

Cloud Data Sharing with IBM Spectrum Scale

Nikhil Khandelwal

Rob Basham

Amey Gokhale

Arend Dittmer

Alexander Safonov

Ryan Marchese

Rishika Kedia

Stan Li

Ranjith Rajagopalan Nair

Larry Coyne



Cloud

Storage





Cloud data sharing with IBM Spectrum Scale

This IBM® Redpaper™ publication provides information to help you with the sizing, configuration, and monitoring of hybrid cloud solutions using the Cloud data sharing feature of IBM Spectrum Scale™. IBM Spectrum Scale, formerly *IBM General Parallel File System* (IBM GPFS™), is a scalable data and file management solution that provides a global namespace for large data sets along with several enterprise features. Cloud data sharing allows for the sharing and use of data between various cloud object storage types and IBM Spectrum Scale.

Cloud data sharing can help with the movement of data in both directions, between file systems and cloud object storage, so that data is where it needs to be, when it needs to be there.

This paper is intended for IT architects, IT administrators, storage administrators, and those who want to learn more about sizing, configuration, and monitoring of hybrid cloud solutions using IBM Spectrum Scale and Cloud data sharing.

Introduction

Cloud data sharing is a feature available with IBM Spectrum Scale Version 4.2.2 that allows for transferring data to and from object storage, using the full lifecycle management capabilities of IBM Spectrum Scale information lifecycle management (ILM) policy engine to control the movement of data. This feature is designed to work with many types of object storage allowing for the archiving, distribution, or sharing of data between IBM Spectrum Scale and object storage.

This paper is organized into the following sections:

- ▶ Technology overview: The target audience for this section is anyone who is interested in this topic.
- ▶ IBM Spectrum Scale Cloud data sharing use cases: This section covers scenarios where Cloud data sharing are typically used, and the target audience is pre-sales and solution architects.
- ▶ Sizing and scaling considerations: This section covers guidance about how to plan for the recommended resources for the sharing service. It also covers suggestions about how to deploy the service. The target audience is solution architects and administrators.
- ▶ Resource requirements and considerations: This sections covers resource planning aspects of using Cloud data sharing. The target audience is solution architects and administrators.

- ▶ **Configuration and best practices:** This section covers recommended settings and tunable parameters when using the Cloud data sharing service. The target audience is solution architects and administrators.
- ▶ **Monitoring and lifecycle management:** This section covers instruction about monitoring the Cloud data sharing service. The target audience is administrators.
- ▶ **IBM Spectrum Control:** This section covers preferred practices for setting up IBM Spectrum Control™ to support Cloud data sharing.

Technology overview

This section provides an overview of IBM Spectrum Scale, cloud object storage, and how IBM Spectrum Scale provides a fully integrated, transparent Cloud data sharing service.

According to 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% of the information in the digital universe at some point in its lifecycle. Thus, organizations are taking on the responsibility for designing, delivering, and maintaining IT systems and data storage systems to meet this demand.¹

Both IBM Spectrum Scale and object storage, including IBM Cloud Object Storage, are designed to deal with this growth. In addition, there's now a fully integrated transparent Cloud data sharing service that will combine the two. IBM Spectrum Scale can work as a peer with storage, such as IBM Cloud Object Storage, by facilitating sharing of data in both directions with cloud object storage.

IBM Spectrum Scale overview

IBM Spectrum Scale is a proven, scalable, high-performance data and 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 can help to reduce storage costs up to 90% and to improve security and management efficiency in cloud, big data, and analytics environments.

¹ Source: EMC Digital Universe Study, with data and analysis by IDC, April 2014

Figure 1 shows a high-level overview of this solution.

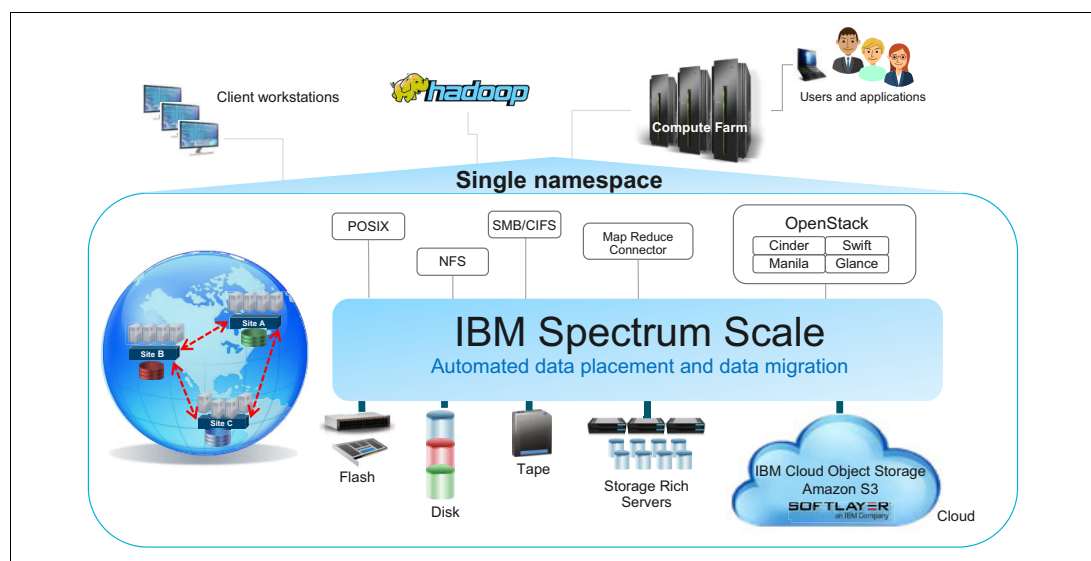


Figure 1 IBM Spectrum Scale overview

Trial VM: 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 this trial version from [IBM developerWorks®](#).

Cloud storage overview

Object storage is the primary data resource used in the cloud, and it is also increasingly used for on-premise solutions. Object storage is growing for the following reasons:

- ▶ It is designed for scale in many ways (multi-site, multi-tenant, massive amounts of data).
- ▶ It is easy to use and yet meets the growing demands of enterprises for a broad expanse 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. However, it is key to remember that most object storage services are only “eventually consistent.” To accommodate the massive scale of data and the wide geographic dispersion, object storage service at times and in places might not immediately reflect all updates. This lag is typically quite small but can be noticeable when there are network failures.

Object storage is typically offered as a service on the public cloud, such as Amazon S3 or SoftLayer®, but is also available as on premise systems, such as IBM Cloud Object Storage (formerly known as *Cleversafe*®).

For more information about object storage, see the [IBM Cloud Storage](#) website.

Due to its characteristics, object storage is becoming a significant storage repository for active archive of unstructured data, both for public and private clouds.

Object storage: 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, such as IBM Cloud Object Storage, is that the former includes a unified file and object access with Hadoop capabilities and is suitable for high performance oriented or data lake use cases. IBM Cloud Object Storage is more of a traditional cloud object store suitable for the active archive use case.

Cloud deployment models

IBM Cloud Object Storage can be deployed both on and off premise. There are numerous private and public cloud offerings.

Off premise cloud

Off premise public clouds, such as IBM SoftLayer Object Storage or Amazon S3, provide storage options with minimal additional capital equipment and datacenter costs. The ability to rapidly provision, expand, and reduce capacity and a pay-as-you-go model make this a flexible option.

When considering public clouds, it is important to consider all the costs and the pricing models of the cloud solution. Many public clouds charge a monthly fee based on the amount of data stored on the cloud. In addition, many clouds charge a fee for data 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 might result in additional 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, can provide flexible, easy-to-use storage with attractive pricing. On premise clouds allow complete control over data and high speed network connectivity to the storage solution. This type of solution might be ideal for use cases that involve larger files with higher recall rates. For a cloud solution with multiple access points and IP addresses, an external load balancer is required for high availability and throughput. Several commercial and open-source load balancers are available. Contact your cloud solution provider for supported load balancers.

Cloud services major components and terminology

This section describes the IBM Spectrum Scale cloud services components that are used by the Cloud data sharing service on IBM Spectrum Scale. A typical multi-node configuration is shown in Figure 2 on page 5 with terminology that will be referenced throughout the paper.

