Scality Ring

Overview

Scality RING is software to create an object storage system on servers with attached storage in a scale-out, distributed shared-nothing architecture. The software includes a number of features as an object storage system. The RING name is from the distributed architecture for the nodes in the configurations, which is based on the Chord protocol for peer-to-peer connections. The RING uses a distributed hash table with key-value pairs. The key format comes from a SHA-1 hash and is 160 bits with 128 for the object ID, 24 for the dispersion location (which node in which location), and 8 bits for the class of service which in this case means the protection policy of either replication or Forward Error Correction information dispersal. The algorithm maps objects to a virtual key (node) spaces and routes access requests to the node owning the data. Each node will have a successor and predecessor to the next node.

The architecture of Scality RING is meant to be high performance, which allows Scality to target replacing traditional NAS and block storage for unstructured and semi-structured data. Usage for performance demanding content repositories, big data analytics storage, and digital media are highlighted usages by Scality.

Data and metadata are distributed across nodes according to the RING algorithm. Forward Error Correction uses the ARC erasure codes with selectable protection and geographical distribution. In addition to information dispersal, Scality offers synchronous and asynchronous replication with the multi-Geo feature.

Highlights

- Software to build large scale object, VM and file storage
- High-performance distributed architecture
- Geographic dispersal
- Full S3 protocol support
- Multiple access methods supported
- Compliance support
- Encryption
- Versioning – file and object
- Multi-tenancy
- Group access control policies
- Management / monitoring through cloud app & as a service from Scality
- Multi-site replication
- Scale Out File System on each node

Access to Scality RING system, which consists of Scality software deployed on x86 servers with attached storage, is through file and object access methods. For file access, Scality supports its own Scale out File System, CIFS/SMB 2.0 (SAMBA), FTP, AppleTalk and NFSv4. Object access is over HTTP/REST protocol for S3, CDMI, Swift and RS2 APIs. Hadoop is implemented using

Scality's CDMI connector where the Hadoop NameNode server is replaced by the Scality architecture enabling the storage node to operate as the Hadoop compute node.

On a Scality storage node, objects and metadata are stored in containers in a local file system to minimize consumption of inodes. Data is automatically rebalanced across nodes when a node is removed or added.

Advanced features are controlled through the S3 API protocols including multi-tenancy, group access control policies, and semantics for versioning.

Scality also has a Multi-Cloud Controller offering called XDM that originated as a separate offering called Zenko. XDM provides data movement, search, and policy management.

Scality offers the HALO Cloud Monitor, which is a cloud-based application to monitor and manage a Scality system. The basic version called Standard Service allows the customer to do monitoring and get metrics (telemetry data) on operations, performance, and configuration. The advanced version called Dedicated Care Service is a services engagement where Scality support engineers will monitor and support the system for an annual fee. The following summarizes the difference between the offerings:

- Standard edition
  - Capacity, features, configuration, problem information collection
- Dedicated Care Service – Scality personnel
  - Alarms on issues, drill downs, problem isolation by Scality eng.
  - Multiple detail metrics and reports
  - Unlimited support calls, 20 engineer requests per year, ann. visit
  - Monitoring – 24x7
  - Quarterly review, assessment, advice
  - Service levels and escalation
  - Guarantee – 100% data access availability

Usage and Deployment

Scality RING is used by cloud service providers in many areas and by clients in vertical markets. Content repositories are a common usage with S3 and CDMI API access for objects and as scale out NAS file storage with NFS.

- Characteristics
  - Scale – No specific defined amount. Scality refers to scaling from petabytes to exabytes of capacity and billions of objects.

