HITACHI
Inspire the Next

Scale out S3 object storage platform, built on microservices architecture that can be deployed as software only or an appliance.

WHITE PAPER

Hitachi Content Platform for Cloud Scale

Architecture Fundamentals

By Hitachi Vantara

November 2021

Overview

With modern object storage use cases, the demand for scale and performance has never been more intense. A new solution is required to deliver intelligent and dynamic data services that will power users' journeys to digital transformation for years to come. Hitachi Content Platform for Cloud Scale (HCP for Cloud Scale) is a next generation object store that builds on insights gained from Hitachi Content Platform (HCP). Launched more than 15 years ago, HCP has established itself as a trusted S3 compatible object store and experienced tremendous market success; analysts continue to rank it a leader in head to head product evaluations. HCP for Cloud Scale is shaped by our experience as well as customer feedback from thousands of enterprise professionals who manage their company's data storage needs.

Today organizations are witnessing a transition in both technology and business models that give IT organizations tremendous agility with respect to how they will operate in the future. Businesses have more flexibility in terms of how they purchase and deploy software that includes numerous public cloud sources offering subscriptions for elastic compute and storage. With the advent of IoT and sophisticated business analytics, organizations are prioritizing scale, performance, hybrid cloud capabilities and investment protection.

HCP for Cloud Scale features a software-defined microservice architecture built for limitless scale and deployment flexibility. A key part of our strategy is to provide a centralized governance hub capable of federating data from multiple disparate S3 sources. Through this federated hub, organizations can efficiently collect metadata from all sources and use it to make decisions and take actions according to policy guidance. This independently scalable metadata repository makes securing and governing data from multiple sources possible. Other foundational capabilities include a modernized API and execution model, scale-out policy engines, and an execution and event notification framework that enable seamless integration with 3rd party on-premises or public cloud storage services.

Who should read this whitepaper?

This white paper is for IT and Storage professionals who are evaluating modern, adaptive object storage technologies for their applications. Stakeholders include application users, legal, accounting, and C-level executives who all demand oversite and accountability for what data is stored, where it is stored, and who has access to it.

Concepts and High-Level Architecture

The Basics

HCP for Cloud Scale is a software-defined, massively scalable, object storage system. It uses a **microservice architecture** that distributes code elements across multiple node **instances**. Each instance may be a physical on-premises server, a virtual machine, or a public cloud server instance.

The basic unit of data is an **object**. An object consists of the data payload (e.g., a file) as well as **metadata** that is composed of system-relevant information (e.g., *creation date*) termed as System Metadata, plus an optional set of user-provided key-value pairs termed User Metadata. **Buckets** serve as the containers for objects and provide the mechanisms necessary to control and access them.

To access HCP for cloud scale, an application or administrator needs a user account with an external identity manager (**IdM**), and membership in an **IdM group**. The IdM groups map to **roles** defined by an administrator; roles precisely define group member permissions and can include a mix of data access and system management capabilities.

Users communicate with HCP for cloud scale over various APIs and UIs to store and retrieve objects, establish data management policies as well as manage the system. As an object is **ingested** (i.e. stored, PUT, or replicated) it is written to **storage components**, from which the object can be later retrieved.

High Level Architecture

HCP for cloud scale's software is organized into dozens of autonomous services, each with custom run-time policies and rules. Most services run multiple copies, scaling horizontally to occupy available infrastructure resources. At a very high level, the architecture can be divided into three main functional areas:

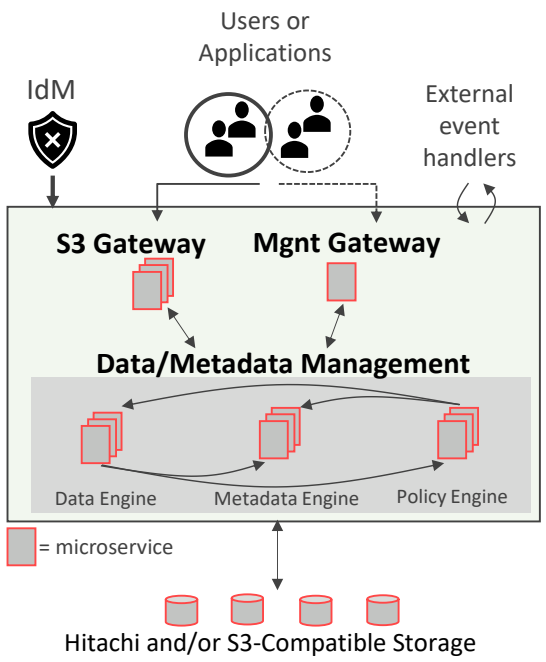


Figure 1. *High-level Architecture*

S3 Gateway(s) handle all application I/O communications. It supports a rich S3 API that is 100% compatible with Amazon Web Services (AWS) S3 API and has API extensions to enable additional functionality not available in AWS to support additional functions

Management Gateway is accessed via a REST API; its functions include system setup and configuration, instance and service health monitoring, reporting, security, upgrades and updates.

The **Data/Metadata Management layer** is a collection of distributed service engines that manage stored data regardless of where it lies. Data can be placed in storage components supplied by Hitachi or S3 compatible devices. Engines scale independently and dynamically to satisfy observed application workloads. Finally, external event handlers are provided to coordinate data flows with 3rd party software and services.

Use-Case Highlights

The use cases for object storage have evolved well beyond the governance and compliance category, which needed immutable storage for rarely accessed data. Here are three modern use cases:

- Batch and Interactive Analytics
- Multicloud XaaS and Hybrid workflows
- High Performance Backup

HCP for cloud scale delivers scale and performance far exceeding earlier generations of object storage. It provides intelligent data management features to handle unstructured data growth stored in **multicloud** environments. For example, most object storage solutions can tier data to public clouds. HCP for cloud scale paves a path for **XaaS** (ANYTHING-as-a-service) with features that make it easy and practical to consume data processing in the public cloud via workflow orchestration. **Interactive big data analytics** can be made more efficient by leveraging the object store's compute capabilities. Finally, HCP for cloud scale's scale-out design not only accommodates terabytes-per-minute throughput, it also offers policies to manage data placement and tools to create and mine metadata for future insights.

S3 in Analytics

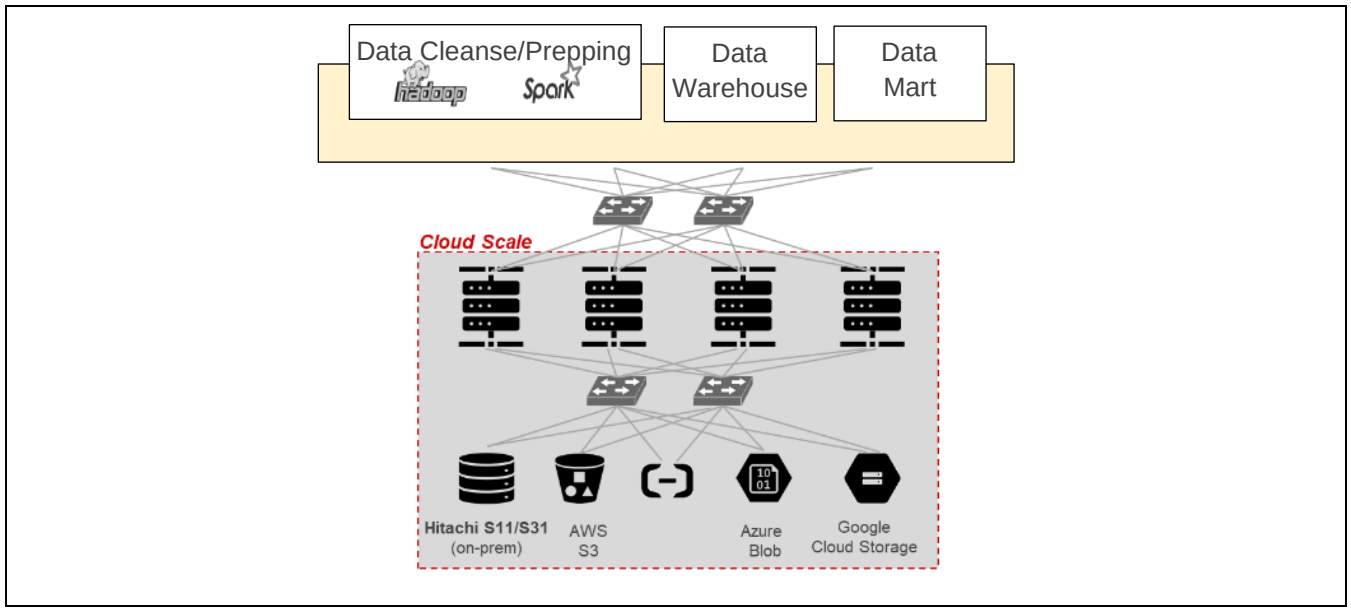


Figure 2. *Scale-out for Analytics Use Case*

Workloads in the analytics space are usually classified as batch, interactive or streaming. Marquee batch products include Cloudera, MapR and Spark. They all seek to produce actionable insights from corporate data, such as web clickstreams, sales data, transaction logs, call center recordings, social media, etc. Business owners enlist data scientists to code SQL expressions that run on a distributed compute cluster to cleanse, join and query data.

A common problem for these products is having more data than the clusters' filesystem can hold. HDFS is particularly inefficient, storing everything in triplicate, with capacity often limited to the disks that physically fit into these data-crunching servers. Scaling storage literally requires adding more compute servers, even though more compute resources are not needed.

A second problem is the infamous scaling limits of the "*name node*" architecture in HDFS. While you can define multiple name nodes for redundancy, they cannot be used for load balancing or scaling file counts. This limits the HDFS index to one node that can only accommodate about 200 million files.

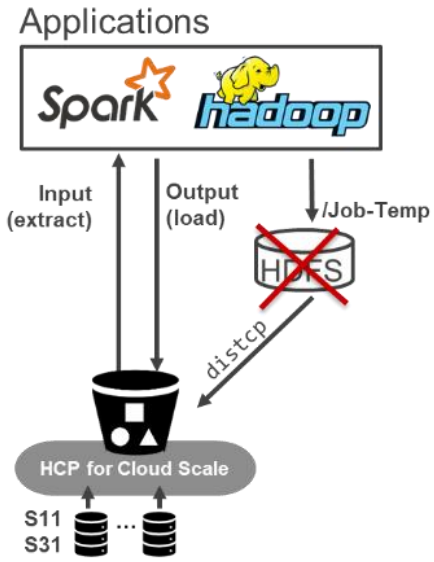


Figure 3. *Scale-out for Analytics*

Key benefits for Analytics:

HCP for Cloud Scale provides the scale and features to build a data lake architecture around an S3 compatible object store to augment or replace HDFS. Its architecture scales in the two dimensions necessary for success: (1) **Scale-out capacity** to accommodate petabytes of historical data that can now grow independently from compute, (2) **Scale out performance** to accommodate hundreds of compute servers which need the object store to deliver the capacity fast enough to avoid bottlenecks.

- Supports the **S3 Select API** to help tools like Spark and Presto prefilter data sets. If datasets can be saved to S3 in parquet format, it preserves their columnar nature. The S3 Select API can then be used to pull pre-filtered sections of data as opposed to the entire file. This vastly reduces data movement and frees valuable compute cycles on the analytics cluster.
- Compatible with the **S3a connectors** built into HDFS, which allows users to easily attach S3 buckets and treat them like HDFS mounts.
- Eliminates the HDFS name node bottleneck which limits the size of system. Instead, HCP for cloud scale distributes metadata evenly across an unlimited number of server instances.
- Stores data using highly efficient erasure coding that provides superior data durability and availability. This technology can survive more disk failures than HDFS, consumes 50 percent less capacity overhead, and spreads data sets across all available disks.
- A "strongly consistent" object store. When an object is written, updated, or deleted, the event is propagated throughout the entire system. This is critical for analytics engines, which divide work in an asynchronous fashion and cannot tolerate an "eventually consistent" solution.

Hybrid Workflows: Mixing On-premises and Cloud

A Hybrid workflow is more than just simple data tiering; it is about leveraging the abundance of cloud-based ANYTHING-as-a-service (XaaS) to achieve desired outcomes at the best cost. Instead of purchasing a software title and using on-premises infrastructure to perform a particular operation, the service may be "leased" in a public cloud as needed. Furthermore, instead of leaving data in public cloud storage, which results in ever-growing monthly storage expenses, a hybrid workflow intelligently transfers only the necessary data between on-premises and public cloud storage and then deletes the unnecessary data to minimize storage and egress costs.

While a good DevOps team can set up data transfers using brute force scripting, the quality of the integrations can vary, and single threaded scripts will not scale with production datasets or adapt to changing needs.

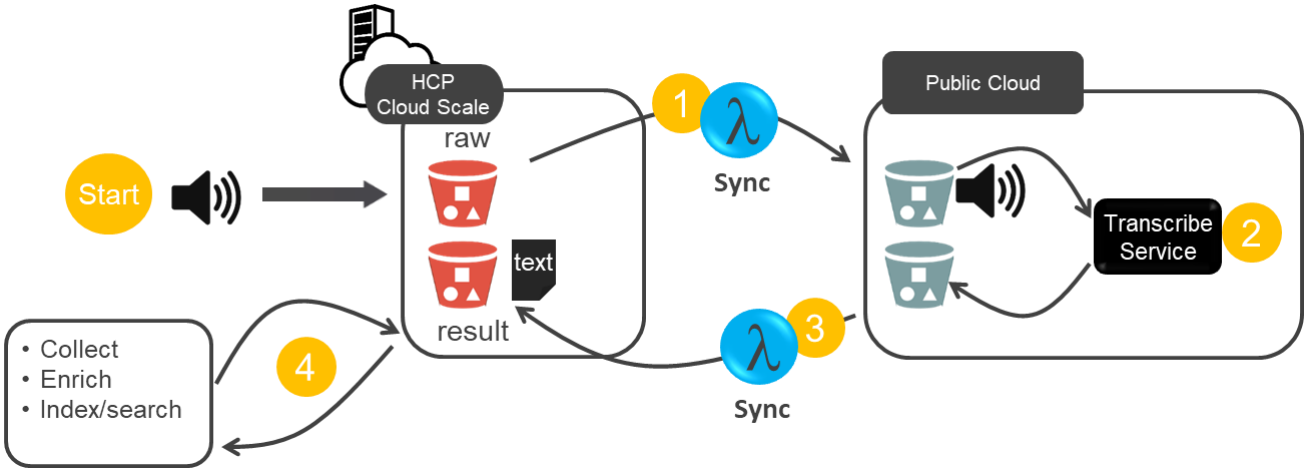


Figure 4. *Hybrid Workflow Use Case – Blending On-Premises and Cloud Infrastructures*

Key benefits for hybrid cloud:

HCP for Cloud Scale offers a variety of built-in features that help users take advantage of the abundance of on demand compute or storage service options that are available. Such services include AWS Lambda (Functions-aaS), AWS Transcribe (Transcription-aaS), AWS Translate (Translation-aaS) and AWS Transcode (Transcoding-aaS).