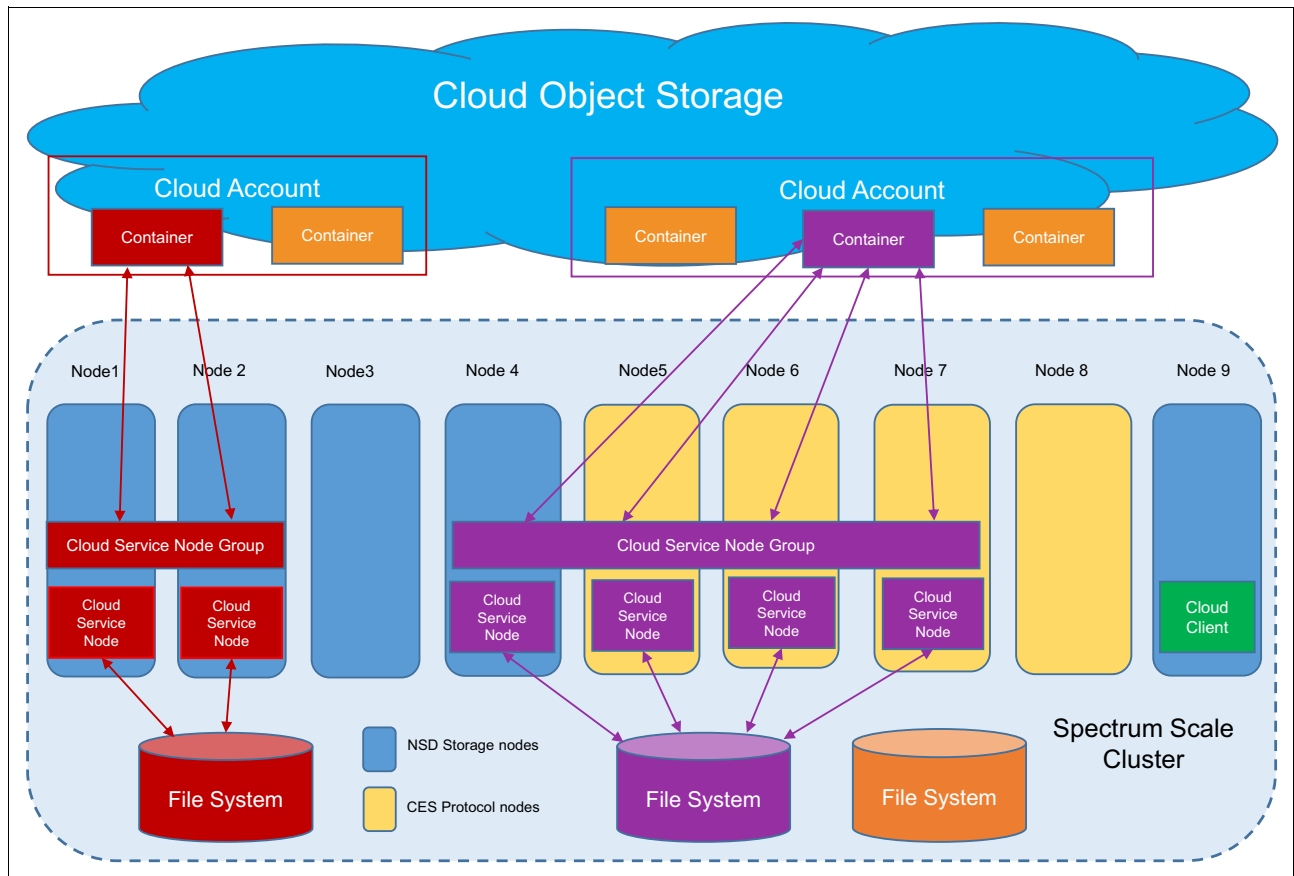


Figure 2 IBM Spectrum Scale cluster with cloud service nodes

The Cloud data sharing service runs on cloud service node groups that consist of cloud service nodes. For reliability and availability reasons, a group will typically have more than one node in it. In the current release, one IBM Spectrum Scale file system and one cloud storage *Cloud Account* can be configured with one associated container in object storage. Figure 2 shows an example of two node groups with the file systems, the cloud service node groups, and the cloud accounts and associated transparent cloud containers colored purple and red.

Cloud services

There are two cloud services currently supported: transparent cloud tiering service and Cloud data sharing. For more information about transparent cloud tiering see *Enabling Hybrid Cloud Storage for IBM Spectrum Scale Using Transparent Cloud Tiering*, REDP-5411 (<http://www.redbooks.ibm.com/abstracts/redp5411.html>).

Cloud service node group

The Cloud data sharing service runs on the cloud service node group. Multiple nodes can exist in a group for high availability in failure scenarios and for faster data transfers by sharing the workload across all nodes in the node group. A node group can be associated only 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 interacts directly with the cloud storage. Transfer requests go to these nodes and they perform the data transfer. A cloud service node can be defined on an IBM Spectrum Scale protocol node (CES node) or on an IBM Spectrum Scale data node (NSD node).

Cloud account

Object Storage is multi-tenant, but cloud services can talk only to one tenant on the object storage, which is represented by the cloud account.

Cloud client

The cloud client can reside on any node in the cluster (provided the OS is supported). This lightweight client can send cloud service requests to the Cloud data sharing service. Each cloud service node comes with a built-in cloud client. A cluster can have as many clients as needed.

Cloud data sharing overview

Cloud data sharing (see Figure 3 on page 7) allows for movement of data between object storage and IBM Spectrum Scale storage:

- ▶ Move data from IBM Spectrum Scale to cloud storage: Export data to cloud storage by setting up ILM policies that trigger the movement of files from IBM Spectrum Scale to cloud object storage.
- ▶ Move data from cloud storage to IBM Spectrum Scale: Import data from cloud storage from a list of objects. (An ILM policy doesn't work for import because policies can only specify files that are already in the IBM Spectrum Scale file system.)

This service provides the following advantages:

- ▶ It is designed for scale in many ways (multi-site, multi-tenant, and massive amounts of data).
- ▶ It is easy to use and yet meets the growing demands for sharing or distribution of data for a broad expanse of applications and workloads.
- ▶ It allows users to easily move large amounts of data by policy between object storage and IBM Spectrum Scale.
- ▶ It provides a manifest and an associated manifest utility that can be used to track what data has been transferred without needing to use object storage directory services which can be problematic as a container scales up.

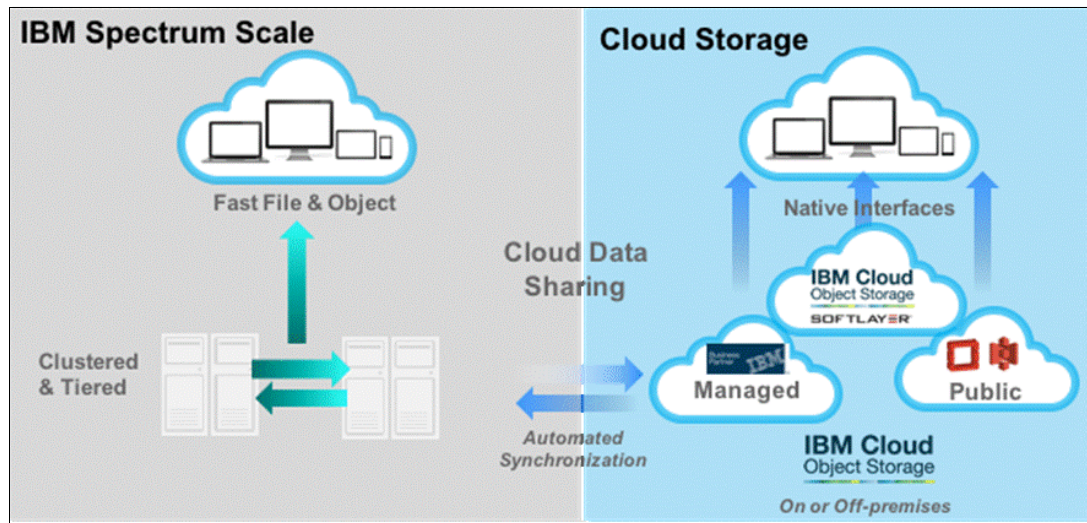


Figure 3 Cloud data sharing overview

Cloud data sharing versus transparent cloud tiering

Cloud data sharing and transparent cloud tiering cloud services both transfer data between IBM Spectrum Scale and cloud storage, but they serve different purposes. Here are some ideas about how to decide which service to use.

You can use Cloud data sharing in the following cases:

- ▶ You need to pull object storage data originating in the cloud into IBM Spectrum Scale. The Cloud data sharing **import** command performs this service.
- ▶ You need to push data originating in IBM Spectrum Scale out to cloud object storage so it can be consumed in some way from object storage. The Cloud data sharing **export** command performs this service.
- ▶ You need to archive IBM Spectrum Scale data out to the cloud and do not want to maintain any information whatsoever on those files in IBM Spectrum Scale. This can be done using the **export** command and then by deleting the associated files that were exported.

You can use transparent cloud tiering in the following cases:

- ▶ You want to use cloud storage as a storage tier, migrating data to it as it gets cool and recalling the data later as needed.
- ▶ You need additional capacity in your file system without purchasing additional hardware.
- ▶ You need to free up space in your primary storage.

How Cloud data sharing works

Cloud data sharing starts a cloud services daemon on one or more cloud service nodes. This daemon communicates directly to a cloud object storage provider. Any node in an IBM Spectrum Scale cluster can send a request to a cloud service node to import or export data. The requests can be driven either directly via the **mmcloudgateway** command or through IBM Spectrum Scale ILM policies. Depending on the application, you can schedule the import or export of data to run periodically or they can be driven by an application or user.

After a file is exported to the cloud provider, it can be accessed either by another IBM Spectrum Scale cluster or by a cloud-based application. The file data is not modified in any way, so an application can access the file directly if desired. An optional manifest file can be specified when exporting a file.

The Export, Import, and the Manifest functions are described as follows:

- **Export**

Exportation of data can be driven by periodic running of a policy on the IBM Spectrum Scale ILM policy engine, with the files to be exported selected by conditions that are specified in the policy. This method is useful for providing consistency between IBM Spectrum Scale files and the object storage over time.

- **Import**

Importation of data cannot be done with the policy engine, because the policy engine works on files that are already in IBM Spectrum Scale. For this reason, the import is driven by directly requesting a set of files to be created. When importing files, the file data can be migrated to the IBM Spectrum Scale cluster during the import operation itself, or the file can be imported as a stub file. If a file is imported as a stub, the file data will not be transferred until the file is accessed on the file system. After the file is present in the operating system as a stub, a policy to import the data can be run separately with the decision on what files to pull in fully, depending on the policy engine.

