

Spectrum Scale OpenStack Integration

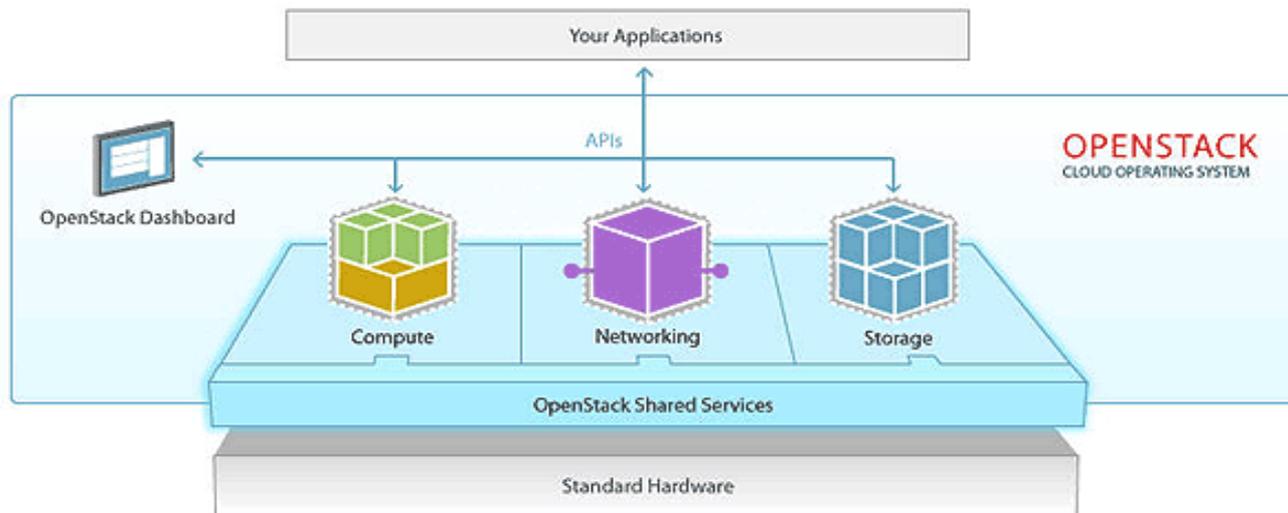


Agenda

- OpenStack overview, Storage components in OpenStack
- Spectrum Scale Object Storage
- Unified File & Object Access (UFO) Feature Deep Dive
- Use Cases Enabled By UFO
- OpenStack Cinder/Manila overview
- UFO and Manila Integration Demo
- Announcements
- Q & A



What is OpenStack ?



OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter



What is OpenStack ?

Open source software for creating private and public clouds

- 50+ projects, and growing
- Think of Amazon AWS, but Open Source, community driven
 - S3 => Swift (object storage)
 - EBS => Cinder (virtual block storage)
 - EC2 => Nova (compute) + Neutron (networking)
 - EFS => Manila (shared file system service)
- Two releases per year, increasing by alphabet:
 - 2015: Kilo (April), Liberty (October)
 - 2016: Mitaka (April), Newton (October)
 - 2017: “O” & “P”



Who uses OpenStack ?