  - Protection / Durability / Resiliency – Forward Error Correction with ARC Erasure Codes provides protection from data element or node failure. Distribution of data is accomplished based on settings for information dispersal and replication. Also included is a disaster recovery mode for site failover and component failures.
  - Index and Search – Included with XDM function.
  - Performance – Focus for Scality is the performance of system based on the RING architecture with the underlying Chord distribution protocol.
  - Access Methods – Object access is through Amazon S3, CDMI, Swift and RS2 APIs over HTTP/REST. File system access through CIFS/SMB 2.0 (SAMBA), FTP, AppleTalk, NFSv4 or by Scality's Scale out File System. Files written using file access can be read using the RESTful interface and for Hadoop processing. VM storage is provisioned through Cinder.
  - Geographic Access – Multi-Geo feature supports synchronous or asynchronous remote replication in addition to the information dispersal with FEC. Multi-site replication is supported
  - Security and Compliance – Multi-tenancy is supported. Various elements of regulatory compliance are supported but no certifications. Encryption with KMIP support is included.
  - Metadata – System and user metadata are added to the object storage. How the metadata is handled is not specified.
  - Integrity and Verification – CRC for data is maintained and checked on access.
  - Longevity of Object Data – Nodes can be removed and have data automatically rebuilt to other nodes. New nodes can be added and data will automatically be rebalanced.
  - Billing and Chargeback – Information is available for the S3 compatible access method.
- Applications
  - Content delivery networks
  - Service providers file storage
  - Big data analytics processing and storage
  - Active archive
  - User generated content and online backup repositories
  - Virtual machine storage
  - Records storage in compliance required environments
- System environments
  - Any environment with web access – either through private cloud or cloud service providers
  - Virtual machines with Cinder provisioning
- Deployment and Administration
  - Central management platform with monitoring of all elements and an activity log.
  - CLI for scripting access.
  - Cloud monitoring application and services offering.

Key Capabilities

Architecture and Deployment

The key architectural element in Scality RING is the ring implementation used for nodes. The RING is based on the Chord peer-to-peer protocol developed at MIT developed to map stored objects onto a virtual circular structure called a Keyspace. Scality has implemented the Chord protocol with additions for data durability, resiliency of nodes, high performance, and management. The Chord protocol with Scality additions yields a distributed storage system capable of scaling to an almost unlimited number of objects and allowing for parallel I/O operations.

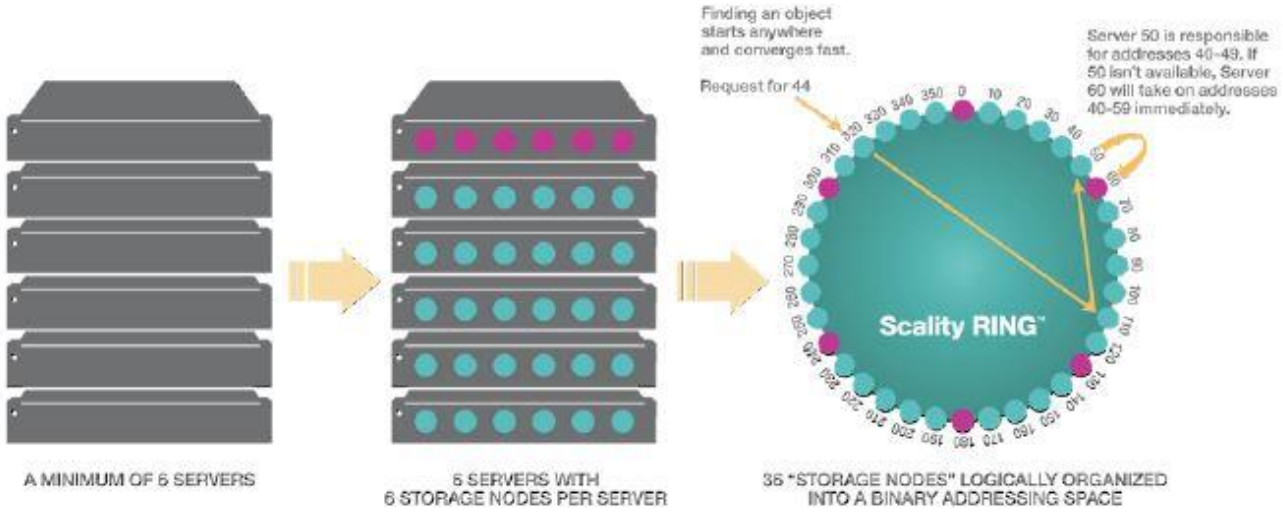

Figure 1: Ring Structure

The ring structure minimizes the number of hops to a node. The following chart from Scality illustrates the nodes and hops required.

