IBM Spectrum Storage

Spectrum Scale

Object, SwiftHLM and Openstack

Sandeep Patil
STSM
Spectrum Scale Dev
(Enablement)

Simon Lorenz
System Health Architect
till End of 2016
working for Object

IBM

The Audience?

- Who is using Spectrum-Scale Object Store?
- Who is using Cloud Object Storage (Cleversafe)?
- Who is using a different Object Store?

- Who is using an Openstack Cloud?
  - including the Cinder GPFS driver?
  - including the Manila GPFS driver?

Object, SwiftHLM and Openstack:

- Object integration advantages
- What‘s new since 4.2.1 (Object)
- 10 cool things you can do with Unified File & Object storage
- SwiftHLM - a middleware that enables Swift to work with tape
- Openstack integration
- What‘s new since 4.2.1 (Cinder & Manila)

Object Integration

Advantages…

Object Integration (since 4.1.1)

Spectrum Scale integration advantages:

- Automated Openstack Swift install and configuration
- Cluster wide Swift management (one cmd updates all nodes and ensures restarts if needed)
- High Availability and Health Monitoring
- GUI integration (Management & Monitoring)
- Unified File and Object Access
- Enhanced Storage Policies (Compression, Encryption, Unified File & Object)
- Automated Tiering of objects (based on Object heat and metadata)
- Keystone integration with AD/LDAP
- Secure communication between Swift services

Object Integration

What's new since 4.2.1

Object Integration (since 4.1.1)

What‘s new since 4.2.1 (major new functionalities):

- OpenStack Liberty Release (4.2.1)
- Execute Object cli commands from any Spectrum Scale Node (4.2.1)
- Enable and Disable Unified File & Object via CLI (4.2.1)
- Enable and Disable S3 Support via CLI (4.2.1)
- Object Encryption on Container Basis (4.2.1)
- Monitor external AD and LDAP Server for Object Authentication (4.2.1)
- External Keystone with SSL support (4.2.1)

Please see the Backup pages for links to the Knowledge Center

Object Integration (since 4.1.1)

What‘s new since 4.2.1 (major new functionalities):

- Best Practice Guide for multi-region Object deployment with HA Keystone (4.2.1)
- Created an Object and Keystone Problem Determination Guide (4.2.1)
- Secure communication between Proxy, Account, Container and Object Server (4.2.2)
- Enhanced PMSwift counters (more Performance Data) (4.2.2)
- Enable an existing! fileset for unified file and object access so the legacy file data can be accessed using object interface (4.2.2)
- Lot's of documentation and How To updates...

Please see the Backup pages for links to the Knowledge Center

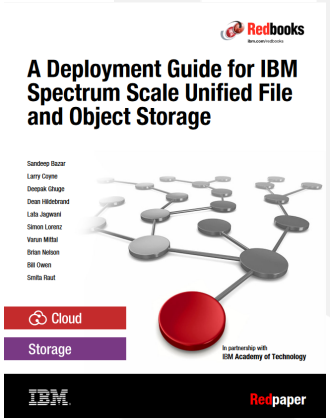
Object Integration

What's new in 4.2.3

Object Integration (what's new in 4.2.3)

- OpenStack Mitaka Release
- Object Store Consistency Tool (OSCT)
- Object data migration tool

- Redpaper update:
  A Deployment Guide for IBM Spectrum Scale Unified File and Object Storage
  http://www.redbooks.ibm.com/abstracts/redp5113.html?Open (already online!)



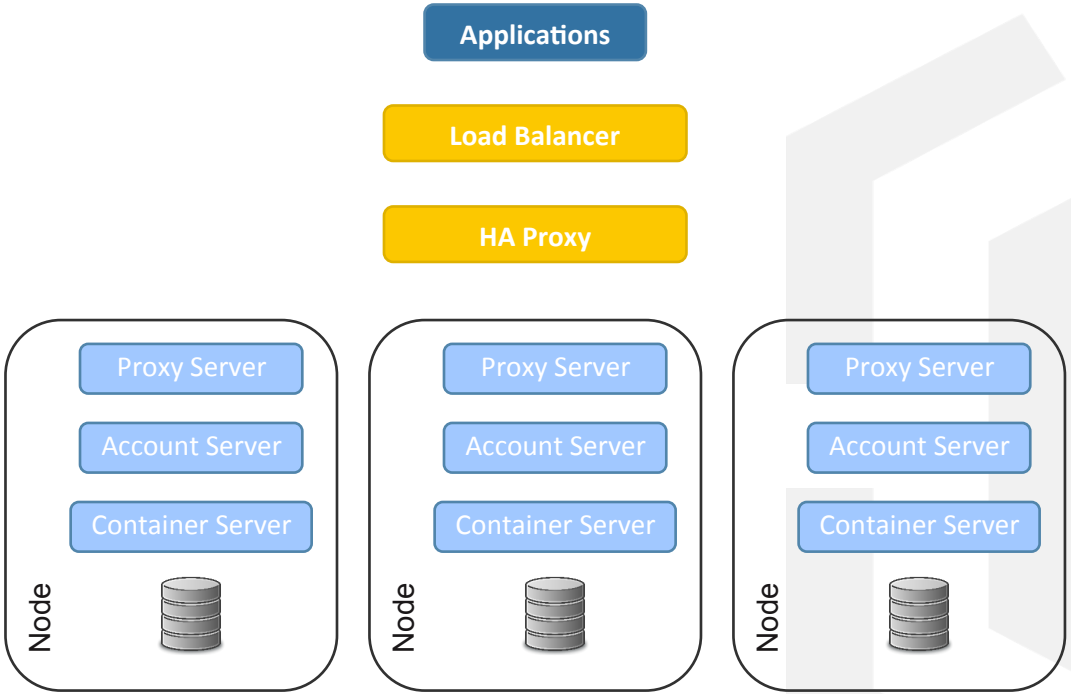
Object Integration

10 cool things you can do with Unified File & Object storage

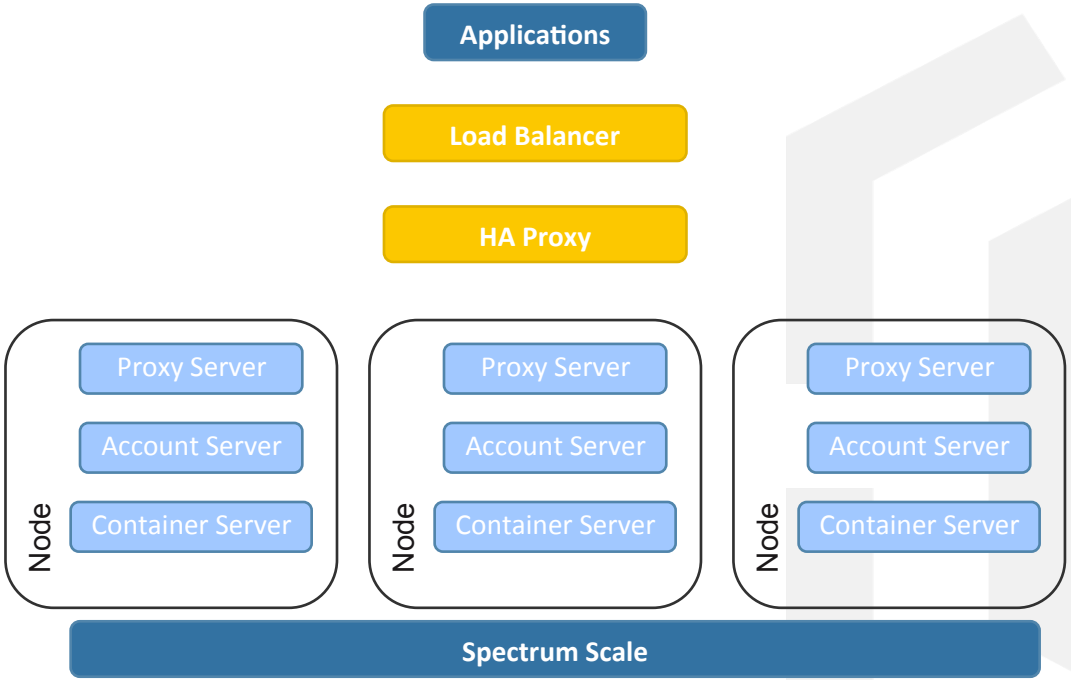
Thanks to our Colleague Smita Raut who summarized it in her blog:
https://www.linkedin.com/pulse/10-cool-things-you-can-do-ibm-spectrum-scale-unified-object-raut