- Provides a bucket-notification capability that ties into AWS SQS. This allows integration of AWS Lambdas and notifications into workflows.
- Cross-region replication (CRR) can not only replicate buckets to any S3 compatible endpoint (Sync-out; Step 1 in Figure 3), it can also monitor AWS S3 buckets and bring new data back to HCP (Sync-in; Step 3 in Figure 3). HCP for cloud scale can also replicate to multiple endpoints.
- Offers a publish-subscribe architecture that is inherently suited to scale. For example, highly transactional workloads are serviced by multiple copies of the S3 gateway, each of which can asynchronously publish jobs to SQS that can then be serviced by multiple subscribers.
- Federates any S3-compatible bucket and uses them as backend storage, be it an on-premises storage target or AWS S3 bucket in the cloud. This capability is more powerful than tiering and provides the flexibility to choose public cloud storage for initial data placement.

Backup Modernization

Most organizations adopt a hierarchical backup strategy that blends local backups and remote vaults. Local backups retain data from the last few weeks or months, while remote vaults provide long-term retention. Remote vaults are the last resort should disaster strike the local backup solution.

Companies have turned to disk based solutions for local backup because information age applications demand RTO (restore time objectives) measured in minutes. Tapes are mostly unable to meet that kind of RTO. Virtual tape libraries (VTL) have had some success, but these tend to be more costly scale-up disk designs that have a proprietary control plane and inherent performance plateaus. Deploying multiple instances help, but this creates management challenges based on tracking ad-hoc indexes of virtual systems and tapes.

While tape has long been perceived as the undisputed king of low-cost, long term vaults, the current trend is to use economical web services like AWS Glacier as a repository for backup datasets. A benefit of maintaining data in the cloud is that in the event that a disaster damages local compute capabilities, the restore environment could also be based in the cloud. However, this strategy requires the backup software to move data multiple times as it ages, across on-premises systems and cloud services.

Key benefits for backup:

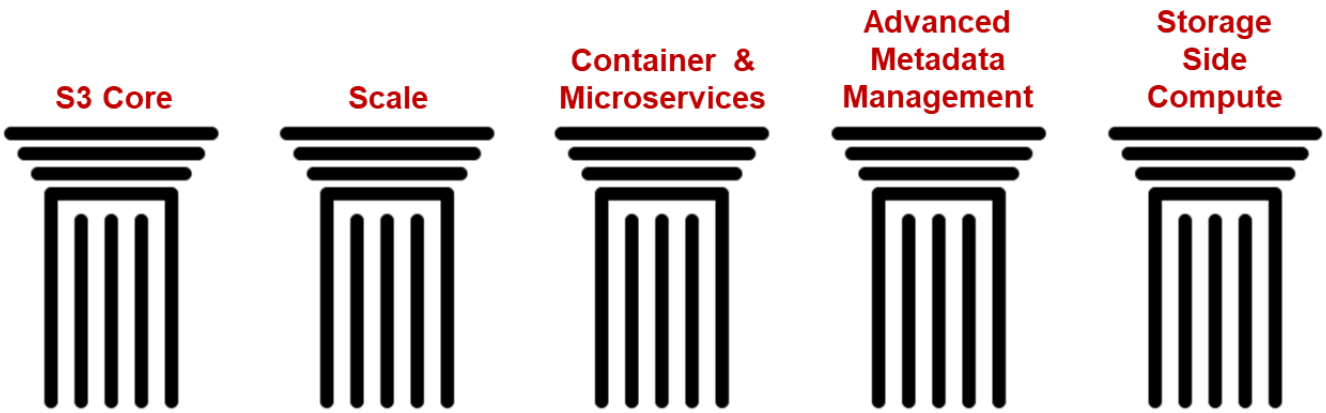
HCP for Cloud Scale accelerates performance for LAN based backup environments and can move large amounts of data within a strict backup window to achieve aggressive terabyte-per-hour (TB/h) goals.

HCP for Cloud Scale also offers new opportunities to monetize the stored data if backup vendors can preserve file instance information. Approaches may vary, but if provided, the backup data could be scanned by analytics engines for analysis and reporting.

- Federate 1000’s of commodity disks to meet any capacity requirement
- Cost effectively protect data with low overhead erasure coding; maintains data availability despite sustaining multiple node or disk failures.
- Performance that scales to the potential of the allocated infrastructure. E.g., for the hardware infrastructure to support 1TB/Min is 10x the infrastructure to support 100GB/Min.
- Policies that leverage metadata to automatically place data to the appropriate S3 endpoint. This frees the backup software from moving data multiple times to ensure long term protection or disaster recovery.
- Manage trillions of files and versions, making it an ideal platform for copy data management software or differential backups.
- Backup with S3 API, which allows authorized applications and ethernet-connected users to restore data.

Five Architecture Design Pillars

New use cases are challenging object stores to scale across all metrics – petabytes of capacity, trillions of files and a seemingly insatiable appetite for throughput and I/O. While fundamentally important, scale is just one of many key factors. Large deployment success will require products and tools that provide flexibility and advanced data management capabilities. Below are five key principles that influenced HCP for Cloud Scale's architecture.



S3 Core

The acceptance of public cloud, and specifically the success of Amazon's AWS S3 API, has made it the de-facto standard for I/O to object storage, with adoption by virtually all object storage vendors.

HCP for cloud scale has an architecture designed for object storage at the core rather than file storage. When file system semantics are required, our strategy is to utilize a gateway for the POSIX presentation layer.

HCP for cloud scale strives for 100% S3 compatibility with every API call implemented on HCP for cloud scale. Moreover, the S3 CLI and SDK tools published by Amazon interoperate with HCP cloud scale without modification. Our goal is that applications are portable and can seamlessly straddle on-premises and cloud infrastructures.

Scale

Scaling is the ability to effectively manage a growing amount of work by adding resources. Scale is hard to perfect. One strategy is to forgo it and pivot to concentrate on modest deployments that scale up to a point. If an enterprise customer needs more resources, advocate for separate deployments as necessary to scale to the required metric. This "sprawl" approach will lead to underutilized resources, create monitoring and management challenges for IT staff, or force client applications to coordinate I/O across multiple endpoints.

HCP for Cloud Scale divides the scale problem into several parts, consciously choosing to tackle metadata and data scaling separately. Both employ automatic and intelligent partitioning strategies to individually manage a finite amount of data, but together deliver a singular global namespace spanning literally 1000's of servers and disks.

Metadata Scale

HCP for Cloud Scale provides a scale-out directory index that maintains associative metadata about every ingested object. The index is non-relational, providing speed, persistence, reliability, and strong consistency. By consolidating all metadata and making it readily available, HCP for cloud scale efficiently carries out lifecycle operations and provides a springboard for data governance and insights.

The partitioning scheme ensures index lookups remain deterministic even as the object store scales to accommodate increased object count. As the business value of data diminishes over time, it may be necessary to reduce the associated storage costs. Oftentimes, this entails moving data to storage with lower performance and availability. To ensure metadata operations are not impacted by data placement choices, HCP for cloud scale utilizes independent layers (one for metadata and one for data), with independent protection domains.

This approach also minimizes impact on data ingest/recall performance. Generally, metadata I/O activity inside an object store exceeds data I/O. A dedicated layer allows background services (e.g., file listings, lifecycle policies, disposition policies) to iterate over metadata, with no need to recall data.

Data Storage Scale

HCP for Cloud Scale effectively spreads application data across multiple storage endpoints (disks) in a fashion that delivers a namespace with aggregate capacity, cumulative performance, and fault tolerance. Endpoints can be Hitachi Content Platform S series storage or any S3 compatible device. The individual endpoints manage a finite amount of data and run autonomous to other endpoints. HCP for cloud scale combines multiple similar endpoints into a pool and apply random placement algorithms to achieve horizontal performance scaling.

Workloads generating smaller files may require metadata scaling to manage object count and IOPS, while larger files may require data scale in terms of capacity and throughput. Customers can size the proper infrastructure ratio to match their precise needs.

Container and Microservices

The value proposition for software defined storage typically highlights: a reduction in costs based on allowing infrastructure to be sourced from low cost suppliers, improved resource efficiency through deployment in hypervisors, or transitioning capital expense (capex) business models into a more agile operational expense (opex) model through the use of public cloud infrastructures like AWS EC2 and S3. All are true to some degree, yet all of this is evolving as well.

The latest variable in this conversation is the emergence of container orchestration services as an alternative to whole machine reservations. Like their hypervisor cousins, container technologies abstract hardware but consume far less overhead to operate. Instead of requiring an entire operating system (~ 2GB) for each virtual machine, containers offer a thin runtime environment (~ 20 MB).

Well-designed software defined storage solutions can load onto Linux or public cloud container services (e.g., EKS) and consume infrastructure with superior efficiencies and thereby reduce costs. HCP for Cloud Scale can be easily deployed in multiple public clouds, on bare metal or in hypervisors. Unrivaled efficiency is achieved through an innovative microservice design and careful logic isolation, where container elements run autonomously on different machines and communicate with each other via event queues and notifications.

Modular Approach

Each service is predominately coded in one major language that is best suited to a particular task or function. With this granular software development approach, it is much easier to isolate, modify, test, and deploy bug fixes and product enhancements. The modularity of HCP for cloud scale's software enables an accelerated release

cadence (quarterly) compared to traditionally designed applications while also streamlining version coordination, providing customers with greater control over when an update should be applied.

Publish/Subscribe Communications

It is important to understand that using a container technology does not by itself guarantee scale. HCP for cloud scale combines container technology with carefully constructed design across the system, including the use of optimized data structures, algorithms, and messaging protocols. Virtually all services follow a publish/subscribe architecture to facilitate loose coupling between container services. When one container service has a task to do, it can publish the work on a queue. Scaling is achieved by spawning multiple receiver services to compete for work, dividing it and linearly increasing the execution of a task backlog.

Precisely Scale Any Service

Services can be programmatically expanded (duplicated) as demand increases. The system management UI provides the ability to inspect and individually alter the runtime policies of any service in (near) real time. Such changes can be permanent or temporary. Moreover, the virtual resources allocated to a service can be similarly expanded or reduced on demand.

Hardware Abstraction

HCP for Cloud Scale incorporates micro-virtualization, where literally hundreds of containers may cohabitate an operating system and share host resources. All interactions with the operating system kernel are read-only, and all storage mountpoints are 100% private to each container. The software can be deployed on virtually any enterprise class Linux distribution, including AMI instances from AWS.

Advanced Metadata Management

A defining characteristic for object storage is the ability to attach metadata to objects. This process enables the system to enforce protection features such as file retention or legal hold capabilities. Next generation object stores must provide integrated index and search technologies to enable fast isolation of object sets that align to a policy, enhanced access protection beyond simple ACL, and easier ways to view the same data by different users. This should be possible whether the applications and storage are located entirely on-premises, in the cloud, or a hybrid deployment connecting multiple clouds.

HCP for Cloud Scale provides built-in data management to govern data. Given policy guidance, it can make decisions and take actions to manage that data directly, without the need for an external data management application that might complicate and slow data access for the sake of data governance.

Versioning saves new copies of an object when changes are made and allows organizations to meet compliance standards. However, preserving an infinite number of versions not only escalates into a capacity problem but must also be considered in the context of an individual's right to dictate the terms for data access or retention. Failure to maintain regulatory compliance can lead to stiff fines and/or loss of business. Moreover, proving full visibility and control over stored data is required by regulators.

Centralized enforcement lets you replace fragmented, inconsistent, or proprietary application-based gatekeeper duties. Policies for consideration include:

- Data Placement – For data protection, replication, cost reduction, processing needs (lambda, XaaS)
- Secure – Control access based on content, PII (personally identifiable information), rate of access
- Lineage – Manage data according to arrival date, location, who accessed, transformation history, author
- Govern – Data retention, legal hold, GDPR or HIPAA compliance, geofencing
- Categorize – Customized or filtered data views, cafeteria-style management policies

- Dispose – Automated disposition, audit logging, support for right-to-be-forgotten (RTBF), cryptographic erase

Storage-Side Compute

Modern use cases demand elastic compute capabilities. Consider the compute load triggered by the S3 Select API. This API can scan semi-structured objects and return sections matching SQL search criteria. When deployed on-premises, HCP for Cloud Scale taps into elastic compute resources using scalable microservices to process the data. It can also access off-premises compute resources using queuing notification built into the product. In both cases the processing resources are invoked only when needed, making infrastructure available to other activities during downtime.

Additionally, organizations routinely set up workflows to mold raw data to match a schema using simple information like date formats, currency, or variable castings. Other curation activities can include:

- Transform – Anonymize, cleanse, or curate data (e.g. PDF to PDF-A)
- Classify – Annotate data with metadata, organizational relationships, customer, or application identification

These are easily accomplished through integration with Hitachi Content Intelligence. Together it is possible to create complex discovery workflows that help organizations adopt business processes to meet country or industry specific regulations. All such work requires a strong storage-side computing resources.

Architecture Deep Dive

HCP for Cloud Scale is designed for massive scalability. During its design, the team carefully considered all assumptions and tradeoffs made in prior generations of object store systems. HCP for cloud scale is built on a microservice foundation on which loosely coupled containers run on loosely coupled server instances. Containers run in effective isolation from one another, yet together they provide a singularly managed global namespace that can deploy on virtually any server hardware or public cloud service.

The collection of services that run on any HCP for Cloud Scale server instance are organized into two groups:

- **Cluster Management Services** monitor the health and continuous availability of all application services loaded into the system. Furthermore, the Cluster Management Services, and not the system administrator, provide the framework to scale the application services that are controlled by the system.
- **Application Services** include essential business logic for the object store to manage data at scale. They deploy as serverless containers with their own virtual resources and are controlled by the system administrators. The **S3 Gateway** service handles User data access. Metadata and Data management are handled by a set of services such as the **Metadata Database**, **Partition Manager**, **Metadata Cache** and **Policy Engine** services. The **UI/MAPI** service is the main interface for System Administrator access. In addition to the above, other essential services such as **Identity Management**, **Logging** and **Event** management services handle access control and event notifications.