- **The Manifest**

The manifest file contains information about the files present in object storage. This manifest can be used by another IBM Spectrum Scale cluster to determine what data to import or it can be used by object storage applications that want to know what data has been exported to object storage. There is a cloud manifest tool that will provide a comma separated value list of files in a manifest. It can also be used to generate a manifest for cloud storage applications that want to have IBM Spectrum Scale Cloud data sharing service use the manifest to determine what data to import.

IBM Spectrum Scale Cloud data sharing use cases

Consider the following key questions which can help determine if a workload is appropriate for cloud sharing:

- Do you have data that needs to be shared between object storage and your IBM Spectrum Scale cluster?
- Does your data consist of larger size files or objects, such as unstructured objects, images, movies, or documents that you might want to archive to cloud storage that you no longer need in IBM Spectrum Scale namespace?
- Do you have requirements for data security, availability of data across multiple sites, scalability, and cost-effectiveness?
- Do you have the infrastructure or ability to procure the infrastructure required to support cloud storage, including network connectivity and load balancing services, and private or public cloud access?

This section showcases the following use cases that can be broadly applied to different workloads:

- ▶ Data distribution or sharing between sites
- ▶ Data export/import from native file systems to object storage by sharing data between applications
- ▶ Active Archive

Use Case: Data distribution or sharing between sites

In multi-site organizations or at digital media stations that typically involve a setup, such as a central office with multiple branch offices, data sharing between sites is typically required. For example, digital media solutions involve data that needs to be shared across geographically distributed sites. Data can originate at the branch office or central office. You can use as many repositories (often referred to as *containers*) as needed with the appropriate access controls for the groups that you need to share them with. Set up as many groupings as you need. See Figure 4.

- ▶ Export data from branch office to central office at periodic intervals (daily or weekly)
- ▶ Share data from central office to branch office (import data)

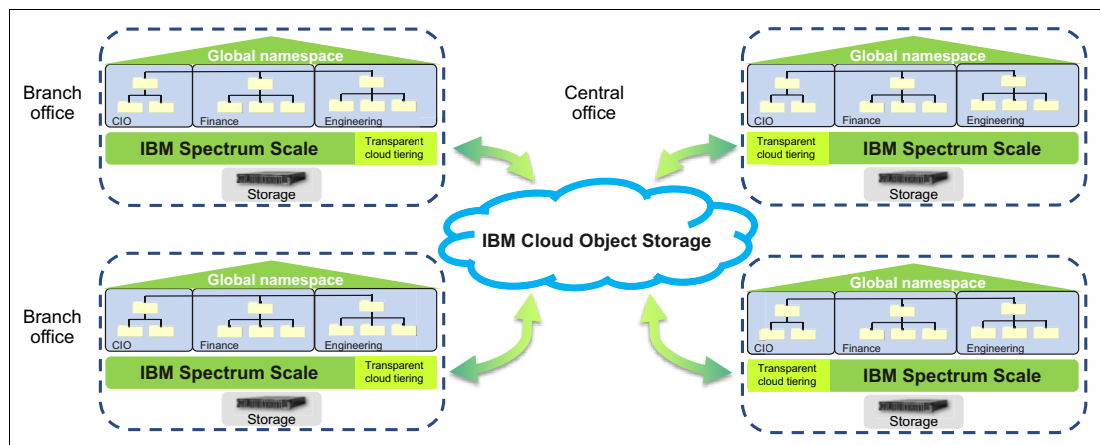


Figure 4 Use case example, sharing data between sites

Use Case: Sharing data between applications

This use case demonstrates Object Storage capability to share data between various applications. The applications in this example are not limited by specific Object Storage data access methods. In this use case, there is a group of content producer applications and a single content consumer application. Figure 5 shows that content producer applications are applications #1, #2, and #3 and that the content consumer application is application #4.

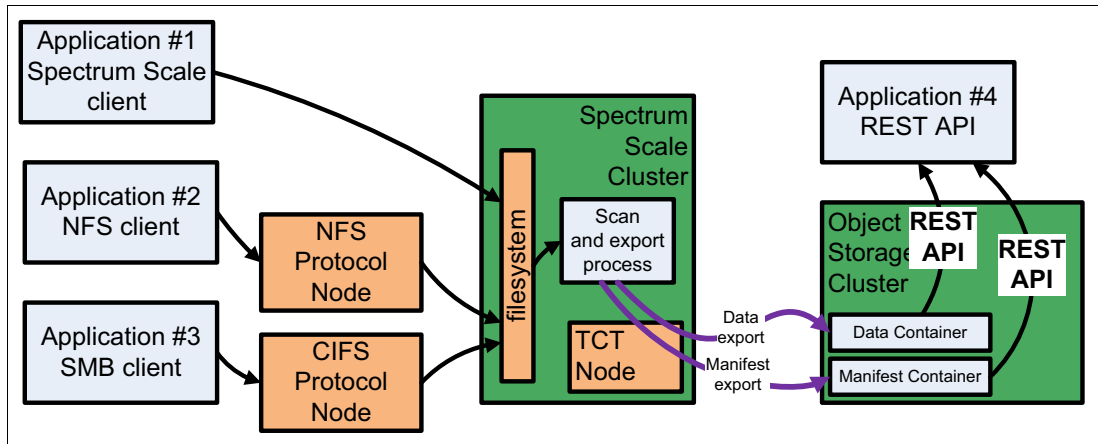


Figure 5 Sharing data between applications with content producer and content consumer applications

Content producer applications access the IBM Spectrum Scale cluster using the standard file system interface. This example is not limited by a specific file system access protocol. It can be either native access or CIFS/NFS access. Applications #1, #2, and #3 produce content and store this content in files written to an IBM Spectrum Scale file system. Each application uses a dedicated directory or file set in the file system to store its content.

Content consumer application #4 in this example is a remote application without direct access to the IBM Spectrum Scale file system. This application might represent a class of workloads that can potentially run anywhere. To implement data sharing for these external applications, we need to find a mechanism that allows this sample data to be stored outside of the IBM Spectrum Scale file system. The characteristics of Object Storage makes it an attractive choice for this type of data externalization.

Use Case: Off-line data archival

The growing amount of unstructured data in the data centers, and the need to manage and store the data efficiently, is a major requirement driving the need for an archival solution. An archival solution allows users to move data offline to cloud storage from IBM Spectrum Scale. With such an archival solution, applications can indirectly recall archived data as needed using a manifest file facility. The Manifest file includes references to all files that have been moved to the cloud. Figure 6 shows the flow of archiving data to the cloud, and Figure 7 shows restoring data from the cloud.

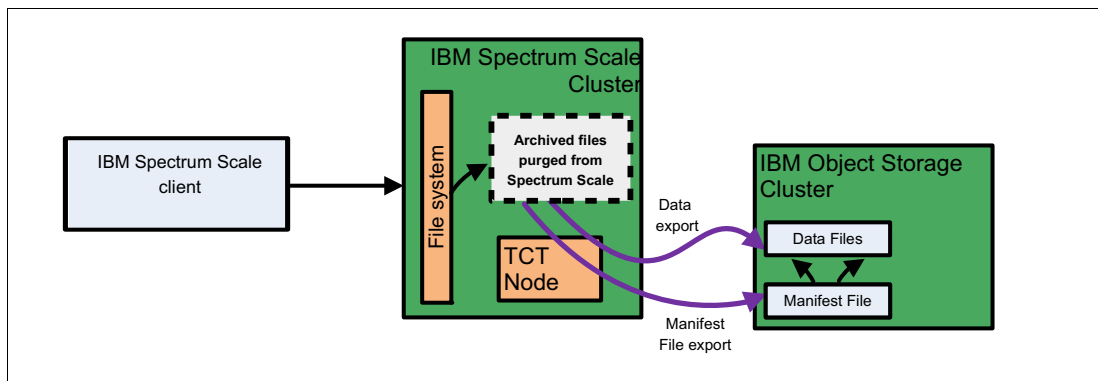


Figure 6 Archiving data to cloud from IBM Spectrum Scale

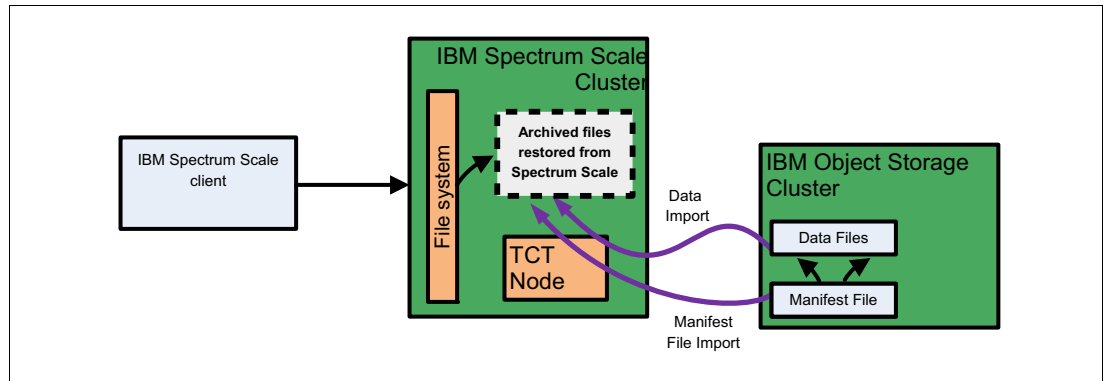


Figure 7 Restoring data from cloud with IBM Spectrum Scale

An Archival solution used for cloud sharing offers the following 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 stretch clusters that allow for support across multiple sites, thereby allowing uninterrupted access or loss of data even if an entire site goes down.
- ▶ Reduced capital equipment costs with off-premise cloud solutions. Archival solution can benefit various industries as shown in Figure 8.