Before we start some insides about Swift and its Integration

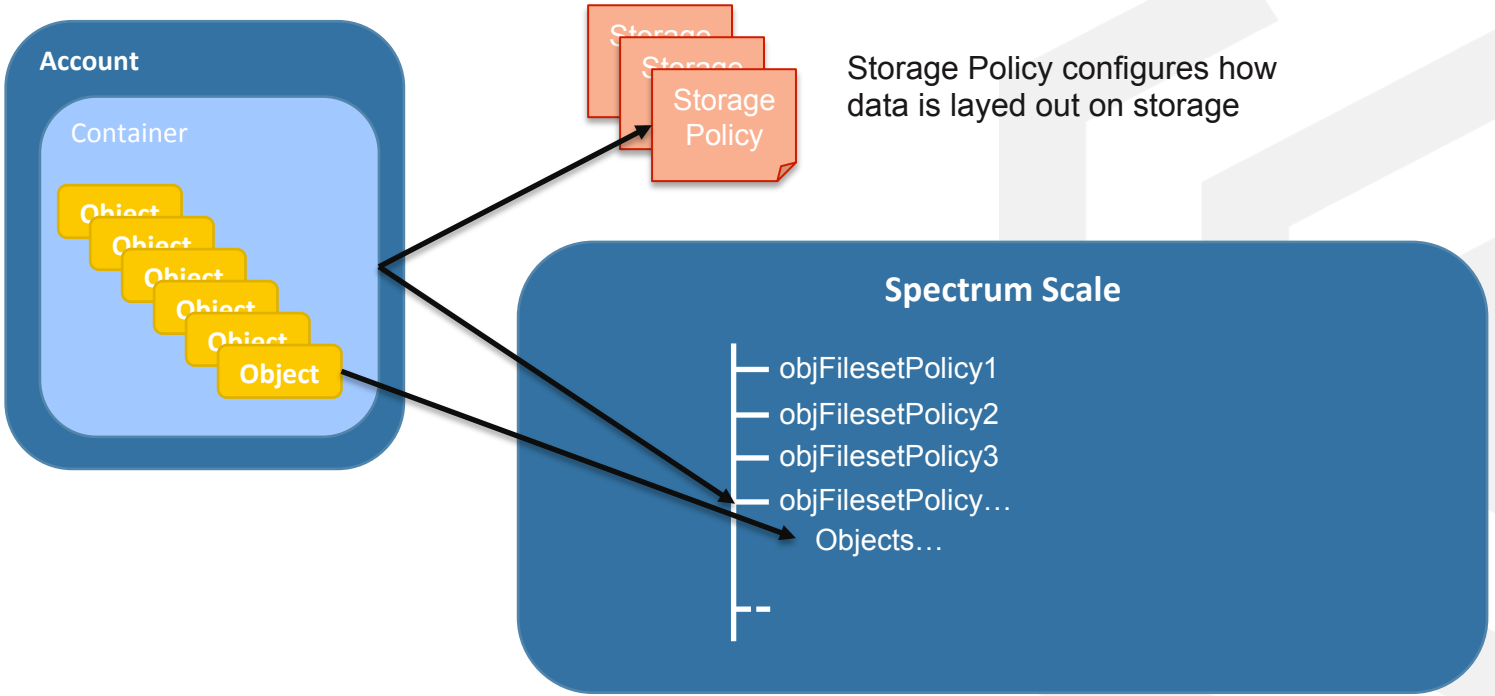
Typical Swift



Spectrum Scale and Swift Integration

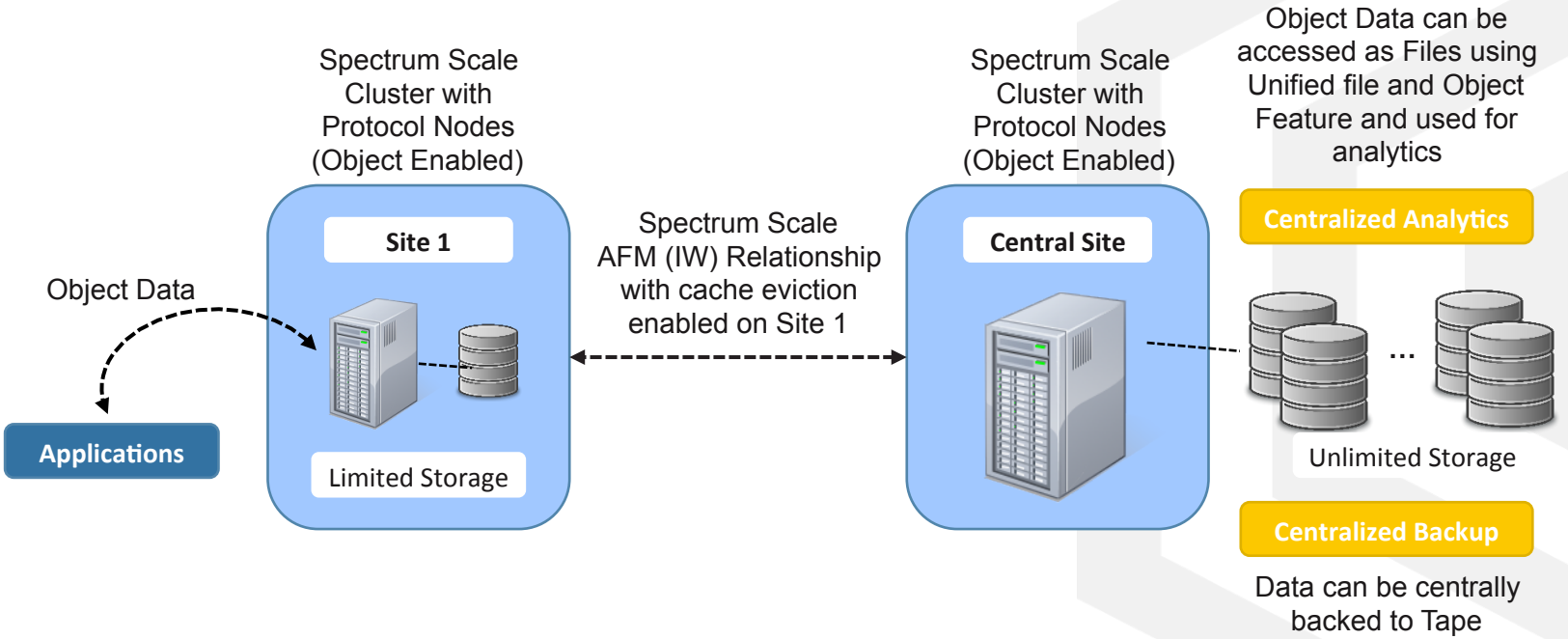


Spectrum Scale and Swift Integration



1. Thin-thick storage capacity site deployments for object data

- Successful deployment at Yahoo Japan (Caching between Japan and US sites)

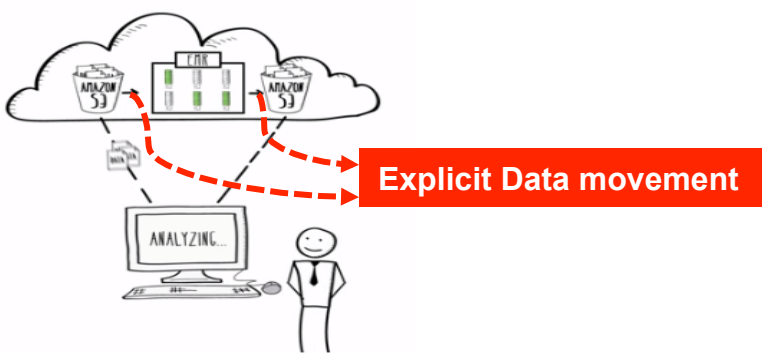


Reference: http://www.linkedin.com/pulse/ibm-spectrum-scale-makes-news-japan-smita-raut?trk=prof-post

2. In-place analytics over object data

- Unified File and Object access to same data makes it possible for analytics systems to access object data via HDFS interface

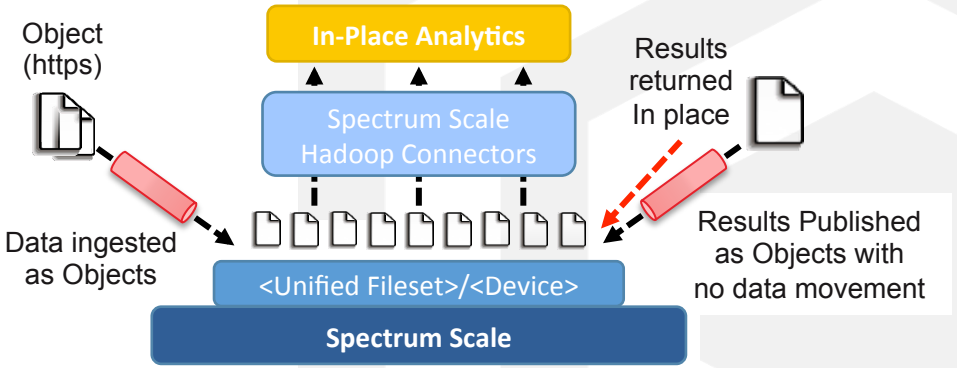
Analytics on Traditional Object Store



1. Data to be migrated from object store to dedicated analytic cluster.
2. Perform the analysis and copy results back to object store for publishing.

Reference: https://aws.amazon.com/elasticmapreduce/

Analytics With Unified File and Object Access



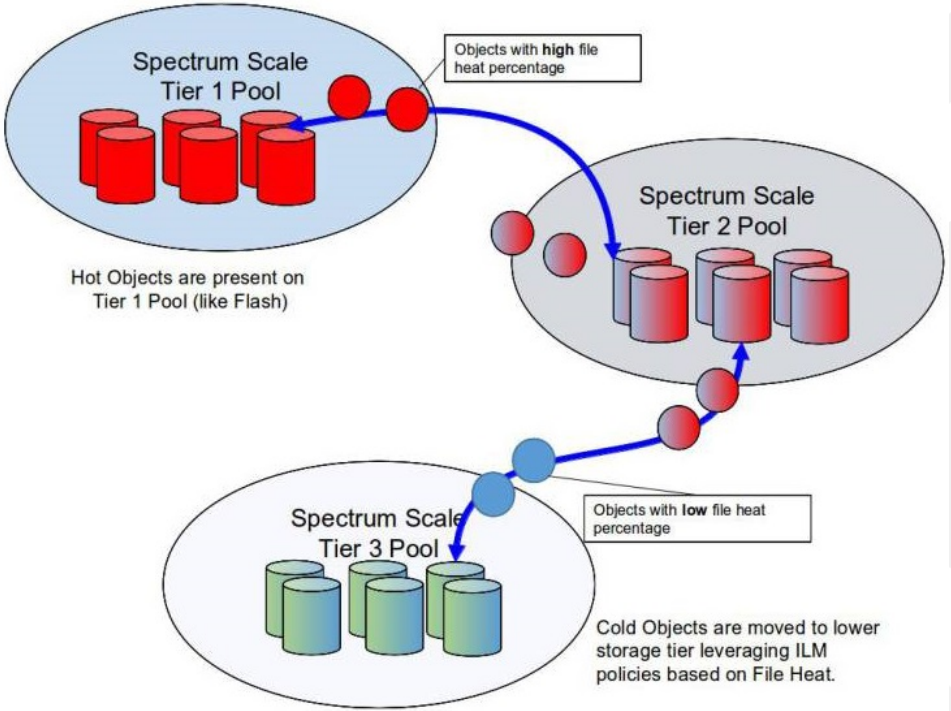
Object data available as "Files" on the same fileset. Analytics systems (Hadoop, Spark) can directly leverage this data analytics.

No data movement / In-Place immediate data analytics

Reference: https://www.openstack.org/videos/video/write-a-file-read-as-an-object

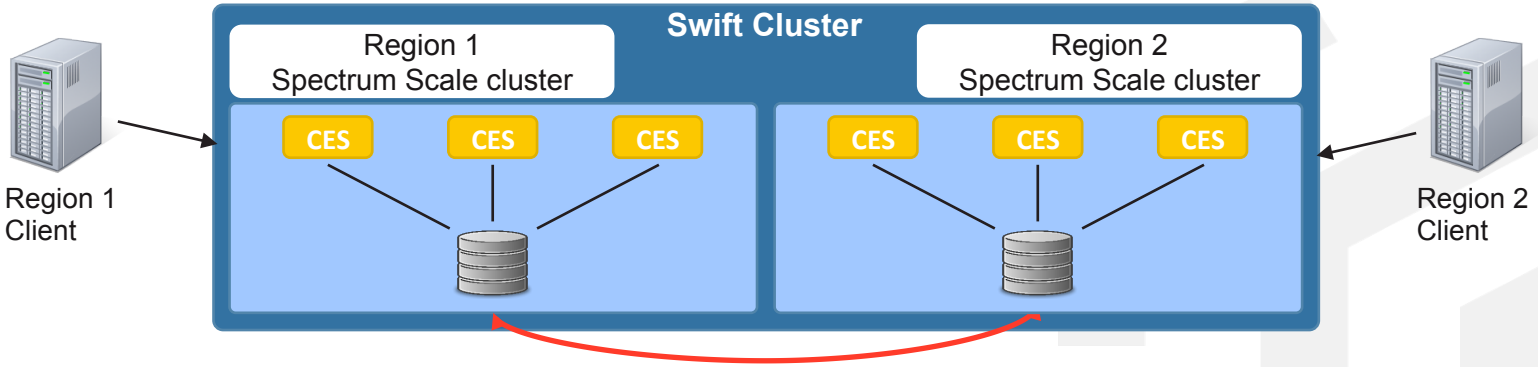
3. Automated tiering of hot objects to faster data pools

- Move infrequently accessed data to slower disks.
- Spectrum Scale provides heatmap tiering policies to free space for hot object data in higher performing storage pools.



