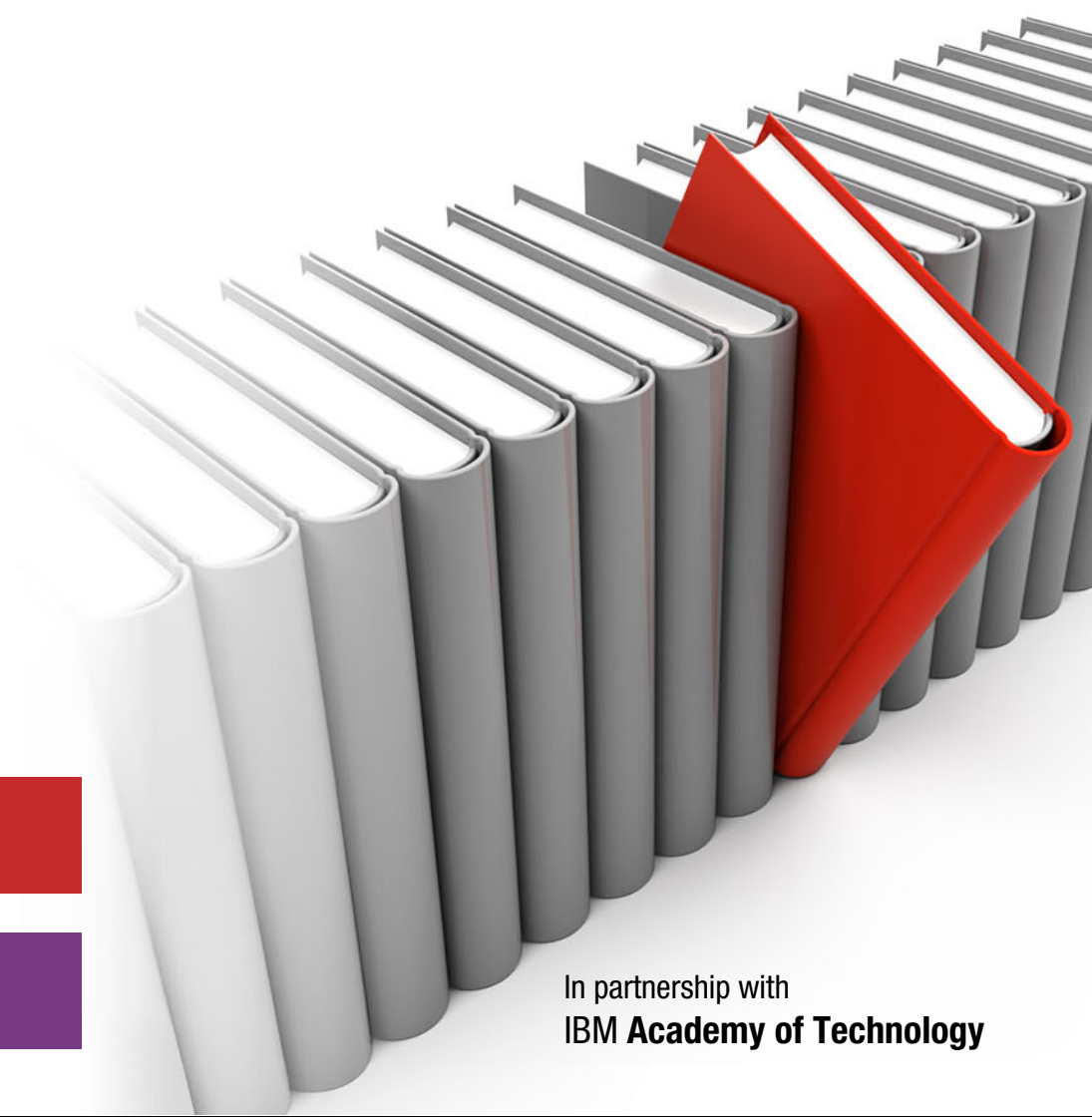
Larry Coyne

Arend Dittmer



Cloud

Storage



In partnership with
IBM Academy of Technology

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.


COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

AIX®	IBM Spectrum™	Redbooks®
Global Technology Services®	IBM Spectrum Accelerate™	Redpapers™
GPFS™	IBM Spectrum Archive™	Redbooks (logo)  ®
IBM®	IBM Spectrum Control™	Storwize®
IBM Cloud™	IBM Spectrum Scale™	System z®
IBM Elastic Storage™	POWER8®	XIV®

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other company, product, or service names may be trademarks or service marks of others.



Enabling Hybrid Cloud Storage for IBM Spectrum Scale Using Transparent Cloud Tiering

This IBM® Redbooks® publication provides information to help you with the sizing, configuration, and monitoring of hybrid cloud solutions using the transparent cloud tiering (TCT) functionality of IBM Spectrum™ Scale. IBM Spectrum Scale™ is a scalable data, file, and object management solution that provides a global namespace for large data sets and several enterprise features.

The IBM Spectrum Scale feature called *transparent cloud tiering* allows cloud object storage providers, such as IBM Cloud™ Object Storage, IBM Cloud, and Amazon S3, to be used as a storage tier for IBM Spectrum Scale. Transparent cloud tiering can help cut storage capital and operating costs by moving data that does not require local performance to an on-premise or off-premise cloud object storage provider.

Transparent cloud tiering reduces the complexity of cloud object storage by making data transfers transparent to the user or application. This capability can help you adapt to a hybrid cloud deployment model where active data remains directly accessible to your applications and inactive data is placed in the correct cloud (private or public) automatically through IBM Spectrum Scale policies.

This publication is intended for IT architects, IT administrators, storage administrators, and those wanting to learn more about sizing, configuration, and monitoring of hybrid cloud solutions using IBM Spectrum Scale and transparent cloud tiering.

Introduction

Transparent cloud tiering is a new Cloud Services feature available with version 4.2.1 of IBM Spectrum Scale that lets you integrate object storage as a storage tier, leveraging the full life-cycle management capabilities of the IBM Spectrum Scale Integrated Lifecycle Management (ILM) policy engine to control the movement of data. This publication covers features available with version 5.0.0 of IBM Spectrum Scale. Transparent cloud tiering works with many object storage providers, and is a near-line storage tier useful as a target for inactive data.

Here are the guidelines on how to read this paper, which is organized into the following sections:

- ▶ **Technology Overview:** Target Audience: Everybody.
- ▶ **IBM Spectrum Scale transparent cloud tiering use cases:** Scenarios where cloud tiering would typically be used. Target Audience: Pre-sales, solution architects.
- ▶ **Sizing Guidelines and Deployment Models:** Guidance on how to plan for the implementation of transparent cloud tiering and recommendations on how to deploy the service. Target Audience: Solution architects, administrators.
- ▶ **Configuration Best Practices:** Recommended settings and tunable parameters when using the transparent cloud tiering. Target Audience: Solution architects, administrators.
- ▶ **Setting Up cloud tiering:** How to set up policies for movement of data and routine administration of the tiering service covering such things as periodic clean-up of deleted and re-versioned files. Target Audience: Administrators.
- ▶ **Monitoring:** How to monitor the tiering service. Target Audience: Administrators.
- ▶ **Security:** Security considerations for the tiering service. Target Audience: Administrators.

Technology overview

This section provides an overview of IBM Spectrum Scale, cloud object storage, and how IBM Spectrum Scale provides a cloud storage tier by using the fully integrated transparent cloud tiering feature.

According to International Data Corporation (IDC), the total amount of digital information created and replicated surpassed 4.4 zettabytes (4,400 exabytes) in 2013. The size of the digital universe is more than doubling every two years and is expected to grow to almost 44 zettabytes in 2020.

Although individuals generate most of this data, IDC estimates that enterprises are responsible for 85 percent of the information in the digital universe at some point in its lifecycle. That means organizations take on the responsibility for designing, delivering, and maintaining information technology systems and data storage systems to meet the demand.

Both IBM Spectrum Scale and object storage, including IBM Cloud Object Storage, are designed to deal with this growth. The transparent cloud tiering function combines the two. IBM Spectrum Scale can take advantage of object storage such as IBM Cloud Object Storage as a storage tier using transparent cloud tiering service.

In a cloud deployment, a hybrid cloud model blends elements of both the private and the public cloud. In the simplest terms, the hybrid model is a private cloud that allows an organization to tap into a public cloud when and where it makes sense. IBM Spectrum Scale transparent cloud tiering supports tiering of data to public cloud object store like IBM Cloud and Amazon S3, allowing you to adopt a hybrid cloud strategy where data is placed in the correct cloud model.

IBM Spectrum Scale overview

IBM Spectrum Scale is a proven, scalable, high-performance file management solution. It provides world-class storage management with extreme scalability, flash accelerated performance, and automatic policy-based storage that has tiers of flash through disk to tape.

IBM Spectrum Scale reduces storage costs up to 90% while improving security and management efficiency in cloud, big data, and analytics environments.

IBM Spectrum Scale provides software-defined storage that can manage quintillions of files and yottabytes of data. It provides high performance, simultaneous access to all of your data in a single global namespace.

IBM Spectrum Scale software has been designed to work across multiple platforms, supporting a mixture of IBM AIX®, Linux, Linux for IBM System z®, and Microsoft Windows clients with support for flash and spinning disk storage. Integrated protocol access allows NFS, SMB, Object, HDFS, and native POSIX clients to seamlessly access a shared global namespace.

With transparent cloud tiering, Spectrum Scale now can seamlessly and transparently tier data to cloud object store like IBM Cloud, Amazon S3, and even to native object stores like IBM Cloud Object Storage.

Figure 1 shows an overview of IBM Spectrum Scale.

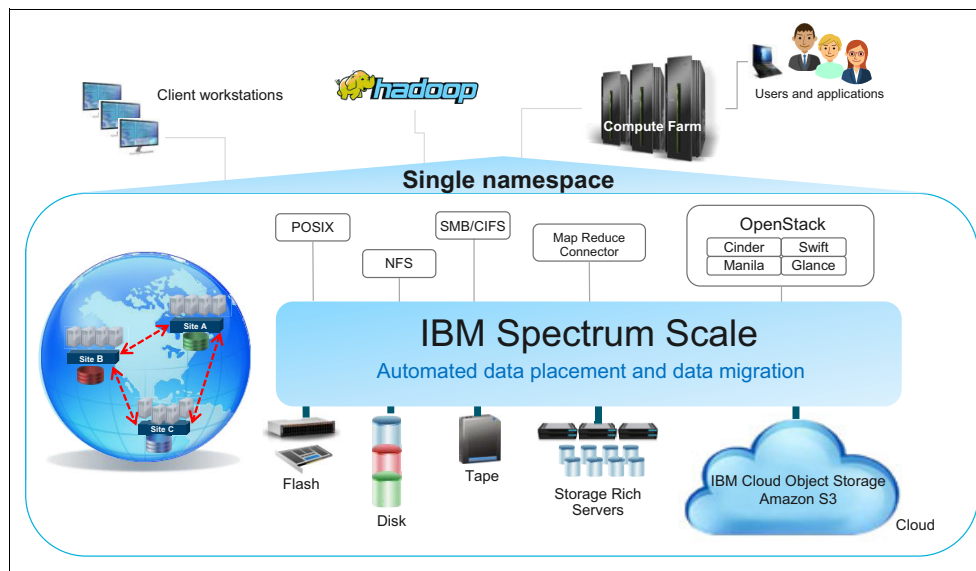


Figure 1 Spectrum Scale Overview

Note: IBM Spectrum Scale offers a no-cost try and buy IBM Spectrum Scale Trial VM. The Trial VM offers a fully preconfigured IBM Spectrum Scale instance in a virtual machine, based on IBM Spectrum Scale GA version. You can download it by clicking the [Start your free trial](#) button.

IBM Spectrum Scale deployment models

IBM Spectrum Scale as software-defined storage provides various deployment models to suit your workload needs. The following deployment models are described in this section:

1. IBM Elastic Storage™ Server (ESS): An IBM specialized hardware solution based on IBM Power with IBM Spectrum Scale declustered RAID technology. The ESS can be deployed as a spinning-disk or a high-density all-flash solution.

2. Shared SAN storage: The cluster is based on traditional shared SAN storage, and can include disk and flash storage systems.
3. Storage-rich servers: The cluster is based on storage rich servers that take advantage of the Spectrum Scale file placement optimizer feature.

IBM Elastic Storage Server

ESS is software-defined storage that combines IBM Spectrum Scale (formerly IBM GPFS™) software and associated clustered file systems with the CPU and I/O capability of the IBM POWER8® architecture.

Elastic Storage Server is a high performance, highly available, and scalable IBM Spectrum Scale building block, meeting today’s needs for high performance and business analytics applications. Replacing hardware-based disk RAID controllers, the IBM Spectrum Scale declustered RAID delivers superior data protection and rebuild times that take a fraction of the time that is needed by hardware-based RAID controllers.

ESS is ideal for large data sets due to its high-density storage options, and superior scalability. Combined with transparent cloud tiering, this deployment model is preferred when you want to minimize the frequency of recalls and data transferred to and from the cloud to save on network and cloud provider costs. IBM ESS comes in two models, GSxS and GLxS, where the key difference is the type and capacity of disk, making it suitable for different workloads. For more information, see the [specifications of IBM ESS](#).