Number of Nodes in RING	RING Capacity*	Number of Hops
100	5.7 PB	3
1,000	56 PB	5
10,000	560 PB	7

*Assumes six (6) Nodes per physical server, 56 HDDs per server, 6TB per drive

Table 1: Hops Required Example (Source: Scality)

A distributed hash table is used to locate the objects in the ring. Key-value pairs are stored in the table so a node can access the information about an object or find the next best location to access for the information. The key used to identify objects is 160 bits, which consists of information to ensure dispersion across nodes, the object ID (called the payload), and the class of service. The class of service (CoS) indicates:

- The number of replica copies for an object (maximum of 6)
- Indicator for geo-distribution

The smallest RING is three physical servers. Each physical server is divided into a set of at least six storage nodes, which is in a ring structure and has an assigned key value.

Connectors provide the interface to the nodes in the RING and provide the protocol services. Requests from clients go to the Connectors and after the protocol handling is done, the Connector will transfer data to the storage nodes. The Connector will select the storage nodes based on a list that is configured for the Connector with key information and the RING topology. The Connector effectively does load balancing between the nodes connected in the RING. Across a larger domain, an external load balancer is expected to be utilized.

Software Architecture

The Scality RING is implemented as a set of storage services that execute on a standard operating system. The top layer is the scalable access services called Connectors that provide the protocol interfaces for the accessing applications. The middle layer includes a distributed file system, data protection software functions, and systems and management software services. The storage nodes use I/O daemons to manage the storage services and device interfaces.

The following diagram from Scality is illustrative of the architecture.