Reference: https://smitaraut.wordpress.com/2016/06/16/hot-cakes-or-hot-objects-they-better-be-served-fast/

4. Multi-Region deployment with active-active configuration

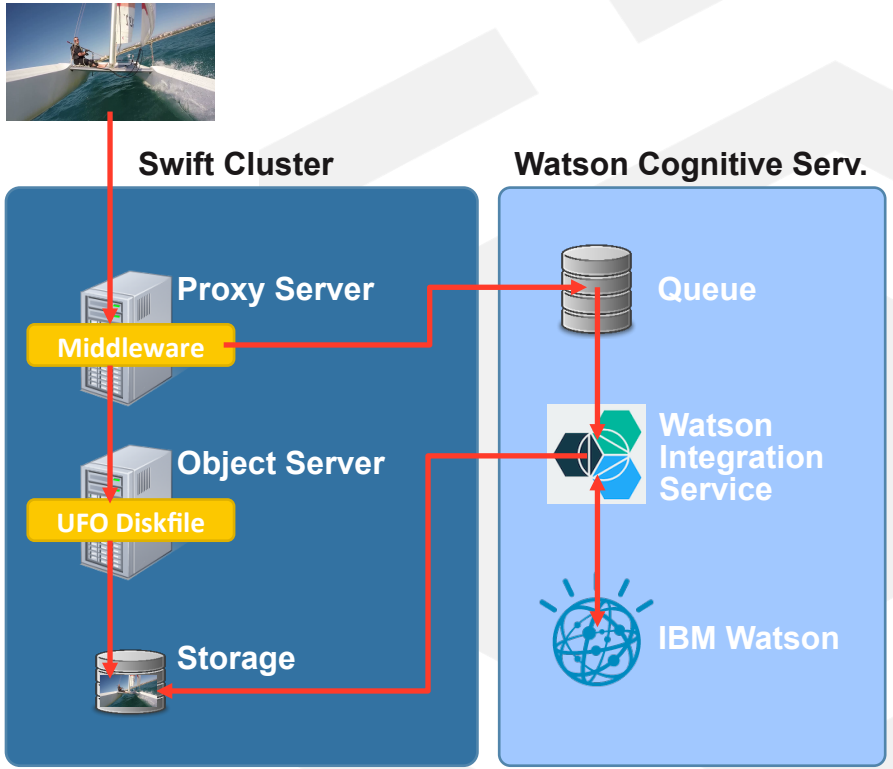


- Provide client access to a local replica of the data to reduce unacceptable high-latency network delays.
- Can be used as active-active disaster recovery configuration.

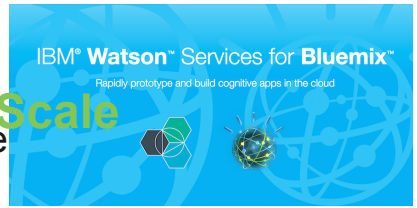
Reference:
https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1ins_multiregionoverview.htm

5. Writing your own pre/post object processing software (middleware)

- Object metadata is typically generated by Devices (e.g. Camera adding pic details) or is added by users or applications.
- An object – e.g. a picture - provides more details that could be viable as metadata. Accurately auto tag the object.
- Done by using



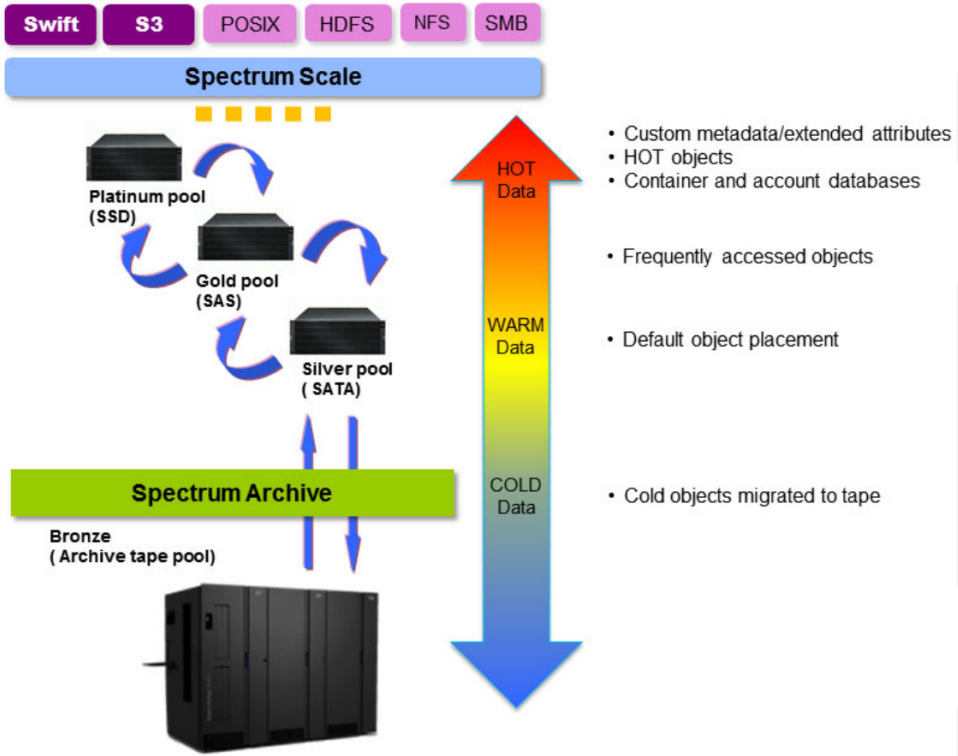
IBM Spectrum Scale Object Storage



Reference: http://www.slideshare.net/SmitaRaut/spectrum-scalecongnitive
Example is open sourced! Try it yourself! https://github.com/SpectrumScale/watson-spectrum-scale-object-integration

6. Moving your object data to tape

- The enormous amount and rapid pace of unstructured data generated poses significant costs and many Organizations mandate tape out even if the data is object.
- Spectrum Scale supports tape integration that works for object data by using storage tiers and Spectrum Archive.

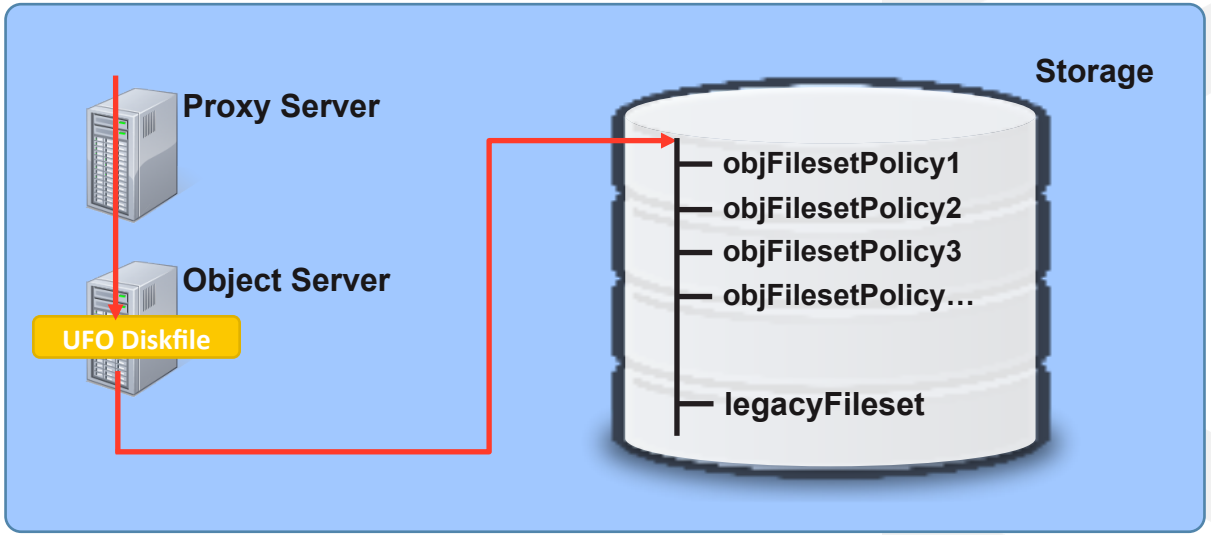


- *See* IceTier and SwiftHLM chapter: Swift high-latency media extensions which ease the Tape usage

Reference: http://www.redbooks.ibm.com/abstracts/redp5237.html?Open

7. Enabling object access on your existing files

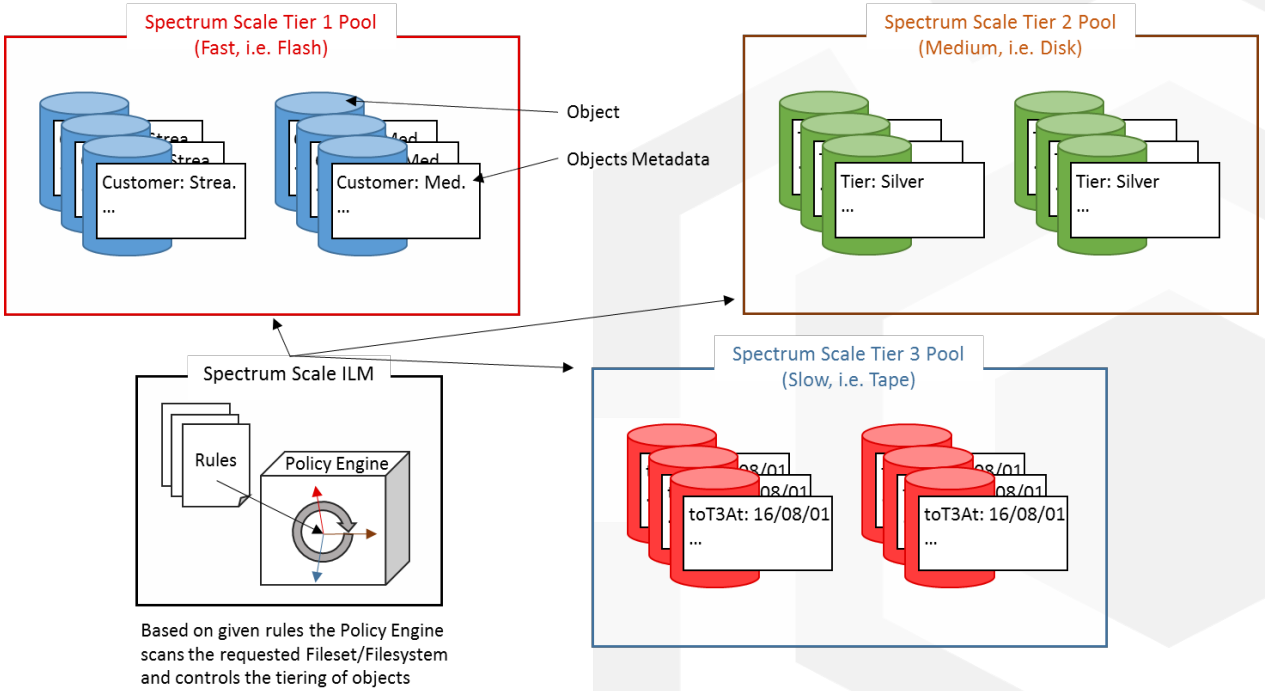
- 4.2.2 release Spectrum Scale provides a capability of enabling object access to existing file data. Use `mmobj file-access link-fileset`.



Reference:
https://www.ibm.com/support/knowledgecenter/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1adm_enablingobjectaccessonexistingfilesets.htm

8. Tiering based on object metadata

- Control tiering by setting object metadata.
  Even update metadata at a later time, to easily modify data placement.
- Add any key value pair as object metadata.
  Cause tiering based on:
  - values
  - the existence of a key
  - any combination of key value pairs.