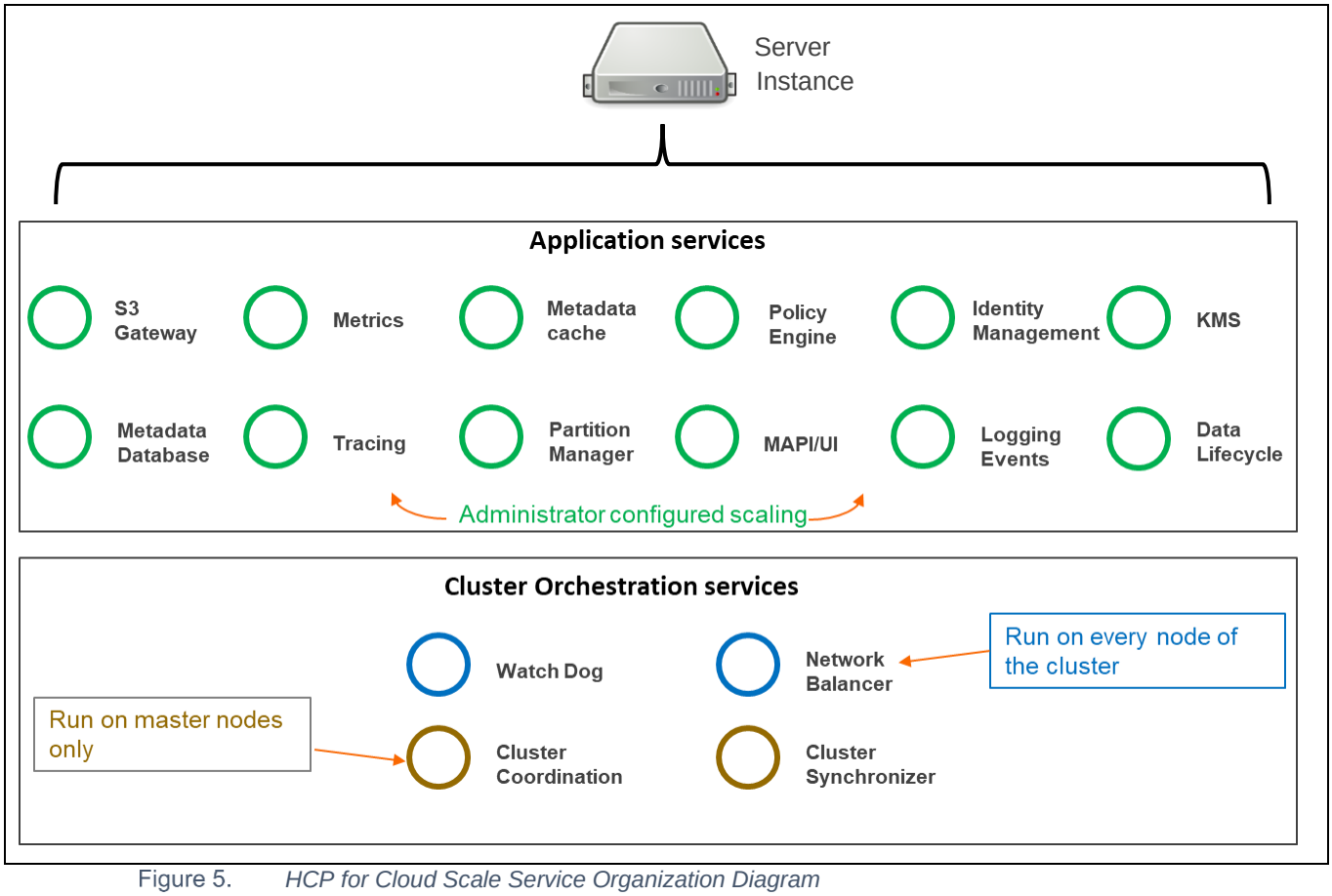


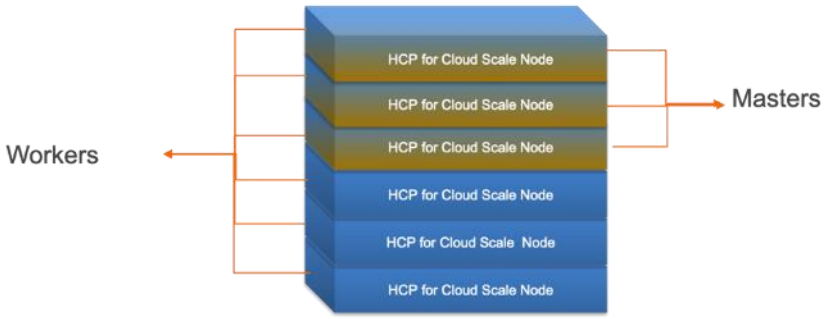
Figure 5. *HCP for Cloud Scale Service Organization Diagram*

Cluster Management Services

Functionally, these services monitor the health of all services and will dynamically restart a failed service on any healthy server instance. This is done by ensuring the number of running copies of each service matches the thresholds established by the system administrator.

The server instances (or nodes) comprising an HCP for cloud scale deployment are classified as masters and workers. Systems must have a minimum set of three **Master** nodes. A loose distinction between **Master** and **Workers** is made to provide the highest level of fault tolerance**.** (*See section Node Resiliency for more details*).

All nodes can be worker nodes, and thus able to host application level services. Master nodes have additional responsibilities for cluster level fault tolerance and are distinguishable by two services called **Cluster Coordination** and **Cluster Synchronization**. The Cluster Coordination service makes decisions about if and when a failed service must relaunch. The Cluster Synchronization service collects and synchronizes the running state of all service instances across the system.



Again, the term Master is loosely defined here to illustrate the special cluster management capabilities that they possess. Master nodes are not precluded from running Application services for optimal resource utilization of the nodes. The ratio of master nodes to worker nodes can be 1 to 50.

All nodes include a lightweight **Watch Dog** service whose sole responsibility is to report the health of services to the Cluster Synchronization service.

Node Resiliency

Multiple master nodes provide the redundancy necessary to ensure cluster availability. For example, a 3-master deployment will continue to operate after a Master node failure as two other Master nodes will remain to monitor and guard the system.

Any number of worker nodes can fail. The master nodes simply relocate the services that were running on the failed worker nodes to any surviving (and healthy) nodes. To provide some context, a 10-node system could sustain the loss of all 7 worker nodes, in which case, services will simply be relocated to the 3 available master nodes, provided they have enough resources to host them.

Service Resiliency

Should any Application service fail, the master nodes will dynamically detect and relaunch that service on any available node. In the event of major resource failures, System Administrators are alerted when the running instances of a given service cannot be matched to configured values. The system will automatically adjust back to the expected configuration when the failed resources, such as nodes or network elements, are replaced.

Application Service Scaling

System Administrators can dynamically adjust the running number of any Application service. The Clustering services will discover the changes made by System Administrators and adjust the Application services to the expected configuration. For instance, System Administrators can appropriately tune the number of Metadata versus S3 Gateway services based on known workload profiles. Indeed, an administrator may choose to run Metadata services on all nodes of the system but include only one instance of the S3 Gateway for small object IO, and vice versa for large object IO.

Application Services

HCP for Cloud Scale's software stack is organized as a set of microservices, each responsible for a specific function and deployed and managed using standard container orchestration technologies.

Figure 6**Error! Reference source not found.** illustrates the functional architecture of HCP for Cloud Scale. The **Network Balancer** is a proxy for all incoming user traffic for that node, and balances the requests across all available Application services that can handle the request. For example, an S3 request arriving on Node 3 might get routed to an S3 Gateway on Node 4 by the Network balancer to achieve optimal user request balancing across the system.

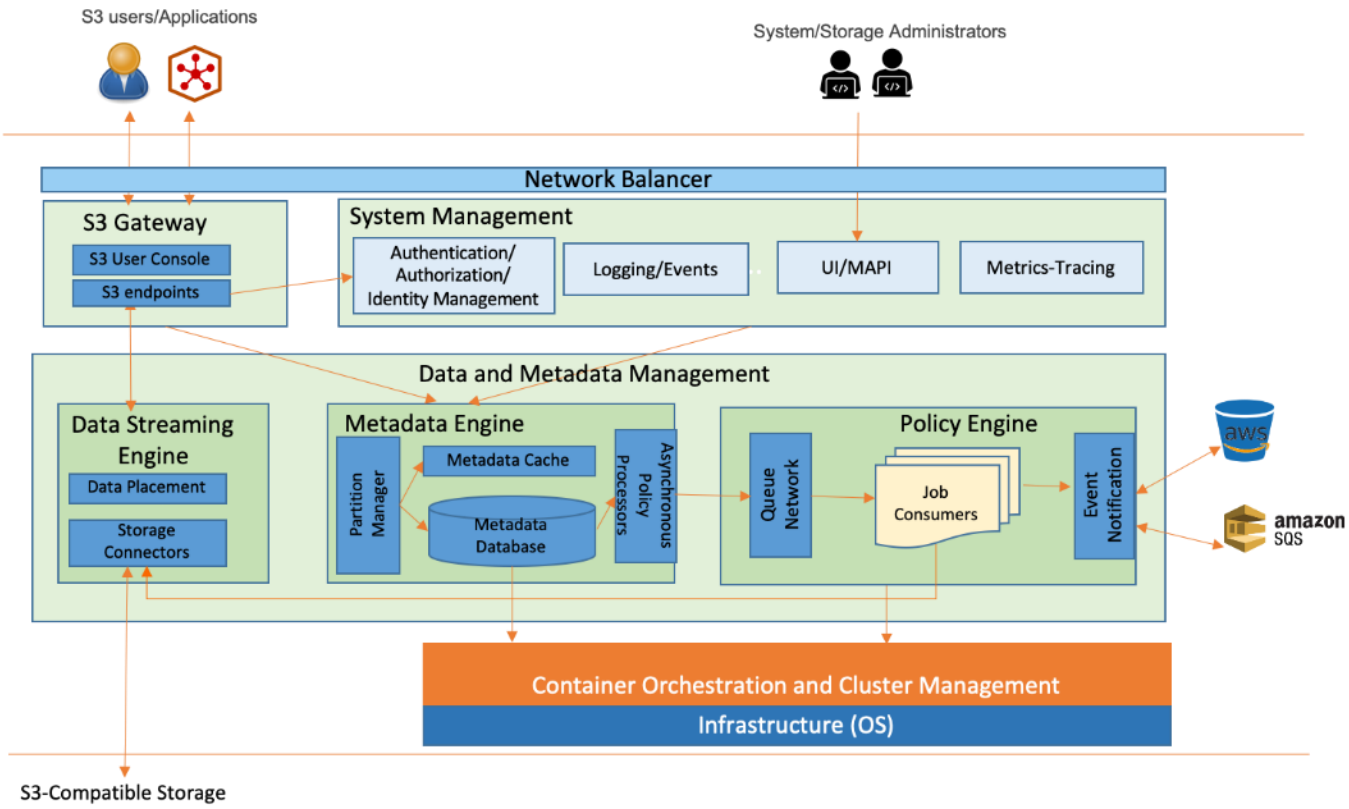


Figure 6. *Cloud Scale Functional Architecture*

System administrators manage HCP for cloud scale using one of many user interface (UI) facilities, including the **CLI (**command line interface**), MAPI (**Message Application Management API which is RESTful**)** or the **GUI (**Graphical User Interface**)**.

Users and Applications manage their data using the standard S3 API (Application Programming Interface). S3 application I/O terminates at the **S3 Gateway.** Users can use the built-in **S3 User Console** for managing buckets or else leverage 3rd party S3 compatible clients such as AWS CLI, Cloudberry, or S3 Browser.

Central to the design of HCP for cloud scale is an extensible set of **Metadata and Data management services** that are carefully composed to scale and connect with multi-cloud infrastructures. With integration tools such as queuing and event notification engines, HCP for cloud scale makes it possible to create data management workflows crossing multiple clouds.

Data ingested by users and applications is stored on one or more attached **S3-compatible storage** targets including the on-premises HCP S series storage nodes or S3-compatible public cloud storage services such as Amazon Web Services (AWS) S3, Microsoft Azure or Google Cloud Storage.

S3 Data Access

The **S3 gateways** are the main entry points into HCP for cloud scale for application data access. They provide **S3 endpoints** to process incoming client requests and coordinate with user and data management container components in HCP for cloud scale. Functionally, the gateways terminate S3 protocol, extract the file and store it in one of the available federated S3 storage targets as objects. For every object stored, the **Metadata Engine** is called upon to record System metadata information, such as the ingest date and owner. Additional Custom metadata (key/value pairs) can be included to guide lifecycle management or governance policies established by the organization. Every new S3 put, post or update will result in a metadata entry being replicated across at least 2 other nodes prior to command acknowledgement, which ensures redundancy and strong consistency. This combination of data and metadata is called an **object**.

A typical system will run multiple copies of the S3 Gateway to process the I/O load; an integrated but basic network balancer service is used to evenly distribute incoming HTTP(S) requests across all available S3 Gateways.

The S3 Gateway consults with the **System Management** Authentication service(s) to authenticate users and validate they have proper permissions to complete the transaction.

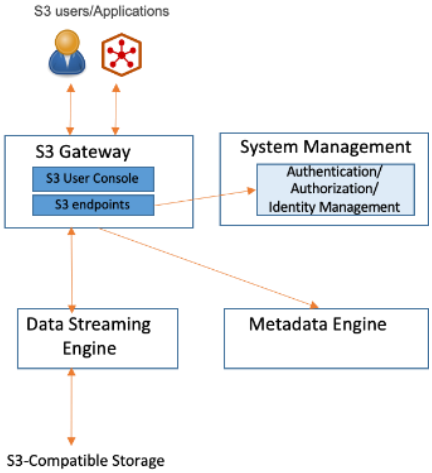


Figure 7. *S3 Data Access components.*

Data Streaming Engine

The S3 Gateway works with the **Data Streaming Engine** to find the optimal placement location across the available storage backends. Several optimization principles are employed during this transaction to reduce I/O latency:

- The number of database transactions that result in disk I/O is optimized by sorting updates as inline or post-process (deferred). Any complicated processing such as updating secondary indices for the ingested object is deferred for background execution.
- All necessary configuration information such as the user configuration and S3 bucket list are cached to minimize disk I/O. The authentication token for a session is also securely cached.