Industry Segment

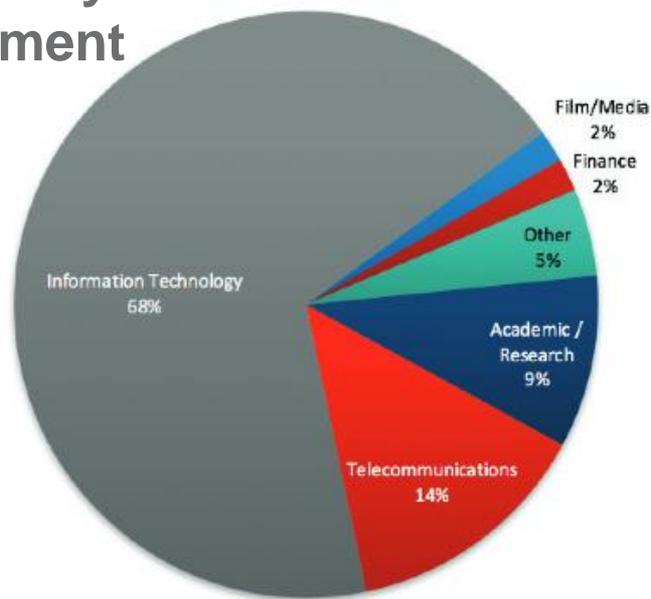


Figure 1.3 n=1402

Company Size

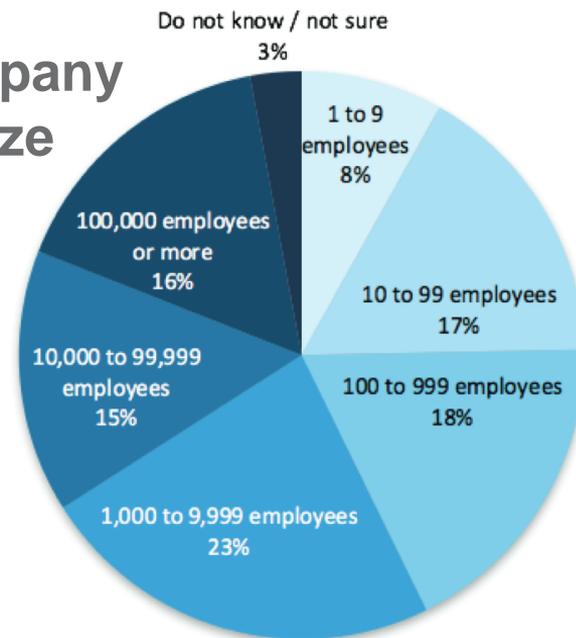


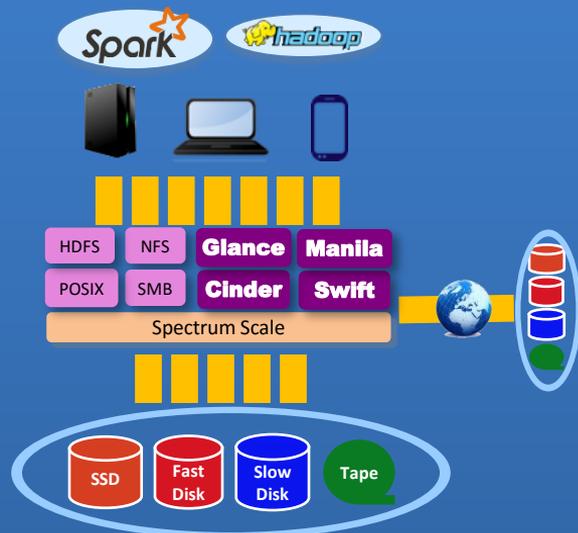
Figure 1.5 n=1402

Source: <http://www.openstack.org/assets/survey/April-2016-User-Survey-Report.pdf>



IBM Spectrum Scale

Data management at scale



- Avoid vendor lock-in with true Software Defined Storage and Open Standards
- Seamless performance & capacity scaling
- Automate data management at scale
- Enable global collaboration

OpenStack and Spectrum Scale helps clients manage data at scale



Business: I need virtually unlimited storage



An open & scalable cloud platform



Operations: I need a flexible infrastructure that supports both object and file based storage



A single data plane that supports Cinder, Glance, Swift, Manila as well as NFS, et. al.



Operations: I need to minimize the time it takes to perform common storage management tasks



A fully automated policy based data placement and migration tool



Collaboration: I need to share data between people, departments and sites with low latency.



Sharing with a variety of WAN caching modes

Results

- Converge File and Object based storage under one roof
- Employ enterprise features to protect data, e.g. Snapshots, Backup, and Disaster Recovery
- Support native file, block and object sharing to data



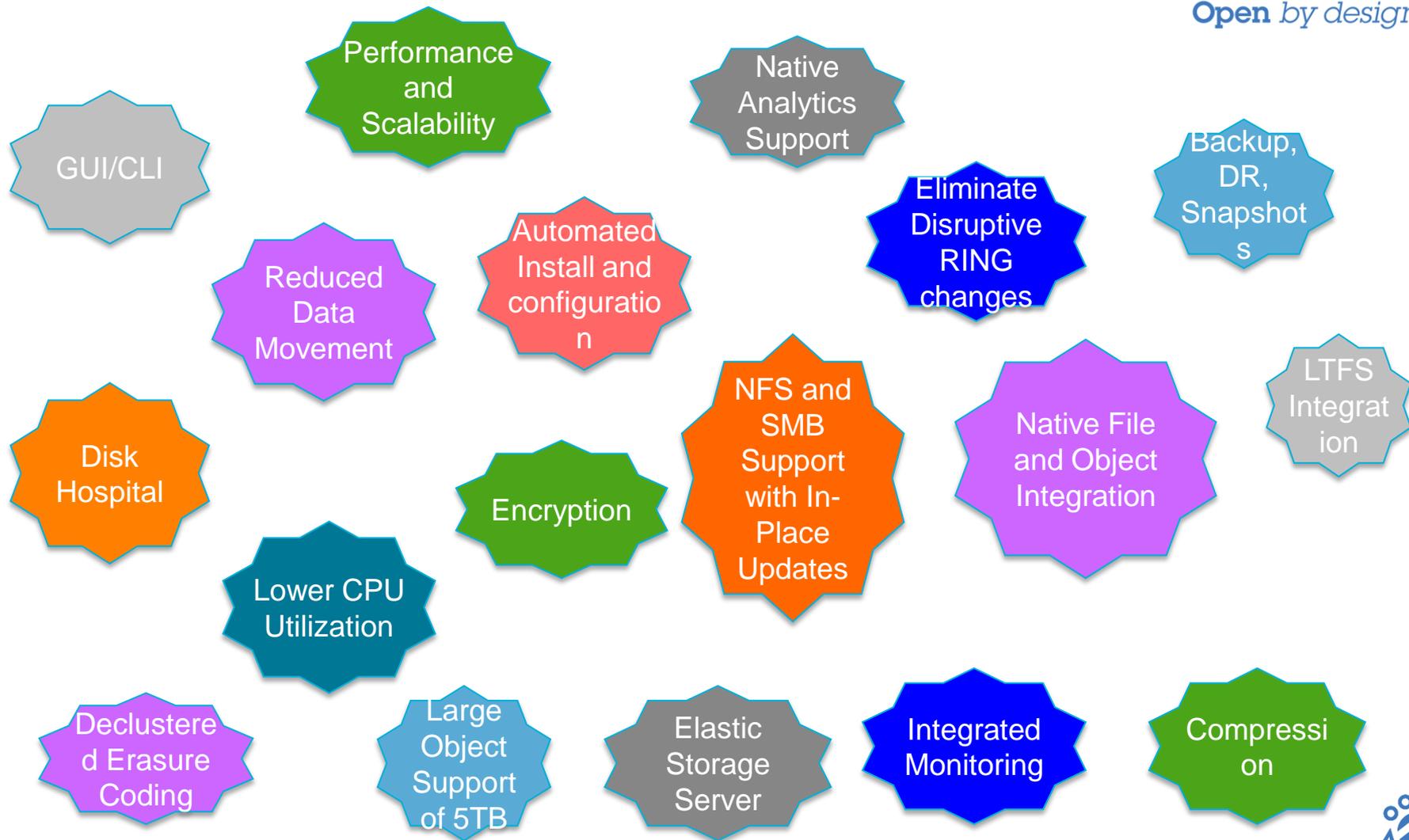
Spectrum Scale Object Storage Introduction