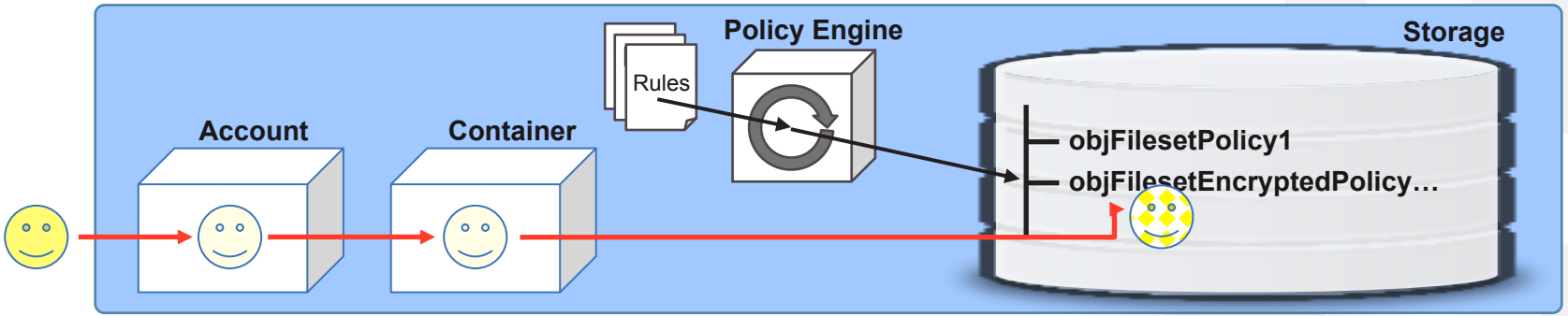


- Sample Usecase:
  Classification: Documents tagged as Confidential, are detected as such and will be placed in a certain tier.

Reference:
https://developer.ibm.com/storage/2016/09/08/are-you-heating-up-your-frozen-daiquiri-and-even-pay-for-it-serve-the-as-you-specified-hot-objects-fast-and-move-the-ones-you-want-to-freeze-into-the-fridge/

9. Encrypting your objects on disk for better security

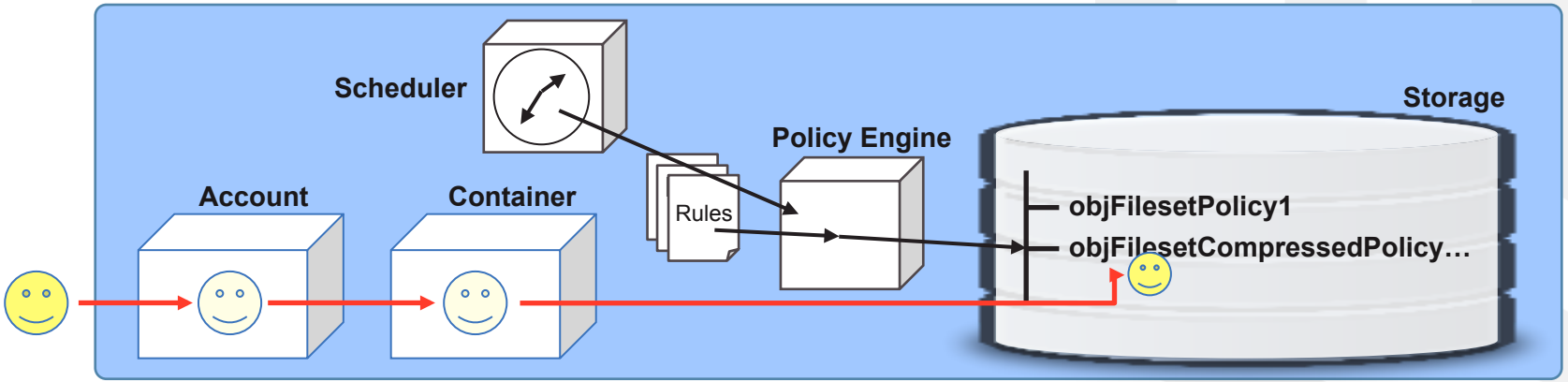
- Objects in Spectrum Scale can be encrypted using Spectrum Scale encryption and ILM policies.
- A new encryption enabled storage policy creates a new fileset.
- An encryption rule for the newly created fileset is applied to the policies.
- Any object that is uploaded into a container that is linked to the encryption enabled policy, will automatically and directly be stored encrypted.
- An object get request will cause a decryption of the data before it is send to the caller.



Reference: https://www.ibm.com/support/knowledgecenter/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1adm_storagepolicyencrypt.htm

10. Compressing your objects on disk for storage space optimization

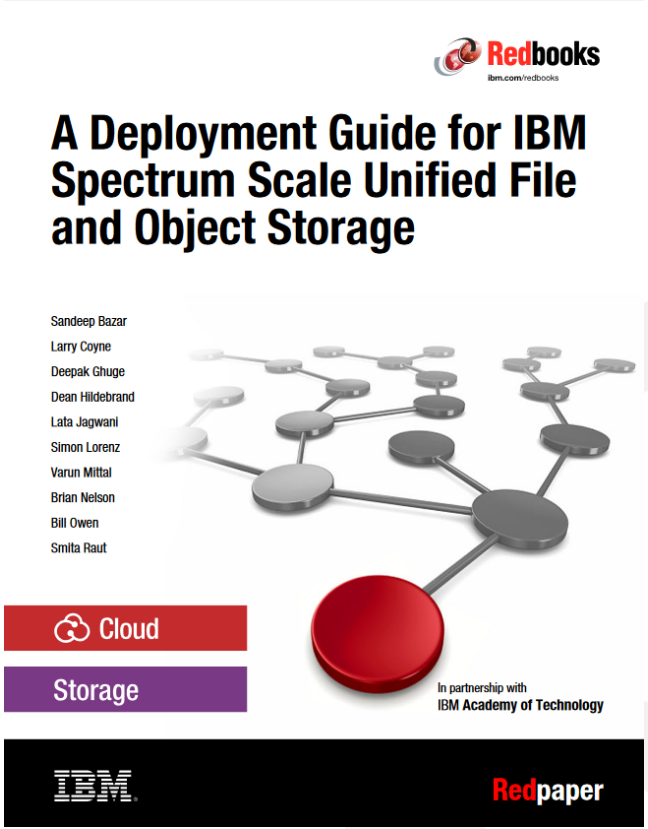
- Objects in Spectrum Scale can compressed using Spectrum Scale compression and ILM policies.
- A new compression enabled storage policy creates a new fileset.
- A migration compression rule for the newly created fileset is applied base on a given schedule.
- Any object that is uploaded into a container that is linked to the compression enabled policy, will be compressed when the given schedule is hit.



Reference: https://www.ibm.com/support/knowledgecenter/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1adm_storagepolicycomp.htm

IBM Redpaper:

- > 900 downloads in 2 weeks since updated
- > 12 000 downloads since the 1st Version was published



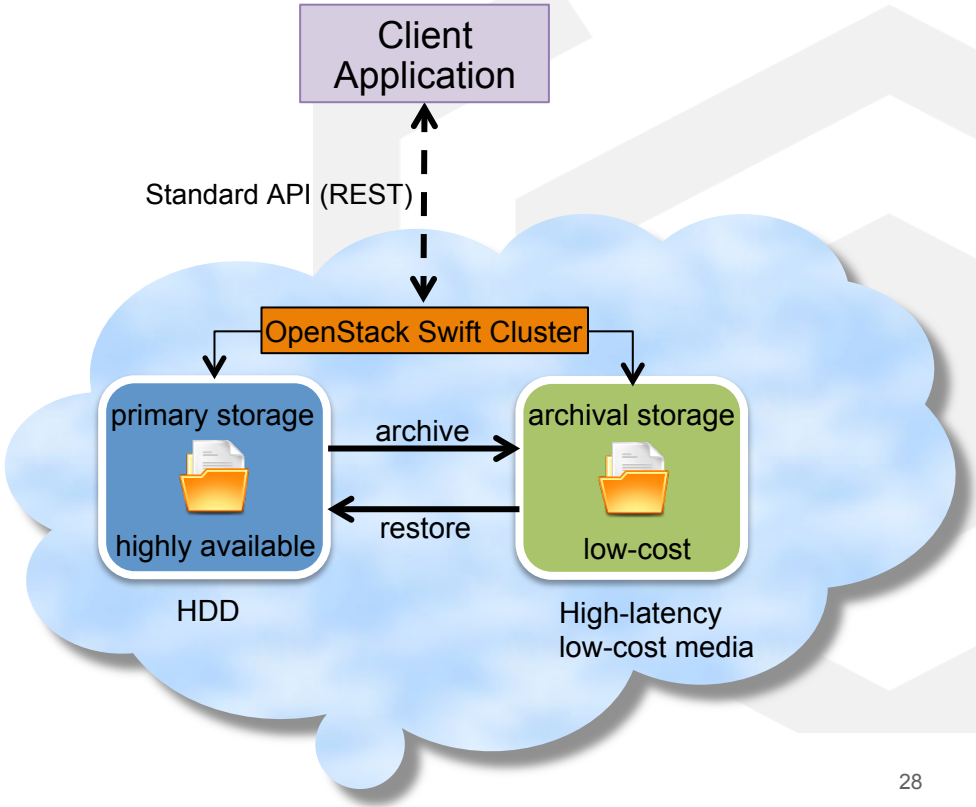
http://www.redbooks.ibm.com/abstracts/redp5113.html?Open

IceTier and SwiftHLM

a middleware that enables Swift to work with tape

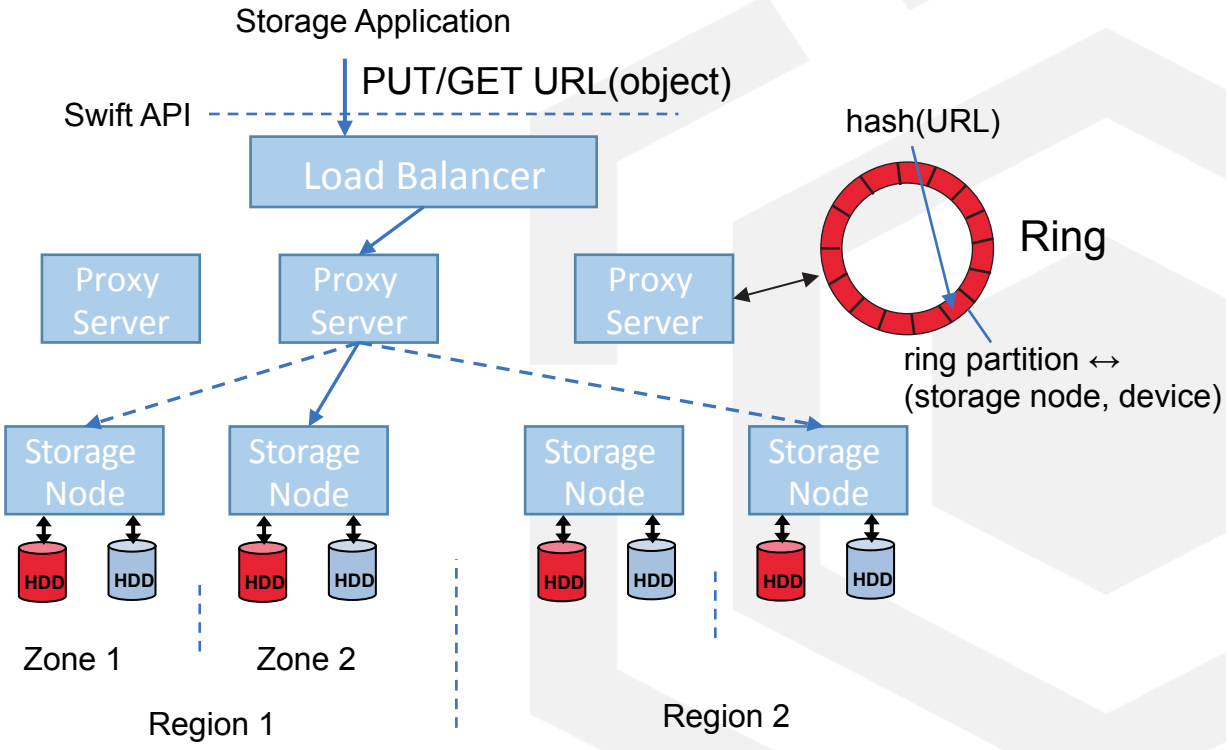
IceTier: OpenStack Swift object storage on tape (or other high-latency media)