- A random distribution data placement algorithm is employed to quickly identify an available storage location for a new object.
- Data is directly streamed from a user to a backend storage target with minimal buffering on HCP for cloud scale.

S3 Gateways

By default, HCP for Cloud Scale runs at least one copy of the S3 gateway service for the whole system. Based on the use case, the S3 gateway service can be scaled by administrators to run multiple instances. The service is responsible for listening and terminating S3 operations sent via HTTPS (port 443), or HTTP (80) if enabled. One of the many factors that determine achievable bandwidth or I/O performance is the total number of gateways working in parallel with service applications. Other factors include the quantity and speed of the network connections installed in the server.

S3 API and Extensions

A modular and iterative software development philosophy aligns well with an API which itself evolves regularly. S3 Select, Object lock and replication filtering were all added in last 24 months. HCP for Cloud Scale includes these with valuable but elective extensions that remain fully compatible with all AWS S3 client tools. For instance, HCP for cloud scale supports multiple replication and notification targets for a single bucket, while AWS currently does not. Our customers also felt the AWS S3 API could be improved in several other areas, such as compliance. Even with such enhancements, our gateway implementation strictly adheres to AWS S3 API syntax and is fully compatible with CLI and SDK tools published by AWS. Our enhancements generally focus on JSON configuration inputs attached to S3 commands; these enhancements pass unhindered through these tools.

System Management and Monitoring

HCP for Cloud Scale provides system administrators with precision delegation tools to control user access to resources, monitoring and configuration. The following subsections describe the system management functions offered by HCP for cloud scale in detail.

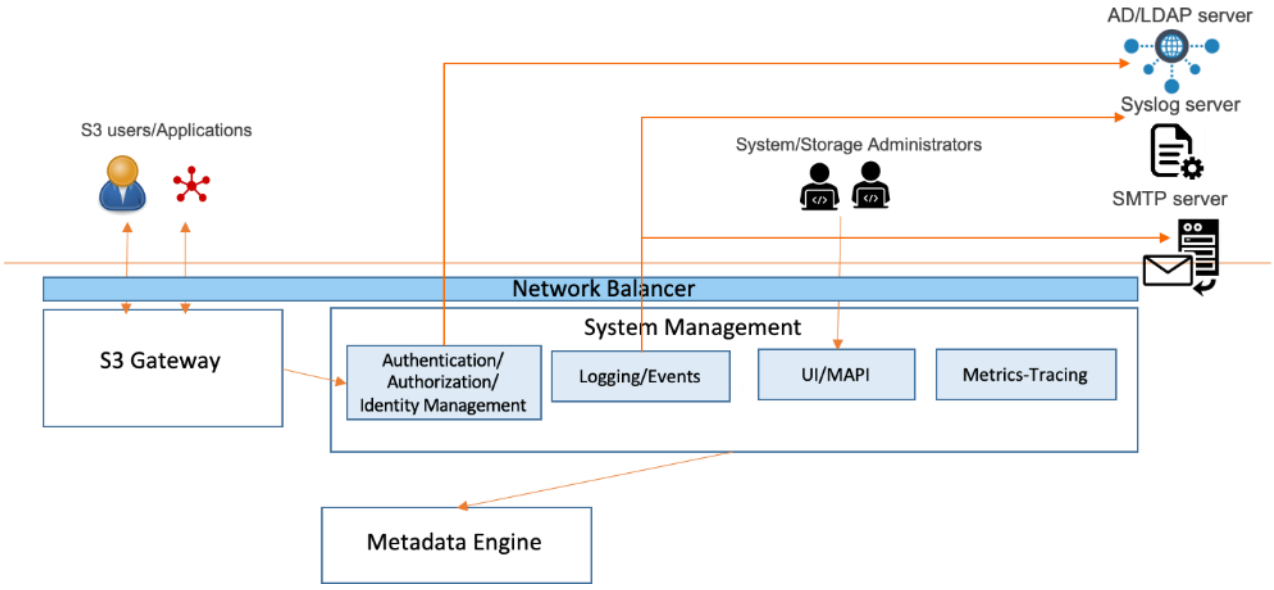


Figure 8. *System Management Components in HCP for Cloud Scale.*

Access Control and Identity Management

The **Authentication**, **Authorization**, and **Identity Management services** integrate with external Active Directory (AD) and/or LDAP Identity Providers to provide authenticated access to HCP for cloud scale. This enables IT organizations to enforce consistent security policies and provide a Single Sign-On (SSO) experience. Users can securely authenticate and gain access to all permitted systems by logging in once with the deployed directory service. The Authentication service in HCP for cloud scale is extensible to accommodate future types of Identity providers.

A given HCP for Cloud Scale system can register with multiple directory server domains or multiple directory servers. However, each user within the system must be associated with only one of the attached directory domains to prevent conflicts.

S3 Authentication

In keeping with 100% S3 fidelity, HCP for Cloud Scale requires users to generate access and secret keys to perform S3 operations. The process of generating these keys requires an account with an identity management system. Specifically, the process requires the user to first authenticate with Active Directory (AD) or Lightweight Directory Access Protocol (LDAP), and secondly to have appropriate group membership with S3 access rights granted by the Administrator.

Roles and Permissions

HCP for Cloud Scale provides a mechanism to divide access and configuration authority among different groups of users. This process begins by creating a **role** and subsequently assigning permissions such as those shown in Figure 9.

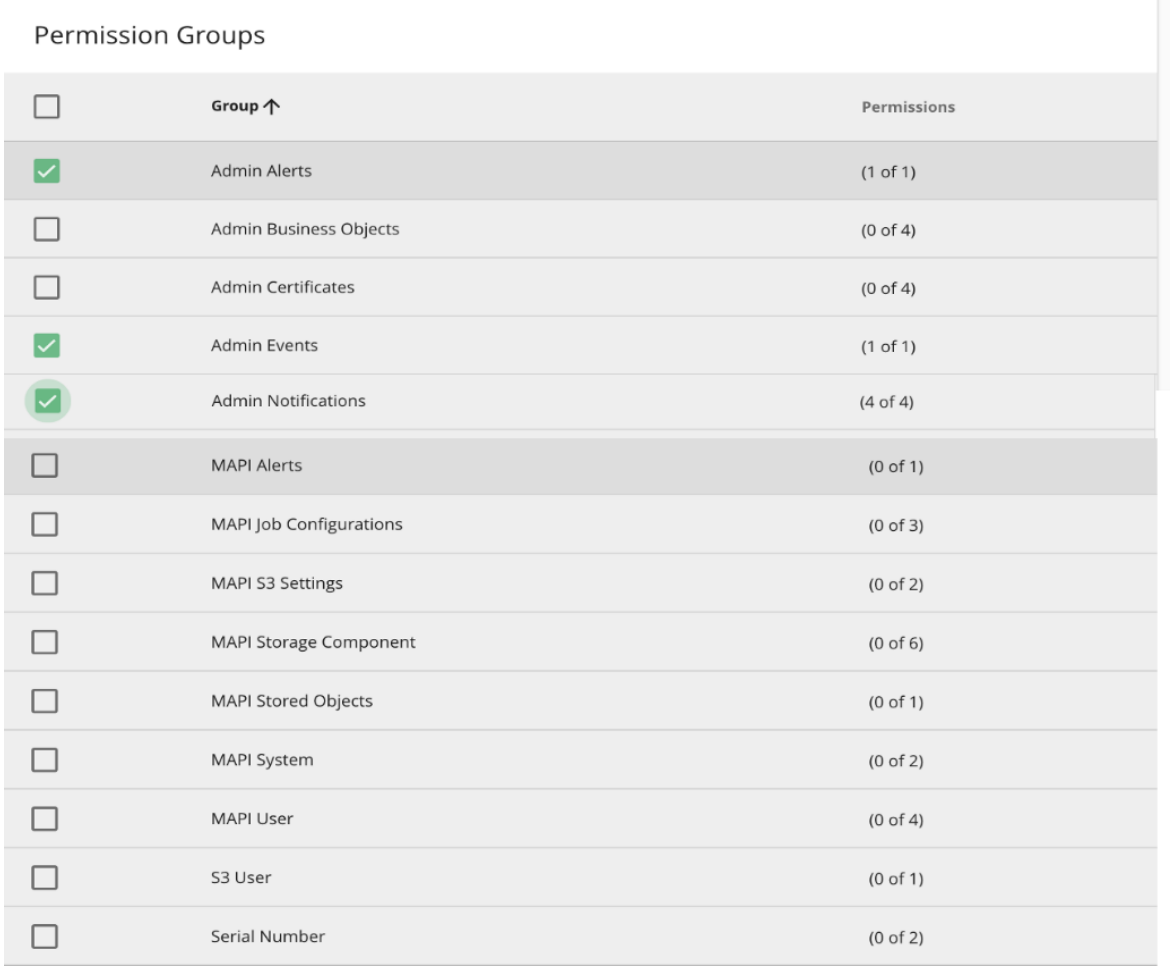


Figure 9. *HCP for Cloud Scale Permissions*

Each role is then mapped with one AD or LDAP group. For example, a “privileged user” role could be created with literally all permissions available, or an “S3 user” role could be restricted to just S3 API, and so on. Users in these groups will only have capabilities explicitly assigned to that role. This allows organizations to compartmentalize sensitive configuration elements of the system. For example, an IT organization may wish to separate the administration of storage versus network versus security versus user and application resources.

It is worth noting that Administrator roles will generally not have access to user data if you limit their permission selections to “MAPI user” and “MAPI S3 setting”. For example, a security administrator within the organization could be restricted to the “Admin certificate” permission that would allow them to update the cryptographic certificates used in the system, but with no access to data.

At system installation time, a “security” user role is created, with all available permissions. To ensure strict best practices for security, only one user within the system can be associated with this role, thus restricting such privileges to just one user. All other users can be assigned a subset of the available permissions.

User Interfaces and Management API (MAPI)

HCP for Cloud Scale provides a collection of interfaces supporting IT automation and ease of management. These include Command Line Interface (CLI), Management API (MAPI), and a Graphical User Interface (GUI).

MAPI

Along with the administrator CLI, a rich set of RESTful Management APIs support IT administrators who are interested in automating their tools. The HCP for Cloud Scale MAPI is therefore extremely rich in that all storage and system administration that can be accomplished through the CLI or GUI can also be accomplished via MAPI.

CLI

The HCP for Cloud Scale administrator CLI (also termed as “admincli”) is an extension of MAPI and is one of the methods through which an IT administrator can manage application service provisioning and scaling, and quickly monitor service health.

GUI

An intuitive web-based GUI enables administrators to easily manage the system and storage without specialized skills. Along with a wealth of capabilities for visualizing various metrics, the GUI also includes a facility to look up necessary reference documentation and allows for test execution of all supported Management APIs.

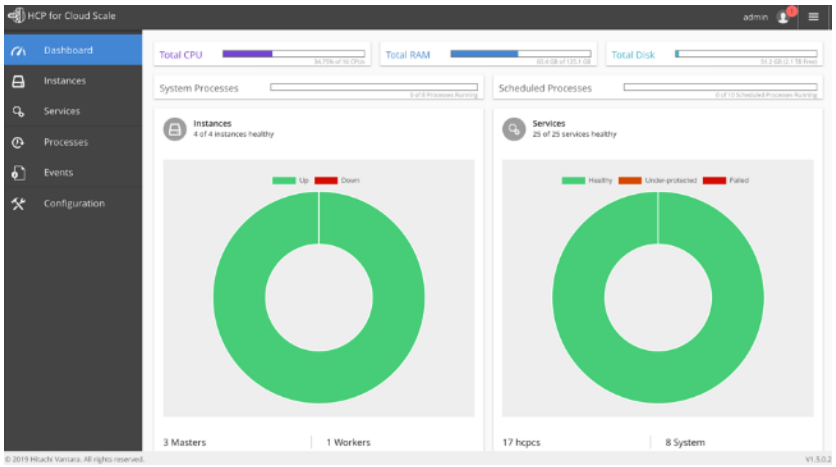


Figure 10. *Cloud Scale's System Management UI.*

Logging and System Administration Events

HCP for Cloud Scale includes a logging framework in which every microservice maintains log files that are rotated and periodically archived. These log files are indispensable for troubleshooting and debugging. Users with the proper permissions can use a configurable download utility offering the option to download all logs or limit the scope to a specific node or service.

System events can be delivered to a remote syslog server. Events can include informational and/or errors which may require more immediate attention.

System monitoring through Hitachi Remote Ops

Real time system health monitoring through Hitachi Remote Ops is included with HCP for Cloud Scale. This phone-home integration can notify Hitachi Support teams or customers and recommend corrective actions. It also facilitates more sophisticated visualization and analysis of the system.

Metrics and Tracing

HCP for Cloud Scale has built-in metrics and tracing utilities to promote better performance assessments and faster resolution of issues. All container services generate logs, metrics and events that can be listed or visualized with graphing tools. The metrics collection service utilizes the open source Prometheus toolkit. A powerful programmatic query interface allows administrators to create complex expressions to gain a deeper understanding of system performance and activity.

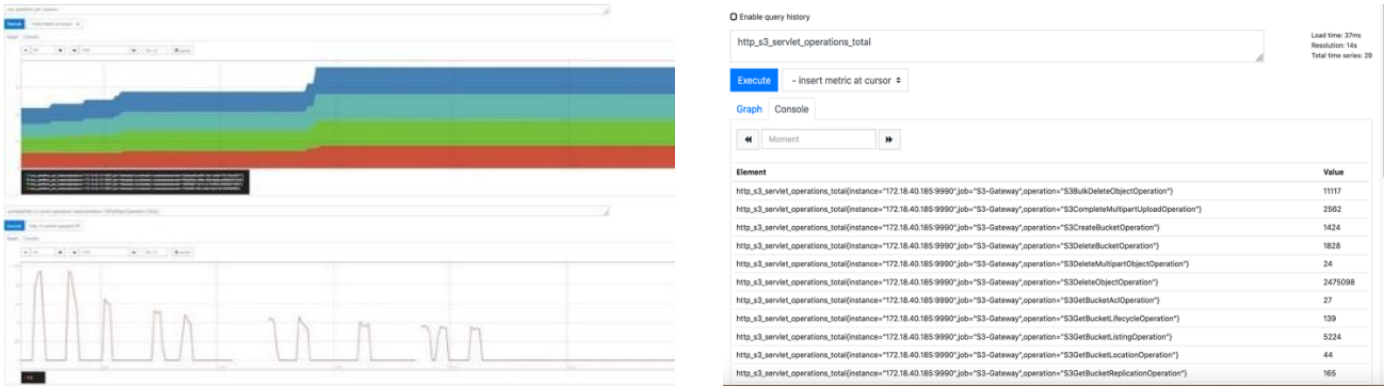


Figure 11. *HCP for Cloud Scale's Metrics Query and Visualization Interface*

The integrated tracing utility was built using the Jaeger toolkit. The tracing framework makes it easy to follow data flow that may span multiple nodes and services (see Figure 12). It presents an intuitive interface to visualize the flow, spot errors in data path across billions of objects, and discover granular latency metrics for every stage of the flow.