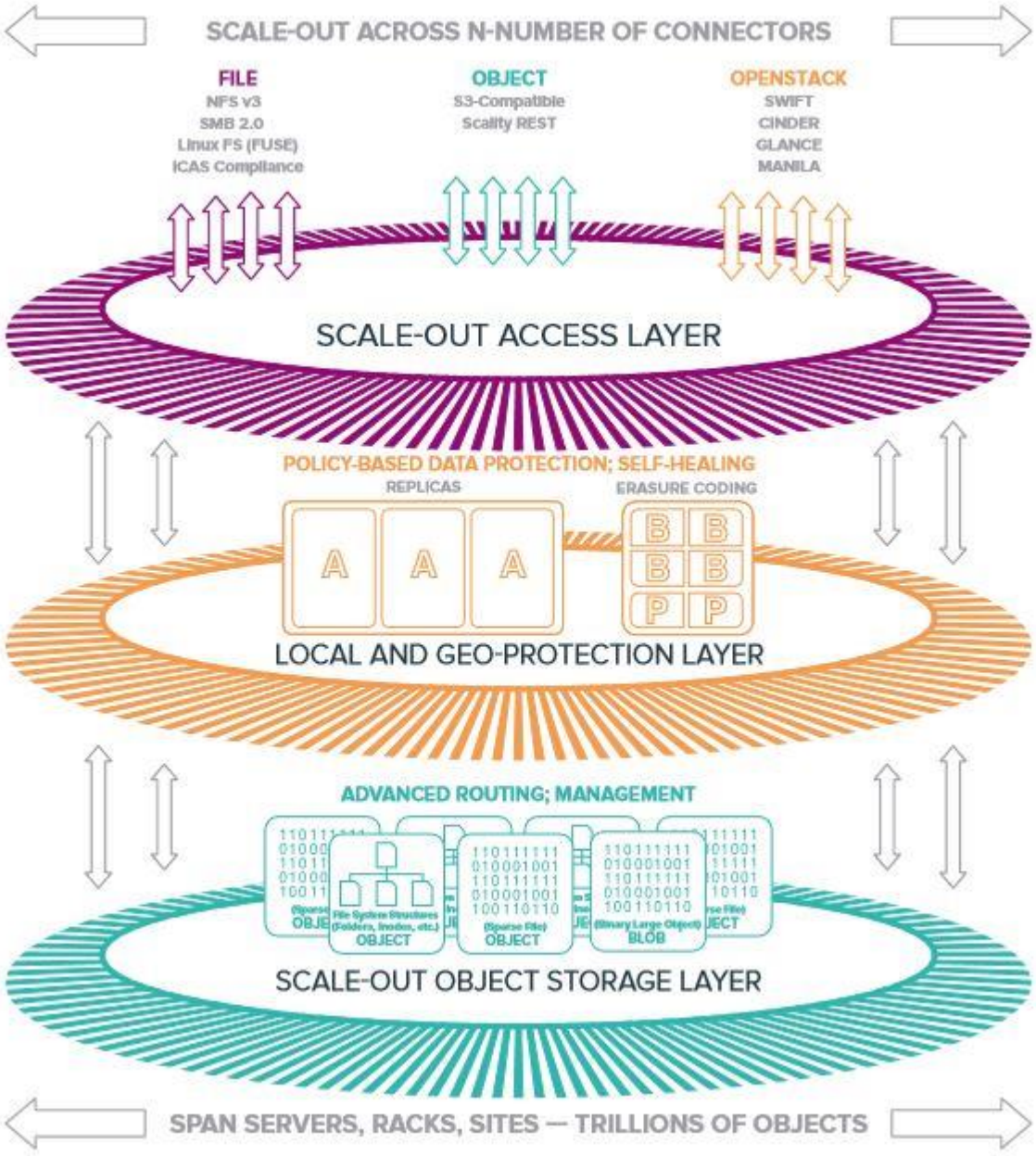

Figure 2: Scality Overview (Source: Scality)

Connectors

Connectors provide the protocols for access to data stored in a Scality system. Objects are supported with S3, OpenStack Swift and a native Scality REST-based interface. File access is

supported for NFS, CIFS/SMB, and FUSE. A block access is supported only through an OpenStack Cinder driver.

The connectors may be used interchangeably to provide access to data. Additionally, multiple connectors can be used for parallel access.

The data transfer path from applications goes to the Connectors and the Connector handled the scheduling to storage nodes. Connector also implement the data protection storage policy. The policy can be replication or error correction using erasure codes.

Database

Internally, Scality uses a NoSQL database called MESA. The MESA database provides an index for the object, identifying where data is store and the abstraction layer for the Scale Out File System.

Storage Nodes

A storage node is a logical construct within a physical server called a storage server. The storage server is usually configured with six storage nodes. The I/O daemons running on the storage nodes are responsible for managing the storing of object fragments called fixed size containers. The standard operating system and standard file system allow administrators to manage the storage nodes with familiar tools and processes.

The mapping of the object to the underlying data on the devices is managed by the I/O daemons. An I/O daemon is configured for each physical device. Small objects are packed into containers. The devices can be segmented into SSDs for indexes and HDDs for data. Scality notes that 10% of the storage capacity for objects should be allocated for metadata, which is used as the guideline for sizing the capacity of SSDs.

Consistency

Scality implements eventual consistency, meaning that it is up to the application to determine if the data is consistent. For Scality, consistency is determined by whether the number of protected copies (in the case of replication) or the number of elements in the dispersed copies for error correction with erasure codes were written before subsequent access. Scality calls their implementation "relaxed consistency" meaning that all writes do not have to be complete before an access is allowed.

The Scale Out File System does not operate the same as the object access and enforces strict consistency. All writes must be complete before the SOFS will consider an operation complete and allow another access to the file.

Data Protection and Security

Device and Node Failure Protection

A device or node failure uses either the replication feature with up to six copies of data or error correction with erasure coding to continue access to data even if a device or a node fails. With erasure coding, up to 64 data and chunks can be defined. Replication and erasure coding are configured at the Connector level. After a failure, the system will automatically rebuild and distribute data to return to the protection state. The erasure codes are a standard Reed-Solomon coding. The number of protection chunks compared to the data chunks is a configuration option that balances efficiency of space used and the number of failures tolerated.