Figure 2 depicts the deployment of the IBM Spectrum Scale cluster over IBM ESS.

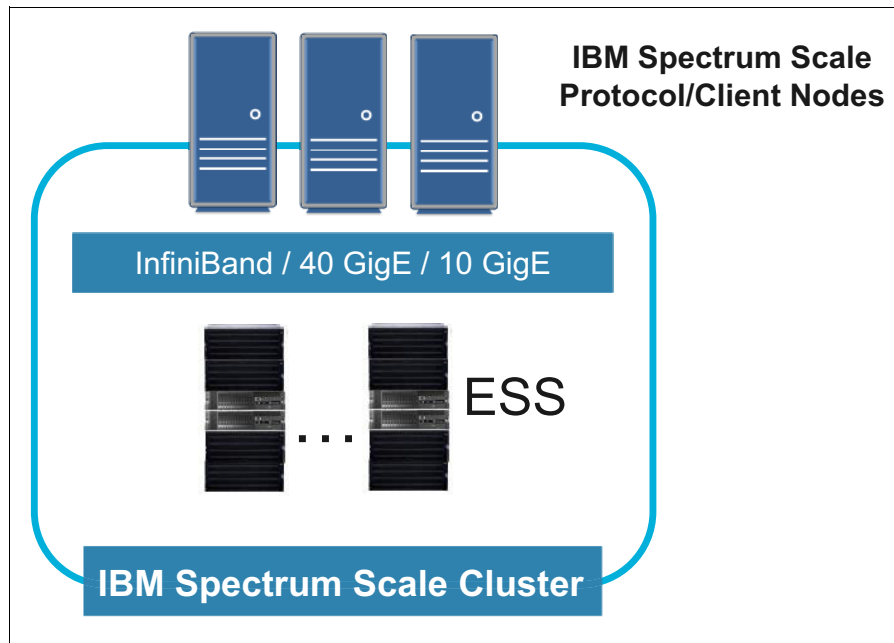


Figure 2 IBM Spectrum Scale with IBM ESS

All Flash ESS

The GSxS models of ESS with high-density SSDs can deliver high IOPS performance for big data, analytics, HPC and other workloads that require high IO rates with low latency. Using high capacity 4 or 15 TB SSDs, the solution provides a large amount of storage capacity in a compact footprint. IBM Spectrum Scale native RAID functionality provides industry-leading reliability, availability, and serviceability (RAS) capabilities and fault-tolerance.

In combination with transparent cloud tiering, the all-flash storage can be configured with a deep cloud storage tier to store cold data in a two-tier solution, or combined with other ESS models to provide a multi-tier solution. All-flash ESS is recommended when you have a large number of files, low application latency, and smaller total data sizes that need to be maintained on the primary storage.

Note: An IBM Spectrum Scale Failure Group is defined by the administrator and contains one or more Network Shared Disks (NSDs). Failure groups are used for replication of data and metadata for reliability.

Shared storage

In this deployment model, a few NSD servers have direct access to shared storage system through a SAN or shared connection, and IBM Spectrum Scale client nodes access the storage through the NSD servers, as shown in Figure 3. This model is typically used with backend storage providing RAID functionality.

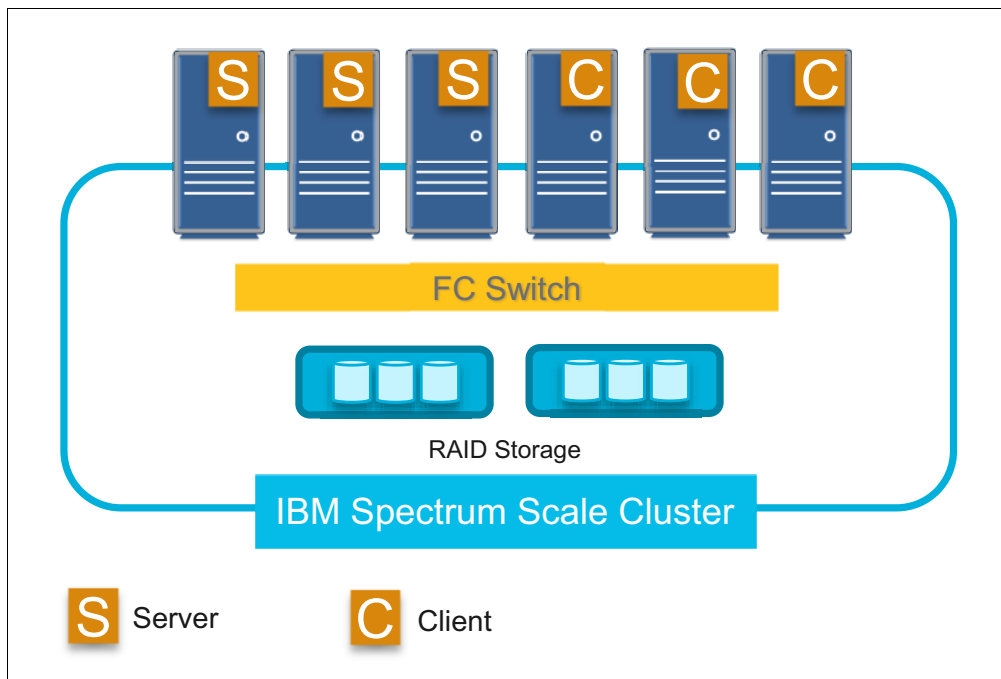


Figure 3 IBM Spectrum Scale with SAN-based shared storage

Storage rich servers

IBM Spectrum Scale File Placement Optimizer (FPO) capability is a share-nothing-cluster deployment model that uses networked storage rich servers, as shown in Figure 4. IBM Spectrum Scale FPO is suitable for workloads that are read dominant or benefit from data locality such as Hadoop. Here the data protection is achieved by using replication of data across the cluster.

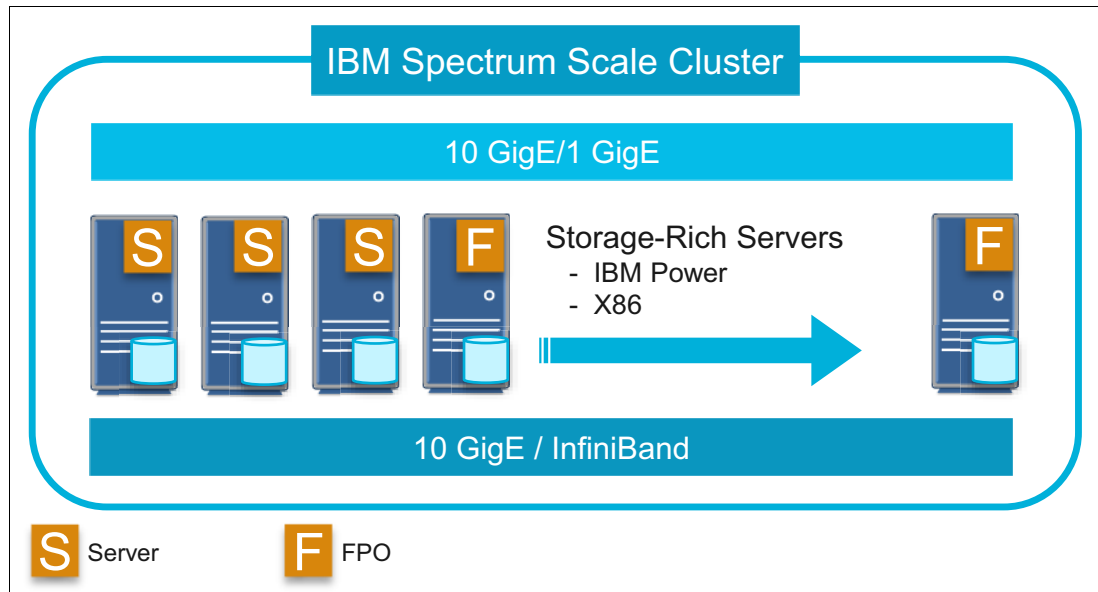


Figure 4 IBM Spectrum Scale with storage rich servers

Cloud storage overview

Object storage is the primary method for accessing data used in the cloud, and it is increasingly used for on-premises solutions. Object storage is growing for these reasons:

- ▶ It is designed for large-scale deployments in many ways (multi-site, multi-tenant, massive amounts of data).
- ▶ It is easy to use and yet meets the growing demands of enterprises across a broad spectrum of applications and workloads.
- ▶ It allows users to balance storage cost, location, and compliance control requirements across data sets and essential applications.

Object storage has a simple REST-based API. Hardware costs are low because it is typically built on commodity hardware. One of its key limitations is that most of the object storage services are only *eventually consistent*. In order to accommodate its massive scale of data and widespread geographic locations, object storage service might not immediately reflect all updates. The lag is typically small, but can be noticeable during network failures.

Object storage is typically offered as a service on the public cloud like IBM Cloud or Amazon S3, but is also available as on-premises systems like IBM Cloud Object Storage. For more information about object storage, see [the IBM Cloud Object Storage Overview](#).

Due to its characteristics, object storage is becoming the predominant storage for active archival of unstructured data both for public and private clouds.

Note: IBM Spectrum Scale also supports object storage as one of its protocols. One of the key differences between IBM Spectrum Scale Object and other object stores like IBM Cloud Object Storage is that IBM Spectrum Scale Object comes with a unified file and object access with Hadoop capabilities. It is also suitable for high performance-oriented or data lake use cases, whereas IBM Cloud Object Storage is more of a traditional cloud object store suitable for the Active Archive use case.

Cloud deployment models

Cloud object storage can be deployed both on and off premise.

Off-premise cloud

Off-premise public clouds, such as IBM Cloud Object Storage or Amazon S3, provide storage options with minimal extra equipment and datacenter costs. The ability to rapidly provision, expand, and reduce capacity, along with a pay-as-you-go model, make off-premise clouds a flexible option. When considering off-premise clouds, it is important to consider all the costs and the pricing models of the cloud solution.

Many cloud providers charge a monthly fee for the amount of data stored in the cloud. In addition, many clouds charge a fee for data that is transferred out of the cloud. This charge model makes public clouds ideal for data that is infrequently accessed. Storing data that is accessed frequently in the cloud can result in higher costs. It might also be necessary to have a high-speed dedicated network connection to the cloud provider.

On-premise cloud

On-premise cloud solutions, such as IBM Cloud Object Storage, provide flexible, easy-to-use storage with attractive pricing. On-premise storage is ideal for cases where you need to have the data under your own control or where you have higher recall rates from on-premises applications that would make off-premise cloud storage expensive.

For a cloud solution with multiple access points and IP addresses, a load balancer is required for high availability and throughput. The transparent cloud tiering service provides a simple load balancing service. Also, several commercial and open source load balancers are available. Contact your cloud solution provider for supported load balancers.

IBM Spectrum Scale transparent cloud tiering: What it is and how it works

Transparent cloud tiering is a fully integrated service of IBM Spectrum Scale that provides a native cloud storage tier. IBM Spectrum Scale administrators can use ILM policy semantics to move files into and out of cloud storage. When migrated to cloud storage, the file name remains in the file system global namespace and the data can be retrieved transparently because the files are accessed by standard applications. Data can be moved between local storage and object storage by policy, on access, or by using command line tools.

Figure 5 shows an overview of how the transparent cloud tiering service is provided.

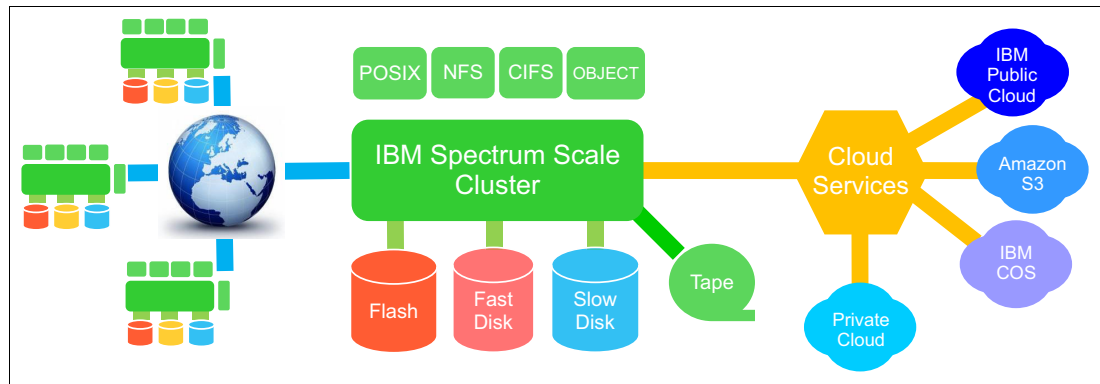


Figure 5 Capabilities with transparent cloud tiering

The transparent cloud tiering service runs on groups of IBM Spectrum Scale nodes and provides the following characteristics:

- ▶ **Scalability:** Scales the service up by defining for the cluster the number of node groups (each of which can have a number of nodes) that are needed to achieve the performance target.
- ▶ **Efficient:** Data moves directly from IBM Spectrum Scale to cloud storage without passing through an intermediate application or appliance. This configuration means less processor burden for the transfer and for tracking what has moved.
- ▶ **Transparent Recalls:** If a file is accessed that has been migrated to the cloud, it can be automatically recalled.
- ▶ **Highly available:** No single point of failure and automatic failover when a node goes down.
- ▶ **Low administrative overhead:** Policy-based movement means that after the system is configured, there is no need for constant intervention. Lifecycle management of deleted and reversioned files can also be automated.
- ▶ **Secure:** All data at rest and on the wire is encrypted. Both a local key manager and IBM Security Key Lifecycle Manager (ISKLM) key management support are available.
- ▶ **Data Integrity:** All data transfers are checked by using object storage integrity checking techniques.