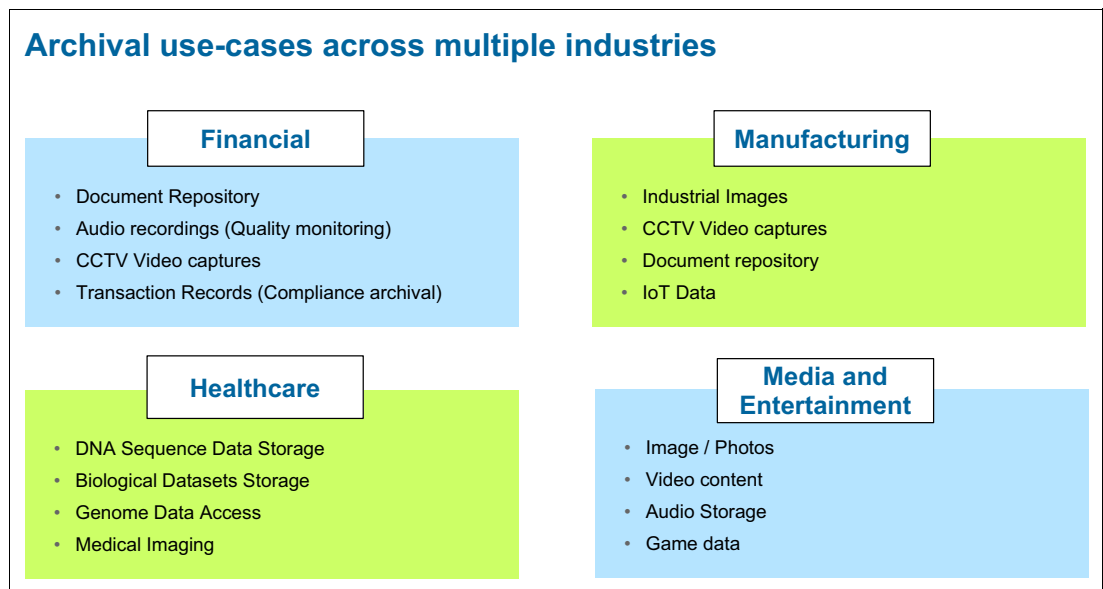


Figure 8 Archival use-cases across multiple industries

Sizing and scaling considerations

This section provides guidance for administrators and architects for sizing the IBM Spectrum Scale infrastructure for efficient Cloud data sharing. Keep in mind the following factors for any Cloud data sharing deployment:

- ▶ The number of cloud service nodes and node hardware
- ▶ The network connecting the IBM Spectrum Scale cluster to the cloud storage provider
- ▶ Active storage and metadata requirements for the IBM Spectrum Scale cluster
- ▶ Capabilities of the cloud object storage used in the deployment

Cloud service nodes

Nodes that host cloud services should be running a current generation Intel server CPU with at least 10 physical processor cores and should be configured with at least 32 GB of memory. To ensure availability, have a minimum of two cloud service nodes. Standard data (NSD) nodes are easier to set up, but if the cloud service nodes chosen are running on Cluster Export Services (CES), hardware requirements as outlined in the IBM Spectrum Scale wiki apply in addition to the minimum requirements.

As a performance consideration, cloud services can scale out up to up to four nodes per file system and parallel export workloads driven by IBM Spectrum Scale policies scale in performance with the number of cloud services nodes.

Use the following recommendations to avoid bottlenecks in communication between the cloud service node and the object storage network:

- ▶ Run cloud services and active CES and other significant workload on different nodes as cloud services can be heavy in resource usage.
- ▶ Separate the IBM Spectrum Scale network from the network connected to the cloud storage system (for internal cloud storage).

Network

Make available a large amount of network bandwidth between cloud service nodes and cloud object storage systems to avoid the network being bottleneck. A simple calculation that factors in the expected amount of data to be migrated or recalled, the time over which the transfer is expected to occur, and the number of cloud service nodes available can provide a good sense of whether the available network bandwidth is sufficient to transfer data within the required time frame. When performing these calculations, keep in mind that only exports driven by IBM Spectrum Scale policies can take advantage of multiple cloud service nodes. With additional scripting effort, imports can be made parallel by splitting manifest files and using tools such as GNU parallel.

When working with off-premise cloud storage contact your cloud administrator to determine the maximum bandwidth for cloud connectivity. The WAN connection to an external network is a typical bottleneck for off-premise storage. Some providers, such as IBM SoftLayer and Amazon, offer dedicated network links to obtain high speed access.

On-site cloud storage systems such as IBM Cloud Object Storage or OpenStack Swift can be configured with multiple redundant access points to the data for accessibility and performance scaling. A cloud service node group doing Cloud data sharing, however, sends all data to a single target address. For that reason, a load balancer is required to take advantage of multiple access points and get the desired scaling and performance. There are several software and hardware load balancers available. DNS round-robin load balancing has

been shown to work well for many customers. Contact your storage administrator for available load balancing options.

IBM Spectrum Scale

Cloud data sharing works with any back-end storage and deployment model supported by IBM Spectrum Scale. To determine the list of files to export to cloud storage Cloud data sharing can use IBM Spectrum Scale policy scans of file system metadata. As the number of files in the file system grows, the size of the metadata increases, and the time required to run these scans will increase.

Configuring IBM Spectrum Scale to include a system storage pool backed by flash based storage for metadata can help reduce scan times and the impact of scans on other system operations.

To further improve IBM Spectrum Scale cloud sharing performance, it might make sense to increase the IBM Spectrum Scale `pagepool` and `dmapiWorkerThreads` parameters. A large page pool size enables more aggressive prefetching when reading large files and directories. In addition, an increased `dmapiWorkerThreads` parameter results in more threads for the data transfer to cloud storage. Set the `dmapiWorkerThreads` parameter to a minimum of 24 and the `pagepool` size to at least 4 GB.

Resource requirements and considerations

This section describes the hardware and resources required for Cloud data sharing.

Cloud service nodes

Cloud service nodes should at least meet the minimum requirements for a CES node on an IBM Spectrum Scale. See the IBM Spectrum Scale FAQ for current protocol node requirements and other hardware restrictions for cloud services. The current IBM Spectrum Scale release supports from one to four cloud service nodes to service a file system. For a redundant solution with high availability, a minimum of two cloud service nodes are required. For cloud service providers with high bandwidth, such as multiple 10 GbE networks, four cloud service nodes might improve throughput to the cloud provider.

If the cloud service node will be used for cloud services only, the minimum memory requirements of a CES protocol node will be sufficient in most cases. However, if the node will also be used for other protocols (for example serving NFS or SMB exports) or will be running other applications, ensure that the system meets the minimum requirements for a node servicing multiple protocols. It should be noted that for the best performance of the cloud service node, there should be no other protocol services running on the node.

In most cases, the cloud service nodes initiates 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 intensive and might have to be run less frequently. If implementing cloud services on a file system with hundreds of millions or billions of files, additional memory devoted to the scan might improve scan times.

Cloud services attempt to make use of as much bandwidth as possible to transfer files to and from the cloud. As a result, it can impact other operations that make use of the same network adapter. Avoid sharing a network that IBM Spectrum Scale is using for internal

communication with cloud services. Also avoid sharing the same network adapter that used by protocol services running on the cloud service node with cloud services.

Network

The network used to communicate with the cloud storage provider is important to consider. The network requirements and bandwidth will vary significantly for on-site versus off-site cloud providers.

It is important to ensure that the network bandwidth is capable of handling the amount of data that will be migrated to the cloud. One way to calculate this is by examining the set of files that will be migrated to the cloud and the expected growth in those files. 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 a file system is growing by 10 TB per day and a 1 Gbps link (approximately 100 MBps) is available to the cloud storage provider, it will take approximately $10000000 \text{ MB} / 100 \text{ MBps} = 100000$ seconds to migrate the data. Because there are only 86400 seconds in a day, it will take more time to migrate the data than is available and a faster network is required.

Similarly, it is important to consider the recall times over the network, especially for large files. For example, if we migrate a 60 GB file over the same 1 Gb network link and later need to read the file, it will take at least $60000 / 100 \text{ MBps} = 600$ seconds to recall the file. This time might grow if multiple files are being read at the same time or if there is other network contention. In many cases long recall times will be acceptable because they occur rarely. However, it is important to consider the needs of your applications and users.

When working with an off-site cloud, work with your cloud provider to determine the bandwidth that can be obtained to the cloud. In most cases this will depend on a WAN connection to an external network. Some providers, such as IBM Softlayer and Amazon, might offer dedicated network links to provide higher speed access.

On-site cloud storage systems, such as IBM Cloud Object Storage, might have multiple redundant access points. Because cloud services currently can be configured only with a single IP address, a load balancer is required for redundancy and might improve system performance by using multiple object endpoints. Several software and hardware load balancers are available. Contact your cloud storage provider for available options.