Spectrum Scale Object Storage

- Basic support added in 4.1.1 release & enhanced in 4.2 release
- Based on Openstack Swift (Kilo Release)
- REST-based data access
 - Growing number of clients due to extremely simple protocol
 - Applications can easily save & access data from anywhere using HTTP
 - Simple set of atomic operations:
 - PUT (upload)
 - POST (update metadata)
 - GET (download)
 - DELETE
- Amazon S3 Protocol Emulation support
- High Availability with CES Integration
- Simple and Automated Installation Process
- Native GPFS Command Line Interface to manage Object service (mmobj commands)



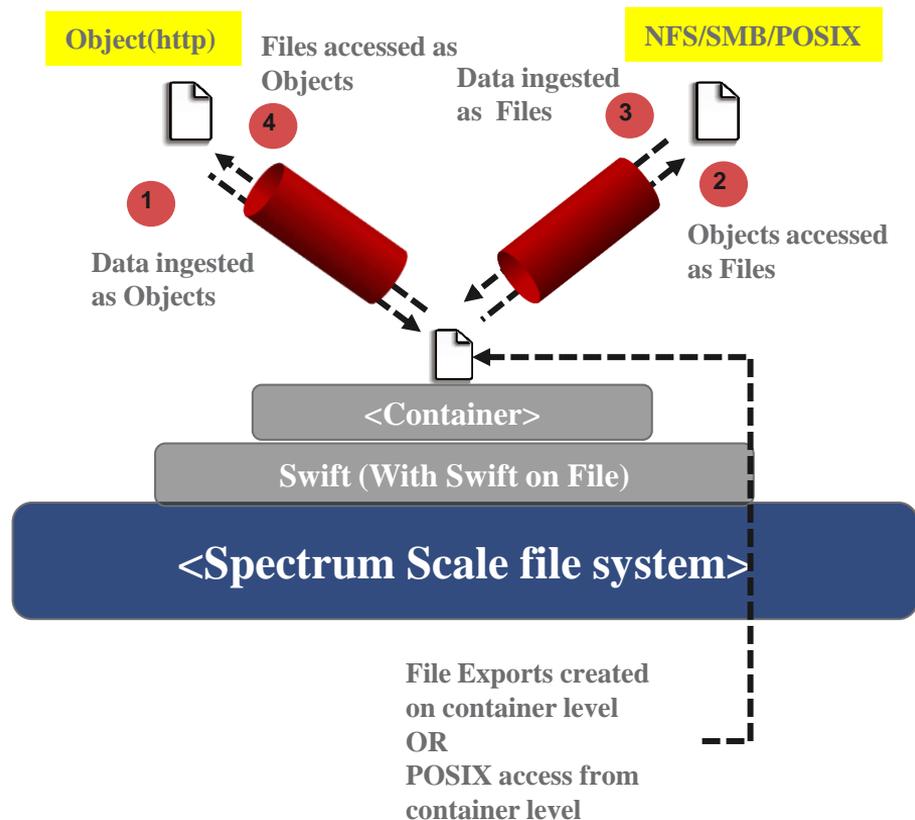


Unified File and Object What is it ?



What is Unified File and Object Access ?

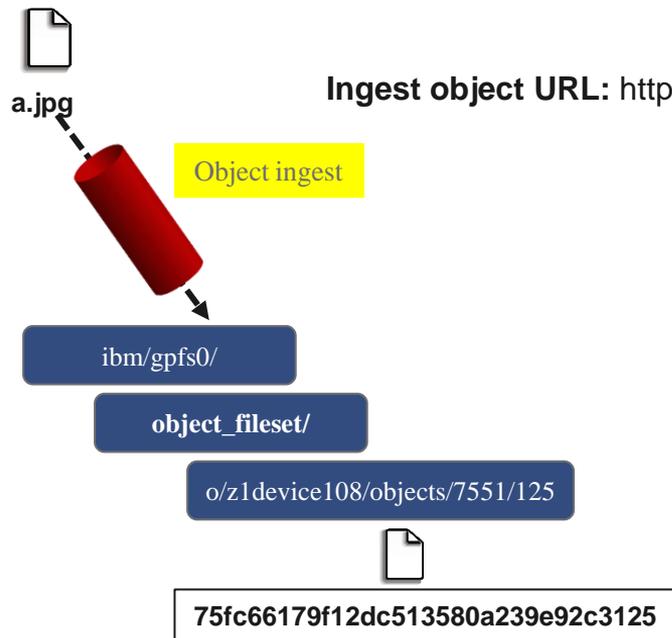
- **Accessing object using file interfaces (SMB/NFS/POSIX) and accessing file using object interfaces (REST) helps legacy applications** designed for file to seamlessly start integrating into the object world.
- **It allows object data to be accessed using applications** designed to process files. **It allows file data to be published as objects.**
- **Multi protocol access for file and object in the same namespace (with common User ID management capability)** allows supporting and hosting **data oceans** of different types of data with multiple access options.
- **Optimizes various use cases** and solution architectures resulting in better efficiency as well as cost savings.