Note: Protocol nodes are also referred to as *Cluster Export Services (CES)* nodes in IBM Spectrum Scale documentation.

Transparent cloud tiering major components and terminology

This section describes the IBM Spectrum Scale transparent cloud tiering components. A typical multi-node configuration is shown in Figure 6 with terminology referenced throughout the document.

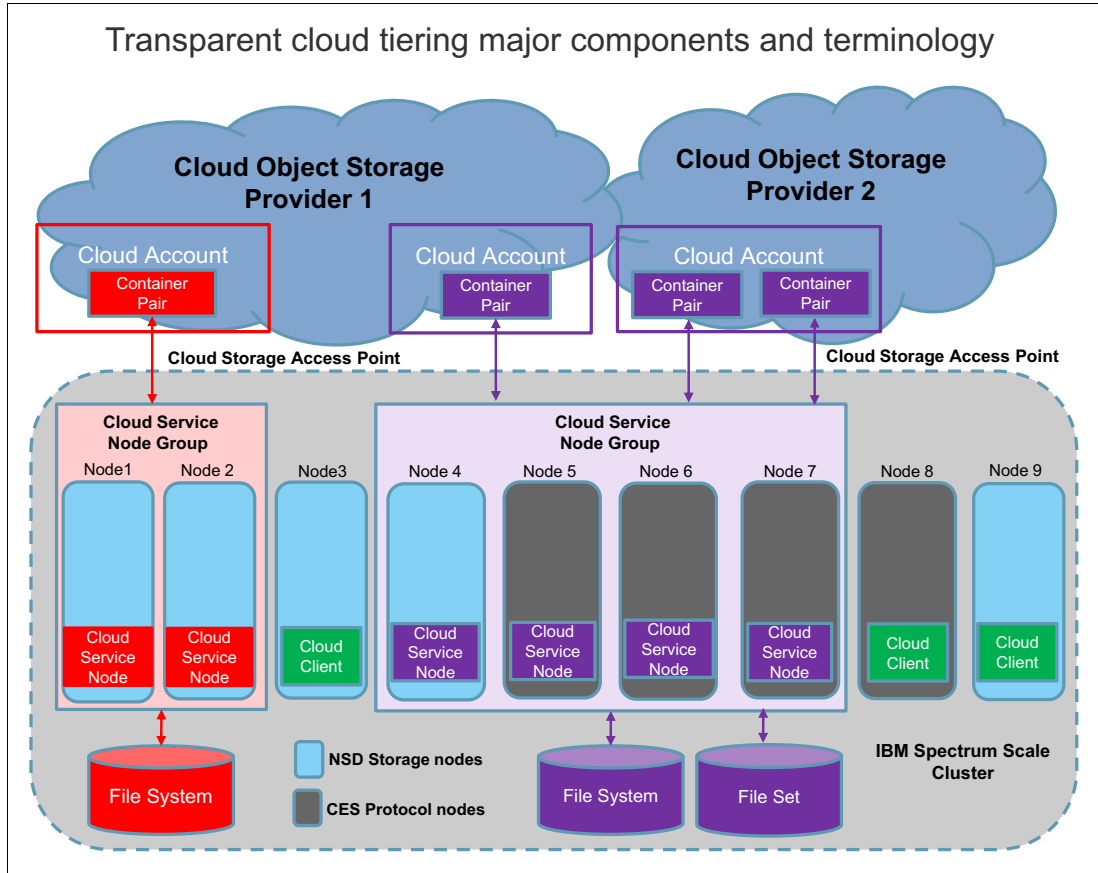


Figure 6 IBM Spectrum Scale cluster with Cloud Service nodes

Note: The transparent cloud tiering service runs on one or more Cloud Service Node Groups that consists of Cloud Service Nodes. For resiliency and availability reasons, a group typically contains more than one node. In Figure 6, the file systems, the Cloud Service Node Groups, and the Cloud Accounts, and associated transparent cloud tiering containers are all colored to show their groupings (connected by the arrows).

The objects in Figure 6 have these definitions:

Cloud Service Node Group

The transparent cloud tiering service runs on the Cloud Service Node Group. The node group allows for high-availability in failure scenarios and balancing the workload across all nodes in the node group, which in turn allows for faster data transfers. A node group can only be associated with one IBM Spectrum Scale file system, one Cloud Account, and one data container on the Cloud Account.

Cloud Service Node	A Cloud Service Node is where the actual work is done. Requests for transfer go to these nodes and they perform the transfer. A Cloud Service Node can only be defined on an IBM Spectrum Scale Protocol Node.
Cloud Storage Access Point	One or more URLs or IP addresses used to access a cloud object storage provider.
Cloud Account	Object Storage is multi-tenant but transparent cloud tiering can only communicate with the tenant on the object storage that is represented by the Cloud Account.
Cloud Client	The remainder of the nodes in the cluster can act as cloud clients (provided the operating system is supported). This light-weight client can be used to send cloud tiering requests to the transparent cloud tiering service. Each Cloud Service Node comes with a built-in Cloud Client. A cluster can have as many clients as needed. Remotely mounted clients are also supported for transparent recall operations.
Migration	Movement of file data from the IBM Spectrum Scale cluster to the cloud object storage.
Recall	Movement of file data from cloud object storage to the IBM Spectrum Scale cluster.

IBM Spectrum Scale transparent cloud tiering use cases

This section showcases use cases that can be broadly applied to a number of workloads:

- ▶ Active Archive
- ▶ Analytics

Before you go into the use cases, consider these key questions that can help determine whether a workload is right for transparent cloud tiering:

- ▶ Do you require long-term retention of data?
- ▶ Does your data consist of larger size files or objects, such as unstructured objects, images, movies, or documents?
- ▶ If you plan to deploy your own object storage for use with transparent cloud tiering, do you have access to sufficient object storage capacity?
- ▶ Do you have requirements for data security, availability, scalability, and cost-effectiveness?
- ▶ Do you have plans to procure the infrastructure required to support cloud storage, including network connectivity and load balancing services, and private or public cloud access?

Active Archive

The growing amount of unstructured data in the data centers, and the need to manage and store the data efficiently is what is driving the need for Active Archive. Active Archive is a method of tiered storage that gives a user access to data with an online or near-online capability.

With an Active Archive, applications can directly recall archived data based on metadata stored in IBM Spectrum Scale, maintaining a global namespace even for data that has been moved to the cloud. An Active Archive can facilitate more frequent and unpredictable data access when compared to a pure Archive solution.

An Active Archive solution using IBM Spectrum Scale transparent cloud tiering has many advantages over other archive products:

- ▶ Flexible deployment options (private cloud, hybrid, or public cloud).
- ▶ Cloud solutions such as IBM Cloud Object Storage are highly available across multiple sites with geo-dispersed erasure coding. Many public cloud providers also provide high availability with multiple site protection. Such storage pairs nicely with IBM Spectrum Scale synchronous replication, which allows for support across multiple sites to allow uninterrupted access to the cloud data tier even if an entire site goes down.
- ▶ Intermittent bulk migrations and recalls can result in faster data access times than a pure archive solution.
- ▶ Reduced capital equipment costs with off-premise cloud solutions.

Active Archive can benefit various industries as shown in Figure 7.

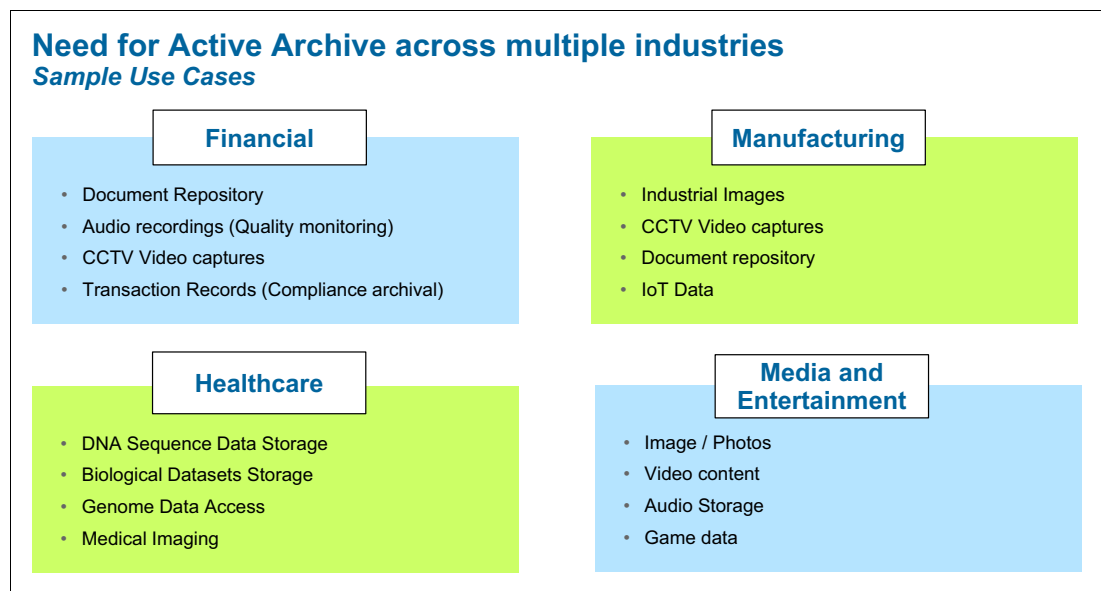


Figure 7 Active Archive use cases

As already outlined, IBM Spectrum Scale with transparent cloud tiering can be a good match to service these Active Archive use cases. See Figure 8 on page 12.

By contrast, IBM Spectrum Archive™ is a more standard archive service that is recommended for standard archive use cases:

- ▶ Store infrequently used data on tape
- ▶ Lowest total cost of ownership (TCO)
- ▶ Long-term retention
- ▶ Tape loading can cause longer access times

Note: Transparent cloud tiering cannot be used to manage the same file system that is being managed by IBM Spectrum Archive.

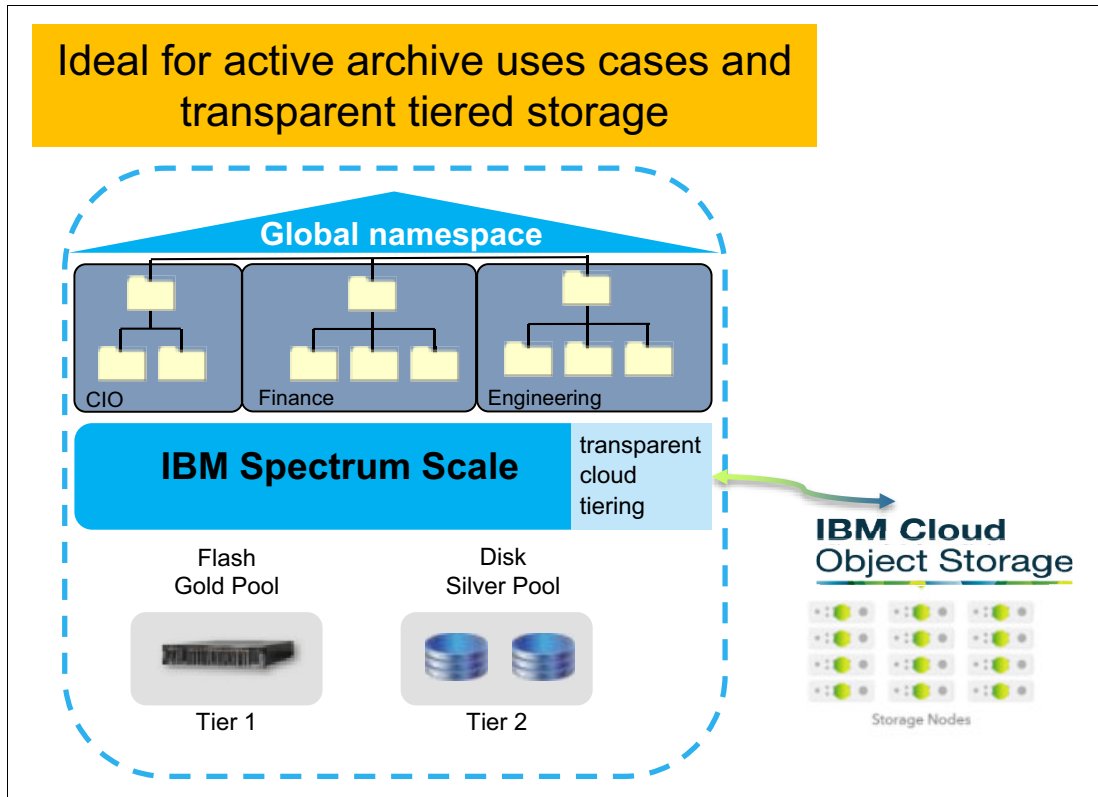


Figure 8 Transparent cloud tiering with IBM Cloud Object Storage

Using transparent cloud tiering along with IBM Cloud Object Storage achieves these goals:

- ▶ Storage Efficiency (primary storage can be freed up for high-performance applications)
- ▶ Cost reduction (infrequently accessed data can be moved to a lower-cost tier, resulting in cost savings)
- ▶ Data Protection (multi-site protection and access for archive data)
- ▶ On-Demand Access (ready for faster recalls and unpredictable data access patterns)
- ▶ Metadata search
- ▶ Limited user intervention (policy-based management)

Active Archive for digital media libraries

A digital media library is a huge repository of unstructured data, predominantly in form of images and videos. The data undergoes various processes throughout its lifecycle, from content capturing, to editing, to automated digital manipulation, and finally to publication and delivery of the content. This process creates a huge number of by-products and different versions of content in the form of images, videos, and project files that are prime candidates for archiving.