IBM Spectrum Scale back-end storage

Cloud services run with any back-end storage type that is supported by IBM Spectrum Scale. Cloud services use IBM Spectrum Scale policy scans to scan file system metadata and determines what files might be candidates to move to cloud storage. As a file system contains more files, the size of the metadata increases, and the time required to run these scans increases. High performing metadata storage helps to reduce scan times and the impact of scans on other system operations. In many cases, using flash storage for metadata can greatly reduce the time and impact of these scans.

Cloud storage

Cloud services support the Amazon S3 and OpenStack Swift protocols for object storage. It has been tested with IBM Cloud Object Storage and IBM Spectrum Scale Object. It has also been tested with IBM SoftLayer 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-premise or dedicated cloud solutions. For on-premise cloud deployments, a load balancer might be required if there are multiple endpoints in order to meet redundancy and performance requirements.

Configuration and best practices

This section guides the administrator with the preferred practices to be followed when deploying IBM Spectrum Scale in Cloud data sharing environments. Details of the attributes are listed in the [IBM Spectrum Scale Wiki](#) on developerWorks.

CES best practices

Although not recommended, cloud services nodes can be run on CES nodes within the cluster. For environments with high bandwidth requirements to the cloud provider, it is beneficial to disable protocol services such as NFS, SMB, or Object on the CES nodes that are running cloud services. If running cloud services with the protocol services on the same node is required, that node might require additional CPU and memory resources. Otherwise, performance degradation for both services might be observed.

Network best practices

Refer to “Sizing and scaling considerations” on page 12 for information about configuring your network.

It is common to bond multiple network adapters on a single system for higher performance or availability. For performance purposes, use such bonding with Cloud data sharing. For availability, active/passive bonds are sufficient for connectivity in the event of a network port or cable failure. For performance, it is common to use a bonding mode, such as 802.3ad. If using a network bond mode for performance, pay attention to the hash mode on the network adapter and on the switch. In most cases, Cloud data sharing communicates with a single endpoint. So hashing modes that rely on a MAC address or an IP address might not distribute network traffic across multiple adapters. Refer to your network switch or OS vendors for details regarding hashing modes.

Cloud service node configuration

To use Cloud data sharing functionality, you first need to configure a cloud service node with a file system and a cloud account, which is similar to setting up cloud services functionality. To configure the cloud service node, use the information listed in the [IBM Spectrum Scale Wiki Quick Start guide](#). Follow the steps up to the section titled “Configure cloud account to be used by Transparent Cloud Tiering.”

Remember to turn off encryption while configuring the cloud account, using the `-enc-enable FALSE` option, when you the cloud account, if Cloud data sharing is to be used. Data to be exported to cloud will not be of any use, if it is encrypted. Other services/applications might not be able to use this encrypted shared data.

Exporting data to cloud using the Cloud data sharing service

When setting up to export data to the cloud, consider how many different groups you need with different read access privileges to that data. The recommended approach is to establish a container (or *vault* as it is called on some object storage) for each group. Set up a different export policy for each group, and target that export at the appropriate container and associated manifest file. You will likely want to run these policies in cron jobs so that the object storage and the IBM Spectrum Scale file system data can be kept consistent.

In addition, exports can be configured and executed via policy either manually or periodically using cron jobs. You can find sample policy that can be applied to export files based on timestamps from a GPFS mount point in the [IBM Spectrum Scale](#) information in IBM Knowledge Center.

Importing data from cloud using Cloud data sharing service

Unlike with the **export** command, you cannot use the policy manager to import files, because policy works against files that are already in the file system. For that reason, **import** is an explicit command.

An import operation allows an administrator to pull objects down from the cloud, without modifying the original object. The **mmcloudgateway files import** command provides this functionality. For more details about the available command line options for importing data from the cloud, go to the [IBM Spectrum Scale](#) information in IBM Knowledge Center.

Apart from the regular command line options, here is a quick code snippet that allows administrators to import all objects from a given container to IBM Spectrum Scale, using this Cloud data sharing service:

```
curl -s http://<cloud-provider-URL>/<container-name> | awk -v RS='<Key>' '{ print $1}' | grep pdf | awk -F'</Key>' '{ print $1 }' | xargs mmcloudgateway files import --container <container-name>
```

This command lists the objects within a container and redirects the object name, including full path, to the Cloud data sharing service **import** command. It allows an administrator to import all objects from a given container into the IBM Spectrum Scale file system in one pass.

Example configuring for sharing data between applications

In this example, Application #1, Application #2, and Application #3 represent a traditional IT environment with predominantly file system interface for workloads running in a data center. However, a traditional file system interface might become too restrictive when planning to deploy an application hosted outside of the data center. Application #4 represents a remote workload. The relative lightweight nature of REST API data access protocol makes it better suited for accessing data via potentially slow WAN links or where the application is running natively in the cloud.

Applications #1, #2, and #3 create data in a POSIX compatible file system. The scan process illustrated in Figure 9 on page 17 as *Scan and export process* detects a new file creation and initiates the data export process. The export process is using the Cloud data sharing export function introduced in IBM Spectrum Scale V4.2.2. A file export process creates an object in the specified Object Storage Data Container. The content of the newly created object is identical to the content of the original file. However, the location of the newly created object is defined based on several criteria such as URL of the Object Storage cluster, name of the Data Container, and so on.

The following example shows a file name and a corresponding object name:

File name: /gpfs/fs1/images/application3/header_t.png

Object name: http://object_cluster_ip/itso-data/gpfs/fs1/images/application3/header_t.png

The following export process is used for exporting the file in this example:

```
/usr/lpp/mmfs/bin/mmcloudgateway files export --tag testtag --container itso-data  
--export-metadata --manifest-file manifest.txt
```

When an export process is completed, the manifest file is updated with an entry that records the tag, container name, time stamp, and location of the object in the container. In addition, the following record is created in the manifest.txt file after the export process creates an object successfully:

```
testtag,itso-data,16 Nov 2016 22:50:16 GMT,gpfs/fs1/images/application3/header_t.png
```

You can find more details about the manifest.txt file format in the [IBM Spectrum Scale](#) information in IBM Knowledge Center.

The manifest file is a source of valuable information for the data consumer application. The application can use this manifest file information to locate a specific object. It can also use time stamps that are recorded in the manifest file to determine when objects were exported. For example, if the application needs to reference all objects that were created in the last 24 hours, it can process the manifest file.

Finding a specific object in the Data Container can be implemented in one of the following ways:

- ▶ Accessing a container index: When the number of objects in a container is relatively small, listing objects in a container performs well. With container indexing, however, as the number of objects increases, the overhead associated with maintaining a container index becomes more significant.
- ▶ Cloud data sharing manifest: The alternative to using a container index is relying on the manifest generated by the Cloud data sharing service that is specifically associated with application data that has moved. In the example, we create a separate manifest for each data producer application. Creating a per-application manifest file can improve overall scalability of the data sharing solution. The advantage of this approach is that it provides a specific and efficient way to know what data has been moved and would be of interest. Figure 9 illustrates the per-application manifest approach.

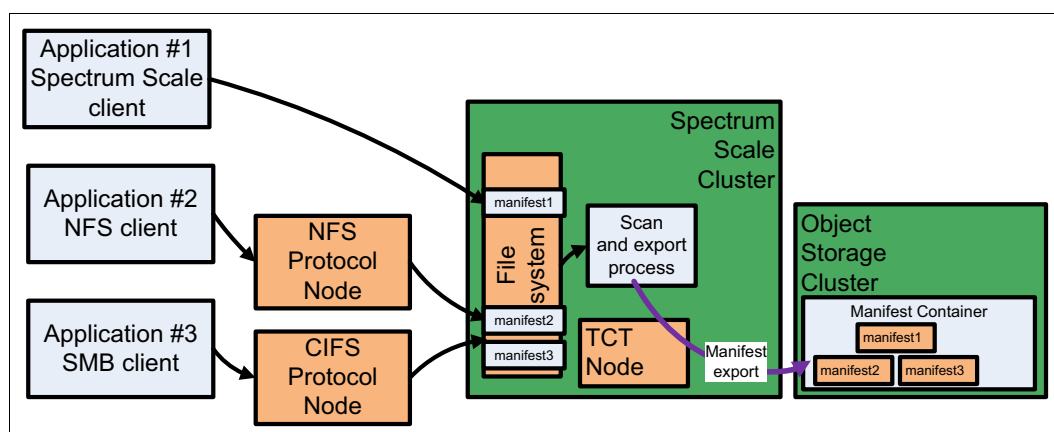


Figure 9 Cloud data sharing: Per-application manifest approach

It is not a function of a content producer application to update a corresponding manifest file when a new data file is created in a file system, because it is not always possible to modify how content producing applications interact with a file system.

For this example, applications #1 through #3 only write data files. The task of exporting data files and updating corresponding manifest files is performed by the scan and export processes. Note that you are not limited by the number of scan processes that will support the data producer applications. If the number of producers is sufficiently high, you can increase the number of scan and export processes to avoid a potential bottleneck. For simplicity, the example shown here is a single scan process. However, if multiple TCT nodes are deployed, one scan process per TCT node is a good starting point. In addition, even on a single TCT node, you can run multiple scan processes if you expect a significant growth in the number of files in the file system.