- **Augment cloud object storage with a low-cost, cold storage tier**
  - Tape, optical, MAID
  - Archive/backup use cases
- **Reduced cost**
  - E.g. tape up to 6x cheaper than disk (current HW/media specs)
  - Future projections in favor of tape
- **Reduced availability**
  - Minutes, 10s of minutes, or hours (depending on use case and SLA)



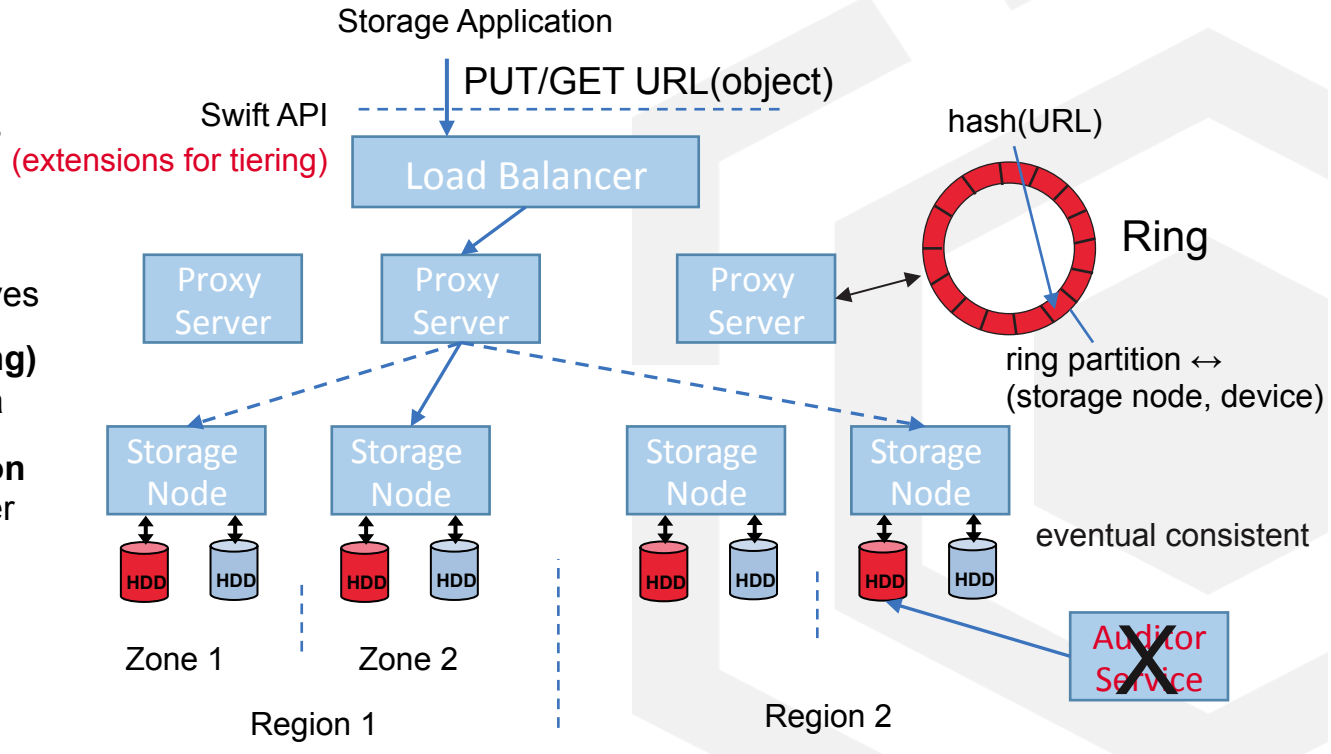
OpenStack Swift object storage on disk

- **Open Source**
  - Increasing adoption
  - Client side solutions
- **Simple REST interface**
  - Swift native
  - Amazon S3
- **Extreme Scalability**
  - Hash-based Data Rings:
    - Hash(URL) -> storage nodes, devices
    - Not storing state (info) per object
    - One ring per storage policy (replication scheme, device set/type)
- **High Availability/Durability**
  - Replication
  - Erasure coding
  - Regular data health checks (auditing)



OpenStack Swift object storage on disk

- **Extend API**
  - Enable explicit archiving operations
  - (**Bulk**) migrate/recall/status
    - Avoid timeouts (return HTTP/1.1 412 Precondition Failed)
    - Cost-efficient use of drives
- **Modify health check (auditing)**
  - to not often recall tape data
- **Customize object distribution**
  - Avoid container spread over too many tapes through collocation
  - Lowers number of mounts and drives usage
  - Low cost => #drives << #tapes

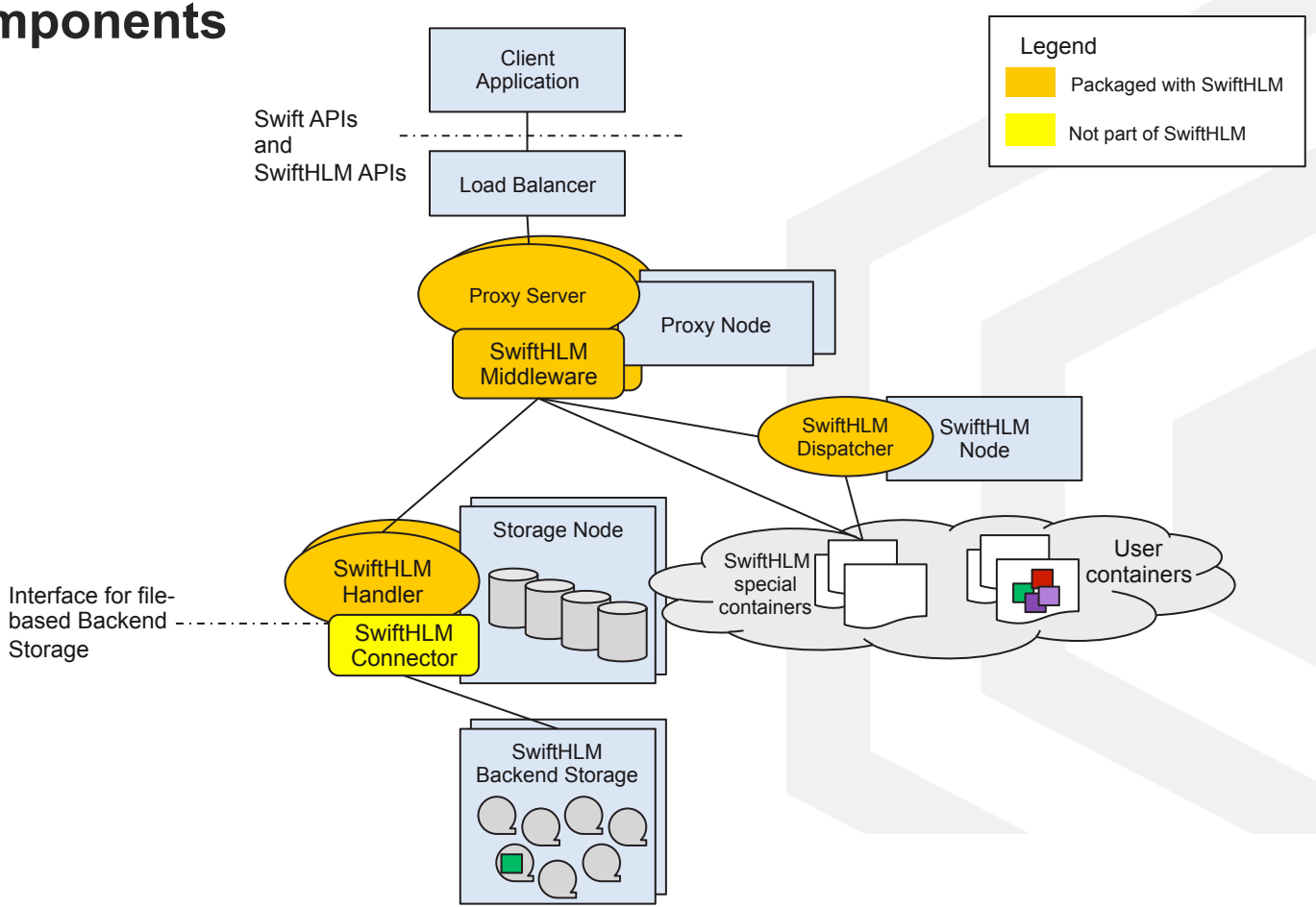


SwiftHLM: Swift high-latency media extensions

- SwiftHLM consists of
  - **SwiftHLM Middleware** (Proxy nodes)
    - Proxy middleware exposing the enhanced API
  - **SwiftHLM Dispatcher** (Swift node)
    - Background daemon creates a list of objects, identify the Storage Node for each object, and dispatches asynchronously to the appropriate Swift Storage Node
  - **SwiftHLM Handler** (Storage nodes)
    - provides/invokes generic interface toward SwiftHLM backend storage. Maps objects to files and submits the mapped list to the backend (via Backend Connector)

- SwiftHLM requires a **backend-specific Connector module**
  - Supplied by the vendor of the backend software/hardware
    - IBM Spectrum Archive EE
    - IBM Spectrum Protect
    - others
  - Note that the Connector is not part of the SwiftHLM packaging

SwiftHLM Components



SwiftHLM: Usage examples

The following four methods are provided for placement control and status monitoring:

- migrate - move an object (or all objects in a container) to HLM tier
- recall - restore an object (or all objects in a container) from HLM tier
- status - query the current placement of object (or objects in a container)
- requests - checks if any pending operation exists for the specified object (or container)

The output of the status command can result in one of three states for the object:

- Resident - The object is only on disk (takes up disk space)
- Premigrated - The object is on disk and on tape (takes up disk space)
- Migrated - The object is only on tape (no used disk space)

Example of a requests command result:

["There are no pending or failed SwiftHLM requests."]
["20170316172233.418--migrate--AUTH_f3013cd87a264a8b87a44b831bfc7579--Redpaper--0--test_object_4--pending"]

SwiftHLM: End to end example

Upload a file as an object:

curl -X PUT -H "X-Storage-Token: $TOKEN" -T movie1.mpg "http://tora-ces:8080/v1/$ACCT/Redpaper/test_object_4"

Use the **status** command to check the state of test_object_4:

curl -X GET -H "X-Storage-Token: $TOKEN" "http://tora-ces:8080/hlm/v1/status/$ACCT/Redpaper" | python -m json.tool

{
"/AUTH_f30.../Redpaper/test_object_1": "resident",
"/AUTH_f30.../Redpaper/test_object_2": "resident",
"/AUTH_f30.../Redpaper/test_object_3": "resident",
"/AUTH_f30.../Redpaper/test_object_4": "**resident**"
}

Migrate the object test_object_4 using the migrate command:

curl -X POST -H “X-Storage-Token: $TOKEN” “http://tora-ces:8080/hlm/v1/**migrate**/$ACCT/Redpaper/test_object_4”

Accepted migrate request.