Such workloads demand high-performance storage and archival of the large amount of data to keep down costs. IBM Spectrum Scale with transparent cloud tiering is an ideal storage system for such workloads. IBM Spectrum Scale provides interfaces (including NFS, SMB, Object (S3/Swift), and POSIX) that can be used for data ingestion and publication in a global namespace.

IBM Spectrum Scale client nodes provide performance unmatched by other protocols, which is ideal for image and video processing.

With transparent cloud tiering, an administrator can easily move processed data, its input, and its by-products to an object store archive (such as IBM Cloud Object Storage) or to public cloud storage (such as IBM Cloud or Amazon S3). An administrator can use the power of IBM Spectrum Scale policies to automatically manage the data to be moved to the object storage provider. Active archiving frees up the primary Spectrum Scale storage so that it is always ready for incoming media.

In Figure 9, the objects (images and videos) are ingested by using IBM Spectrum Scale Object or file interfaces. Because IBM Spectrum Scale provides a single global namespace, the ingested media objects are immediately available to media processing compute farms running the IBM Spectrum Scale client. The final product, which resides on IBM Spectrum Scale, is then made available to the consumer through one of the Spectrum Scale interfaces like object or file (NFS, SMB).

After the project has been delivered, it can be moved through transparent cloud tiering to a cost efficient, durable Active Archive such as IBM Cloud Object Storage, IBM Cloud, or Amazon S3. This solution can be deployed at a single site as shown in this diagram, or across multiple sites using Spectrum Scale synchronous mirroring and IBM Cloud Object Storage erasure coding. This configuration ensures that uninterrupted access to data continues during a total site failure.

Figure 9 shows the Active Archive *media* use case.

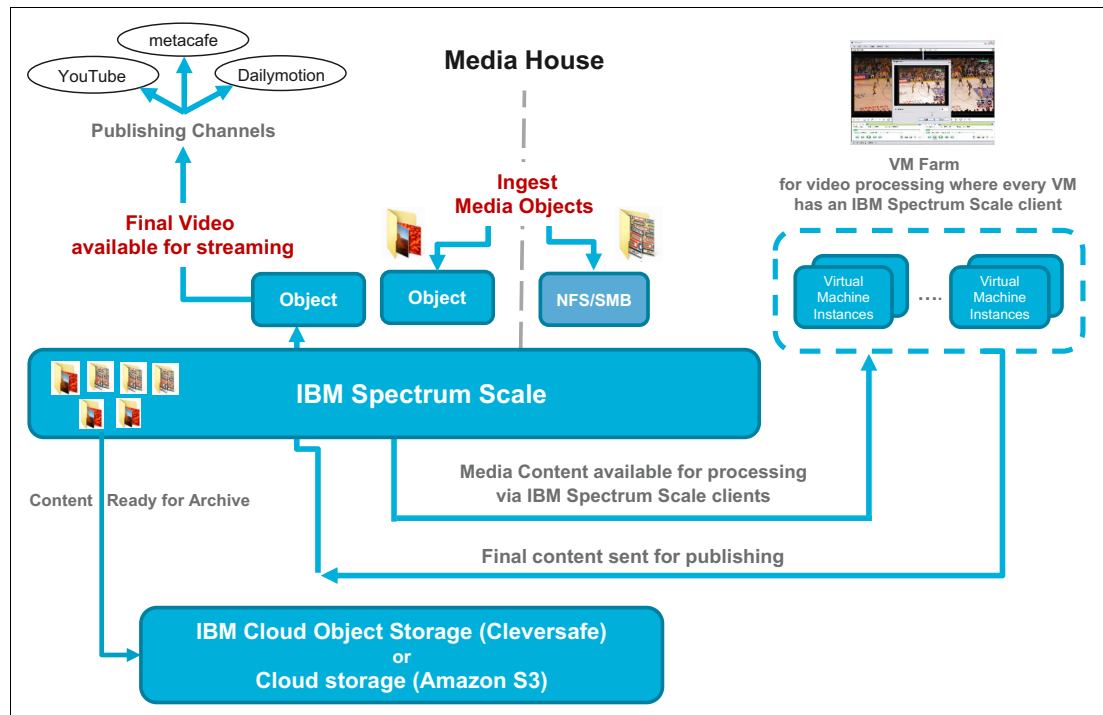


Figure 9 Active Archive for Media

Analytics

With features such as high-speed clustered data access, HDFS transparency, and native POSIX access, IBM Spectrum Scale is an ideal storage platform for analytics workloads. It can deliver results fast with in-place data analytics and can provide a single data repository for content creation, as well as servicing analytics applications such as Hadoop and Spark.

Using IBM Spectrum Scale, you can create a custom architecture to match your analytics workload. You can use storage rich servers, ESS, and flash mixing storage technologies as needed. Transparent cloud tiering can be used to extend these capabilities. In this example, IBM ESS GSxS is the primary storage building block, providing low-latency high-density flash storage for analytics. In addition, IBM Cloud Object Storage provides a repository for cold data, which transparent cloud tiering can recall if it is required for further analysis at any time.

The solution, as shown in Figure 10, scales to multiple petabytes of data, with the active data on high-speed flash. This high-speed, low-latency flash solution provides an ideal analytics platform to deliver fast results. Inactive data is automatically moved to object storage, which frees up space on the high-performance flash and delivers significant cost savings. The tiering is handled by IBM Spectrum Scale policies, so the administrative effort needed to manage the data placement is low.

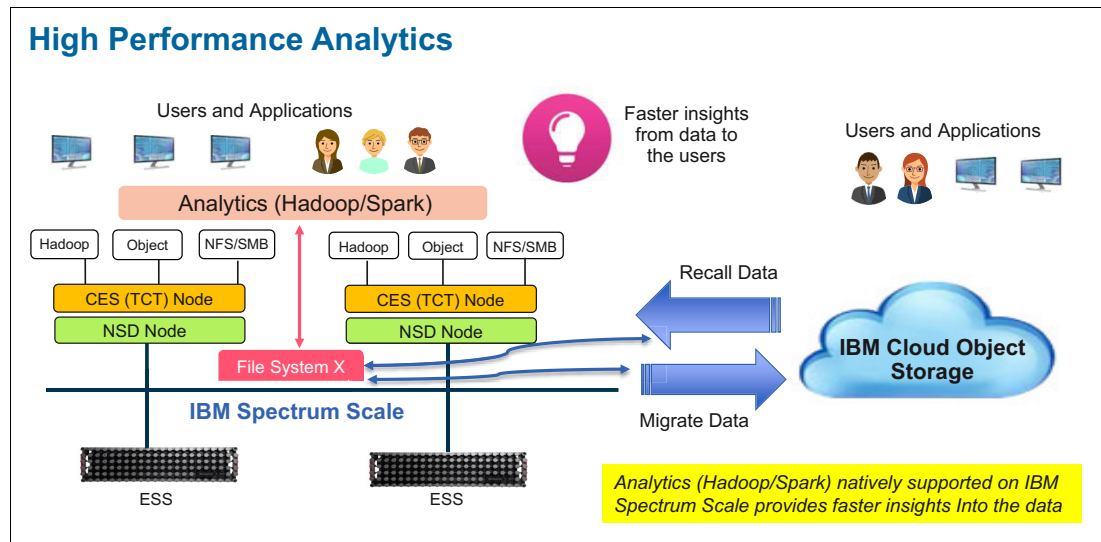


Figure 10 Analytics use case with transparent cloud tiering

Unsupported use cases of transparent cloud tiering

Transparent cloud tiering is not recommended for a certain set of use cases:

- ▶ Using transparent cloud tiering to migrate/recall hot (active) data.

The cloud storage tier should be used as a repository for infrequently accessed data. Migrating frequently accessed data causes multiple recalls, which can result in unacceptable latency and unwanted cloud provider data recall access costs.

- ▶ Using transparent cloud tiering as a backup mechanism.

Transparent cloud tiering is designed to place data in less expensive storage. It is not a replacement for a backup solution. With proper planning, transparent cloud tiering can be used to restore previous versions of a file or deleted files from a cloud tier. See "Restoring files from the cloud tier" on page 22 for additional information.

- ▶ Using transparent cloud tiering for disaster recovery scenarios.
Metadata on the IBM Spectrum Scale file system is required for transparent cloud tiering. Simply placing the data in the cloud is not a complete DR solution.
- ▶ Using transparent cloud tiering when the average file size is smaller than the recommended minimum average file size, which is currently 1 megabyte.
When a file is migrated, some metadata remains on the IBM Spectrum Scale file system. Space savings for small files are not as great with transparent cloud tiering. In addition, the additional processor cycles required to create and move small objects can result in lower throughput.

Sizing Guidelines and Deployment models

This section is designed for architects and administrators to help with sizing and deploying a Spectrum Scale solution with transparent cloud tiering. It outlines what can affect the load of a cloud tiering solution and how it can affect hardware requirements.

On any transparent cloud tiering deployment, it is important to consider these items:

- ▶ The number of cloud service nodes and node hardware
- ▶ The network connection between the IBM Spectrum Scale cluster and the cloud storage provider
- ▶ The active storage and metadata requirements for the IBM Spectrum Scale cluster
- ▶ The capabilities of the cloud object storage that is used in the deployment

Things to consider

The following pieces of information can help you determine the hardware and software needed to successfully run transparent cloud tiering.

Amount of data to migrate

Consider the amount of data that will be moved to and from the cloud. Analyze the existing size of the file system or data set that you are considering when tiering to the cloud, and how the data is used.

Number of files in file system or file set

Examine the size and number of files that are in the file system. The number of files affects how long it will take to execute a policy to migrate data to the cloud. Also, consider future plans for file system growth.

File size distribution

Consider the size of the files that will be migrated to the cloud. After a file has been migrated, a small amount of metadata will remain on the file system. The minimum metadata size is the size of the file system inode (4096 bytes by default). As a result, small files experience less capacity savings than larger files. In addition, larger files make better use of network bandwidth, although very large files can also lead to excessive recall times.

Recall frequency

Consider how often data that is migrated to and from the cloud provider is likely to be accessed. Most public cloud providers charge for data as it is copied from the cloud. Data that is migrated to cloud storage should be accessed infrequently. For existing IBM Spectrum Scale file systems, the policy engine can be used to generate lists of files that have not been accessed recently. For policy examples, see the [IBM Spectrum Scale wiki on the IBM developerWorks site](#).

Application tolerance for latency

If an application accesses a file that has been migrated to the cloud, it can be recalled transparently. However, because the file is located in the cloud, additional latency occurs as a result of this recall. Larger files and slower network speeds result in higher recall latency because the transfer of files from the cloud storage to the IBM Spectrum Scale cluster takes longer to complete.

Transparent cloud tiering checks the integrity of each file as it is recalled. The integrity check requires the entire file, so the tiering service does not allow an application to access a file until the file has been recalled in its entirety. Table 1 provides a sample of latency versus file size for a reasonably fast cloud service configuration.

Note: This example is intended to get you thinking about latency. Your actual performance might be better or not as good as the numbers listed here, depending on your specific environment.

Table 1 makes the following assumptions:

- ▶ Overall recall performance is file size dependent, with the best performance occurring at larger file sizes. To keep the example simple, a fixed number of 700 MBps is used.
- ▶ For this example, assume that 32 files are being recalled on a node, dividing the overall throughput between all 32 parallel transfers.

Note: The presence of intermittent heavy recall workloads where recall requests get queued up can lead to longer recall times.

Table 1 is an example of how latency increases directly and linearly with file size.

Table 1 Latency related to file size

File Size	Latency
2 MB	0.1 seconds
20 MB	1 second
200 MB	10 seconds
2 GB	100 seconds
20 GB	17 minutes
200 GB	170 minutes

Encryption/key manager

For environments that encrypt data that is sent to the cloud, transparent cloud tiering can use either a local key or an external key manager, such as IBM Security Key Lifecycle Manager (ISKLM). If you use a local key manager, ensure that the key is backed up properly to allow access to encrypted data during a node failure.