Example 1 includes sample code fragment that illustrates the **scan** and **export** process module functions.

Example 1 Scan and export process module functions

```
#!/bin/bash

MANIFEST_CONTAINER='itso-manifest'
DATA_CONTAINER='itso-data'
CLOUD_URL='http://Object_Storage_Cluster_IP'
BASE_DIR=/gpfs/fs1/images
SCAN_FREQUENCY=5;

while true
do
cd $BASE_DIR

for applications in `ls -l` ;
do cd $BASE_DIR/$applications ;
> manifest.list ;
find . -type f -exec basename {} \; | grep -v manifest | sort -u >
manifest.list ;
done

cd $BASE_DIR
for applications in `ls -l` ;
do cd $BASE_DIR/$applications ;
> manifest.presort ;
for i in $(cat manifest.txt | awk -F ',' '{ print $4}') ;
do basename $i >> manifest.presort ;
done ; sort -u manifest.presort > manifest.extract ;
done

cd $BASE_DIR
for applications in `ls -l` ;
do cd $BASE_DIR/$applications ;
for i in $(diff manifest.extract manifest.list | grep '>' | awk '{ print $2
}') ;
do echo exporting $BASE_DIR/$applications/$i ;
/usr/lpp/mmfs/bin/mmcloudgateway files export --tag testtag --container
$DATA_CONTAINER --export-metadata --manifest-file manifest.txt $i ;
done ;
```

```

        if [[ $(diff manifest.extract manifest.list | wc -l) -gt 0 ]] ;
        then
            echo manifest.txt is changed. Exporting... ;
            /usr/lpp/mmfs/bin/mmcloudgateway files export --container
$MANIFEST_CONTAINER manifest.txt ;
        fi ;
    done

    sleep $SCAN_FREQUENCY ;
    # The process continuously scanning subdirectories.
    # To increase scan frequency – decrease SCAN_FREQUENCY value in seconds.
done

```

The scan and export processes traverse the directory structure where applications #1 through #3 store data. The directory structure in this example looks as follows:

```

/gpfs/fs1/images
/gpfs/fs1/images/application1
/gpfs/fs1/images/application2
/gpfs/fs1/images/application3

```

The process scans directories and compares the list of files stored in those directories with information in a corresponding manifest file. If it detects that there were new files created, it exports the newly created files by initiating the **mmcloudgateway files export** command. The export process automatically updates the `manifest.txt` file. At the end of the scan cycle, if there were any changes detected and an export process updated a manifest file, the updated manifest file is also exported to a manifest cloud storage container. In this example use case, the manifest export is the mechanism that notifies the content consumer application that new content has been produced.

Figure 10 demonstrates the concept of the new content detection based on interpreting manifest file records.

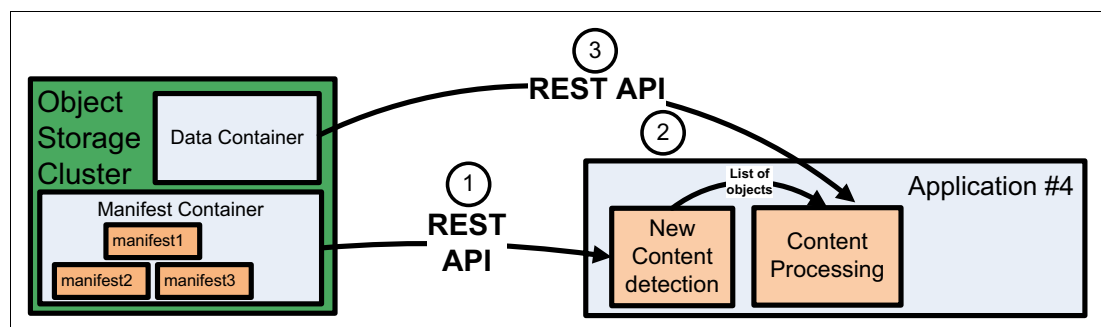


Figure 10 New Content Detection based on interpreting manifest file records

The New Content Detection process starts with extracting records from manifest files stored in the Manifest container. It is a two-step process:

1. First, enumerate all manifest files.
2. Then, extract records from all enumerated manifest files.

The New Content Detection module also maintains a previous state of manifest file records. After the extraction is done, the process compares the current and previous states of manifest records and initiates a data processing process for every new record detected. In this example, the simplistic data processing utility is the Linux viewer `eog`. The data processing utility is invoked every time a new data object is detected.

Example 2 includes a sample code fragment that illustrates the New Content Detection module function.

Example 2 Sample code fragment, New Content Detection module function

```
#!/bin/bash
#Finding all manifest.txt objects in $MANIFEST_CONTAINER
# and extracting manifest content
MANIFEST_CONTAINER='itso-manifest'
DATA_CONTAINER='itso-data'
CLOUD_URL='http://Object_Storage_Cluster_IP'
SCAN_FREQUENCY=5
while true
do
for manifest in `curl -s $CLOUD_URL/$MANIFEST_CONTAINER | awk -v RS='<Key>' '{
print $1}' | grep manifest.txt | awk -F'</Key>' '{ print $1 }'` ; do
    curl -s $CLOUD_URL/$MANIFEST_CONTAINER/$manifest ; done | sort | awk -F ','
'{ print $4}' | sort -u > manifest.current

if [[ $(diff manifest.current manifest.previous | wc -l) -gt 0 ]] ;
    then for j in $(diff manifest.previous manifest.current | grep '>' |
        awk '{ print $2 }') ;
        do echo displaying $j;
            curl -s $CLOUD_URL/$DATA_CONTAINER/$j > /tmp/$(basename $j) ;
            eog -n /tmp/$(basename $j);

            # “eog” is a simple image viewer. This is just an example of a workload.
            # Replace “eog” with your data processing utility.

done
        mv manifest.current manifest.previous ;
else
    echo Nothing to process;
    sleep $SCAN_FREQUENCY;
    # The process continuously scanning manifest files content.
# To increase scan frequency – decrease SCAN_FREQUENCY value in seconds.
fi
done
```

The REST API request is issued via the Linux **curl** command. Note that in the example there is no authentication used between the content consumer application and object storage. This approach might be useful for some publicly available data that does not require any access authorization. In most actual production environments, however, there is a layer of security between object storage and an external application.

After New Content Detection identifies all newly created manifest file entries it initiates some data processing action. The New Content Detection module provides a list of new objects to the Content Processing module. The list includes the exact location of the objects in the Object Storage Data Container. Figure 11 depicts this three-step process. The last step in the process is data extraction and processing. The Content Processing module initiates REST API GET request and receives object data content. The data content is the passed to the data processing utility in the eog example.

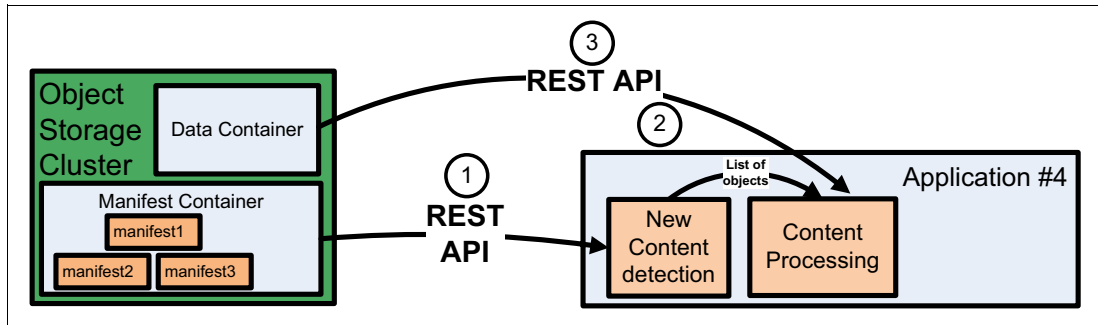


Figure 11 Data processing action, three-step process

In this example, there is only one data consuming application. However, this model is not limiting the number of applications interacting directly with the Object Storage. You can run any number of data consumer applications against the same data set.

The following examples demonstrates how to use a tag attribute in a manifest file to further enhance data consuming application. Assume that an implementation grew beyond scalability limitations of a single data consuming application and that you need to divide the workload between two instances of an application to meet performance requirements. In the example shown in Figure 12 on page 21, this use case is represented by Application #4 and Application #5. A simple workload distribution is established between the two instances of data consuming application based on even and odd numbers of files. When files are exported by Scan and Export, two new tags are employed—“even” and “odd”—so that when Scan and Export invokes the mmcloudgateway utility, it specifies the tag based on whether a file number in a queue was even or odd. One can imagine cases where the tag can be useful for allowing consumer applications to filter out data that is not needed.

As an example:

```
usr/lpp/mmfs/bin/mmcloudgateway files export --tag odd --container $DATA_CONTAINER
--export-metadata --manifest-file manifest.txt random_file1.jpg
usr/lpp/mmfs/bin/mmcloudgateway files export --tag even --container
$DATA_CONTAINER --export-metadata --manifest-file manifest.txt random_file2.jpg
```

All you need to do to split the workload between Application #4 and Application #5 is to change the logic of a New Content Detection of an application in a way that one instance will only process objects with even and the other instance will only process objects with odd tags.