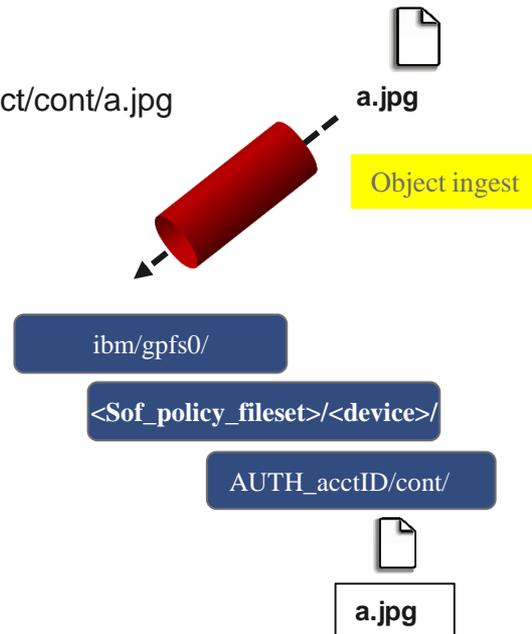
Filesystem Layout (Traditional Vs Unified File and Object Access)

- One of the **key advantages of unified file and object access is the placement and naming of objects when stored on the file system**. In unified file and object access stores objects following the same path hierarchy as the object's URL.
- In contrast, the default object implementation stores the object following the mapping given by the ring, **and its final file path cannot be determined by the user easily**.

Traditional SWIFT

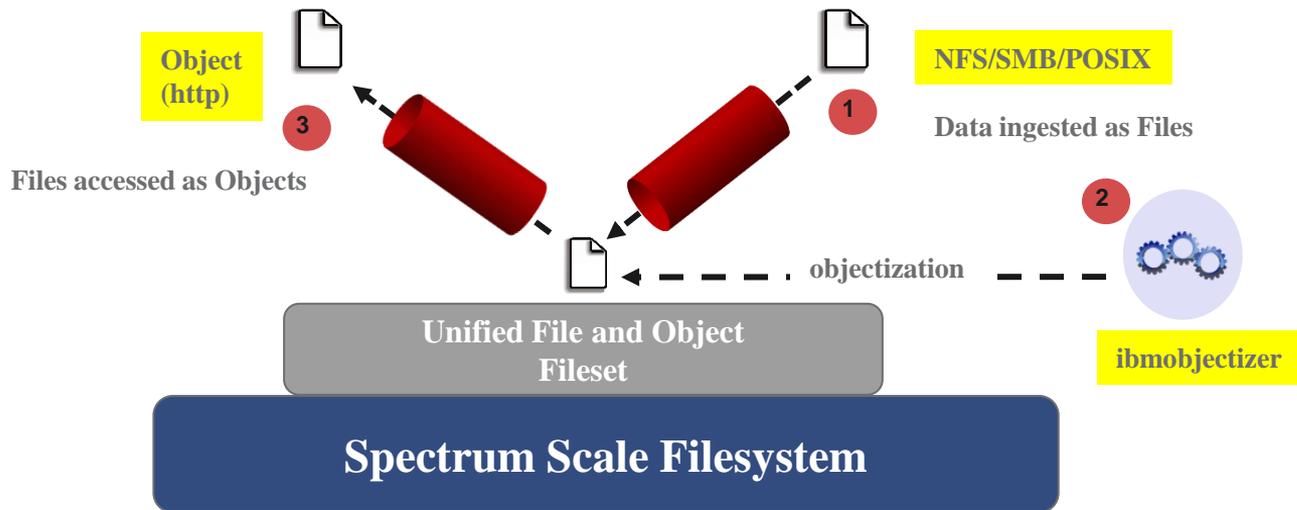


Unified File and Object Access



Objectization – Making Files as Objects (Accessing File via Object interface)

- Spectrum Scale 4.2 features with a system service called **ibmobjectizer** responsible for objectization.
- Objectization is a **process that converts files ingested from the file interface** on unified file and object access enabled container path **to be available from the object interface**.
- When new files are added from the file interface, they need to be visible to the Swift database to show correct container listing and container or account statistics.



Unified File and Object Access – Policy Integration for Flexibility

This feature is specifically made available as an “object storage policy” as it gives the following advantages:

- **Flexibility for administrator** to manage unified file and object access separately
- Allows to **coexists with traditional object and other** policies
- Create multiple unified file and object access policies which can vary based on underlying storage
- Since policies are applicable per container , it gives **end user the flexibility to create certain containers with Unified File and Object Access policy and certain without it.**