Resource requirements and considerations

This section describes the hardware and other infrastructure that are required for transparent cloud tiering.

Cloud service nodes

Cloud services run on CES nodes, and must meet the minimum requirements for CES protocol nodes on IBM Spectrum Scale. See the IBM Spectrum Scale FAQ for current CES protocol node requirements and other hardware restrictions for cloud tiering. IBM Spectrum Scale 4.2.2.4 supports up to 4 file systems, each managed by 1 - 4 cloud service nodes.

For a redundant solution with high availability, a minimum of two cloud service nodes per file system are required. For on-premises clouds, or cloud service providers with high-bandwidth networks such as multiple 10 GbE networks, four cloud service nodes might improve throughput from a file system to the cloud provider.

If the cloud service node is used for only cloud services, the minimum memory requirements of a protocol (CES) node are sufficient. However, if the node is used for other protocols (for example, serving NFS or SMB exports) or for running other applications, ensure that the system has sufficient CPU and memory capacity for a node servicing multiple protocols.

In most cases, the cloud service nodes start scans of the file system to determine which files to migrate to the cloud. As a file system contains more files, this scan becomes more I/O intensive due to the increased metadata, and might have to be run less frequently. If implementing cloud tiering on a file system with hundreds of millions or billions of files, extra memory devoted to the scan might improve scan times.

Transparent cloud tiering attempts to use as much bandwidth as possible when transferring files to and from the cloud. As a result, it might impact other operations that use the same network adapter. Avoid sharing a network that IBM Spectrum Scale is using for internal communication with cloud tiering. It is also best to avoid sharing the network adapter used by protocol services running on the cloud service node with cloud tiering.

Network

Network requirements and bandwidth vary significantly for onsite versus off-site cloud providers. Data volumes and application performance requirements also play a major role when sizing the network used for transparent cloud tiering.

Ensure that the network bandwidth is capable of handling the amount of data that will be migrated to the cloud. You can calculate the projected network requirements by examining the data that is generated and changed in your environment over time. Using this approximation, divide the amount of data by the expected network bandwidth to ensure that the data can be migrated in the expected time.

For example, if your applications generate 10 TB of data to transfer to the cloud per day and the data link between the IBM Spectrum Scale cluster and your cloud provider is 1 Gbps (approximately 100 MBps), it will take approximately $10,000,000 \text{ MB} / 100 \text{ MBps} = 100,000$ seconds to migrate the data. With only 86,400 seconds in a day, it will take more time to migrate the data than is available, so a faster network is required.

Recall latency times are equally important to consider, especially for large files. For example, if you recall a 60 GB file over the same 1 Gb network link, it will take at least $60,000 / 100 \text{ MBps} = 600$ seconds to recall the file. This time can increase if multiple files are being read at the same time, or if the network has other traffic. In many cases, long recall times are acceptable because they occur infrequently. Always consider the needs of your applications and users.

When working with an offsite cloud, work with your cloud provider to determine the bandwidth that can be obtained to the cloud. In most cases, this amount depends on a WAN connection to an external network. Some providers, such as IBM Cloud and Amazon, might offer dedicated network links to provide more predictable, higher speed access.

Onsite cloud storage systems, such as IBM Cloud Object Storage, can have multiple redundant access points, each with a different IP address. Because transparent cloud tiering can currently only be configured with a single IP address, a load balancer is required for redundancy that can improve system performance by using multiple object endpoints. Several software and hardware load balancers are available. Contact your cloud storage provider for a list of recommended options.

IBM Spectrum Scale backend storage

Transparent cloud tiering works on a standard IBM Spectrum Scale file system so it can be used with any backend storage. Cloud tiering uses IBM Spectrum Scale policy scans to scan file system metadata and determine what files might be candidates to move to cloud storage. As the number of files in the file system grows, the size of the metadata increases and the time that is required to run a policy scan increases.

High-performing metadata storage can help reduce scan times and the impact of scans on other system operations. With larger deployments of this service, using flash storage for metadata can greatly reduce the time and impact of these scans.

Cloud storage

Transparent cloud tiering supports the Amazon S3 and OpenStack Swift protocols for object storage. It has been tested with IBM Cloud Object Storage and IBM Spectrum Scale Object, as well as with IBM Cloud Object and Amazon S3 public clouds. Depending on the cloud solution, it might be necessary to work with your solution provider to ensure that the solution offers the capacity and throughput required by the file workload, particularly for on-premises or dedicated cloud solutions. For on-premises cloud deployments, a load balancer might be required if there are multiple endpoints to meet redundancy and performance requirements.

When connecting to public clouds, generally one Cloud Storage Access Point connection is sufficient because public storage always comes with built-in load balancing. For private on-premise object storage, it is advisable to have more than one Cloud Storage Access Point connection which will allow the transparent cloud tiering service to provide basic load balancing. An alternative would be to get an on-premise load balancer though in our experience the cost vs. the benefit trade-off doesn't pay off and the built-in load balancer will be the best option.

Configuration guidelines

This section provides the administrator with guidelines for deploying the cloud tiering solution. It includes recommended tunable parameters in different components of IBM Spectrum Scale and transparent cloud tiering.

File system guidelines

Cloud tiering uses native IBM Spectrum Scale APIs to read and write data from the file system. Therefore, it is important to ensure that the file system read/write performance is good. Setting the following attributes with proper values is key in achieving optimal performance for tiering.

All of these attributes can be set by using the `mmchconfig` IBM Spectrum Scale command. The following settings can improve the performance of cloud service nodes:

Note: If the cloud service node is also being used for other workloads, such as protocol exports or application I/O, then the configuration needs to take into account those workloads as well.

pagepool This setting is the IBM Spectrum Scale memory that is used for caching file data and file system metadata. Set an appropriate value based on the IO pattern (Random, Sequential, Random Direct, NSD servers).

Tiering many smaller files (Less than 1 MB) to cloud storage results in a random IO pattern, whereas tiering larger files (1 MB or greater) to cloud storage results in a sequential IO pattern. For random IOs, this setting should be set to higher values depending upon the overall memory available on the cloud service node.

dmapiWorkerThreads

Cloud tiering accesses data from the IBM Spectrum Scale file system through portions of the data management API (DMAPI) layer. This setting defines the performance of this layer. The default value is 12. Set this attribute to a higher value, such as 60, to improve migrate and recall performance.

workerThreads This setting is used to tune the file system performance in environments that are capable of high sequential or random read/write workloads, or small file activity. It automatically sets the values of 20 other parameters. As of Spectrum Scale 4.2.1, it is recommended to not change any other thread-related parameters if the value of this attribute is changed to a non-default value.

For details about these attributes, see the [IBM Spectrum Scale wiki](#).

CES preferred practices

Cloud services can be run on IBM Spectrum Scale CES nodes. For environments with high bandwidth requirements to the cloud provider, it might be beneficial to use a non-CES node, or to ensure that protocol services, such as NFS, SMB, or Object on the CES nodes, run on different network adapters than cloud services. If you are running cloud services with the protocol services on the same node, that node might require extra CPU and memory resources to avoid performance degradation.

Network preferred practices

The two network communication paths that are used by transparent cloud tiering are shown in Figure 11. Cloud service and cloud client nodes communicate through the cluster-wide network to coordinate the migration and recall of files from a cloud service provider.

Transparent cloud tiering traffic along this network is minimal, and places little extra demand on the cluster-wide IBM Spectrum Scale network. Cloud service nodes are required to communicate with the cloud provider to perform migrations and recalls of file data. The cloud network can generate significant traffic when migrations or recalls are in progress.

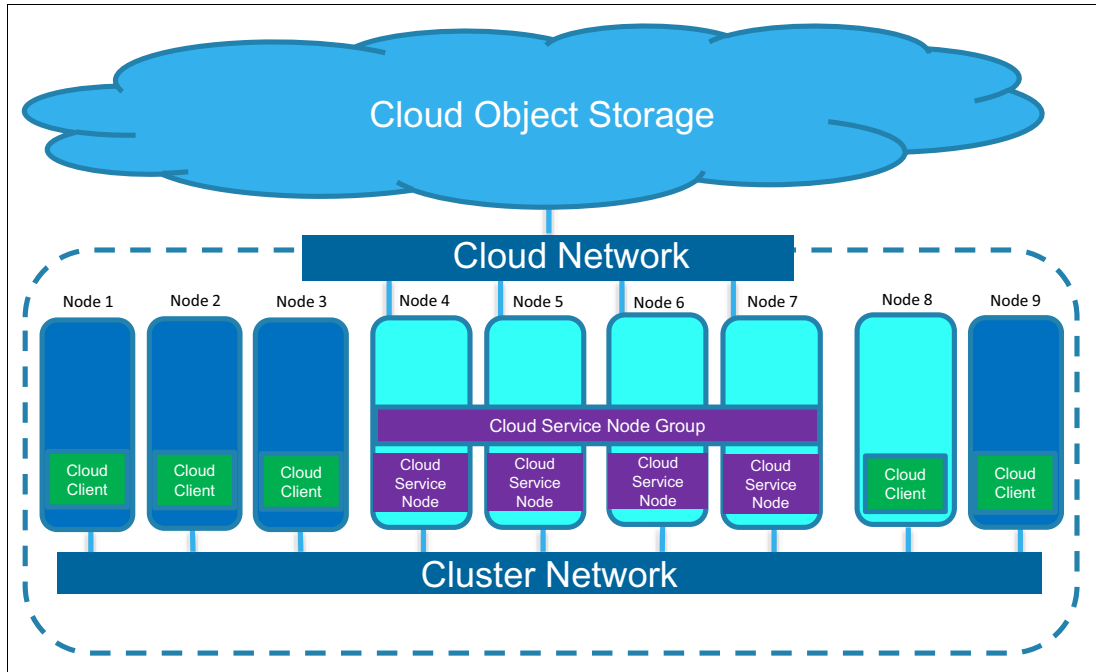


Figure 11 Transparent cloud tiering network

Although both the cloud network and cluster network can coexist on the same adapters, it is best to have a separate adapter running on a different subnet than cluster traffic to communicate to the cloud object storage. If protocol services, such as NFS, Object, or SMB, are running on the cloud service nodes, keep protocol communication and cloud communication on separate adapters as well to avoid performance degradation due to network contention.

On many networks, it is common to bond multiple network adapters on a single bonded adapter for higher performance and availability. Transparent cloud tiering supports such bonding. For availability, active/passive bonds are sufficient for connectivity during a network port or cable failure. For performance, it is common to use a bonding mode such as 802.3ad. If you use a network bond, pay particular attention to the hash mode on the network adapter and on the switch.

In most cases, Transparent cloud tiering communicates with a single endpoint. Therefore, hashing modes that rely on a MAC address or an IP address might not distribute network traffic across multiple adapters evenly. Refer to your network switch or Operating System vendors for details about hashing modes.

Policy execution guidelines

Policies can be used to migrate files from local storage to the external cloud storage pool. The following section describes guidelines when working with IBM Spectrum Scale cloud tiering policies. For sample policies, see “Setting up cloud tiering” on page 24. It also describes the two ways in which policies can be applied: Manual policy execution or threshold-based execution.

Policy parameters

To use multiple cloud service nodes efficiently, the following settings can be applied when running any policy using the `mmapplypolicy` command:

- ▶ `-m` is the number of threads that are used within each `mmapplypolicy` process during the policy execution phase. The default value is 24. It is recommended to set this value to a number equal to the number of available CPU cores to speed file system scan times.
- ▶ `-B` is the number of files that are passed for each invocation of the transparent cloud tiering migration command. The default value is 100. Based on the average file size, this number can be changed so that each node processes an equal and reasonably large number of files in a single iteration.

For example, when migrating 200 files through four cloud service nodes, the default value of `-B (100)` tries to distribute 100 files to each node. In this case, only two nodes are used to transfer files. Decreasing this value to 50 forces each node to receive 50 files, which equalizes the load across all of the nodes.

Running policies on multiple nodes

When running migration or recall policies, it is important to consider which nodes are used to execute the policies. By default, the policy is executed on the node where the `mmapplypolicy` command is run. More than one node can be passed to the `mmapplypolicy` command with the `-N` flag to improve scan performance. This section describes how to choose those nodes.

Data migration is performed by all of the nodes that participate in executing the policy. So if a cloud gateway node runs the policy, that node performs the migration of the file to the cloud provider. If a non-cloud gateway node participates in the migration, a file migration request is forwarded to a random cloud gateway node for processing. Therefore, it might optimize bandwidth and improve balance to just use the cloud gateway nodes in the node group that manages the file system to execute the policy.

Invoking mmapplypolicy

The `mmapplypolicy` command can be invoked to either run a policy that has been established for a file system using `mmchpolicy`, which is usually the case in threshold migrations, or can be run with a policy file to perform migrations. Using multiple nodes to run a policy can improve the speed of the policy run. For guidance on which nodes to use, see the “Running policies on multiple nodes” on page 21. When running a policy using more than one node, a global work directory is required in order for the nodes to coordinate work.