Use the **status** command to check the state of the object test_object_4:

curl -X GET -H "X-Storage-Token: $TOKEN" "http://tora-ces:8080/hlm/v1/status/$ACCT/Redpaper" | python -m json.tool

{
"/AUTH_f30.../Redpaper/test_object_1": "resident",
"/AUTH_f30.../Redpaper/test_object_2": "resident",
"/AUTH_f30.../Redpaper/test_object_3": "resident",
"/AUTH_f30.../Redpaper/test_object_4": "**migrated**"
}

Recall test_object_4 to Spectrum Scale Object storage using the recall command:

curl -X POST -H “X-Storage-Token: $TOKEN” “http://tora-ces:8080/hlm/v1/**recall**/$ACCT/Redpaper/test_object_4”

Accepted recall request.

SwiftHLM: End to end example

Use the **status** command to check the state of the recalled object test_object_4:

curl -X GET -H "X-Storage-Token: $TOKEN"
"http://tora-ces:8080/hlm/v1/status/$ACCT/Redpaper" | python -m json.tool

{
"/AUTH_f30.../Redpaper/test_object_1": "resident",
"/AUTH_f30.../Redpaper/test_object_2": "resident",
"/AUTH_f30.../Redpaper/test_object_3": "resident",
"/AUTH_f30.../Redpaper/test_object_4": "**premigrated**"
}

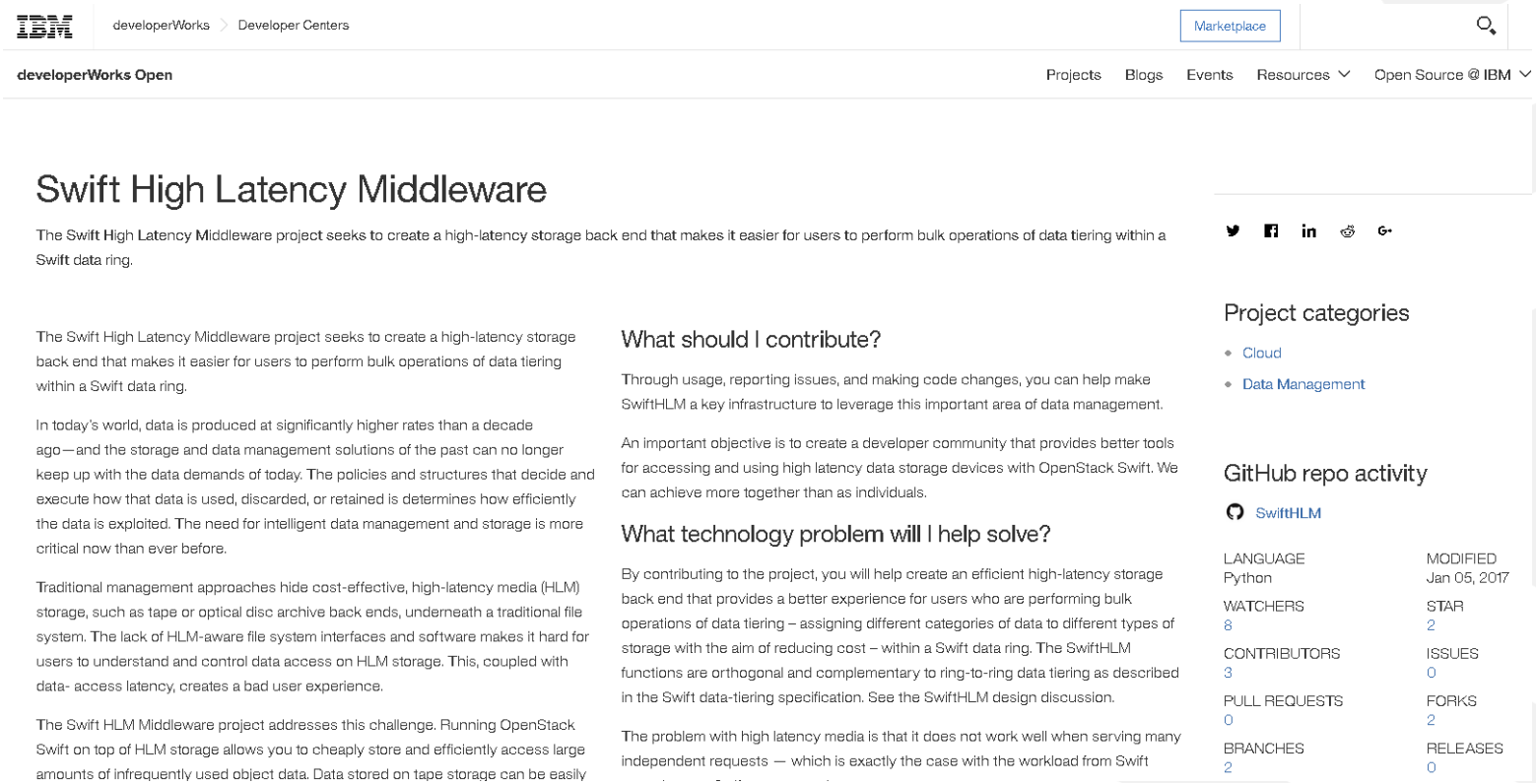
The following curl command **downloads** the recalled object as a file:

curl -X GET -H "X-Storage-Token: $TOKEN" -o movie1.mpg
"http://tora-ces:8080/v1/$ACCT/Redpaper/test_object_4"

If you try to download an object that is still in the migrated state you will get the following precondition failure:

HTTP/1.1 412 Precondition Failed
Content-Length: 118
Content-Type: text/plain
X-Trans-Id: tx783ae4a6288a4cfe872f0-0058cb0a08
Date: Thu, 16 Mar 2017 21:56:25 GMT
Object /AUTH_f3013cd87a264a8b87a44b831bfc7579/Redpaper/test_object_4 needs to be RECALL-ed before it can be accessed.

SwiftHLM on developerWorks – Your source for latest information



IBM developerWorks > Developer Centers

Marketplace

developerWorks Open

Projects Blogs Events Resources Open Source @ IBM

Swift High Latency Middleware

The Swift High Latency Middleware project seeks to create a high-latency storage back end that makes it easier for users to perform bulk operations of data tiering within a Swift data ring.

The Swift High Latency Middleware project seeks to create a high-latency storage back end that makes it easier for users to perform bulk operations of data tiering within a Swift data ring.

In today's world, data is produced at significantly higher rates than a decade ago—and the storage and data management solutions of the past can no longer keep up with the data demands of today. The policies and structures that decide and execute how that data is used, discarded, or retained is determines how efficiently the data is exploited. The need for intelligent data management and storage is more critical now than ever before.

Traditional management approaches hide cost-effective, high-latency media (HLM) storage, such as tape or optical disc archive back ends, underneath a traditional file system. The lack of HLM-aware file system interfaces and software makes it hard for users to understand and control data access on HLM storage. This, coupled with data- access latency, creates a bad user experience.

The Swift HLM Middleware project addresses this challenge. Running OpenStack Swift on top of HLM storage allows you to cheaply store and efficiently access large amounts of infrequently used object data. Data stored on tape storage can be easily

What should I contribute?

Through usage, reporting issues, and making code changes, you can help make SwiftHLM a key infrastructure to leverage this important area of data management.

An important objective is to create a developer community that provides better tools for accessing and using high latency data storage devices with OpenStack Swift. We can achieve more together than as individuals.

What technology problem will I help solve?

By contributing to the project, you will help create an efficient high-latency storage back end that provides a better experience for users who are performing bulk operations of data tiering – assigning different categories of data to different types of storage with the aim of reducing cost – within a Swift data ring. The SwiftHLM functions are orthogonal and complementary to ring-to-ring data tiering as described in the Swift data-tiering specification. See the SwiftHLM design discussion.

The problem with high latency media is that it does not work well when serving many independent requests — which is exactly the case with the workload from Swift

Project categories

- Cloud
- Data Management

GitHub repo activity

SwiftHLM

LANGUAGE Python	MODIFIED Jan 05, 2017
WATCHERS 8	STAR 2
CONTRIBUTORS 3	ISSUES 0
PULL REQUESTS 0	FORKS 2
BRANCHES 2	RELEASES 0

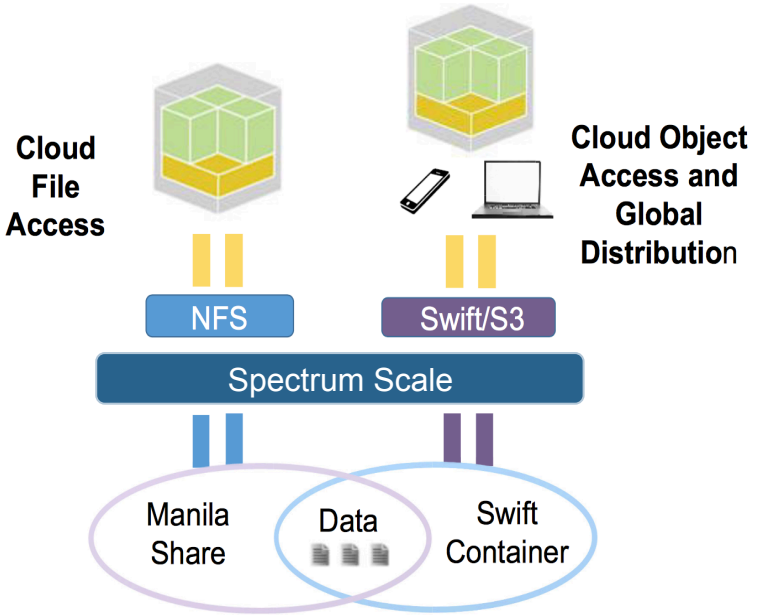
https://developer.ibm.com/open/openprojects/swift-high-latency-middleware/