Example: `mmobj policy create SwiftOnFileFS --enable-file-access`

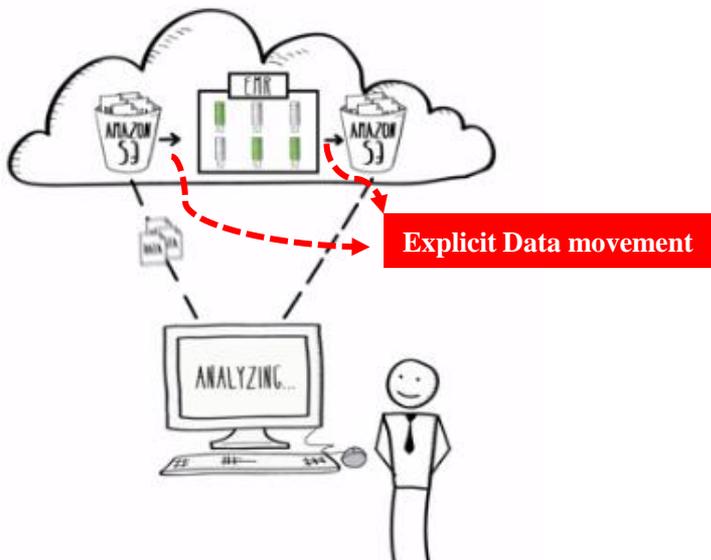


Use Cases



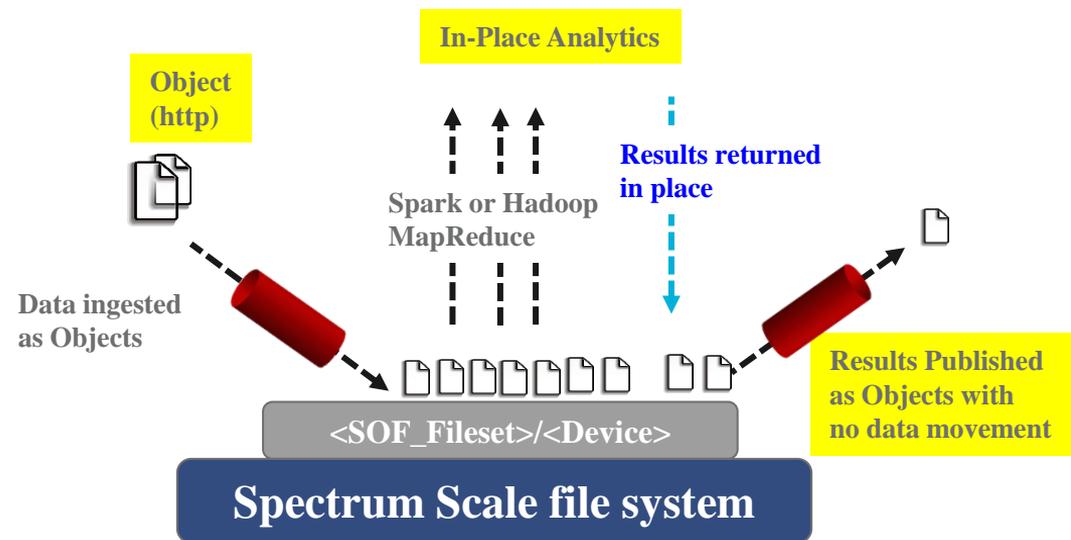
Use case 1 – Enabling “In-Place” analytics for Object data repository with analytic results available as objects

Analytics on Traditional Object Store



Traditional object store – **Data to be copied from object store to dedicated cluster**, do the analysis and **copy the result back to object store for publishing**

Analytics With Unified File and Object Access



Object store with Unified File and Object Access – Object Data available as File on the same fileset . Analytics systems like Hadoop MapReduce or Spark allow the data to be directly leveraged for analytics.

No data movement / In-Place immediate data analytics.



Use case 2 : Process Object Data with File-Oriented Applications and Publish Outcomes as Objects

Final processed videos available as Objects in container which is used for external publishing

Media House OpenStack Cloud Platform
(Tenant = Media House Subsidiaries)

Dailymotion

YouTube metacafe

Publishing Channels

Final Video (as objects) available for streaming



Subsidiary 1



Subsidiary 2

Ingest Media Objects



VM Farm for Subsidiary 1 for video processing



VM Farm for Subsidiary 2 for video processing

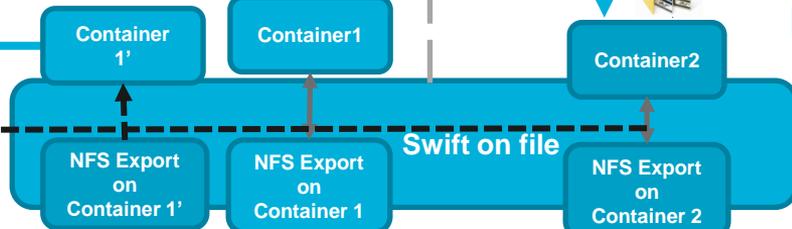


Raw media content sent for media processing which happens over files (Object to File access)