Typically, this can be a temporary directory on the file system being scanned, for example:

```
/gpfs/gpfs0/.mmapplypolicy
```

To invoke `mmapplypolicy` using cloud service nodes, a typical command would be:

```
mmapplypolicy gpfs0 -g /ibm/gpfs0/.mmapplypolicy -N TCTNodeclass
```

Where **TCTNodeclass** is the configured node class for the cloud service nodes, and the file system in this case is named **gpfs0**. This invocation would run the established policy, but a file containing an arbitrary policy can be passed by using the **-P** flag.

Threshold migration

If threshold migrations are being used, it is useful to automatically run **mmapplypolicy** when a pool crosses a threshold. To do so, a callback must be added that invokes **mmapplypolicy** during low space and out of space events. See the threshold migration section for more details about enabling these events. The callback for low space and out of space events should be configured to run **mmapplypolicy** on all TCT nodes. This configuration can be done by using the **callback** parameters. For example:

```
mmaddcallback migration --event lowDiskSpace,noDiskSpace -N TCTNodeclass --command /usr/lpp/mmfs/bin/mmapplypolicy --parms "%fsname -N TCTNodeclass -g /ibm/gpfs0/.mmapplypolicy --single-instance"
```

The invocation for this configuration contains the **TCTNodeclass** twice. The first command specifies which nodes the callback is run on. In this example, all of the nodes in the **TCTNodeclass** class are notified of the event to ensure that the policy is run even if only a single node is unavailable. The second command specifies which nodes will be used to run the policy itself. In this example, the nodes in the **TCTNodeclass** class are again used to run the policy. Also, note the **--single-instance** option, which ensures that only one instance of the migration runs at a time.

Restoring files from the cloud tier

While transparent cloud tiering is not a replacement for a backup solution, it can be used to restore previous versions of files or files that were accidentally deleted from a IBM Spectrum Scale file system.

Transparent cloud tiering maintains a Cloud Directory database, which tracks files in the IBM Spectrum Scale file system that have been migrated to the cloud tier. Transparent cloud tiering does not delete files in the cloud tier that have been deleted on the IBM Spectrum Scale file system or remove older versions of files stored in the cloud until a reconcile process is run. As a result, older versions of files can be restored from the cloud.

File versions available in the cloud tier can be listed with the following command:

```
mmcloudgateway files cloudList --path <path> --file-versions <filename>
```

If a file is lost, or an older version of a file is required that exists in the cloud tier, it can be recovered using the **mmcloudgateway files restore** command.

If a significant number of files or an entire directory tree are removed and need to be restored, it might be necessary to recreate the directory structures as well as restoring multiple files in parallel. For guidance and assistance with this, see “Bulk file restore” on page 28 for further guidance and examples.

IBM Spectrum Scale transparent cloud tiering guidelines

Transparent cloud tiering has several configurable options. In most cases, the default options work well. If needed, the following tunable parameters can be changed for transparent cloud tiering.

These parameters are configured by using the `mmcloudgateway config set` command. For more information about this command, see [IBM Knowledge Center](#):

migrate-threadpool-size

This parameter directly corresponds to the number of CPU cores on the cloud service node, which processes the incoming migrate requests. Generally, set this value equal to the number of CPU cores on a cloud service node. The default value is 32. Having a smaller value queues up the incoming data migration requests.

recall-threadpool-size

This parameter directly corresponds to the number of CPU cores on the cloud service node, which processes the incoming recall requests. Generally, set this value equal to the number of CPU cores on a cloud service node. The default value is 32. Having smaller value queues up the incoming data recall requests.

IBM Cloud Object Storage transparent cloud tiering guidelines

IBM Cloud Object Storage (COS) is an ideal private or managed cloud object solution for use with transparent cloud tiering. COS is an ideal target for inactive data. If you use COS as the cloud object provider with transparent cloud tiering, it has these considerations:

- ▶ Synchronize the time on the COS Accessors and the cloud service nodes by using Network Time Protocol (NTP) servers. If the Accessor and cloud service nodes are not in sync, credential errors might occur when authenticating with the accessor.
- ▶ Transparent cloud tiering creates a vault on COS using the COS default vault template. A specific vault template can be specified in the `mmcloudgateway account create` command.
- ▶ The COS user requires the **Vault Provisioner** role, which is assigned through the dsNet Manager UI.
- ▶ Make sure that an access pool is created and available on the COS system. It must have **Cloud Object Storage** as the API type.

The recommended vault settings for COS and TCT are shown in Table 2.

Table 2 Recommended vault setting for COS and TCT

Configuration	Recommended Values	Comment
Width	As per IBM Cloud Object Storage documented configuration for production	
Threshold	As per IBM Cloud Object Storage documented configuration for production	
Write Threshold	As per IBM Cloud Object Storage documented configuration for production	
Alert Level	As per IBM Cloud Object Storage documented configuration for production	
SecureSlice Technology	Disabled	Transparent cloud tiering performs the client-side encryption of all data before the data is stored on the object storage. Therefore, encryption is disabled on the vault.
SecureSlice Algorithm	Not applicable because SecureSlice is disabled	

Configuration	Recommended Values	Comment
Versioning	Disabled	Transparent cloud tiering has built-in versioning capability, so IBM Cloud Object Storage versioning can be disabled.
DeleteRestricted	Yes/No	Gateway does not attempt to delete the vaults, so this setting can be set to Yes or No.
Name Index	Disabled	Disabling this setting can result in improved Vault performance.
Recovery Listing	Enabled	When Name Index is disabled, this setting is enabled to be able to conduct a disaster recovery if needed.

For the latest information about COS with transparent cloud tiering, see [IBM Cloud Object Storage considerations](#).

Setting up cloud tiering

This section describes how to configure IBM Spectrum Scale with transparent cloud tiering.

Cloud tiering policies

The IBM Spectrum Scale policy engine is a key part of file management for transparent cloud tiering. Policies can be used for batch file migration and recall operations. A new policy feature introduced with transparent cloud tiering is a transparent recall policy that is applied for automatic data recall on file access.

Batch migrate policies

IBM Spectrum Scale offers a rich policy language to help identify files to migrate. Many policies have identify files based on parameters, such as last access time, file heat, fileset, file path, and file extension. For a full list of attributes available, see the policy section of [IBM Knowledge Center for IBM Spectrum Scale](#).

A policy is executed by using the `mmapplypolicy` command. There are three ways to start a policy scan:

- ▶ Based on a capacity threshold
- ▶ On a schedule, for example daily or weekly
- ▶ Manually by a system administrator by running the `mmapplypolicy` command

Transparent cloud tiering includes several sample policies in the `/opt/ibm/MCStore/samples` directory. These policies include both threshold and directory-based migration. These are only samples. You can create custom policies with whatever rules and attributes meet your needs.

Before running any migration policy for the first time or installing a new rule on a file system with `mmchpolicy`, run `mmapplypolicy` with the `-I` test flag. When the test flag is used, the scan phase of a policy is run, but no files are migrated. This process allows you to see how much data would have been migrated and a list of files that will be migrated.

Use this data to determine whether the policy is performing as expected, or if changes are required before applying the policy. Testing a policy can protect against accidental file migrations and recalls.

Threshold-based migrations

A threshold-based migration typically specifies a file system storage pool, a high-water mark, and a low-water mark, with an additional optional premigration threshold. If the storage pool usage exceeds the high-water mark, the policy attempts to migrate files until the pool usage is at the low-water mark. The optional premigrate threshold attempts to migrate more files to the cloud provider, and leaves a copy of the file in the local file system. Premigrates can be used to speed future migrations by transferring file data while still leaving the file in active storage.

When using a threshold policy, a migration is complete when the low water mark is reached. Because this amount is usually not all of the files that could be migrated, use a weight to determine which files go first. The weighting parameter (`weight()`) can be based on many file attributes, including file heat, file size (example: move largest files first) or last access time (example: move oldest files first). Example 1 is a sample threshold migration policy that is weighted on last access time (`ACCESS_TIME`).

Example 1 Sample threshold migration policy

```
/* Define a cloud pool, using the 'mmcloudgateway' command to perform migrations to */
/* the pool */
RULE EXTERNAL POOL 'cloudpool' EXEC '/usr/lpp/mmfs/bin/mmcloudgateway files' OPTS '-F'

/* Define a migrate rule, in this example we will migrate from the system pool */
RULE 'MigrateToCloud' MIGRATE FROM POOL 'system'
  /* Define the threshold to use, in this example we will migrate when the pool is */
  /* 85% full, and attempt to migrate files until the pool is 75% full */
  THRESHOLD (85, 75)

  /* Next, define a weight for the file to determine which files to migrate first */
  /* Here we will use the last access time for the file by subtracting it from the */
  /* current time, giving a higher weight to older files */
  WEIGHT(CURRENT_TIMESTAMP - ACCESS_TIME)

  /* Define that we want to move to the cloud pool */
  TO POOL ('cloudpool')
  /* We don't want to migrate files that are smaller than 8k in size, because there */
  /* are little to no space savings due to file overhead */
  WHERE (KB_ALLOCATED > 8)
  /* Do not migrate cloud services internal files, which reside in the */
  /* .mcstore directory */
  AND NOT(PATH_NAME LIKE '%/.mcstore/%')
```

The rule in Example 1 migrates all of the files in a file system, from the least recently accessed to the most recently accessed, until the system pool is less than the 75% threshold. Threshold-based migrations can be run manually, on a schedule (by using cron for example), or in response to an event such as low disk space. To configure a threshold policy to automatically run when a file system is low on disk space using a callback, complete the following steps:

1. Ensure that the `enableLowspaceEvents` configuration parameter is set to yes in `mm1sconfig`. If not, run the `mmchconfig enableLowspaceEvents=yes` command.

Note: The IBM Spectrum Scale cluster must be restarted for this change to take effect.

2. Create a migration callback. See the migration callback guidance in “Running policies on multiple nodes” on page 21.
3. Create a file that contains the threshold policy, the recall policies for the file system, and other policies required for the file system (such as a placement policy). Apply this policy to the file system by using the `mmchpolicy` command or the Spectrum Scale GUI.

Note: It is always recommended to test a new policy by using `mmapplypolicy -I test` before applying it to a file system.

Remember that when you apply a threshold policy to a full file system for the first time, the initial data movement might take a long time. For example, if a 100 TB file system is 95% full, and cloud tiering is introduced, applying an 80% threshold results in 15 TB of data movement. Based on network speeds and other factors, the data movement can take a significant amount of time, almost 2 days over a 1 Gbps link. In cases such as this, it might be best to gradually reduce the threshold over time to avoid the long initial migrations.

Ad hoc file migration

At any time, you can migrate a specific set of files or directories, such as an archive or a content repository, to a cloud provider by using a policy file and the `mmapplypolicy` command. Example 2 shows how a policy can be used to migrate PDF files in an archive directory.

Example 2 Migrating PDF files in an archive directory

```

/* Define a list of files to exclude from migration. The .mcstore directory should */
/* not be migrated */
define(
    exclude_list,
    (
        FALSE
        OR PATH_NAME LIKE '/gpfs/.mcstore/%'
    )
)

/* Define a cloud pool, using the 'mmcloudgateway' command to perform migrations to */
/* the pool */
RULE EXTERNAL POOL 'cloudpool' EXEC '/usr/lpp/mmfs/bin/mmcloudgateway files' OPTS '-F'
ESCAPE '% -'

/* Migrate files from the system pool to the cloud storage pool */
RULE 'archivemigrate' MIGRATE FROM POOL 'system' TO POOL 'cloudpool' WHERE
(
    (
        /* Migrate any .pdf files in the application/archive directory */
        PATH_NAME LIKE '/gpfs/application/archive/*.pdf%'
    )
    AND NOT (exclude_list)
)

```

A policy like this one can be manually run to trigger a bulk migration of these files.

Considerations when running backups

Transparent cloud tiering should not be used as a replacement for a backup strategy. Files that are migrated with transparent cloud tiering might be lost if accidentally deleted, or during a complete site failure. File system backups can protect from these types of events. Transparent cloud tiering can be run along with file system backups for data protection.

When running transparent cloud tiering with file system backups, a file must be backed up before it is migrated to cloud storage. Failure to back up the file first results in a file recall when the backup program is run. This can cause unnecessary recall traffic, and add significant time to the backup run.

To ensure that files are not migrated before being backed up, it is best to define a migration policy that explicitly excludes files that have been recently modified. The amount of time to wait to migrate a file should be greater than the frequency of file system backups. For example, if nightly backups are run, at a minimum, the migration policy should exclude files that have been modified within the past day.

This setting ensures that they can be backed up before being migrated. In most cases, it is best to wait longer to avoid recalls if a backup window is missed for some reason. If there is an issue running backups for a longer time, it might be necessary to disable a migration policy until the problem with backups has been resolved to avoid many recalls during the next successful backup.

Excluding recently modified files can be accomplished with a policy statement in the **WHERE** clause, such as:

```
WHERE (DAYS(CURRENT_TIMESTAMP) - DAYS(MODIFICATION_TIME)) > 5
```

The threshold migration policy that is given above can be extended with this rule, as shown in Example 3.