Openstack Integration

Nova, Glance, Cinder, Manila

Shared File Systems and OpenStack Storage

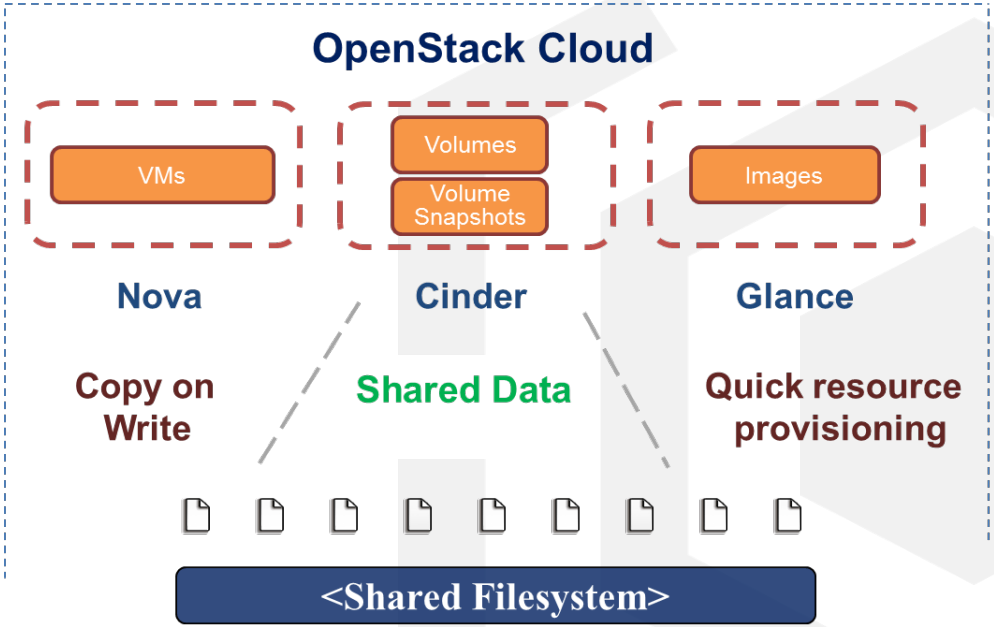
- Common data plane for OpenStack storage
- Provides enterprise features like:
  - Snapshots & backups
  - Automatic tiering/migration of data across storage pools
  - Local & multi-site Replication
- Enables Unique Features for OpenStack services
  - Nova live migration
  - Efficient data sharing between Glance, Cinder, Nova using COW



Nova/Glance/Cinder Integration

Shared Filesystem can be used by:

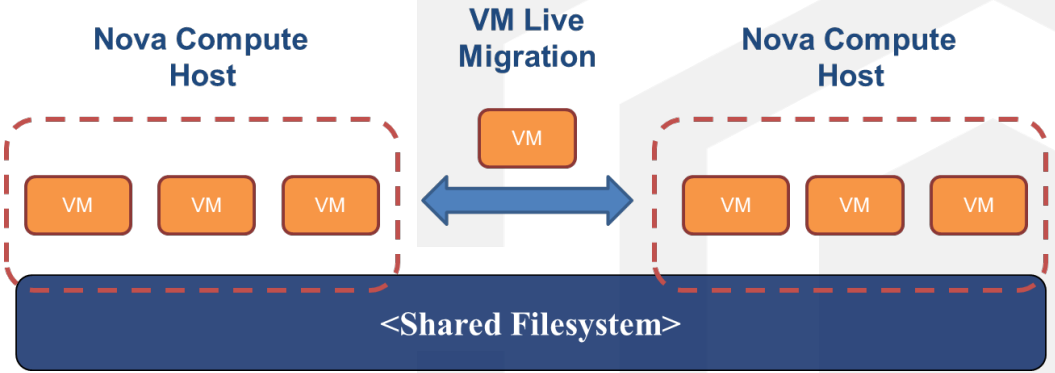
- Nova: by VM instances for ephemeral storage
- Glance: store glance images
- Cinder: store volumes (can be bootable volumes) that nova uses or persistent data volumes used by workloads running in Nova VMs



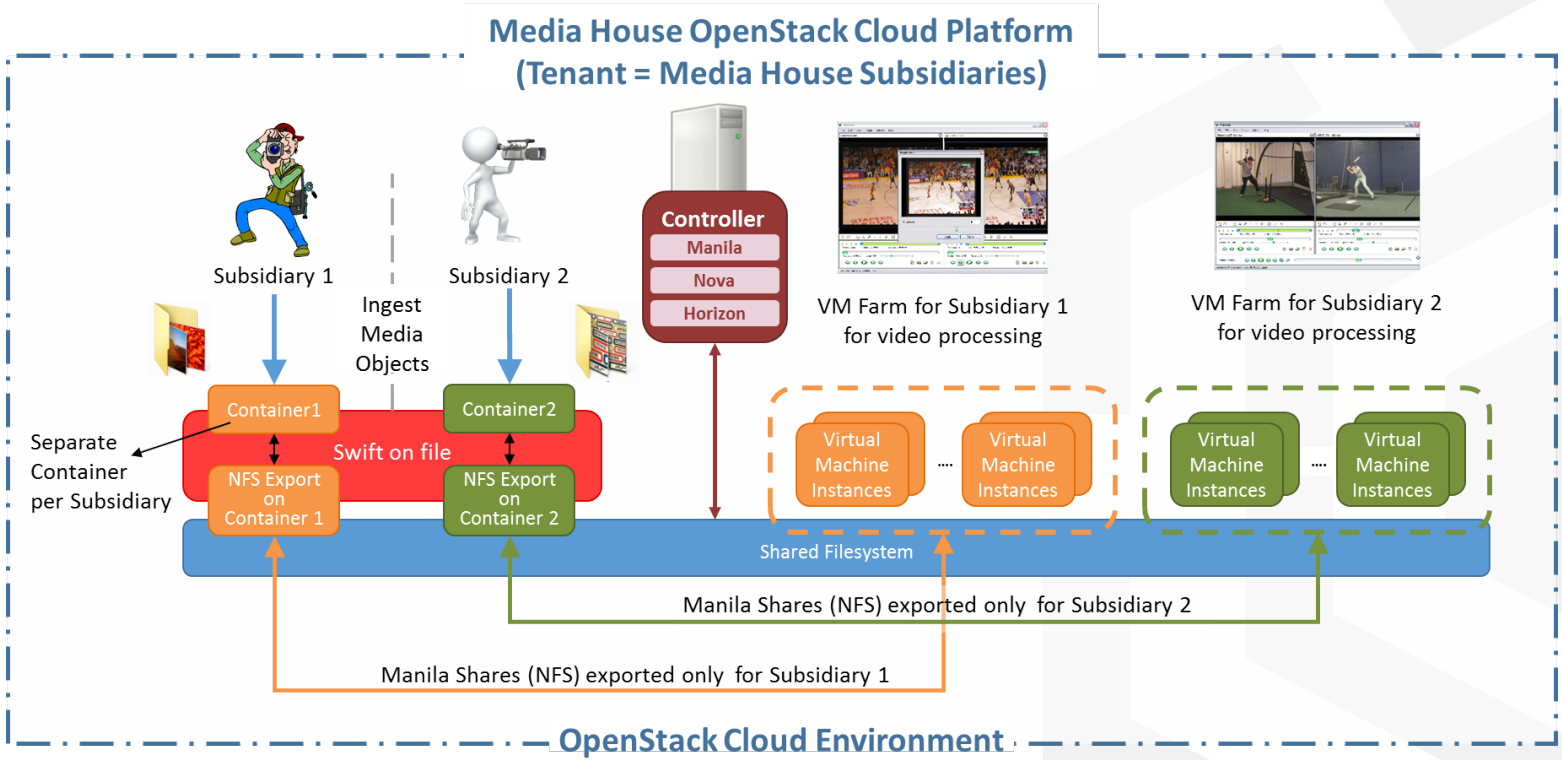
Nova/Glance/Cinder Integration

A shared storage backend for nova also enables live migration for instances, between the compute hosts.

As the filesystem is clustered among the compute hosts, the VM storage is available on the other nodes and VMs can be migrated seamlessly.

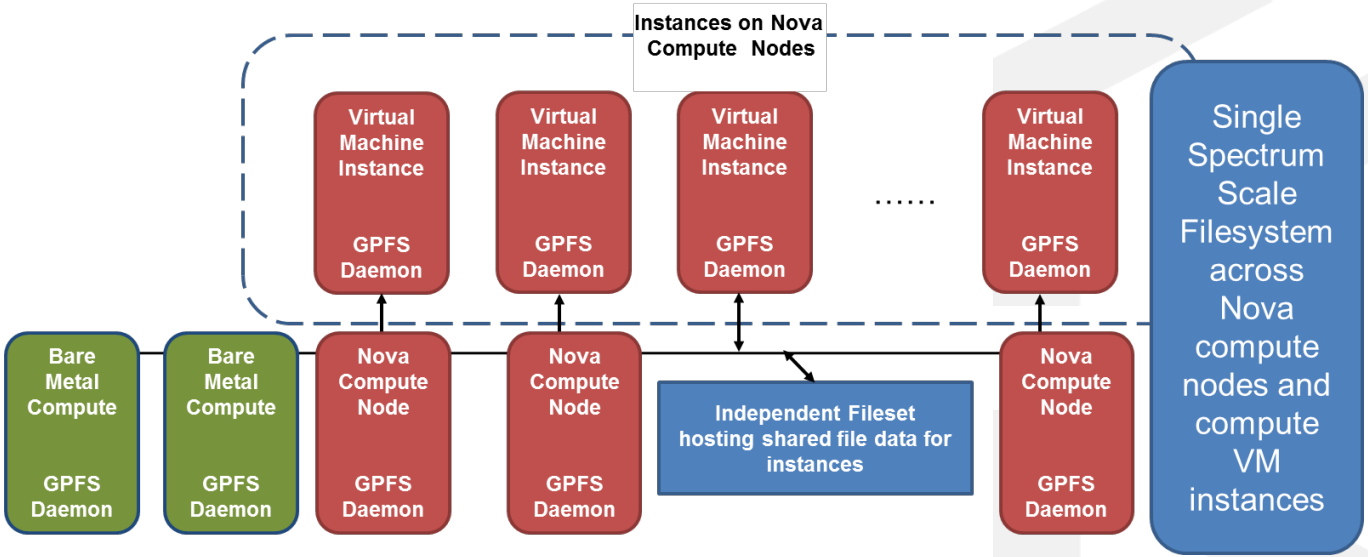


Swift/Manila Integration



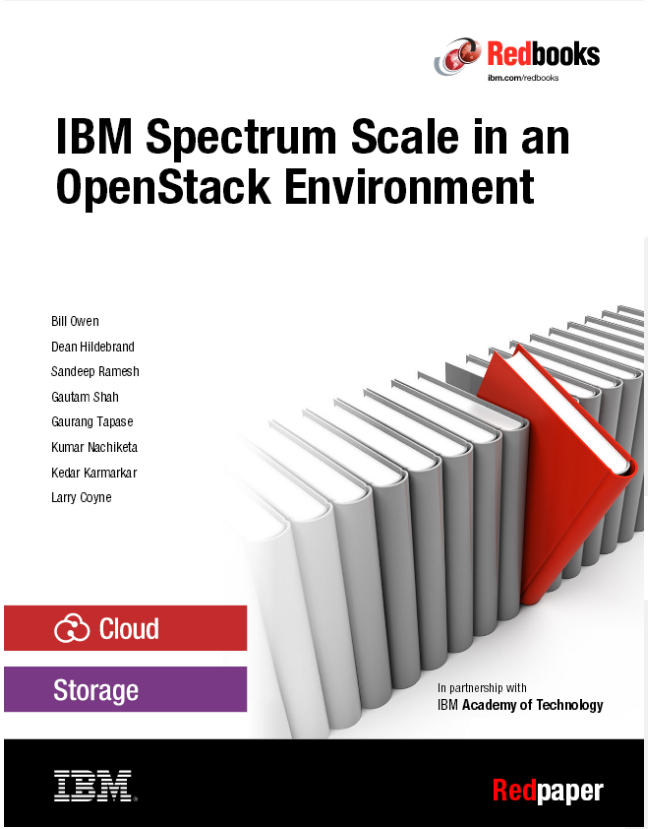
Reference: https://www.openstack.org/videos/video/amalgamating-manilla-and-swift-for-unified-data-sharing-across-instances

Share data between the workloads running in the compute VMs, with those running on Physical nodes.