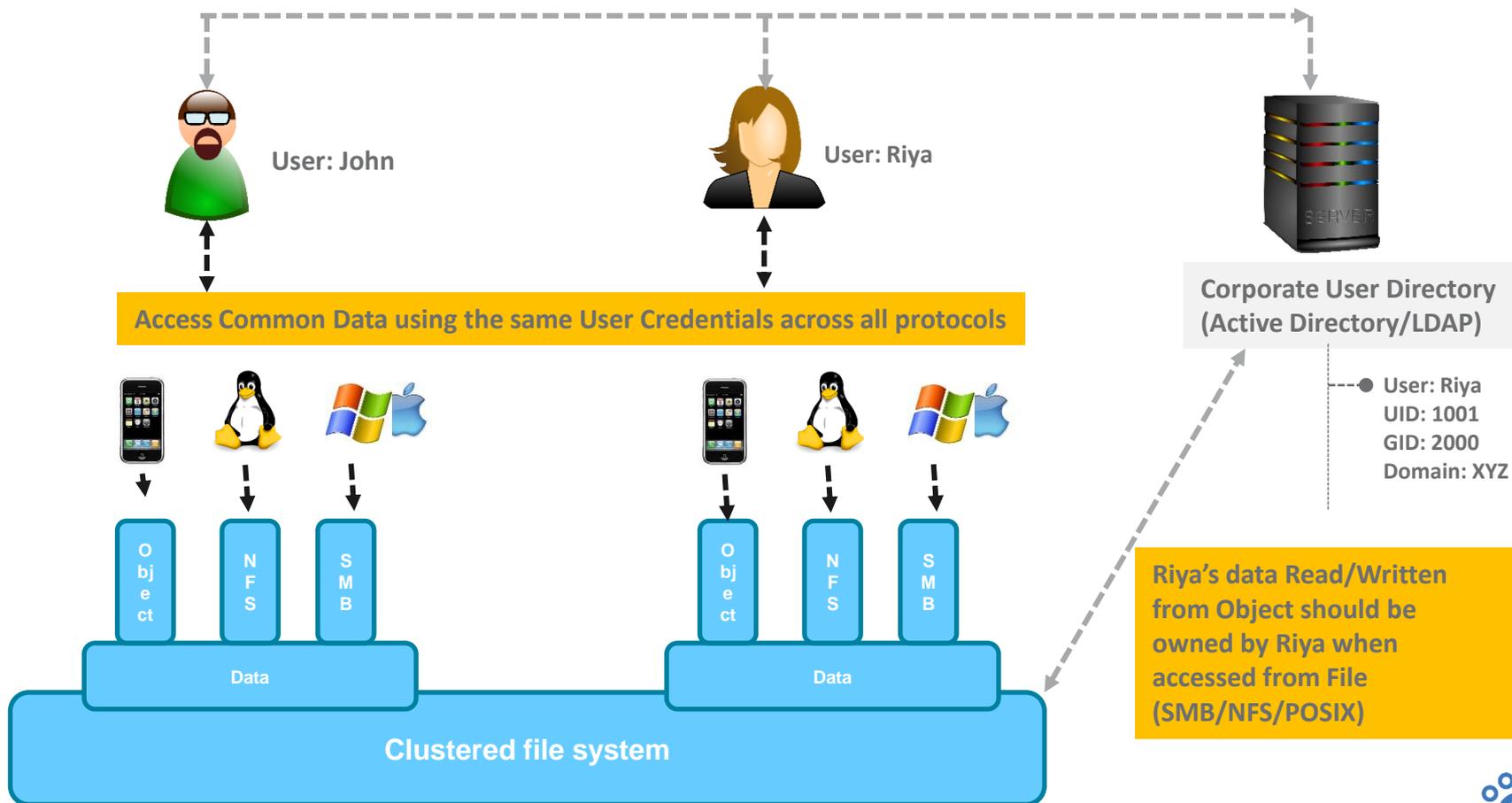
Manila Shares (NFS) exported only for Subsidiary2

Manila Shares (NFS) exported only for Subsidiary1

Files converted into objects for publishing (File to Object access)



Use case 3 : Users read/write data via File and Object with Common User Authentication and Identity



OpenStack Cinder - Block Storage Service



What is Cinder ?

- Architected to provide Block storage as a service
- Presents persistent block-level storage volumes for use with OpenStack Nova compute instances
- Manages the creation/deletion, attaching and detaching of these volumes between a storage system and different host servers
- Plug-in architecture (Use your own vendor drivers)
- Consistent API regardless of backend selection, backend devices can be invisible to end-users
- Specific placement based on volume-type selection
- Expose differentiating features via volume types and extra specs

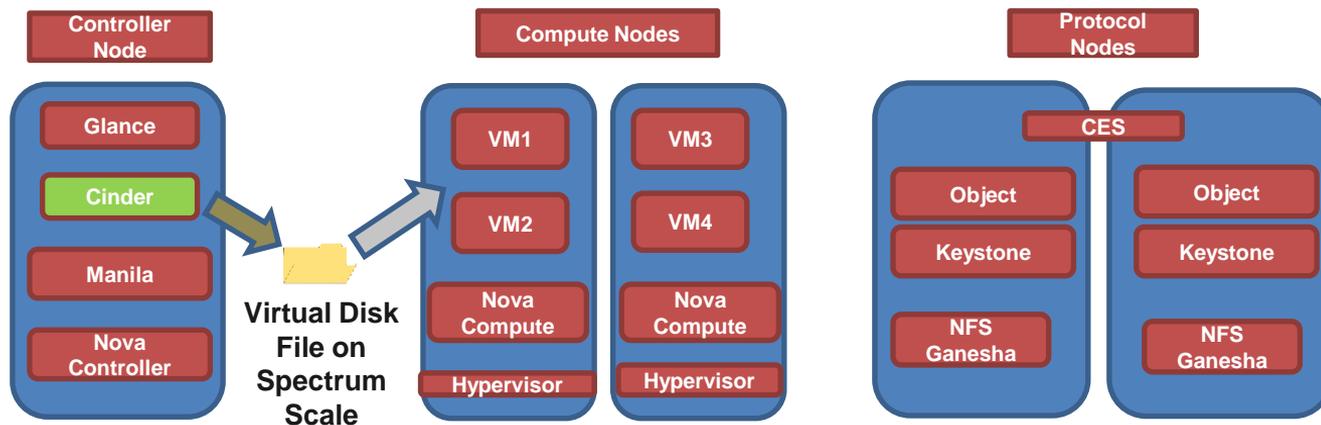


Spectrum Scale Cinder driver

- Added in OpenStack Havana release
- Supports Local/NFS based deployments
(Cinder service may or may not run on Spectrum Scale node)
- **Supported Operations by driver:**
 - ✓ Create/Update/Delete volumes
 - ✓ Create/Update/Delete volume snapshots
 - ✓ Create volume from Snapshot/Volume/Glance Images (uses copy-on-write)
 - ✓ Attach/Detach volumes to an instance (Nova VM)
 - ✓ Extend volume
 - ✓ Volume Backup/Restore
 - ✓ Volume Migration
 - ✓ Consistency Groups support
- **Roadmap**
 - ✓ Replication support (sync, async)
 - ✓ Consistency Groups enhancements
 - ✓ Volume Migration enhancements
 - ✓ Volume Attach/Detach to Docker containers
 - ✓ Certification of Spectrum Scale Cinder driver with OpenStack distributions