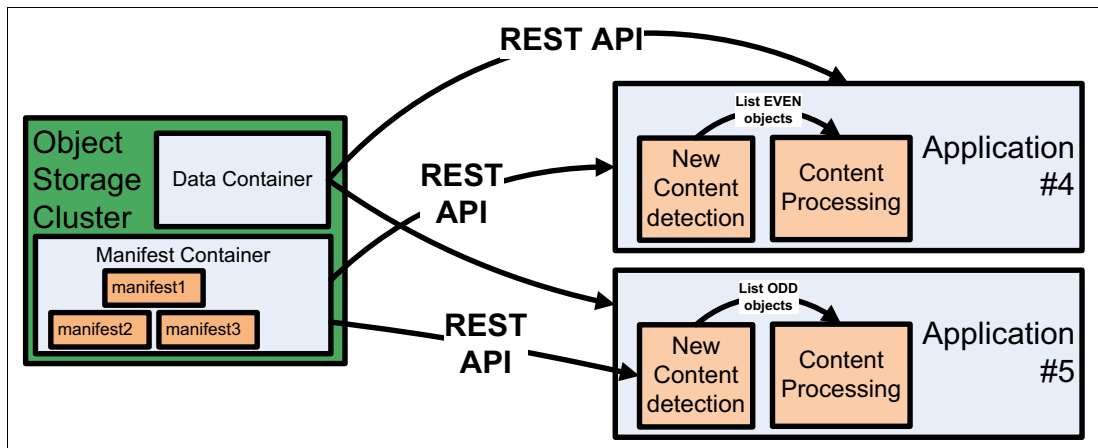


Figure 12 Dividing the workload between two instances to meet performance requirements

This method of using a tag attribute in metafile is a simple example of using the value of the attribute to divide the workload between two instances of a data consumer application. For additional information about the use of the tag attribute, see the [IBM Spectrum Scale](#) information in IBM Knowledge Center.

In the example the use of tag attribute is a form of passing parameters between Scan and Export New Content Detection modules. The interpretation of this parameter is passed via a tag attribute, and it defines behavior of a data consumer application instance. It further illustrates the data sharing feature and how it can be used.

IBM Spectrum Scale Cloud data sharing best practices

Cloud services have several configurable options, with most of them being common to both transparent cloud tiering and Cloud data sharing. In most cases, the default options are acceptable. Depending on the hardware environment, you can set the following tunable parameters using the `mmcloudgateway config set` command:

- ▶ `migrate-threadpool-size`

This parameter directly corresponds to the number of CPU cores on the cloud service node, which processes the incoming import and export requests (it used for migrates as well). Set this value equal to the number of CPU cores on a cloud service node. The default is 32. Cloud services rely heavily on multi-threading for performance. So setting it to less than the number of CPU cores usually has a significant negative impact on performance. Setting it higher is also not helpful, and in many cases might degrade performance.

- ▶ `slice-size`

This parameter directly corresponds to the block size that is set for the file system. The default is 256 KB. Increase the block size of the file system and this slice size value to 1 MB or higher for higher average file sizes and for improved performance.

Set these parameters using the `mmcloudgateway config set` command. For more details about this command, refer to the [IBM Spectrum Scale](#) information in the IBM Knowledge Center.

IBM Cloud Object Storage Cloud data sharing best practices

The preferred practices when using IBM Cloud Object Storage Cloud data sharing include:

- ▶ Synchronize the time on the accessor or accessors and the cloud services node group via NTP servers.
Make sure that the time and date for cloud services node group and the COS accessor or accessors are synchronized. If they are not, there is a possibility of credential errors when accessing the accessor via the IBM Spectrum Scale CES node.
- ▶ For cloud services to be able to create the vault, the user (whose access key is specified in the `mmcloudgateway account create` configuration command) should have the “Vault Provisioner” role assigned through the dsNet Manager UI.
- ▶ Make sure an access pool is created and available on the IBM Cloud Object Storage system, and it must have “IBM Cloud Object Storage” as the API type.
- ▶ Do not run highly demanding NFS/SMB services and cloud services on the same CES node.

- Use the vault setting configurations for IBM Cloud Object Storage and cloud services shown in Table 1.

Table 1 Preferred vault setting configurations for IBM Cloud Object Storage and cloud services

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	
WriteThreshold	As per IBM Cloud Object Storage documented configuration for production	
Alert Level	As per IBM Cloud Object Storage documented configuration for production	
SecureSlice Technology	Sharing with other Scale Clusters: disabled (because encryption is provided by Cloud data sharing) Sharing with other applications and services: enabled (because no encryption is provided by Cloud data sharing in this case)	When using Cloud data sharing, <i>turn off</i> encryption generation by the cloud service if there is an application in the cloud consuming the shared data. If an IBM Spectrum Scale cluster is sharing with another IBM Spectrum Scale cluster, <i>turn on</i> encryption by the Cloud data sharing service.
SecureSlice Algorithm	Not applicable because SecureSlice is disabled	
Versioning	Disabled	Cloud services have built-in versioning capability; thus, IBM Cloud Object Storage Versioning can be disabled.
DeleteRestricted	Yes/No	Gateway does not attempt to delete the vaults; thus, 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 IBM Cloud Object Storage with cloud services, go to the [IBM Spectrum Scale](#) information in IBM Knowledge Center.

Monitoring and lifecycle management

Cloud data sharing service shares the same Java virtual machine (VM) with transparent cloud tiering. Thus, status of the service can be monitored using the same **mmcloudgateway service status** command. This command shows whether the file system is configured correctly with the service and indicates the cloud account connectivity that the sharing service uses. You can also start the service if it is stopped using the **mmcloudgateway service start** command. For details, go to the [IBM Spectrum Scale](#) information in IBM Knowledge Center.

Each Cloud data sharing operation is audited and an entry is made in the `/var/MCStore/ras/audit/audit_events.json` file, which is a TCT audit file. Each entry in this

file is compliant with the cloud auditing industry standard (CADF). Example 3 shows a sample entry that can be found in the `/var/MCStore/ras/audit/audit_events.json` file for a given file sharing export operation. A similar entry is found for an import operation as well. Each entry maintains all attributes, such as initiator, target, action, and so on that is mandated by CADF, which completely describes the event.

Example 3 Sample CADF compliant entry for export operation

```
{ "typeURI": "http://schemas.dmtf.org/cloud/audit/1.0/event", "eventType": "activity",
  "id": "600b85e6-2103-4b69-9d01-2410df2d7298", "eventTime": "2017-01-13T11:46:26.942
  UTC", "action": "backup/export", "outcome": "success", "initiator": { "id": "9b101868-fec8
  -4687-a032-972976efb49c", "typeURI": "data/security/account/admin", "name": "root", "ho
  st": { "address": "10.114.96.64" } }, "target": { "id": "23ba8684-30c7-4de9-abfb-b09ddb37be
  e8", "typeURI": "service/storage/object", "name": "s3", "observer": { "id": "17bb34bf-8fe
  5-441b-920c-b8b7f32810b4", "typeURI": "service/storage/object", "name": "gpfs/mcstore"
  }, "measurements": [ { "result": "1", "metric": { "metricId": "c8b45022-9f5c-4665-b52e-cd50
  a648b83c", "unit": "file", "name": "The number of files that are successfully
  exported", "annotations": { "requested_files_list": "[/gpfs/export1.txt]" } } }, { "result"
  : "28", "metric": { "metricId": "df14a4f2-82e2-43d1-8288-172365d8d3e5", "unit": "byte", "n
  ame": "File size in
  bytes" } } ], "attachments": [ { "content": "accountname=proxy", "name": "cloudName", "conten
  tType": "text" } ] }
```

Cloud data sharing service also maintains metrics about the data being sent to or retrieved from the cloud. Monitoring the metrics is integrated with the Spectrum Scale Performance Monitoring tool. For more information about this tool, go to the [IBM Spectrum Scale](#) information in IBM Knowledge Center.

Given that tiering and sharing are both cloud operations, separate metrics are not available for each feature. For example, if an administrator is running migrate (tiering) as well as export (sharing) over set of files, cloud metrics would indicate the aggregate number of PUT requests sent to cloud as well as the combined number of bytes sent to cloud. Details about how to view these metrics using the CLI and the IBM Spectrum Scale GUI are provide in IBM Knowledge Center as well.

IBM Spectrum Control

Use IBM Spectrum Control to monitor IBM Spectrum Scale with Cloud data sharing and IBM Cloud Object Storage deployments. Using IBM Spectrum Control can give a single point of control in hybrid-cloud deployments using these two products.

IBM Spectrum Control can list cloud object pools, for example the mcstore pool in Figure 13, which lists the account and cloud containers.

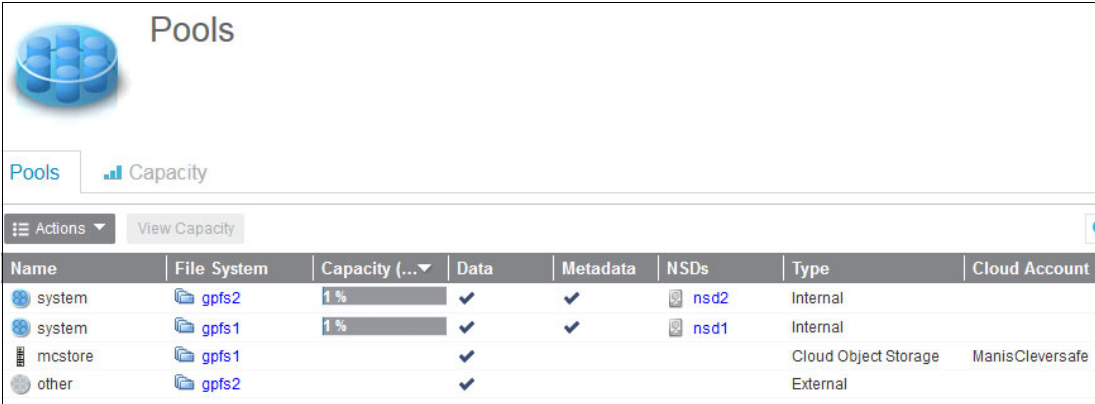


Figure 13 IBM Spectrum Control and cloud storage pools

In addition, the health of cloud services nodes is shown (Figure 14).