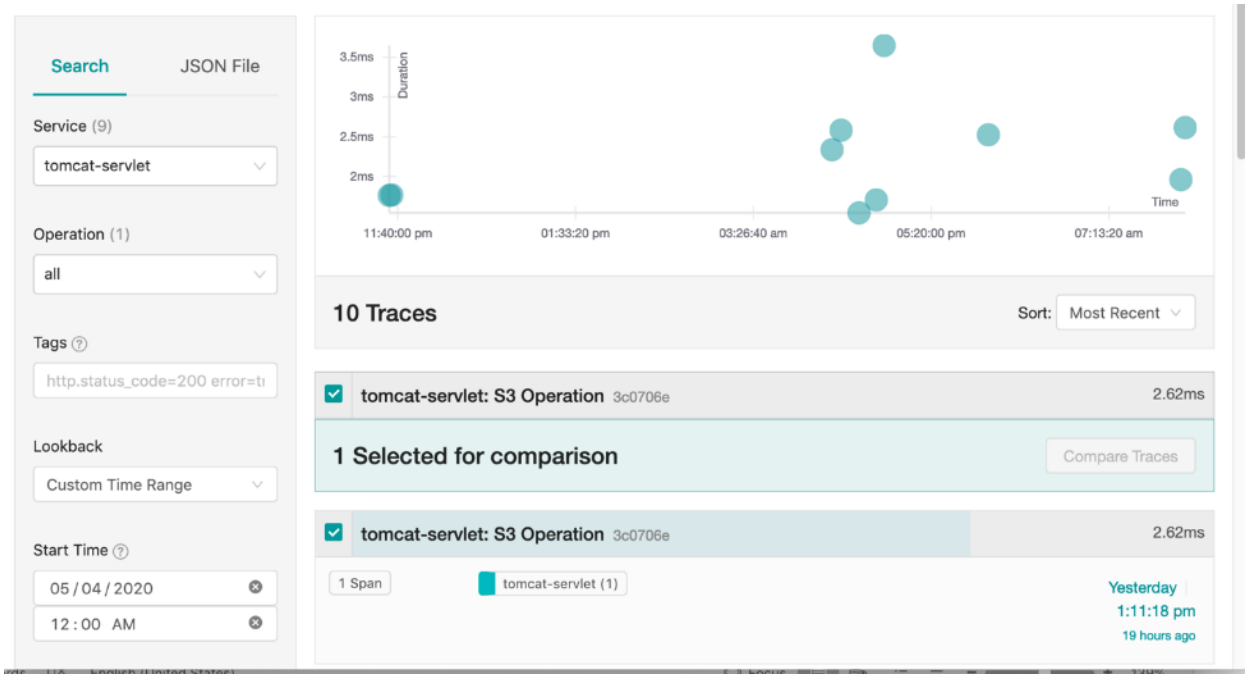


Figure 12. *HCP for Cloud Scale's Tracing Query and Visualization Interface.*

Metadata Management

The Metadata Management engine is unique in the industry and a fundamental investment area. Its importance becomes evident once massive volumes of data begin to be stockpiled. IT departments must then tackle securing, governing, and providing a unified management view of data spread across departmental and multi-cloud silos. Through scale, the metadata engine was designed to capture all of the metadata necessary such that, when combined with advanced and customer specific policies, the data itself can be managed directly where it is stored. HCP for Cloud Scale provides organizations the tools to exploit the power of metadata to the fullest and delivers a competitive advantage for solving advanced data management problems at scale. This is referred to as Intelligent Data Management (IDM).

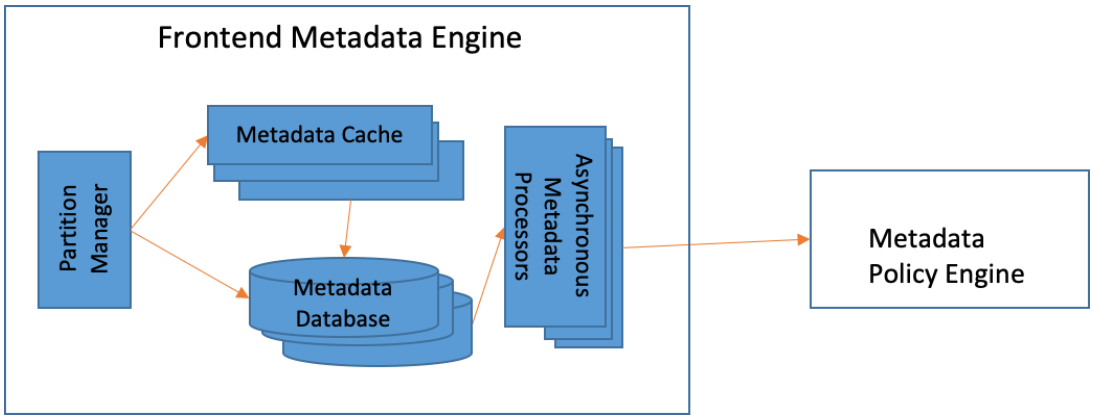


Figure 13. *HCP for Cloud Scale Metadata Engine.*

The **Metadata Database**, **Metadata Cache** service, **Partition Manager,** and the **Asynchronous Metadata Policy** (see Figure 13) components work together to collectively provide the necessary metadata functions.

The following sections describe the design and unique strengths of each of the components that ultimately comprise the powerful data management platform that is HCP for Cloud Scale.

Patent Pending Distributed Metadata Database

Many distributed databases are either 'strongly consistent' or 'scalable and performant' but not both at the same time. HCP for cloud scale tackled this problem by inventing a purpose-built database that boasts strong consistency while maintaining performance at scale. The Metadata Database and Partition Manager form the core of the metadata persistence, load distribution and protection functionality.

Specifically, metadata storage is subdivided into smaller pieces termed **"partitions"**. Partitioning enables efficient metadata operations and balanced load distribution across the system. Each partition maintains a key/value-based structure that is formatted on disk as a "*Log Structured Merge (LSM) Tree*".

Metadata Partitioning, Fault Tolerance and High Availability

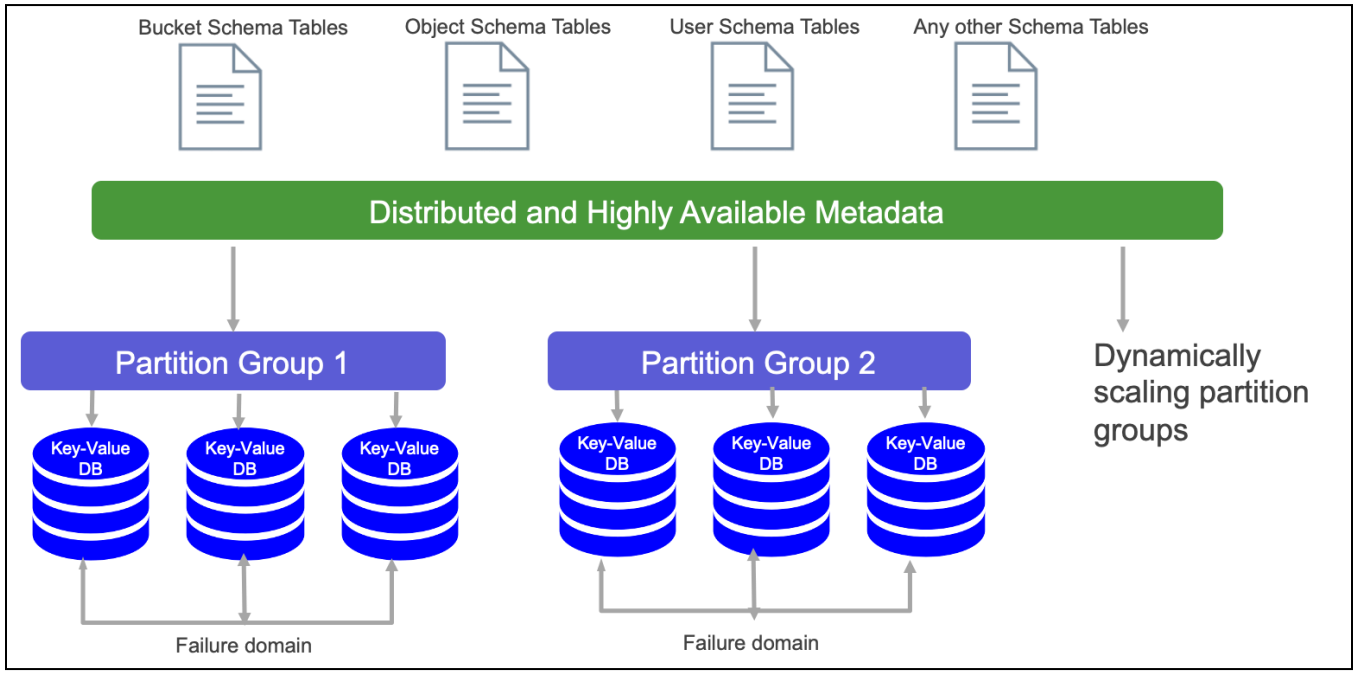


Figure 14. *Highly Available Metadata Partition Groups*

Redundancy and availability are achieved by combining independent partitions to form "*partition groups*". As illustrated in Figure 13, each partition group operates independently, providing a protection domain with no reliance on other partitions. All communications within a partition group such as heartbeats and metadata communication remain isolated within that group. Partition groups use the RAFT consensus algorithm to ensure strong consistency and high availability through a concept called "*Leader*" and "*Followers"*. Writes and Reads for a given partition are always channeled through the Leader, which provides strong consistency and avoids the scale challenges posed by global locking schemes.

The number of partitions in a group is established by the system administrator. This allows organizations to choose a failure tolerance appropriate to their needs. The redundancy factor is easily expressed as

(group_members - 1) / 2

A three-member partition group can tolerate at least one failure yet remain available to service read/write operations. Groups with five members, can tolerate two failures, and so on.

Because each partition group operates independently, a complete failure of all members would have no impact on other partitions. The advantages of layered fault tolerance include allowing: i) at least one failure for each partition and ii) continued system operation (e.g., partial availability) should a whole partition group fail.

Together, the partition groups across the system provide a robust foundation for a strongly consistent, distributed and fault tolerant key/value database. A set of specific schemas/tables are created on top of this foundation for organizing and classifying metadata further (see top of Figure 13). For example, a few basic schemas/tables employed by HCP for cloud scale include the objects, buckets, and user information. Lastly, every schema can be partitioned using different criteria to match different growth or load distribution characteristics.

Metadata Primary and Secondary Indices

The Metadata Management engine will establish multiple indices, each using different key criteria. This offers the ability to support a variety of queries with very performant response times. The primary key index is derived from the object path and is the only index updated in real time as part of a user transaction. This allows for immediate access of objects and up-to-date listing of buckets and object versions as soon as objects are written.

The cost to immediately-update bucket listings is that the primary table is sorted lexicographically. This does allow for specific situations where continuous writes of objects with incremental names (folder/object0001, folder/object0002, folder/object0003, etc.) could resolve to the same partition, with a potential detriment to small object performance. It is recommended that for best performance, objects not be written serially with incremental naming suffix patterns (timestamps, for example). In this example, better distribution is achieved by randomizing the first character(s) of the file name (folder/a-object0001, folder/b-object0002, folder/c-object0003, etc.). Note that this does not include use cases with shared prefixes. Even in cases where names are close together, if many of them are written, eventually that prefix will be split due to the partitioning scheme described in the next section. Several secondary indices are built asynchronously, updated post ingest, and indexed with criteria suited to their purpose. The secondary indices help HCP for Cloud Scale deliver responses, such as searching based on custom metadata with unmatched speed. Architecturally, any number of secondary indices can be created; none are updated in real time; thus the quantity of indexes has no impact on data transaction performance.

Dynamic Partitioning and Partition Balancing

The metadata engine employs a dynamic partitioning scheme (and hence dynamic key ranges) to achieve linear scalability. Partitions split automatically when conditions meet pre-programmed watermarks. Moreover, new partitions gravitate towards the nodes with adequate spare resources, which fosters optimal rebalancing and infrastructure utilization.

Specifically, the **Partition Manager** service performs on-demand rebalancing of partitions across nodes. There is no need to set some artificial quantity ahead of time. When a partition is split, sections are removed and used to seed new partition(s). Metadata movement is accomplished by taking a snapshot of the split metadata to minimize network and disk I/O. The partitions are purposely kept small to minimize the amount of metadata moved during splits. Figure 15 illustrates how 4 partition groups might be distributed across nodes.

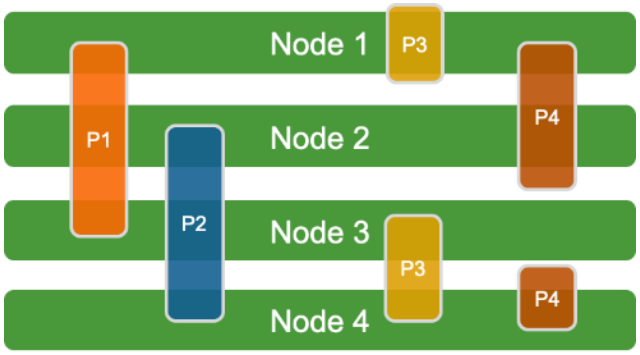


Figure 15. *Partitions Balanced Across Nodes*

The benefits of HCP for Cloud Scale's dynamic partitioning design are multi-fold:

Static Partitioning	**Dynamic partitioning**
An administrator must preconfigure some number of static partitions. The administrator must possess complex knowledge of properly tuning the system with the appropriate number of partitions based on available resources. Failure to configure the system correctly could cause certain partitions to become hot spots and others to be underutilized.	The software provides a uniform distribution of load across the system by adapting to actual observed conditions.
Reconfiguring the system with static partitions can create a sudden backlog and a surge of rebalancing activity that competes with regular traffic, which can degrade system performance until complete.	With dynamic partitioning, only split partitions are impacted. Furthermore, partitioning events are spread over time to reduce the impact on system performance.