Example 3 Extending the use of the threshold migration policy

```
/* Define a cloud pool, using the 'mmcloudgateway' command to perform migrations to */
/* the pool */
RULE EXTERNAL POOL 'cloudpool' EXEC '/usr/lpp/mmfs/bin/mmcloudgateway files' OPTS '-F'

/* Define a migrate rule, in this example we will migrate from the system pool */
RULE 'MigrateToCloud' MIGRATE FROM POOL 'system'
  /* Define the threshold to use, in this example we will migrate when the pool is */
  /* 85% full, and attempt to migrate files until the pool is 75% full */
  THRESHOLD (85, 75)

  /* Next, define a weight for the file to determine which files to migrate first */
  /* Here we will use the last access time for the file by subtracting it from the */
  /* current time, giving a higher weight to older files */
  WEIGHT(CURRENT_TIMESTAMP - ACCESS_TIME)

  /* Define that we want to move to the cloud pool */
  TO POOL ('cloudpool')
  /* We don't want to migrate files that are smaller than 8k in size, because there */
  /* are little to no space savings due to file overhead */
  WHERE (KB_ALLOCATED > 8)
  /* Do not migrate cloud services internal files, which reside in the */
  /* .mcstore directory */
  AND NOT(PATH_NAME LIKE '%/.mcstore/%')
  /* Ignore files that have been modified in the past 5 days */
  AND (DAYS(CURRENT_TIMESTAMP) - DAYS(MODIFICATION_TIME)) > 5
```

Bulk file restore

While `mmcloudgateway files restore` is well suited for restoring individual files or file lists, its performance is limited because it only runs as a single execution thread. Moreover, the `restore` command requires that the corresponding directory structure in the IBM Spectrum Scale file system exists before the files in the respective directory are restored. The script below addresses both limitations, allowing for restoring entire file system trees from the cloud tier in parallel with a single command.

The `ptctrecall.bash` script is invoked with the number of parallel processes and the path of the directory that is to be recovered as arguments. The actual degree of parallelism is limited by the number of cores available on the machine where `ptctrecall.bash` is invoked. The script compares the list of files available in the file system with the list of files stored in the Cloud Directory to identify the files that need to be restored.

It then splits and formats the corresponding file list so that they can be passed to the `mmcloudgateway files restore` command. Directories that contain files are automatically created by the script enabling the re-creation of lost file system trees. Finally `ptctrecall.bash` launches the restore operations in parallel through GNU Parallel.

The `ptctrecall.bash` script is intended as a starting point for readers who want to be able to restore data from cloud storage tiers. See Example 4. It can be extended to leverage multiple cloud service nodes for faster recovery through remote execution, for example with the `ssh` command. Moreover, recovery with `ptctrecall.bash` does not preserve metadata associated with a directory.

The metadata of a directory could be captured through a `find -type d <path> | cpio` pipeline redirected to a file. That file can then be migrated to the cloud tier and retrieved at recovery time so that directories can be created with the correct metadata information.

Example 4 The `ptctrecall.bash` script

```
#!/bin/bash

# First arg is the number of processes, second arg is the directory to restore
numprocs=$1
dirpath=$(readlink -f $2)

mkdir /tmp/ptct.$$

# Find files stored in cloud storage that are not found in directory
find $dirpath -type f | sort -u > /tmp/filesindir.$$
/opt/ibm/MCStore/bin/mcstore dblist -p $dirpath -r | sort -u > /tmp/filesincloud.$$
comm -3 /tmp/filesincloud.$$ /tmp/filesindir.$$ > /tmp/filestorestore.$$

# (Re-)create directories that don't exist
/usr/bin/awk 'BEGIN {FS="/";OFS=FS} { $NF=""; print $0 }' /tmp/filestorestore.$$ > /tmp/dirs.$$
sort -u /tmp/dirs.$$ > /tmp/ptct.$$/newdirs
while read -r line
do
    if [ ! -d $line ]; then
        mkdir $line
    fi
done < /tmp/ptct.$$/newdirs
rm /tmp/ptct.$$/newdirs
```



```

# Create chunked file list to distribute restore workload
numlines=$(wc -l < /tmp/filestorestore.$$)
if [ $(( $numlines % $numprocs )) -ne 0 ]; then
    chunklines=$(( $numlines / $numprocs + 1 ))
else
    chunklines=$(( $numlines / $numprocs ))
fi
split -l $chunklines /tmp/filestorestore.$$ /tmp/ptct.$$/tcchunk

# Create mcstore file lists from chunk files
for i in $(ls /tmp/ptct.$$/tcchunk*)
do
    migrationchunk="/tmp/ptct.$$/mcstore.$(basename $i)"
    touch migrationchunk
    while read -r line
    do
        echo "filename=$line" >> $migrationchunk
        echo "target=$line" >> $migrationchunk
        echo "%%" >> $migrationchunk
    done < $i
    rm -f $i
done

# Launch parallel restore using gnu parallel - max one restore process per core
/usr/local/bin/parallel /opt/ibm/MCStore/bin/mcstore restore -l {} ::: /tmp/ptct.$$/*

# Clean up
rm -f /tmp/filesincloud.$$
rm -f /tmp/filestorestore.$$
rm -rf /tmp/ptct.$$

```

Interoperability with IBM Spectrum Scale features

For the latest information about interoperability of Cloud Services and other IBM Spectrum Scale features, see the following resources:

- ▶ [IBM Spectrum Scale FAQs](#)
- ▶ [Using transparent cloud tiering with other IBM Spectrum Scale features](#)

IBM Spectrum Scale native compression and encryption

IBM Spectrum Scale can natively compress and encrypt files that have been stored to a file system. Encryption and compression affect data at rest and is transparent to the user. Transparent cloud tiering migrates and recalls files that have been encrypted or compressed natively in IBM Spectrum Scale. However, it is worth noting that files are decrypted and decompressed before being migrated to a cloud storage provider.

In order to ensure at rest encryption of migrated files, the Cloud Service connection can be configured to encrypt data before sending it to cloud storage. Alternatively, some cloud storage providers, such as IBM Cloud Object Storage, can be configured to encrypt data at rest. If a file is recalled from the cloud provider, recalled data is reencrypted using the active IBM Spectrum Scale encryption policy at the time of the recall.

Reconcile

Transparent cloud tiering automatically removes reversioned or deleted files from cloud object storage. However, the process is not guaranteed to remove files in the event of system failures or unexpected power loss. The reconcile process will check for such missed files and absolutely guarantee that they will be deleted, which may be important if you have a regulatory requirement to guarantee deleted files are destroyed.

The reconcile process takes care of keeping the local file system and its cloud tiered data in sync by removing such files. The frequency of reconcile (weekly, bi-weekly, or monthly) should be set depending on how frequently files from the active file system are being removed. Details of the reconcile process can be found in the “Reconciling files between IBM Spectrum Scale file system and cloud storage tier” topic at [IBM Knowledge Center](#).

If reconcile is not run periodically, the tiering service might accumulate orphan objects (for which the corresponding local file from the IBM Spectrum Scale file system is removed) on the cloud storage. An orphan object is an object in cloud object storage that does not have a corresponding file in the active IBM Spectrum Scale file system.

Orphaned or previous versions of files can be restored from the cloud tier if they were accidentally deleted or modified. It can be useful to wait between running reconciles, or to retain files for several days with the `--days-retained` flag when running the reconcile process to preserve these files for some time. See the “Restoring files from the cloud tier” on page 22 for details on restoring previous versions of files.

Monitoring

This section is intended for administrators. It provides guidance about where performance metrics and events of cloud tiering service can be viewed on the IBM Spectrum Scale GUI, and how to get an end-to-end capacity view of IBM Spectrum Scale and IBM Cloud Object Storage by using IBM Spectrum Control™.

Service status

The status of the transparent cloud tiering service and, various entities such as cloud service, cloud account, cloud storage access point and container pair, on the cloud service node can be monitored by using the `mmcloudgateway service status` command to ensure that the process is running on the cloud service nodes. See the “Transparent cloud tiering status description” topic in [IBM Knowledge Center](#) for details of each status value.

To ensure no disruptions to file access, the cloud service must be running on at least one of the cloud services nodes at any time. From IBM Spectrum Scale release 5.0.0 onwards, cloud service is also monitored by IBM Spectrum Scale health monitoring service periodically. In case of any change in the health state of the service as well as the cloud service entities, appropriate events are raised and can be monitored through IBM Spectrum Scale GUI. Details of “Transparent Cloud Tiering events” can be found in [IBM Knowledge Center](#).

The transparent cloud tiering service can also be monitored by using the `mmhealth` command. This command provides a single view of multiple IBM Spectrum Scale processes to check the overall health of a node or set of nodes.

The **CLOUDGATEWAY** component in **mmhealth** (Example 5) shows the status of the transparent cloud tiering services.

Example 5 Sample mmhealth output

```
[root@tct-ces1 ~]# mmhealth node show
```

```
Node name:          tct-ces1
Node status:        HEALTHY
```

Component	Status	Reasons
GPFS	HEALTHY	-
NETWORK	HEALTHY	-
FILESYSTEM	HEALTHY	-
DISK	HEALTHY	-
CES	HEALTHY	-
CLOUDGATEWAY	HEALTHY	-
GUI	HEALTHY	-
PERFMON	HEALTHY	-

For detailed status of cloud services component, invoke **mmhealth node show cloudgateway**. It displays a detailed health view of all of the container pair sets, and any recent events raised.

For additional details, see the “Cloud services: Transparent cloud tiering and Cloud data sharing” topic in [IBM Knowledge Center](#). It covers monitoring cloud service, administering files managed by cloud service, recommendations for periodic transparent cloud tiering service maintenance tasks, and so on.

Cloud connectivity

Connectivity to a cloud storage provider can be verified from any cloud service node by running the command **mmcloudgateway account test**. This command verifies connectivity and credentials on the cloud provider. Verify this information if there are any network or object storage provider changes, to ensure continued connectivity to the cloud.

Performance Statistics

Transparent cloud tiering performance statistics are available through the IBM Spectrum Scale GUI, the **mmperfmon** command, and can be exported to external monitoring tools, such as Grafana. To view performance statistics of cloud tiering service, it must be enabled in the IBM Spectrum Scale GUI and monitoring tool. For more information, see these resources:

- ▶ [Enabling transparent cloud tiering performance monitoring metrics on the GUI](#)
- ▶ [Integrating transparent cloud tiering metrics with performance monitoring tool](#)

The metrics available for transparent cloud tiering include such items as migrate and recall times, data transferred, and many others. A full “List of performance metrics” is available in the performance monitoring section of the IBM Spectrum Scale topic in [IBM Knowledge Center](#).

Each metric has different aggregation levels. Cloud service node, cloud service node group, and file system can be used to give node level or cluster-wide views of various statistics.

During migrations, you can monitor these metrics in the UI for potential problems.

IBM Spectrum Control

IBM Spectrum Control can be used to monitor IBM Spectrum Scale with transparent cloud tiering and IBM Cloud Object Storage deployments. as shown in Figure 12. IBM Spectrum Control can be a single point of control in hybrid-cloud deployments using these two products.

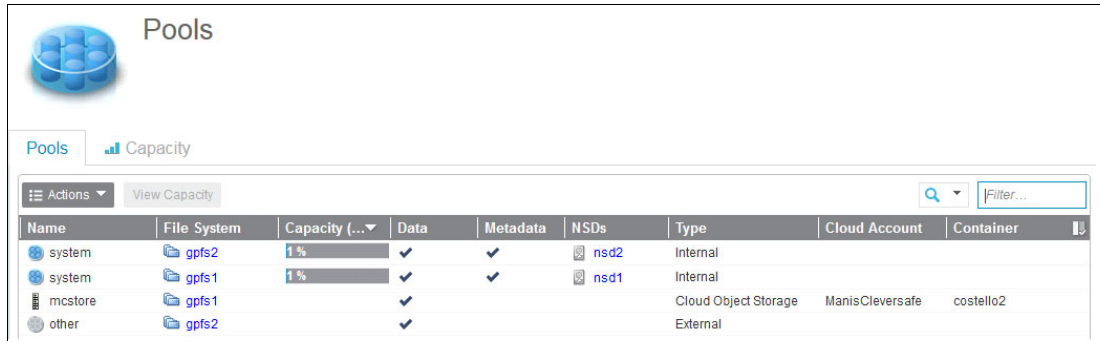


Figure 12 IBM Spectrum Control and cloud storage pools

IBM Spectrum Control can list cloud object pools, for example the mcstore pool in Figure 13, listing the account and cloud containers. In addition, the health of cloud service nodes is included.