Running clustered Filesystem in Nova instances to share data with workloads running on Physical nodes

IBM Redpaper:



https://www.redbooks.ibm.com/Abstracts/redp5331.html?Open

Cinder & Manila Integration

What‘s new since 4.2.1, ongoing efforts/future goals

Cinder Integration (what's new since 4.2.1)

- Integration with OpenStack Juju Charms for automated Spectrum Scale Cinder/Glance/Nova backend services deployment.

  As a part of this work, 4 charms (spectrum scale manager, spectrum scale client, cinder spectrumscale, glance spectrumscale) got introduced into the charm store.
  GPFS cinder driver is enhanced to work with cinder services deployed inside a Linux container. This change is specific to Juju charm deployment.

Cinder Integration (ongoing efforts/future goals)

- Tighter integration with Juju charms. Enhance Spectrum Scale charms for automatically creating filesystems with limited supported configurations. Currently, filesystem creation is a manual step.
- Active Active support in cinder. This enables management of cinder volumes through other hosts in the cluster. Currently, the cinder volume service host which has created the volume is responsible for managing it.
- Replace consistency group support with generic groups. (cinder component upstream change)

Manila Integration

What‘s new since 4.2.1:

- Manila driver update with Spectrum Scale CES support to create Ganesha NFS exports
- Manila driver update to support manage/unmanage share. Existing filesets can be brought under Manila management
- Versions care and Bug fixes

Ongoing efforts/future goals:

- Manila share network support PoC, evaluating if we can support multi-tenancy with GPFS + Manila integration
- Compression support in Manila. i.e. be able to create compressed filesets as Manila shares

Spectrum Scale

Questions?

Spectrum Scale

Feedback is very welcome!

Spectrum Scale

Thank you

Legal notices

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectually property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER OR IMPLIED. IBM LY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. 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 in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, ed or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 1 0504- 785
U.S.A.

Information and trademarks

Notes:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.

Special notices

This document was developed for IBM offerings in the United States as of the date of publication. IBM may not make these offerings available in other countries, and the information is subject to change without notice. Consult your local IBM business contact for information on the IBM offerings available in your area.

Information in this document concerning non-IBM products was obtained from the suppliers of these products or other public sources. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. Send license inquires, in writing, to IBM Director of Licensing, IBM Corporation, New Castle Drive, Armonk, NY 10504-1785 USA.

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

The information contained in this document has not been submitted to any formal IBM test and is provided "AS IS" with no warranties or guarantees either expressed or implied.

All examples cited or described in this document are presented as illustrations of the manner in which some IBM products can be used and the results that may be achieved. Actual environmental costs and performance characteristics will vary depending on individual client configurations and conditions.

IBM Global Financing offerings are provided through IBM Credit Corporation in the United States and other IBM subsidiaries and divisions worldwide to qualified commercial and government clients. Rates are based on a client's credit rating, financing terms, offering type, equipment type and options, and may vary by country. Other restrictions may apply. Rates and offerings are subject to change, extension or withdrawal without notice.

IBM is not responsible for printing errors in this document that result in pricing or information inaccuracies.

All prices shown are IBM's United States suggested list prices and are subject to change without notice; reseller prices may vary.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Some measurements quoted in this document may have been estimated through extrapolation. Users of this document should verify the applicable data for their specific environment.

Spectrum Scale
Backup

Object Integration

What's new since 4.2.1

Object Integration (since 4.1.1)

What‘s new since 4.2.1 (major new functionalities):

- OpenStack Liberty Release (4.2.1)
- Execute Object cli commands from any Spectrum Scale Node (4.2.1)
- Enable and Disable Unified File & Object via CLI (4.2.1)
  https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1adm_enablefileaccess.htm
- Enable and Disable S3 Support via CLI (4.2.1)
  https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1adm_ChangeconfigurationenableS3.htm
- Object Encryption on Container Basis (4.2.1)
  https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1adm_storagepolicyencrypt.htm
- Monitor external AD and LDAP Server for Object Authentication (4.2.1)
- External Keystone with SSL support (4.2.1)
  https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1adm_configextkeystoneforobject.htm

Object Integration (since 4.1.1)

What‘s new since 4.2.1 (major new functionalities):

- Best Practice Guide for multi-region Object deployment with HA Keystone (4.2.1)
  https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1ins_multiregionplanning.htm
- Created an Object and Keystone Problem Determination Guide (4.2.1)
  https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1pdg_Object_relatedissues.htm
- Secure communication between Proxy, Account, Container and Object Server (4.2.2)
  https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1ins_securecommunicationproxyserver.htm
- Enhanced PMSwift counters (more Performance Data) (4.2.2)
  https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1adv_perfmonforobjmetric.htm
- Enable an existing! fileset for unified file and object access so the legacy file data can be accessed using object interface (4.2.2)
  https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1adm_enablingobjectaccessonexistingfilesets.htm
- Lot's of documentation and How To updates...
  https://www.ibm.com/support/knowledgecenter/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1adm_mngobjectstorage.htm

IceTier: OpenStack Swift object storage on tape (or other high-latency media)

- Further details:
  - https://www.slideshare.net/SlavisaSarafijanovic/swifthlm-openstack-swift-extension-for-high-latency-media-such-as-tape-and-optical-disc-internals-and-interfaces
  - https://etherpad.openstack.org/p/ptg_atlanta_swifthlm

Cognitive Object Storage

Problem: Lack of meaningful metadata

We need a way to cognitively auto-tag heterogeneous unstructured Object data to leverage its benefits….

For leveraging the power of user defined object metadata, the object has to be appropriately tagged

Inhibitors of meaningful tagging of object metadata

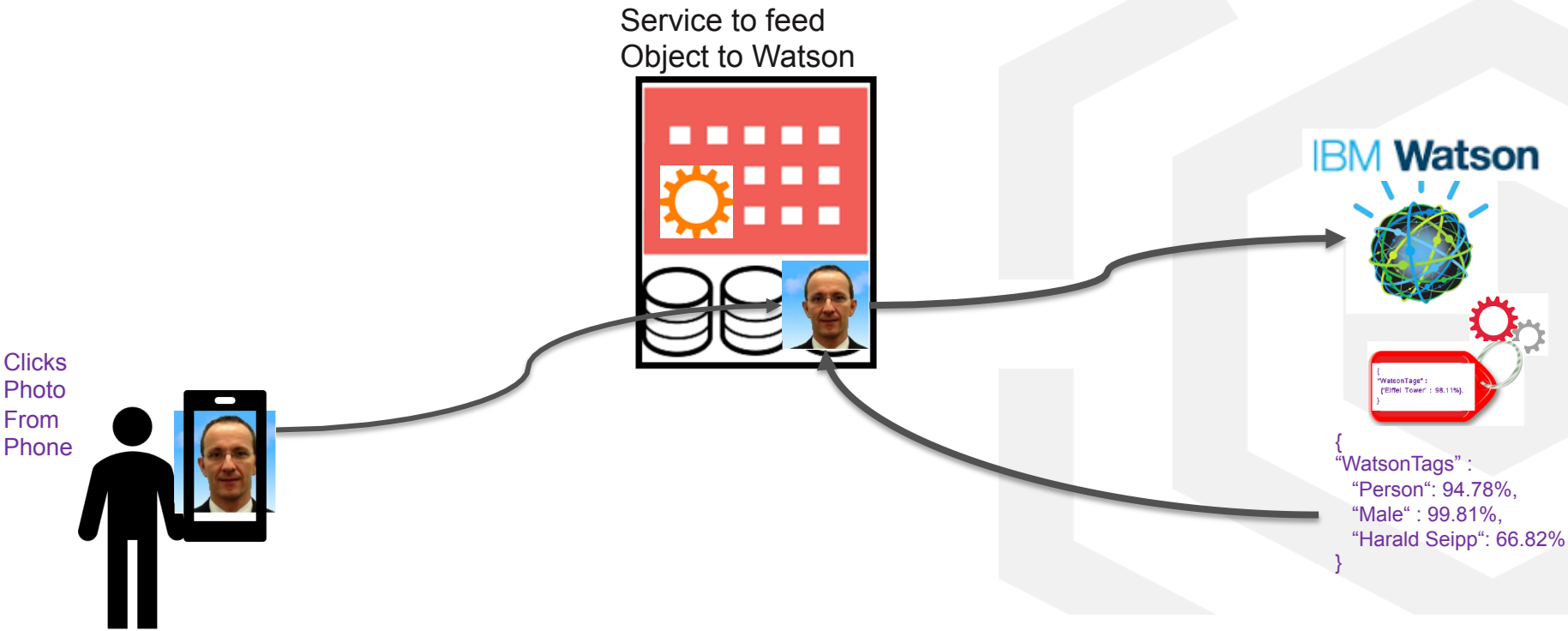
- **Device-based** – e.g.: Digital camera tagging basic info to pics
  - primitive and low level attributes that provide raw data about that object → less or constrained value for analytics
- **Manual** – given by user or applications
  - overhead for the end user at time of object generation → no tagging
  - unnecessary or misleading metadata attributes → no value-add for further processing
- There could be many dimensions of a object, and not all may be added when the object is generated
  - Image may have many faces, but user might tag only few.
  - A song might mapping to two genres but in metadata only one is captured

Cognitive Object Storage

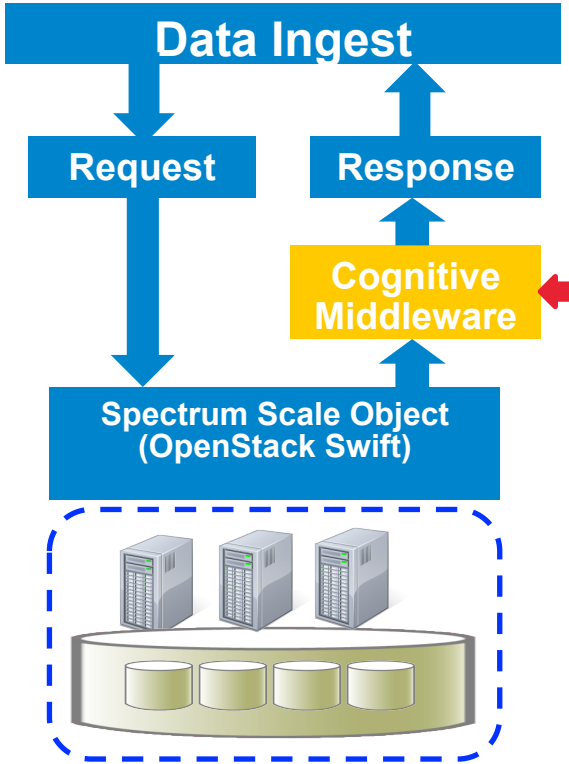
We need a way to cognitively auto-tag heterogeneous unstructured Object data to leverage its benefits….

Solution: Integration of Cognitive Computing Services with Object Storage for auto-tagging of unstructured data in the form of objects

Deriving Object tags using Cognitive services



How it works



Send Objects

Receive Cognitive tags and updates Objects

IBM Watson

Cognitive tags to objects as user defined metadata for applications to leverage.

Try it yourself!

- Sample open source middleware and service code for auto-tagging image objects
- Spectrum Scale Object which is based on OpenStack Swift supports custom middleware like this
- Code and instructions are available on GitHub under Apache 3.0 license.

https://github.com/SpectrumScale/watson-spectrum-scale-object-integration