The system administrator sets the criteria for splitting partitions. Factors can include location, data access patterns, or partition size. A split strategy that results in smaller partitions helps to promote lower and deterministic latency and makes it easier to start new partitions on servers with modest resources. Figure 16 illustrates how partition growth will expand in proportion to object count. Individual object table partitions grow naturally over time based on the traffic conditions specific to their key range. Moreover, new partitions adapt dynamically to fit the infrastructure as it expands. (Please refer to the Metadata Key Routing section to learn more about key range).

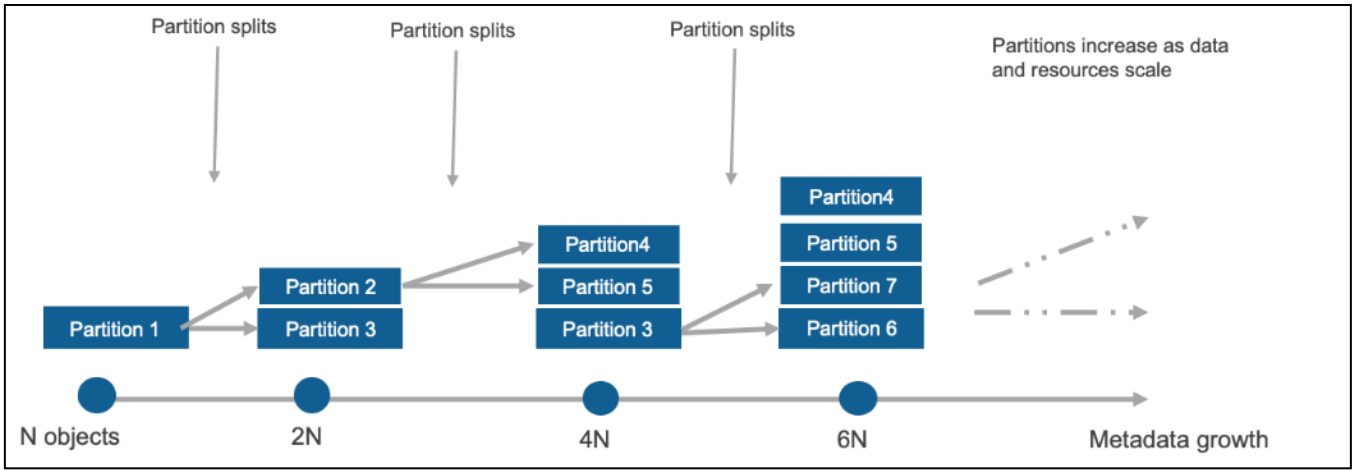


Figure 16. *Dynamically Scaling Metadata Partition Groups*

In addition to scaling and balancing, the Partition Manager works to automatically heal the system after failures. When failures occur, the partition manager automatically detects that nodes have stopped responding and works to rebalance metadata across the remaining healthy nodes.

Metadata Key Routing

With dynamic partitioning, metadata indices scale and rebalance in a steady and continuous fashion, sidestepping the rebalancing avalanche common in competing architectures. However, this continuous churn in partitions poses a new challenge for request routing: locating the right partitions in a constantly changing environment. The patent-pending metadata routing design in HCP for cloud scale makes it possible to efficiently route metadata at scale in this dynamic environment.

All schema tables start as one huge partition that is responsible for the entire (0-2^{256}) range. If the partition grows to surpass predefined thresholds, the key range is split down the middle, resulting in two partitions. When this happens the **Partition Manager** service updates a "*cached*" ***Partition Map*** to reflect the new number of partitions and a new key-range-to-partition mapping. This simple update efficiently propagates the new routing rules. The cached partitions are updated via the publish-subscribe messaging mechanism described in the Distributed Metadata Cache section.

When an index is queried, it will consult the cached Partition Map to quickly locate the Metadata leader node associated with that key range. If the cache was stale due to a pending update, the Partition Map might point to the wrong partition. This situation is easily detected. In such cases, the client's write/read operation is rejected by partition leader, whose key-range might have changed recently. The client triggers the cache to rediscover the partition map by explicitly calling for a partition map update, and the query is retried. This process of updating a dynamically changing partition map through both proactive and reactive discovery mechanisms minimizes overall system chatter.

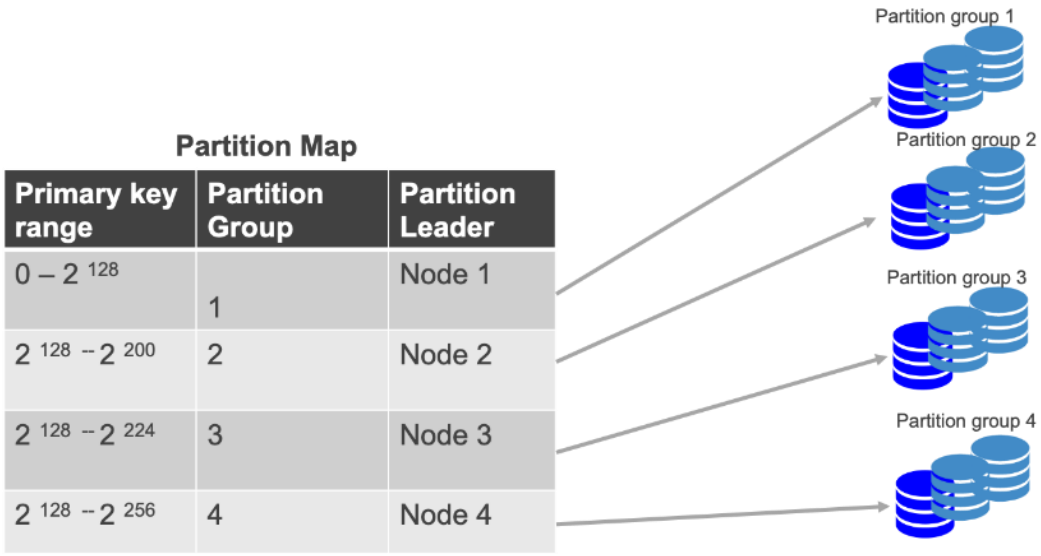


Figure 17. *Metadata Routing*

Metadata Policy Engine

HCP for Cloud Scale implements a set of background functions that automatically clean up expired object versions, reclaim space by purging deleted objects from storage, automatically copy user buckets to remote S3 targets for disaster recovery and more. The patent-pending **Metadata Policy Engine** service in HCP for cloud scale is designed specifically to support processing of these I/O and compute intensive background functions at scale. One can easily envision that the types of supported background functions must adapt and evolve to meet future business needs.

The flexible Metadata policy engine employs a "*data processing pipeline*" architecture to interconnect processing functions, and/or seamlessly integrating with remote processing engines such as public cloud services. The processing engine consists of three key elements: a source (***asynchronous job producers*** in Figure 18), a messaging framework (**queue network** in Figure 18), and a processing engine (***asynchronous* job consumers** in Figure 18) that includes components to execute background functions. Each of the pipeline elements can be scaled independently to match workload requirements, which is another benefit of the pipeline architecture.

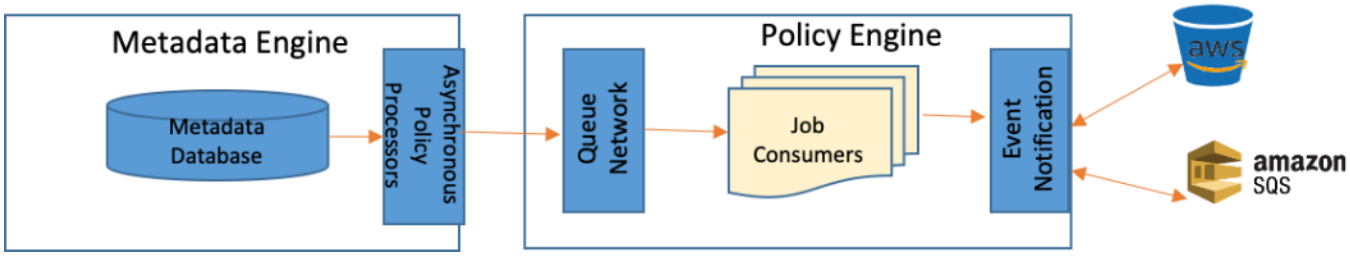


Figure 18. *Metadata Policy Engine*

The background functions can be triggered to run by predefined event(s) or they can be scheduled. An "***Event Notification engine***" can also notify a remote service, such as AWS SQS, after completion of a background function or again based on a user driven event.

The following set of functions, which can be easily augmented with new ones in the future, are currently supported by the job consumer engine.

Background Function	Description
Retention expiration	Expire locked objects that have reached their retention maturity. These expiration dates are set via S3 API using S3 object lock configuration at the bucket level or object lock applied to the object directly.
Disposition	Automatic cleanup of expired versions. An S3 bucket lifecycle policy authorizes HCP for Cloud Scale to automatically delete versions after their retention period expires.
Replication (Sync-to)	Automatic copy of user buckets to remote S3 targets.
Replication (Sync-from)	Automatic copy of remote S3 bucket in AWS to one hosted within HCP for Cloud Scale.
Event Notification	Send notifications upon completion of a background function or a user driven event, such as an object put to remote notifiable targets (e.g., AWS SQS).

Certain object storage products claim hybrid cloud support where the only relevant feature is an ability to use public cloud as a secondary tier. While this allows some amount of storage flexibility and cost savings, true hybrid cloud power comes from being able to utilize compute, storage, and applications/services across a hybrid cloud environment. This is encouraged and easily achieved by leveraging the replication and notification features of HCP for Cloud Scale.

Event Notifications

Users can construct complex workflows utilizing multiple AWS service queues and lambdas today, which will be extensible to GCP and Azure in the future.

HCP for Cloud Scale can notify remote services such as AWS SQS of events happening locally. Notifications include any number of basic S3 operations such as file/file part upload, delete or a failure to replicate an object. Currently, HCP for Cloud Scale is the only object store capable of sending a single event to multiple remote targets.

Event notification is performed as part of the policy engine processing pipeline. HCP for Cloud Scale can easily publish to multiple consumers, providing more efficient and cost-effective scaling.

Distributed Metadata Cache

HCP for Cloud Scale includes a distributed in-memory cache for frequently accessed metadata such as system, user and S3 bucket configuration. Cached entities include all frequently accessed items such as bucket lists, user information, access control lists, partition maps, and more.

A set of *cache servers* act as a central messaging hub communicate between the database and the *cache clients*. Cache clients are in-memory data structures that reside on each of the HCP for Cloud Scale service instances. Cache clients refresh stale entries based on a timer or by subscribing to the cache servers for specific changes.

Cache coherency across the system is achieved with a lightweight publish-subscribe messaging protocol. The *cache servers* subscribe to the database for changes to specific information stored in the database. The Metadata database notifies the cache clients when updates to subscribed information are made.

At a system level, a small set of cache servers can buffer the database from a large number of client queries, significantly reducing the I/O load on the database.

User Data Replication

A replication policy is used to asynchronously copy objects to/from a remote S3 bucket. The functionality leverages the Policy Engine framework discussed in the previous sections. This framework is designed to efficiently spread work across the entire infrastructure and can scale to handle millions of buckets. The following replication features can be integrated into a disaster recovery plan:

- **One to many Sync-To**: Unlike AWS S3 support, HCP for Cloud Scale can replicate objects across multiple destination buckets.
- **Many to One Sync-From**: Users can bring data from external S3 sources back to the HCP for Cloud Scale system. This functionality is unique and an essential tool for migrating data from remote systems.

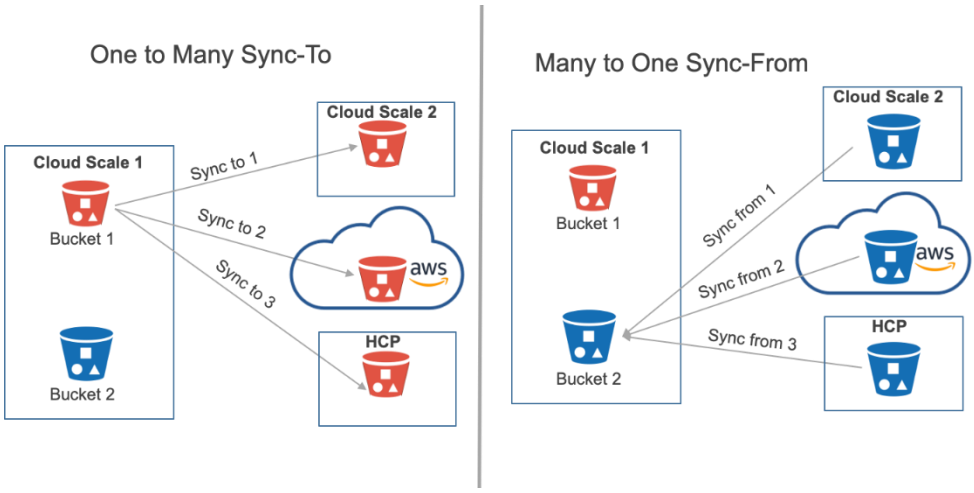


Figure 19. *Advanced Replication Capabilities Available in HCP for Cloud Scale*

Administrators grant or deny replication authority to users by adding replication-specific permissions for "*bucket sync-to*" and "bucket *sync-from*" to their roles. Users with the proper permissions can control what objects get replicated using the standard S3 API's for bucket replication. They may also provide detailed filter rules and tags to define the replication policy. HCP for Cloud Scale optimizes replication by providing single instancing capability for version updates. If it recognizes an overwrite, HCP for Cloud Scale does not re-upload versions of objects that match the content hash of previously uploaded versions. If a version matching an object to be uploaded exists remotely, HCP for Cloud Scale ensures a new link to the previous version is created using the S3 put-copy API instead of duplicating all of the content again.

Infrastructure Requirements

HCP for Cloud Scale deployments are separated into two infrastructure layers as depicted below. This approach allows for precise sizing of an infrastructure to meet client application needs.