Spectrum Scale Cinder driver deployment configuration



Spectrum Scale Cluster



OpenStack Manila - File Share Service

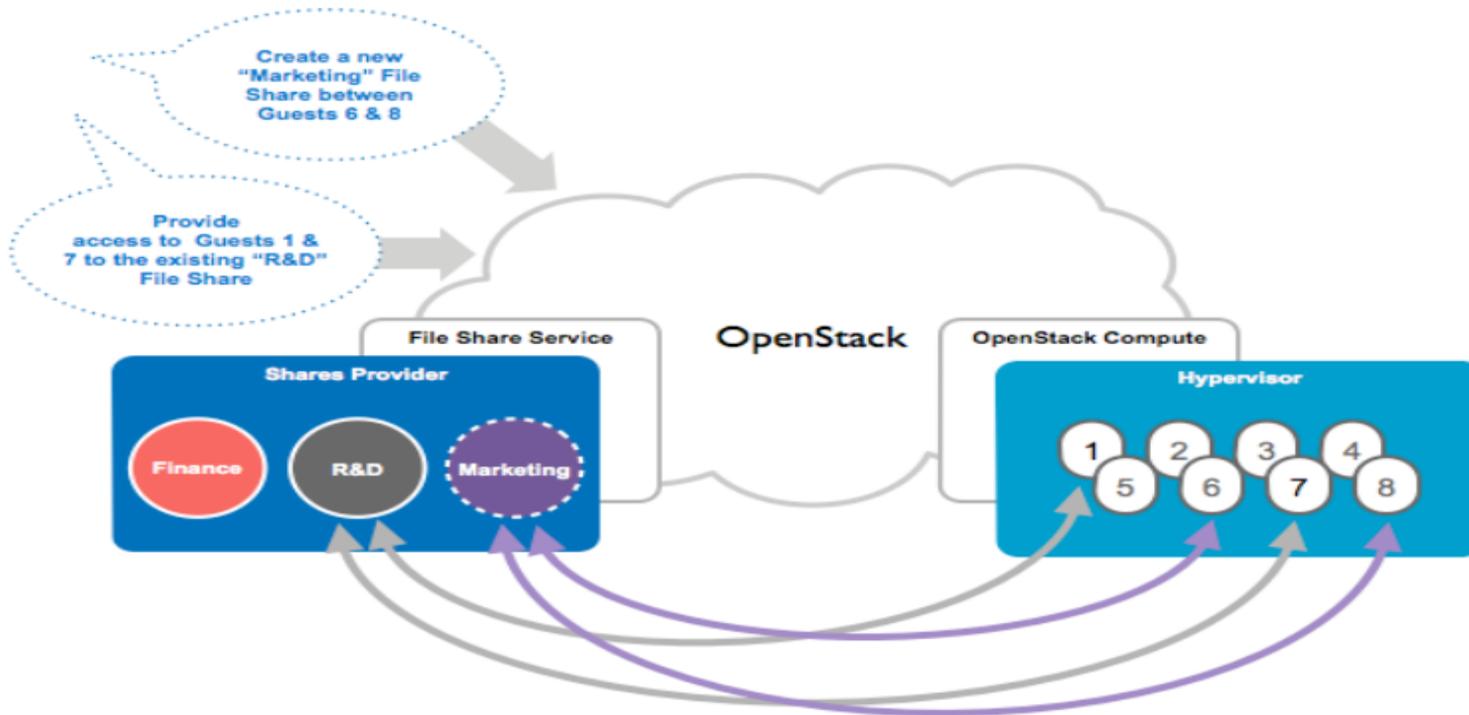


What is Manila ?

- Shared **FileSystem as a service**
- **Provision file shares** to Nova (OpenStack Compute) instances
- **Vendor neutral API** for provisioning and attaching filesystem-based storage such as NFS, CIFS, and other network filesystems
- Supports **Multi-tenancy**