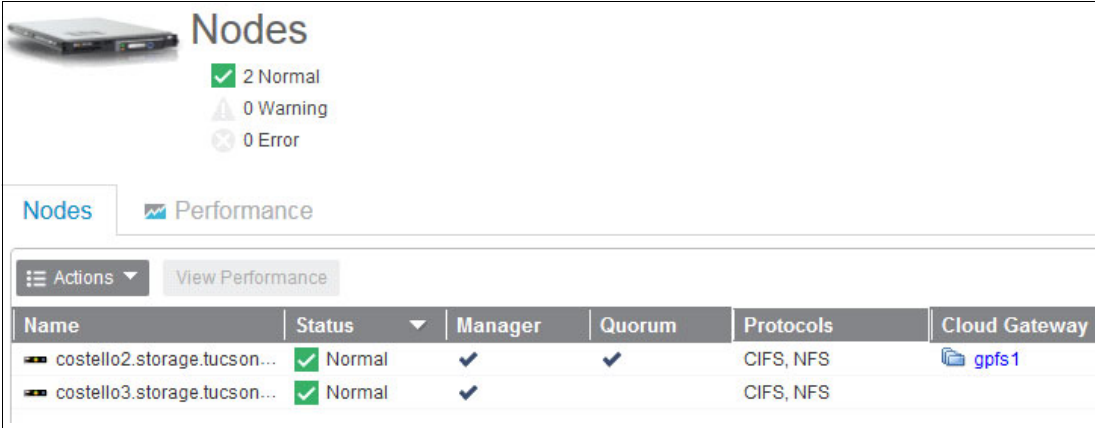


Figure 14 IBM Spectrum Control node status

IBM Spectrum Control also displays the status of IBM Cloud Object Storage, as shown in Figure 15, which provides a single convenient point to view system status.

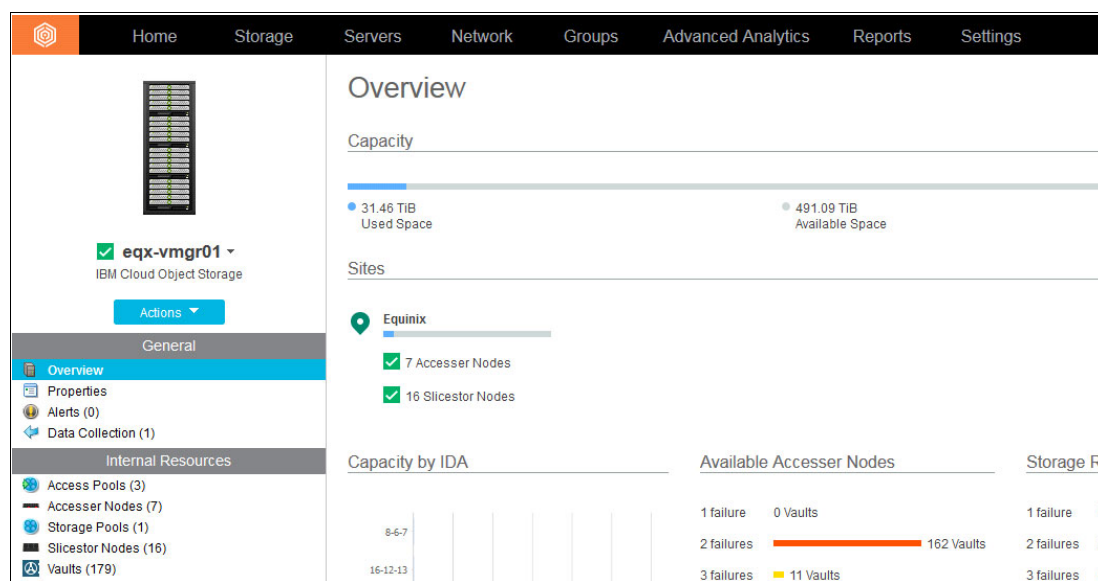


Figure 15 IBM Spectrum Control and IBM Cloud Object Storage

Summary

IBM Spectrum Scale Cloud data sharing allows IBM Spectrum Scale to share data with cloud object storage, thereby meeting data availability, site accessibility, and data scaling requirements. The Integrated Lifecycle Management Policy Engine allows IBM Spectrum Scale to be configured to share files with cloud object storage by policy periodically, allowing existing file applications to take advantage of cloud object storage data and allowing object storage based applications to take advantage of file system data.

Related publications

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

IBM Redbooks

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

- ▶ *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 Redbooks, 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 V4.2: Concepts, Planning, and Installation Guide*, GA76-0441
- ▶ *IBM Spectrum Scale V4.2: Administration and Programming Reference*, SA23-1452
- ▶ *IBM Spectrum Scale V4.2: Advanced Administration Guide*, SC23-7032
- ▶ *IBM Spectrum Scale V4.2: Data Management API Guide*, GA76-0442
- ▶ *IBM Spectrum Scale V4.2: Problem Determination Guide*, GA76-0443

Online resources

These websites are also relevant as further information sources:

- ▶ IBM Elastic Storage™ Server
<http://www.ibm.com/systems/storage/spectrum/ess>
- ▶ IBM DeepFlash 150
<http://www.ibm.com/systems/storage/flash/deepflash150/>
- ▶ IBM Cloud Object Storage
<https://www.ibm.com/cloud-computing/infrastructure/object-storage/>
- ▶ IBM Spectrum Scale
<http://www.ibm.com/systems/storage/spectrum/scale/index.html>
- ▶ IBM Spectrum Scale resources
<http://www.ibm.com/systems/storage/spectrum/scale/resources.html>
- ▶ IBM Spectrum Scale in IBM Knowledge Center
http://www.ibm.com/support/knowledgecenter/SSFKN/gpfs_welcome.html
- ▶ IBM Spectrum Scale Overview and Frequently Asked Questions
<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 International Technical Support Organization, Tucson Center.

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 currently is 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, 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.

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 IBM Spectrum Scale and is leading the ISDL development team responsible for productization, installation, serviceability, and configuration.

Arend Dittmer is a Solution Architect for Software Defined Infrastructure. He has over 15 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.

Alexander Safonov is a Storage Architect with IBM Systems Storage in Canada. He has over 25 years of experience in the computing industry, with the last 20 years spent working on Storage and UNIX solutions. He holds multiple product and industry certifications, including SNIA, IBM Spectrum Protect™, and IBM AIX®. Alex spends most of his client interaction time building solutions for data protection, data archiving, storage virtualization, automation and migration of data. He holds an M.S. Honors degree in Mechanical Engineering from the National Aviation University of Ukraine.

Ryan Marchese is a Staff Software Engineer, currently working as a developer on the transparent cloud tiering team. He began working at IBM in 2013 as a developer for the Cloud Systems Software team and has since been focused on IBM Storage. Ryan earned his master's degree in computer science from the University of Florida.

Rishika Kedia is a Senior Staff Software Engineer with IBM Systems Storage in India. She has over 10+ years of experience in various domains such as Systems Management, Virtualization and Cloud technologies. She holds a Bachelor's degree in Computer Science from Visvesvaraya Technological University. She is an invention plateau holder and has publications also in Virtualization and Cloud areas. She is currently a leader for CloudToolkit for Transparent Cloud Tiering and Data Sharing.

Stan Li is a software engineer working as a developer on the transparent cloud tiering team. He has been employed at IBM for 3 years and has worked on numerous cloud and storage related products. Stan earned his bachelor's degree in computer engineering from the University of Illinois at Urbana-Champaign.

Ranjith Rajagopalan Nair has over 14 years of industry experience (12+ in IBM) in various domains such as z/VM®, Linux on IBM z® Systems, Systems, Networking, and Storage. He has worked on various IBM Products such as IBM Systems Director VMControl™, IBM z/VM MAP agent, and Network Control. He is currently a developer working on cloud storage services, including cloud tiering and cloud sharing.

Larry Coyne is a Project Leader at the ITSO, 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.

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 from two to 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 at:

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>

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 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 Control™	Redpapers™
developerWorks®	IBM Spectrum Protect™	Redbooks (logo)  ®
GPFS™	IBM Spectrum Scale™	Systems Director VMControl™
IBM®	IBM z®	z/VM®
IBM Elastic Storage™	Redbooks®	
IBM Spectrum™	Redpaper™	

The following terms are trademarks of other companies:

Cleversafe, and C device are trademarks or registered trademarks of Cleversafe, Inc., an IBM Company.

SoftLayer, and SoftLayer device are trademarks or registered trademarks of SoftLayer, Inc., an IBM Company.

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds 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.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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



REDP-5419-00

ISBN 0738456004

Printed in U.S.A.

Get connected