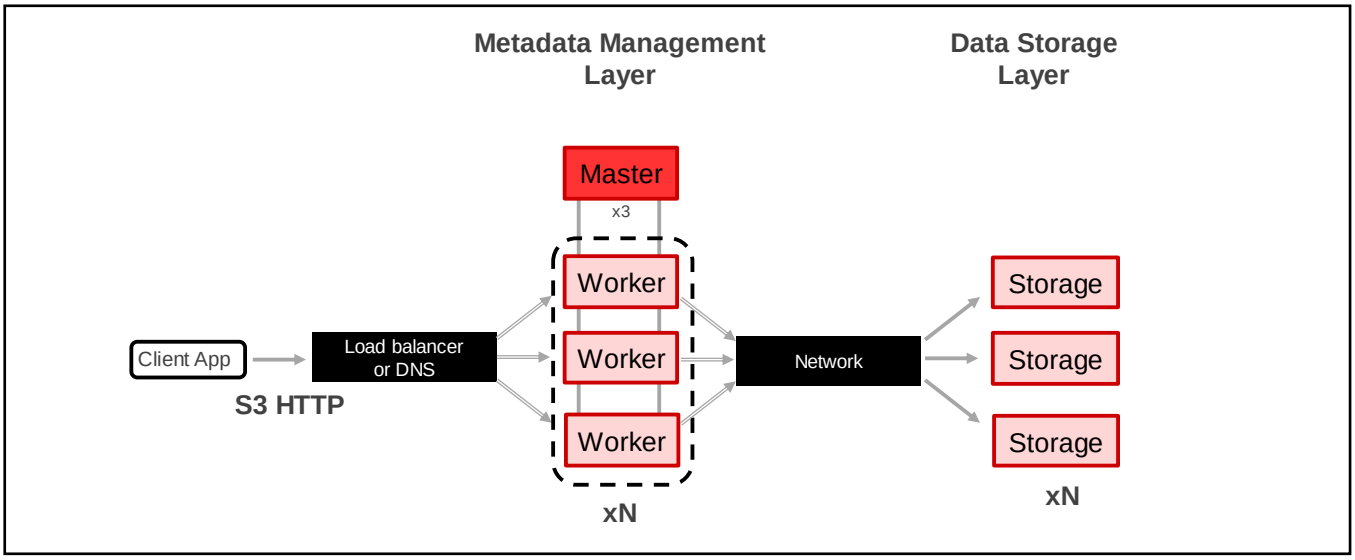


Figure 20. *HCP for Cloud Scale Infrastructure Diagram*

Infrastructure Requirements for Metadata Management Layer

Physically, this layer requires clustered servers to host the metadata management portion of the HCP for Cloud scale software. It holds ALL object metadata and applies policies to distribute and govern data. The layer can be deployed on servers provided by customers, Hitachi servers, or the Cloud (e.g., AWS EC2) environments. This layer is entirely software defined with few prerequisites beyond having networked Linux servers with a container-based execution environment installed.

The layer can be scaled to accommodate more object count, more custom metadata, or more storage-side compute in order to handle the scaling of services, such as replication, encryption, lifecycle, and query (S3 select).

Server Organization

The layer requires a minimum of 4 server instances that provide the container execution muscle. Architecturally, 3 instances are designated master nodes, with all additional instances designated as workers.

Server Compute Requirements

HCP for cloud scale has adopted a *scale-out strategy* aimed at leveraging modest servers that can be extended horizontally to linearly increase performance. Servers can be physical, virtual, or a combination so long as they are networked and visible to each other. If not visible, virtual instances should be created on different physical servers for redundancy. They need not be identical but should meet these recommended minimums:

Resource	Minimum	Recommended
RAM	32 GB	128 GB
CPU	8-core	24-core
Available Disk Space	500 GB 10k SAS RAID	2x 1.92 TB SSD
Network Connectivity	1x 10 Gigabit Ethernet NIC	4 x 10 Gigabit Ethernet NICs

Server Network Requirements

Modular service components regularly communicate with each other and necessarily require a network with routing. While it is permissible to have server instances on different subnets, we suggest they be on a LAN providing <1 ms latency. This environment best supports services that employ quorum based 'voting' semantics for availability within a site or availability zone.

Each server must have at least one network which can be shared by all services. While the least expensive approach of having all traffic share the same physical port might trigger security concerns in your organization, most of the services running in HCP for Cloud Scale need not be exposed externally.

Any number of additional network ports can be added to improve resiliency, performance or security. However, each network port adds a cost burden with more cables and switches that must be considered against the benefits. Regardless of supplier, we generally recommend servers with four Ethernet ports that are 25Gb capable but wired with 10Gb switches as an economical trade-off. Motivations for more ports include:

- **Bonding** networks: Electively, system architects can bond two (or more) physical ports together to improve resiliency and performance of a network. For more information, please refer to Link Aggregation Control Protocol (LACP) for Ethernet defined in IEEE 802.1AX.
- **Isolate application traffic**: The software supports elective traffic segregation using network label designations of **External** and **Internal**. Electively, these allows you to physically isolate external customer traffic (e.g., S3 commands) from internal cluster services traffic (e.g., heartbeat). Each microservice identifies the ports and network it utilizes, and reports this in the GUI for admin review.
- **Isolate services traffic**: Support services such as Identity management, DNS and NTP can be located on separate networks for improved security.

Server Disk Requirements

Generally, the persistent storage within a server is utilized exclusively for metadata storage (approximately 2.5KB per object). Solid state disks (SSD) are recommended since this storage directly hosts the metadata database.

- **Disk Size:** A 1TB SSD can hold system metadata for approximately 400 million objects; less if users add custom metadata. Architects should consider the number of stored objects in the entire deployment, with the understanding that HCP will distribute object metadata evenly across the server cluster.

Infrastructure Requirements of Data Storage Layer

The resources in this layer are called “storage components”. The resources can be HCP S series storage nodes or any S3-compatible endpoint such as AWS S3. The storage components can be scaled out horizontally to gain the benefits of cumulative capacity and performance. HCP for cloud scale storage is presented as single unified global namespace for a “bottomless bucket” experience no matter if there is one storage component or 1,000. Specifically, its management layer spreads data written to a logical S3 bucket across all connected storage components.

The Data Storage layer can be scaled to accommodate more capacity, bandwidth or IOPS.

Storage Network Requirements

Storage components connect with the management layer through the network as URL endpoints. The network can be any speed, bonded, dedicated or shared. The only real requirement is that the components must be visible to the metadata management layer.

HCP S Series Nodes

Each Hitachi Content Platform (HCP) S series storage appliance provides a highly durable and available storage component managed by HCP for Cloud Scale. The most recent appliance models are among the densest storage platforms available today with more than 15 PB of disk capacity per rack. HCP for Cloud Scale can be configured with one or more HCP S node storage appliances, which ensure high availability with redundant controllers and components, redundant data paths and dual port disks, that collectively eliminate single points of failure. They use commodity server and JBOD hardware, along with enterprise class disks that connect internally using a SAS disk topology. For fast data access, HCP S series nodes takes advantage of as many as eight 10GbE network ports.

Hardware and Architecture

The HCP S software that runs on the HCP S series node hardware provides all capabilities and storage functionality. It also features hardware monitoring and component installation and replacement wizards. Its data protection is based on software defined Erasure Coding (EC) across all disks in the appliance, which is configured for an optimal balance between data durability, fastest repair time and storage cost efficiency. HCP S software is designed to easily add disks and automatically abandon disks that start to fail. Disks that are added do not require formatting and are immediately available for use. The data that was stored on abandoned disks will be recreated (repaired) using EC data protection, and then redistributed across the remaining disks in the appliance. The repair feature offered by HCP S software uses intelligent algorithms to find the fastest path to re-establish maximum data protection and reduce vulnerabilities. HCP S software can safely handle multiple disks failing at the same time; up to six hard disks can fail concurrently with the default EC data protection class configuration. Once the data on a failed disk has been repaired by the software, that disk no longer counts among the six.

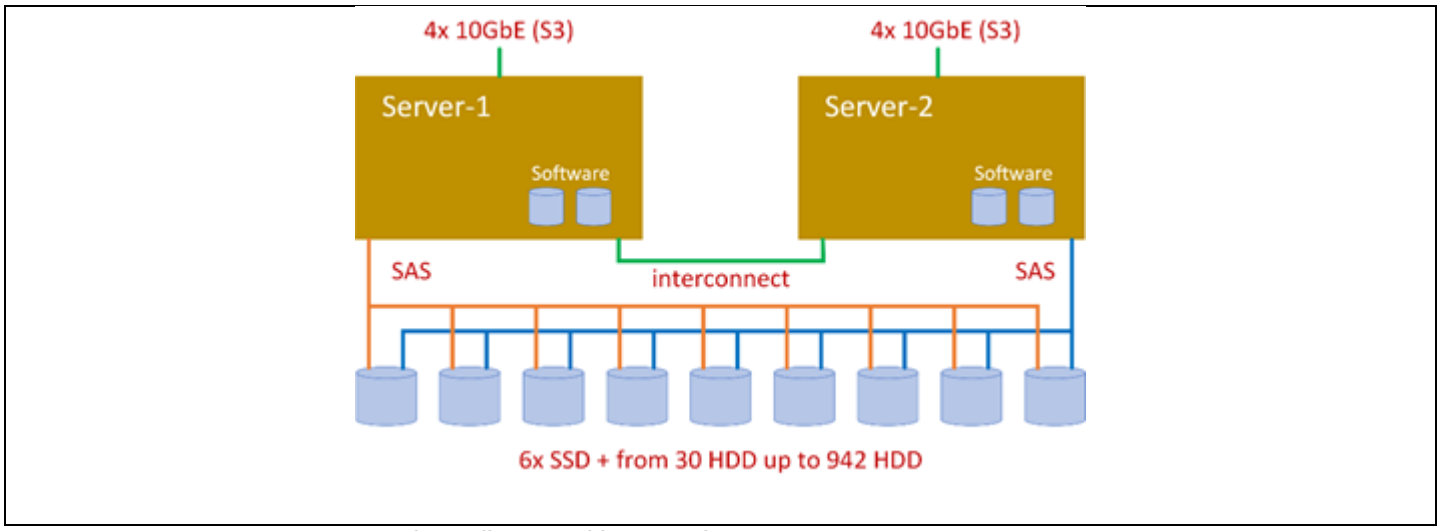


Figure 21. *HCP S Node Appliance Architecture Diagram*

Hardware Architecture

The HCP S series is purposely designed to manage a large but finite amount of storage. The fully redundant architecture of an HCP S node is depicted in Figure 21. Each appliance operates autonomously, with individual durability and availability domains. Availability is provided via redundant controllers, redundant Ethernet connections and redundant SAS connections (two ports per drive). Durability is provided via EC algorithms running in software to protect data stored on large form-factor (LFF) SAS disk drives. HCP for Cloud Scale is designed to consume any number S-Nodes by federating and striping data across them to achieve cumulative capacity and throughput goals.

Software Architecture

HCP S nodes protect stored data with Erasure Code (EC) algorithms for superior fault tolerance and configuration flexibility; this technology protects a stored data stream as opposed to a whole disk. This scheme is particularly beneficial when recovering from a disk failure, since the repair focuses only on written data, and can ignore the unwritten areas of a disk. However, small objects workloads often challenge EC implementation in other object storage systems. Many lack a scalable index to manage a truly large volume of stored objects. Secondly, when EC is applied to small objects individually, the overall storage efficiency can substantially drop, and the time to repair following a disk failure can substantially increase vs an equal capacity composed of large files.

In the following sections, we discuss several innovative concepts that allow HCP S software to address challenges associated with managing large numbers of objects of different sizes.

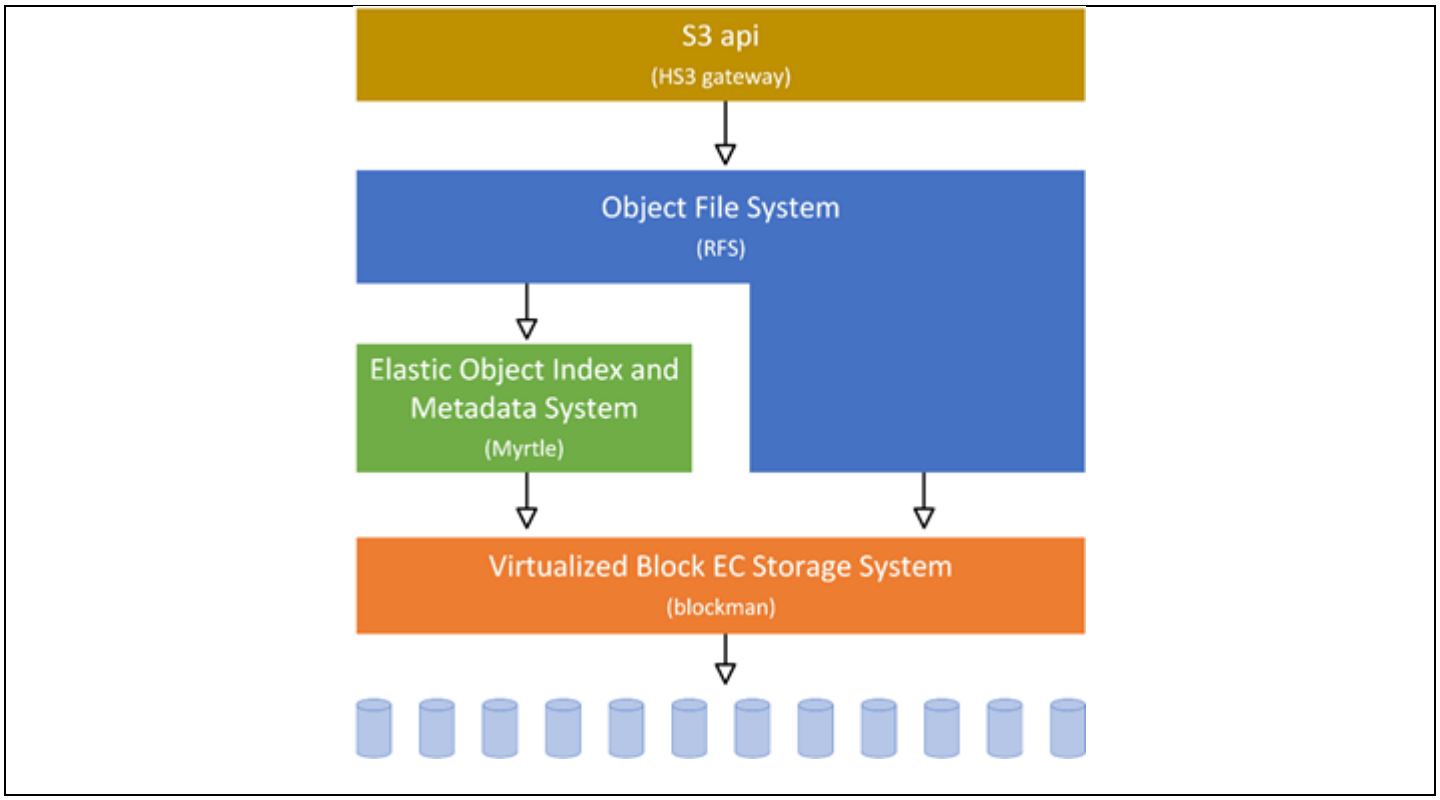


Figure 22. *HCP S Software Architecture*

Elastic Object Index and Metadata System

HCP S software includes a purpose-built Elastic Object Index and Metadata System using log-structured merge-tree (LSM tree) principles, a design also used by well-known high-performance no-SQL databases. This provides HCP S storage with a high-performance object index with unlimited scale to store, find and retrieve billions of objects. The Elastic Object Index also enables full utilization of all available capacity, irrespective of the object sizes and amount of system metadata. The capacity that an index may need can greatly vary with the number of objects and system metadata. For typical object storage products that use a traditional database for the object index, the maximum required disk capacity needs to be pre-allocated for the worse case. However, the Elastic Object Index does not require pre-allocation. Therefore, the number of objects an HCP S storage appliance can store is only limited by the available capacity.