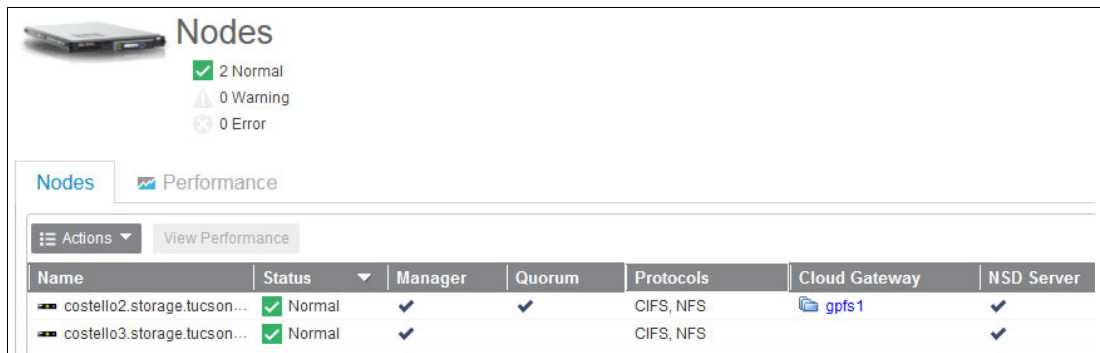


Figure 13 IBM Spectrum Control node status

IBM Spectrum Control displays the status of IBM Cloud Object Storage, as shown in Figure 14. This window provides a single convenient point to view system status.

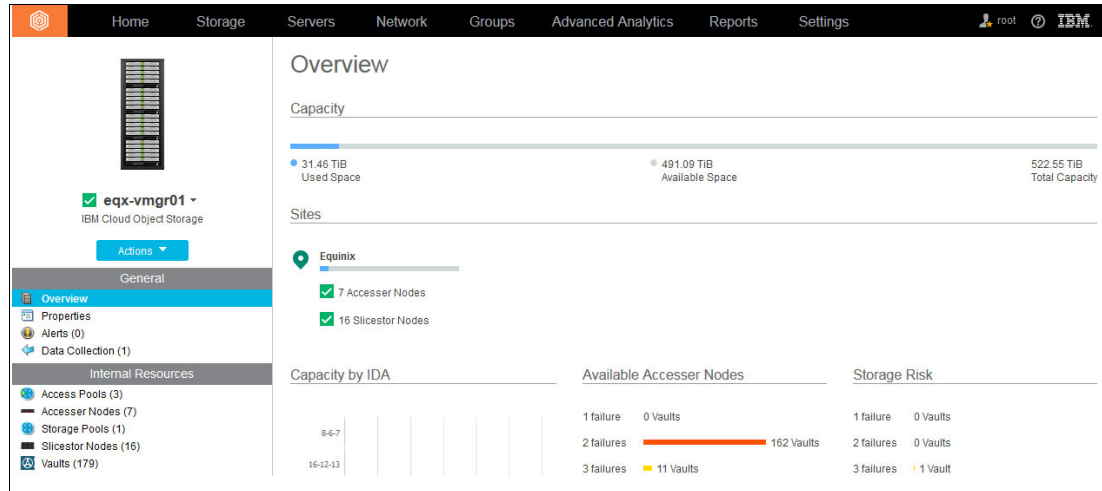


Figure 14 IBM Spectrum Control and IBM Cloud Object Storage

Security

Cloud services on IBM Spectrum Scale send and receive data from external systems. Because many of these systems are on a public cloud, data security is extremely important to consider when configuring a cloud tiering solution. Consider both network security for data that is in flight, and data security for data that is being stored on the cloud.

Network security

IBM Spectrum Scale network security is described in the IBM Spectrum Scale section of the IBM Knowledge Center, along with firewall settings that are required to secure the IBM Spectrum Scale cluster. Cloud service nodes run a service that other nodes in the cluster communicate with to coordinate data migration and recalls.

By default, this service runs on TCP port 1817. The port can be changed by using the `mmcloudgateway` command. Only systems that are a part of the IBM Spectrum Scale cluster require access to this port, which can be blocked from systems external to the cluster itself.

Cloud services communicate with the external cloud service provider through HTTP or HTTPS. If you use HTTP, login credentials are sent in plain text. HTTP is susceptible to malicious users who monitor network traffic to find these credentials. Therefore, use HTTP only on private cloud systems on secure networks.

To avoid man-in-the-middle attacks, where a malicious user intercept traffic between the cloud service node and the cloud provider, cloud tiering performs strong checks on any HTTPS server certificates for authenticity. If you use a private cloud, or a nonstandard configuration, it might be required to manually update the server certificate by using the `mmcloudgateway service` command.

Cloud services initiates TCP connections with the cloud object provider. In many cases, this process involves connecting through a WAN to a public cloud provider. Under no conditions do cloud service nodes require inbound connections from the WAN, so all inbound connections from the WAN should be blocked.

Data security

Cloud services offer native encryption to ensure that data is secure when it is at rest on cloud storage. Data is encrypted by using a secure private key on the cloud service node before sending it to the external cloud object store. The private key can be managed by using a secure key server, such as an ISKLM server, or can be configured directly on the cloud service nodes, where it is stored in a local Java keystore.

Local keys can be used for testing and possibly for some small deployments. A secure server, such as an ISKLM server, is advised when possible to manage encryption keys. Encryption keys should be handled carefully, and you should keep a secure backup of the encryption key. Loss of the key results in the loss of access to any data that has been sent to the cloud provider.

ISKLM servers also offer key rotation, where a new key can be generated to encrypt new data. For key rotation, data that has already been migrated by using the old encryption key will not be reencrypted. For this reason, cloud services store both the old and new encryption key. The old key is used to decrypt data that has already been migrated to the cloud provider. The new key is used to encrypt any new data that is being sent to the cloud provider.

Summary

IBM Spectrum Scale transparent cloud tiering allows your existing applications to take advantage of the flexibility, scalability, and low cost of cloud object storage. Transparent cloud tiering allows you to migrate data that is accessed less frequently to object storage and recall data transparently as it is requested by applications.

The IBM Spectrum Scale ILM policy engine provides administrators with a powerful and flexible tool to manage data being moved to the cloud. End-to-end file validation and available encryption ensure that data remains secure.

This paper described how transparent cloud tiering can be used to create a hybrid cloud solution for Active Archives and high-performance workloads such as analytics. It covered sizing, configuration guidelines, monitoring, and security to help with planning and implementation. This paper also covered how to configure IBM Cloud Object Storage for use with transparent cloud tiering, and how IBM Spectrum Control provides integrated management of a hybrid cloud solution.

Related publications

The publications that are listed in this section are considered suitable for a more detailed discussion of the topics that are covered in this paper.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Some publications referenced in this list might be available in softcopy only.

- ▶ *A Deployment Guide for IBM Spectrum Scale Object*, REDP-5113
- ▶ *IBM Private, Public, and Hybrid Cloud Storage Solutions*, REDP-4873
- ▶ *IBM Spectrum Scale (formerly GPFS)*, SG24-8254
- ▶ *Introduction Guide to the IBM Elastic Storage Server*, REDP-5253

You can search for, view, download, or order these documents and other IBM Redbooks publications, IBM Redpapers™, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- ▶ *IBM Spectrum Scale V5.0 Command and Programming Reference*, SC27-9222
- ▶ *IBM Spectrum Scale V5.0: Administration Guide*, SC27-9220
- ▶ *IBM Spectrum Scale V5.0: Concepts, Planning, and Installation Guide*, GC27-9219
- ▶ *IBM Spectrum Scale V5.0: Problem Determination Guide*, SC27-9221
- ▶ *IBM Spectrum Scale V5.0: Big Data and Analytics Guide*, SC27-9223

Online resources and references

The following resources and references are available online:

- ▶ IBM Client Demonstration Center
<https://www.ibm.com/systems/clientcenterdemonstrations>

Note: The IBM Client Demonstration Center (for Business Partners, IBMers, and anyone with an IBMid) provides a catalog of remote demonstrations (video or live connection) that consists of self-contained material for customer demonstrations of IBM solutions. Most of the demonstrations are provided with predefined scenarios, and some also allow for the development of new scenarios. Demonstrations can also be considered as “ready to use” material for enablement or training.

- ▶ IBM Cloud Object Storage
<https://www.ibm.com/cloud/object-storage>
- ▶ IBM Elastic Storage Server
<https://www.ibm.com/us-en/marketplace/ibm-elastic-storage-server>

- ▶ GNU Parallel: The Command-Line Power Tool - O. Tange, February 2011:42-47:
<https://www.usenix.org/system/files/login/articles/105438-Tange.pdf>
- ▶ IBM Spectrum Scale
<https://www.ibm.com/us-en/marketplace/scale-out-file-and-object-storage>
- ▶ IBM Spectrum Scale in the IBM Knowledge Center
http://www.ibm.com/support/knowledgecenter/SSFKCN/gpfs_welcome.html
- ▶ IBM Spectrum Scale Overview and Frequently Asked Questions (FAQ)
<http://ibm.co/1IK06PN>
- ▶ IBM Spectrum Scale wiki
<https://ibm.biz/BdXVxv>

Authors

This paper was produced by a team of specialists from around the world working at the IBM Storage and SDI development lab, Pune India.

Nikhil Khandelwal is a senior engineer with the IBM Spectrum Scale development team. He has over 15 years of storage experience on NAS, disk, and tape storage systems. He has led development and worked in various architecture roles. Nikhil is currently a part of the IBM Spectrum Scale client adoption and cloud teams.

Rob Basham is a Senior Engineer working as a Storage Architect with IBM System Labs. He has an extensive background working on systems management and storage architecture and standards such as SCSI and Fibre Channel. Rob is an IBM Master Inventor with many patents relating to storage and cloud storage. He is currently working as an architect on cloud storage services including transparent cloud tiering.

Sandeep R. Patil is a Senior Technical Staff Member who works as a Storage Architect with IBM System Labs. He has over 15 years of extensive product architecture and design experience. Sandeep is an IBM Master Inventor, an IBM developerWorks® Master Author, and a member of the IBM Academy of Technology. Sandeep holds a Bachelor of Engineering (Computer Science) degree from the University of Pune, India.

Anbazhagan Mani is a senior engineer and cloud storage architect. He is an IBM master inventor and also a member of Technical Experts Council, India. He has over 16 years of experience working at IBM and has been primarily working on systems management, storage management, and cloud technologies including OpenStack and IBM Cloud Object Storage. He is one of the architects of transparent cloud tiering.

Amey Gokhale has over 15 years (9+ years with IBM) of industry experience in various domains including systems, networking, medical, telecom, and storage. In IBM, he has mainly led teams working on systems and storage management. Starting with IBM Storage Configuration Manager, which managed internal LSI RAID controllers in System x, he led a team that provided storage virtualization services within IBM System Director VMControl, which met storage requirements in virtualized environments. In his current role, he is a co-architect of transparent cloud tiering in Spectrum Scale and is leading the ISDL development team responsible for productization, install, serviceability, and configuration.

Jinesh Shah is a Senior Solution Architect, member of IBM GIST (India) Team. He has worked on various platforms, datacenter activities, and consulting projects. In his current role, he supports IBM partners for global RFPs and outsourcing deals as a Storage Pre-sales consultant. Jinesh has over 19 years of IT experience. He is a science graduate from University of Mumbai, India.

Kedar Karmarkar is a senior engineer and solution architect with IBM Spectrum Scale development team. Kedar is part of the IBM Spectrum Scale Client adoption team and was the IBM Storwize® V7000 Unified Level 3 support lead in his prior role at IBM. He has over 20 years of infrastructure software, storage development experience in management, and architect roles. He has led development of NAS Storage, Block level virtualization, replication, systems, and storage management products. Kedar has a Bachelor of Engineering (Computer Science) degree from the University of Pune, India.

Donald Mathisen is an IT professional with over 25 years of experience who is currently a Senior Managing Consultant for Systems Lab Services. He is IBM Certified for Systems Storage, an IBM Level 2 Certified IT Specialist, and a The Open Group Master Certified IT Specialist. He has held several roles of increasing responsibility with previous companies before coming to IBM. Don started with IBM in Strategic Outsourcing as an Advisory Technical Services Professional in March of 2006. He is the lead person in Systems Lab Services for IBM Object Storage, transparent cloud tiering, Elastic Storage Server, IBM XIV®, and IBM Spectrum Accelerate™ platforms. In addition, he has led many cross-functional worldwide IBM teams on customer facing engagements. He has provided worldwide education and training for services delivery on such products as XIV, FlashSystems, IBM Spectrum Accelerate in the Cloud, and IBM Spectrum Accelerate. Don holds a BA in Computer Science from Mt. Saint Mary College in Newburgh, NY.

Larry Coyne is a Project Leader at the International Technical Support Organization, Tucson Arizona center. He has 35 years of IBM experience with 23 in IBM storage software management. He holds degrees in Software Engineering from the University of Texas at El Paso and Project Management from George Washington University. His areas of expertise include client relationship management, quality assurance, development management, and support management for IBM storage software.

Arend Dittmer is a Solution Architect for Software Defined Infrastructure. He has over fifteen years of experience with scalable Linux based solutions in enterprise environments. His areas of interest include virtualization, resource management, big data, and high performance parallel file systems. Arend holds a patent for the dynamic configuration of hypervisors for portable virtual machines.

Thanks to the following people for their contributions to this project:

Tito Hernandez

IBM Cloud

Keith D. Ball

Michael Daubman

Scott Fadden

Pallavi Galgali

Arati Ranade

Harald Seipp

Frederick Stock

Christine Telford

Carl Zetie

IBM Systems

Lisa Shepard
IBM Global Technology Services®

Now you can become a published author, too

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time. Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run two - six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online:

ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>



REDP-5411-01

ISBN 0738456861

Printed in U.S.A.

Get connected