Manila Example

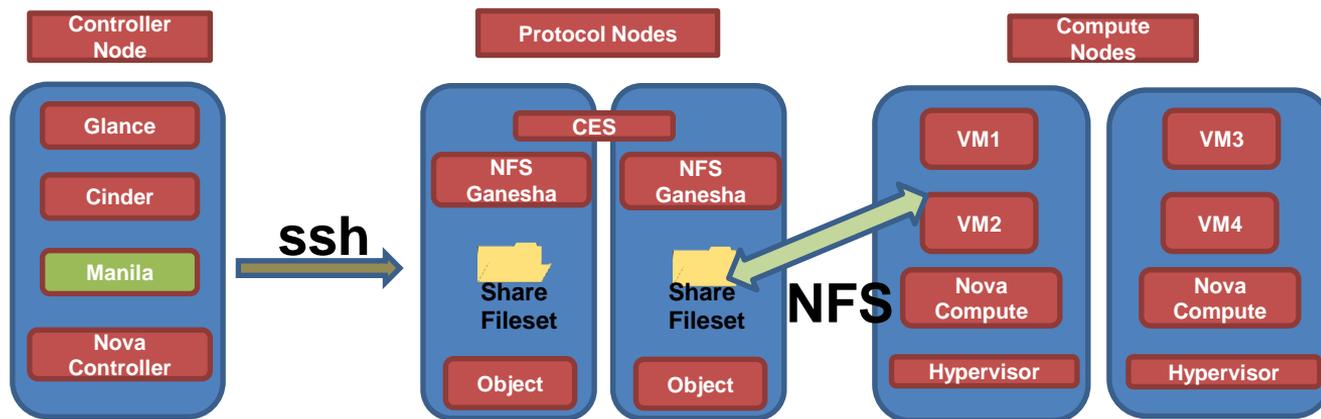


Spectrum Scale Manila driver

- Added in OpenStack Kilo release
- Today supports kNFS & Ganesha v2.1.
- **Supported Operations by driver:**
 - ✓ Create/Delete/List shares
 - ✓ Allow/Deny access to shares (Create NFS exports and allow/deny access)
 - ✓ List share access rules
 - ✓ Create/Delete/List share snapshots
 - ✓ Create share from snapshot
 - ✓ Extend share
- **Roadmap**
 - ✓ Support for NFS (Ganesha server) through Spectrum Scale CES (2H2016)
 - ✓ Manage/unmanage share/share snapshot
 - ✓ Compression/Encryption Support through driver capabilities and extra specs
 - ✓ Certification of Spectrum Scale Manila driver with OpenStack distributions
 - ✓ CIFS support



Manila service running with Spectrum Scale CES (NFS Ganesha)



Spectrum Scale Cluster



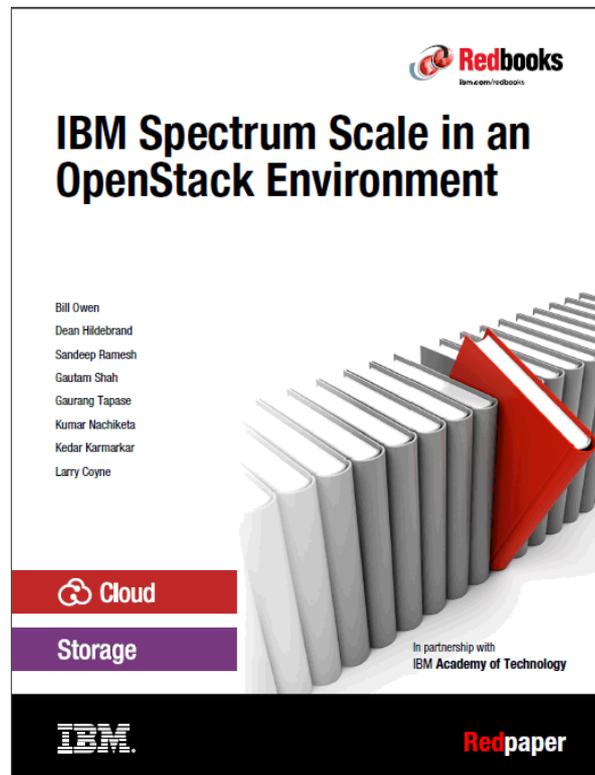
Demo

**Combining Manila and Swift for Unified Data Sharing
across Nova instances**



OpenStack Austin Summit 2016

IBM Spectrum Scale in an OpenStack Environment Redpaper Published



<http://www.redbooks.ibm.com/abstracts/redp5331.html>



OpenStack Austin Summit 2016

Speaker Sessions

Amalgamting Manila and Swift for Unified Data Sharing Across Instances
(Bill Owen, Simon Lorenz, Sandeep Patil, Gaurang Tapase, Brian Nelson)

https://www.youtube.com/watch?v=3MMrMUaA_Mg

Write a file, Read as an Object
(Bill Owen, Sandeep Patil, Smita Raut, Sasikanth Eda)

<https://www.youtube.com/watch?v=6ovLb6aktbM>



OpenStack Usage Survey



Questions ?



References

Write a File, read as an Object: Openstack Summit, Austin, TX Apr 2016

<https://www.youtube.com/watch?v=6ovLb6aktbM&feature=youtu.be&t=2>

Amalgamating Manila and Swift for Unified Data Sharing: Openstack Summit, Austin, TX Apr 2016

https://www.youtube.com/watch?v=3MMrMUJaA_Mg

Hadoop HDFS Vs Spectrum Scale: <https://www.youtube.com/watch?v=kOeEbdO8F4A>

From Archive to Insight: Debunking Myths of Analytics on Object Stores – Dean Hildebrand, Bill Owen, Simon Lorenz, Luis Pabon, Rui Zhang. Vancouver Summit, Spring 2015.

<https://www.youtube.com/watch?v=brhEUptD3JQ>

Deploying Swift on a File System – Bill Owen, Thiago Da Silva. BrownBag at OpenStack Paris, Fall 2014

<https://www.youtube.com/watch?v=vPn2uZF4yWo>

Breaking the Mold with OpenStack Swift and GlusterFS – Jon Dickinson, Luis Pabo. Atlanta Summit, Spring 2014

<https://www.youtube.com/watch?v=pSWdzjA8WuA>

SNIA SDC 2015

http://www.snia.org/sites/default/files/SDC15_presentations/security/DeanHildebrand_Sasi_OpenStack%20SwiftOnFile.pdf

Spectrum Scale Infocenter

http://www.ibm.com/support/knowledgecenter/#!/STXKQY_4.2.0/com.ibm.spectrum.scale.v4r2.adm.doc/b1adm_manageunifiedaccess.htm



Thanks