Virtualized 4 KiB Block System with Erasure Code Data Protection

The HCP S software writes all objects to a purpose-built virtualized 4KiB (kibibytes) block system that obfuscates the EC data protection from the objects that are stored. The underlying disks are sliced in fixed-length data streams of 64 MiB (mebibytes) called extents. When the virtualized 4 KiB block system needs more blocks to store objects, the software will group several extents together based on the EC data protection class and create more 4 KiB blocks. The extents in a single extent group are spread across distinct disks such that one disk failure only affects one extent. The number of extents in each extent group depends on the configured EC data protection class. The software is configured to use 20+6 extents. The 20 extents * 64 MiB of each extent group is used to store the object data. The 6 extents * 64 MiB is used for EC data protection of that data.

An extent group may contain multiple small objects or only a portion of a large object. The capacity of an extent group is fully utilized irrespective of the object size.

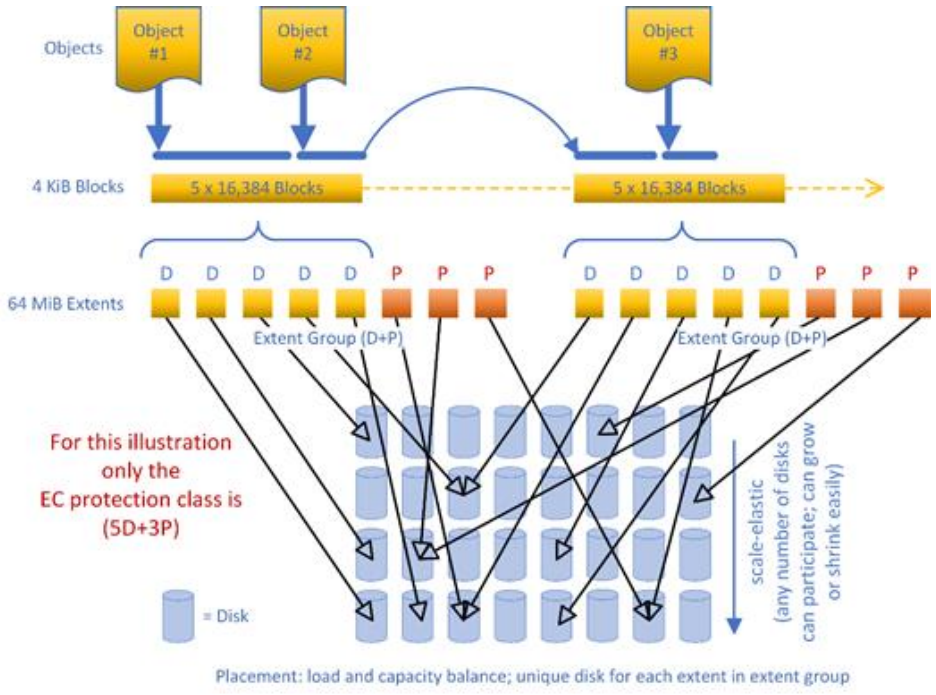


Figure 23. *Virtualized 4 KiB Block System with EC Protection*

Our design offers two main benefits: (1) small objects do not have to be individually sliced into EC fragments, which may lead to inefficient storage, and (2) the number of EC fragments that need to be repaired in the event of a failed disk do not increase with smaller objects. This dramatically reduces the time it takes to recover from a disk failure where the data consists of small objects.

HCP S software supports multiple EC data protection classes concurrently across the same set of disks. There are three protection classes. The protection class cannot be configured by the end-user as we have selected an optimal setting for all workloads to meet storage efficiency and data durability expectations, which enhances ease of use by eliminating complex configuration concerns for users.

For EC data protection we configured 20+6 extents with priority assigned to HDD storage. For the Elastic Object Index (LSM tree) write-ahead-log storage, we use 3+3 extents with priority assigned to SSD storage. For second tier (index tree) storage, we use 10+6 extents with priority given to HDD storage. Based on its software defined architecture, HCP S software allows for reconsideration of these EC data protection classes, or the addition of new ones whenever necessary.

Erasure Code Data Protection Across Nodes and Sites

Other object storage products may implement a single EC data protection function covering all disk, node and site failure domains. However, HCP for Cloud Scale does not combine these failure domains; they are addressed at different layers in the solution. The loss of data due to disk failure is statistically expected to occur substantially more often than data loss due to node and site failures. Hence, recovering lost data due to node and site failures is best accomplished in a way that addresses their specific characteristics rather than one design for all type of failures. The HCP S software layer focuses on recovery from data loss with maximized efficiency. By keeping the repair process as close as possible to the disks, it is not exposed to network and other latencies, which is key for performance as millions of fragments may need to be repaired when a 16 TB disk fails. This, combined with avoiding a dramatic increase of EC fragments for small objects, allows HCP for Cloud Scale to optimize performance for repairing lost data.

Operation

HCP S software can process many concurrent (write) operations at high speed. Each HCP S series node has two servers; both servers accept write/read requests arriving over Ethernet networks. For optimal performance, prior to accepting new objects, each server has already created a write buffer in the virtualized 4KiB block system from several extent groups spread across all disks. When a new object is received, it is written to 4 KiB blocks that are free in the buffer. The extents that these 4 KiB blocks are mapped to are then committed to disk. The EC data protection extents of the extent group are then also recalculated and committed to disk. Every commit of an extent to disk may include multiple object writes. For data consistency a journal is kept and will be replayed after an unexpected outage occurs during pending commits.

When new objects are stored by the Object File System (RFS), the Elastic Object Index is updated with the necessary object system metadata and block addresses for later retrieval. The Elastic Object Index itself uses the same virtualized 4 KiB block system used by objects, albeit with different EC data protection classes.

Lean Transactions with Self-Management Services

HCP S software is architected to minimize the number of operations to complete a write, read or delete request. Anything not directly contributing to the consistency and protection requirements is deferred. The leaner the transaction, the better the performance. Operations such as system and data health, storage efficiency and hygiene are performed outside of the write, read and delete transactions. They are referred to as “smart services” which are self-configured by the software to execute in the background. Twenty-nine (29) such services exist, none of which require user management, thus enhancing efficiency and ease of use.

Security Highlights

In a software defined world, you may choose to supply and maintain the “platform”; a server cluster running the Linux operating system of their choosing. Advantages include cost savings delivered via negotiating a bulk purchase of “whitebox” servers that can run all your data center applications. The CIO and DevOps team also maintain direct control over the electronic security of the operating system (OS). They can ensure it is hardened to meet specific corporate standards and control when Common Vulnerabilities and Exposures (CVE) are applied to the base OS.

Product Security Features

With the hardware and OS architecturally abstracted from the software, good security requires cooperation between the HCP for Cloud Scale administrators and the server administrators. The following features are supported out of the box with HCP for Cloud Scale.

Domain Certificates

A domain SSL certificate may be loaded so S3 applications can validate identity and minimize risks for man-in-the-middle spoofing. The domain certificates are employed for all secure communications between HCP for Cloud Scale and external services (i.e., users/applications, Identity providers, storage components and more). HCP for Cloud Scale supports uploading separate domain certificates for each type of communication (i.e., storage components versus users/applications).

Additionally, administrators can also upload specific certificates for client-side and server-side communication. For example, the S3 Gateway uses the server-side domain certificate when authenticating with its users/applications, while specific client-side certificates are used for communication with storage components and identity providers. With such granular certificate management, administrators have the flexibility to further secure and restrict access across various infrastructure within their environments.

Data at Rest Encryption (DARE)

HCP for Cloud Scale includes an embedded KMS (key management system) to securely store encryption keys. The microservice implementation provides redundancy across the cluster to ensure both availability and durability of the stored secrets. When DARE is enabled by an admin user, the software creates a unique key for every file called a DEK: Data Encryption Key. Groups of DEK are encrypted with master keys called KEKs (Key encryption key). The encrypted DEK ultimately becomes part of the object’s system metadata, while the KEK is stored in the KMS. With a unique key created for every object, there is a built-in mechanism for cryptographic erasure.

Data in Flight Encryption (DIFE)

HCP for cloud scale employs the strictest level of security, while remaining flexible for the future. With DARE and DIFE together, HCP for Cloud Scale bundles the necessary components for a holistic data protection strategy. All end-user data access is secured via HTTP(S). Administrators can turn off unwanted ciphers and only enable protocols that are secure against known vulnerabilities, such as TLS 1.2/1.3.

User Access

All user access, whether for data or management, is authenticated with Microsoft Active Directory (AD) and/or LDAP servers. This lets IT teams dictate password complexity, password rotation policies and leverage corporate auditing tools to detect login anomalies. Please refer to the role-based access control in the Access Control and Identity Management section for detailed information on user access and control.

Layered Permission Strategy

Users are associated with a LDAP or AD group. A group is associated with a HCP for Cloud Scale role. A role defines a specific set of permissible actions. The software provides over 100 permissions that can be associated with a given role. For example, administrators can create a “root-like role” with all the permissions, or an “S3 user role” restricted to just the S3 API, and so on. Any number of roles can be defined that align 1:1 with an AD/LDAP group. Users in these groups will have ONLY the capabilities explicitly associated with the role.

Bucket Privacy

When individual users create an S3 bucket and subsequently write data, there is no other user that can programmatically read it. While this suits many applications, HCP for cloud scale also supports S3 ACLs (access control lists) to share objects with other users. Such access must be explicitly granted by the owner. Additionally, HCP for cloud scale supports a “Canned ACL” to grant wider access. For example, a read ACL may be added to an object for authenticated users; the grant will allow any user appearing in any AD or LDAP with S3 user role privileges to read the file.

Network Isolation

All services in HCP for Cloud Scale can be configured by administrators to run on internal or external networks. Administrators can secure HCP for Cloud Scale by restricting the services exposed to end-users (e.g., S3 Gateway, MAPI, Metrics and Tracing) on external networks, and firewall all other external TCP ports except for traffic destined for those services. Typically, most organizations will also restrict administrative services such as MAPI, Metrics and Tracing to an internal network, exposing only the S3 Gateway and thereby ports 80/443 to external networks.

CORS Allowed Origins

Administrators can further restrict access to the system by entering a comma-delimited list of acceptable origins. Organizations can be challenged when they have no control over browser setups that prevent a random script from being invoked as part of an HTTP request. Cross-Origin Resource sharing support secures organizations and prevents unforeseen access via scripts.

Product Security Process

Security Architecture reviews

HCP for Cloud Scale is routinely reviewed by a Hitachi Vantara product security team that uses industry standard security rubrics to ensure security related quality expectations from our customers are met. The review process focuses on security touchpoints such as ciphers employed, transport protocols adopted and flexibility in managing permissions, It is covered in the previous sections, along with other important considerations such as secure coding practices.

Scanning for various best practices

HCP for Cloud Scale follows all industry-recommended best practices such as scanning software for software composition analysis, dynamic application security testing, threat modeling and penetration testing. Tools used for static code analysis include BlackDuck, Nessus, HCL ASoC, and Owasp Zap, which provide a wide range of security analysis coverage. High priority and critical issues found during regular scans are constantly reviewed and fixed before the delivery of new releases.

Summary

For over 15 years, Hitachi has evolved Hitachi Content Platform – initially to address traditional archiving and compliance use cases, and then with the HCP portfolio products, to address collaboration, remote office file serving, NAS replacement, modernized backup, search, big data analytics, and more. With HCP for Cloud Scale, Hitachi Vantara maintains its commitment to cutting edge object storage software innovation.

HCP for Cloud Scale is a software defined S3-compatible object storage solution with built-in data security and protection, advanced metadata-based intelligence, and modern hybrid cloud workflow capabilities. It has been designed from the ground up to address the growing requirements and broader benefits of software defined storage. Its innovative microservices design enables high performance and massive scalability to support 100's of nodes and billions of objects, and it eliminates classic database and network bottlenecks that plague distributed system designs. With its hardware agnostic architecture, HCP for Cloud Scale can be deployed with ease on servers and cloud platforms, including Hitachi Unified Compute Platform configurations and generic "white box" servers capable of running docker and a compatible Linux distribution.

Object storage can redefine the purpose and value of data, as well as significantly enhance data management and governance while drastically lowering storage costs. HCP for Cloud Scale is not only a modern object store but also a data management platform to enrich data with the context necessary to secure, protect, control, and analyze it. It is the ideal foundation for intelligent and dynamic data services that you need to power your journey to digital transformation.

Hitachi Vantara

Corporate Headquarters
2535 Augustine Drive
Santa Clara, CA 95054 USA
hitachivantara.com | community.hitachivantara.com

Contact Information
USA: 1-800-446-0744
Global: 1-858-547-4526
hitachivantara.com/contact

WP-602-B BTD November 2021