Remote Protection

For site protection, geographic dispersion is supported as well as asynchronous or synchronous remote replication. Two or more sites can be included in the geo-distribution protection.

Data Integrity

Data integrity is assured by adding a checksum added to the data and index files written to the storage nodes as individual files. Every access checks the checksum. Repairs are made with the protection data whenever an error is detected. A background scan of all data is run continuously to find and correct data errors.

Multi-Tenancy

Multi-tenancy is not specifically handled by the Scality RING software. Instead, the application must implement its own tenancy controls or the S3 protocol can be used to isolate access.

Encryption

AES-256 server-side encryption is used to encrypt data at rest and uses KMIP for key control.

Advanced Features

As a system that has been available for some time, Scality has added a number of valuable advanced features. The advanced features for performance are included in the performance section of this document.

Versioning

Versioning is supported for file access and for S3 protocol and allows retention of multiple versions. For object, the standard S3 API access allows selection of the individual objects.

Compliance

Compliance to a level to meet most regulatory requirements are incorporated into Scality RING. Encryption, WORM, legal holds and audit trails are a few of the compliance features. Additionally, Scality supports a secure delete of stored data.

WORM

WORM mode is supported as part the of the compliance implementation.

RING Supervisor Management Portal

The RING Supervisor Management Portal offers a unified management UI to manage and monitor RING's S3 and file interfaces.

RING Installer

The RING Installer emphasizes simplicity by allowing RING deployments to be installed with a target time of under an hour.

Index and Search

No built-in index and search feature is available. An external database (MongoDB) contains metadata extracted from Scality and indexed for searchability. The feature is called Extended Data Management or XDM. Data is extracted from multiple Scality systems to create a global metadata search capability.

Data Reduction

No data reduction capability is available currently Scality.

File Access

For file access, Scality provides the Scale-Out File System (SOFS). SOFS is a file system supporting parallel access with support for NFSv3, CIFS/SMB (Samba 3.5), AFP, and FTP. For the file system, files are mapped to objects on the storage nodes. The control information for files including metadata is stored in the Mesa Database. The file system is executing on Connectors (sometimes referred to as access nodes).

- On each node, single namespace
  - Files stored in object buckets, metadata in database
  - Geo-models: 3 site active, 2 site active with witness, active/passive
- Unlimited files sizes, 4B filesystems
- WORM mode after set period of time – based on a policy setting by the administrator

Features of SOFS File Access

Included with the file access:

- Global namespace
- Multi-tenancy
- Unlimited number of files and file size
- Auto-scaling of filesystem size
- Parallel transfers
- Distributed metadata in database for performance
- Quotas
- Versioning with self-service undelete
- Customizable ACLs
- Policy engine – retention, multi-site protection
- Tiering from flash devices to disk.

Significant Announcements

- Dec 2023 – start with announcement coverage

Futurum Group EvaluScale – Object Storage

The Futurum Group product review methodology "EvaluScale" assesses each product within a specific technology area. The evaluation of each product is based on its capabilities, with capabilities for each technology segment grouped into distinct categories. The products are evaluated based on the following 4 criteria categories:

- Performance / Capacity
- Basic Functionality
- Advanced Capabilities
- Ability to Execute

The full Object Storage EvaluScale can be found here.

The Futurum Group Opinion and Outlook for Scality Ring

Scality RING gives customers the opportunity to build a high-performance object and file storage system. Other vendors can OEM the software and build integrated systems based on Scality. The performance expands usage beyond archive or content repositories. Scality has the opportunity to work with integrators on even more high value solutions.

Having been an early entrant into the market with object storage, Scality has had time to mature the system and add advance capabilities. The relationship with HPE has given Scality access to a new set of customers.

Scality continues to add more customers in diverse industries. Support and increasing sales will be their focus areas.