FusionStorage 8.0 object storage

Technical White Paper

Issue 01

Date 2019-07-30

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://e.huawei.com

Email: support@huawei.com

Contents

1 Overview

With explosive growth of data and boom in Internet services, ever-changing and uncertain storage requirements of newly emerging applications bring huge challenges to storage systems. Specifically, the finance industry is facing a host of new opportunities and challenges brought by e-Banking and mobile Internet finance in particular. Such challenges include needs for precise user requirement analysis and day- or even hour-level service rollout periods.

Beyond the finance industry, a surge in new services and an exponential increase in service data can be seen in the fields of governance, manufacturing, and the carrier industry. These developments pose the following new challenges for storage systems in enterprise data centers:

- Tension between long system construction periods and short rollout periods of new services
- Inability of storage systems to meet increasing concurrent data processing requirements
- Demand for big data and cloud computing technologies that facilitate customer requirement analysis, service data analysis, and decision making

Figure 1-1 New challenges for storage systems

If you are faced with the preceding challenges, then your ideal storage system may be something like this:

- The system is agile. Resources can be deployed flexibly and acquired on demand. The rollout periods of new services are shortened.

- A variety of methods for accessing large volumes of unstructured data are supported.
- Implementing quick and large-capacity expansion is a piece of cake.
- Superb performance is delivered to process data concurrently.
- The total cost of ownership (TCO) is decreased.

If this is what you are looking for, FusionStorage object storage may be the answer. As a distributed object storage product that supports large-scale horizontal expansion, FusionStorage object storage integrates local storage resources of general-purpose servers through software to form a distributed resource pool. It delivers enterprise-class reliability and availability as well as provides a variety of service functions and value-added features.

FusionStorage object storage can be flexibly purchased and deployed based on service requirements. It enables enterprises to quickly provide private or hybrid cloud storage services and is a suitable data access solution for ever-changing service scenarios in which flexibility and efficiency are key requirements.

2 Product Highlights

FusionStorage object storage boasts the following highlights:

- **Distributed storage for on-demand use**

  Using distributed technologies, FusionStorage object storage organizes storage media, such as hard disk drives (HDDs) and solid state disks (SSDs), into different types of large-scale storage pools. It provides standard application programming interfaces (APIs) compatible with Amazon S3 for upper-layer applications and clients. FusionStorage object storage supports integration with mainstream cloud computing ecosystems and is applicable to cloud backup, cloud archiving, and private cloud service operation.

- **Elastic scalability and high efficiency, meeting future data access requirements**

  FusionStorage object storage adopts a fully distributed architecture. It enables a linear growth in system capacity and performance by increasing storage nodes, requiring no complex resource requirement plans. It can be easily expanded to contain thousands of nodes and provide EB-level storage capacity. This helps meet your future storage demands.

  FusionStorage object storage implements automatic load balancing to evenly distribute data and metadata onto nodes, eliminating metadata access bottlenecks and ensuring system performance after capacity expansion.

  To optimize node performance, FusionStorage object storage leverages an efficient distributed hash table (DHT) routing algorithm, concurrent I/O processing techniques, distributed cache techniques, as well as hardware, such as non-volatile memory express (NVMe) SSDs and Intel QuickAssist Technology (QAT) acceleration cards. This better supports mission-critical cloud services such as big data analysis.

  In addition, FusionStorage object storage enables you to expand I/O- and bandwidth-intensive services as well as services that require large capacities in your data centers based on your service needs.

- **A wealth of enterprise-class features, helping you build highly available data centers**

  FusionStorage object storage provides a variety of enterprise-class features to meet requirements of different application scenarios. Multiple storage pools，as well as node- and cabinet-level security help you easily build solution-level data protection mechanisms. Erasure coding (EC) and object-level deduplication enable you to develop appropriate hardware resource utilization plans. Multi-tenant and quality of service (QoS) allow you to flexibly and effectively allocate internal cloud storage resources.

- **Openness and wide compatibility, making FusionStorage object storage the ideal choice for the next-generation cloud infrastructure and big data platform**

Based on an open architecture, FusionStorage object storage provides standard S3 and scale-out data storage tiers for private and hybrid cloud data centers as required. This helps you easily build open cloud platforms without worrying about vendor lock-in.

FusionStorage object storage is also compatible with standard Hadoop HDFS APIs. It provides high-throughput data access for cloud data centers, helps enterprises gradually build applications based on large-scale datasets, and fully exploits the values of information.

- **Automated data services and O&M**

  FusionStorage object storage provides an automatic management system to easily complete service configuration and implement hardware platform monitoring and management, such as alarms, topology management, and performance reporting.

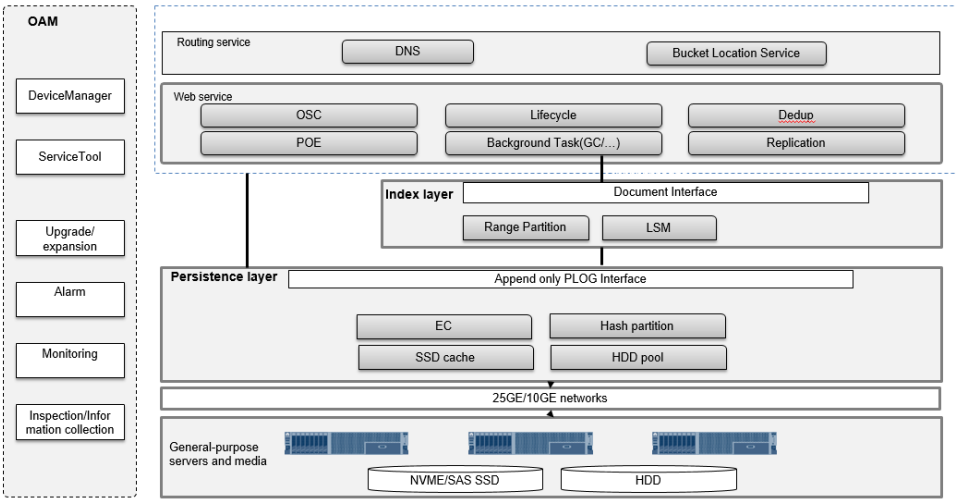
  In addition, FusionStorage object storage also supports eSight, a unified management platform, to monitor storage status and alarm information. The automated and comprehensive management functions relieve data center O&M personnel from complex software and hardware resource management and shorten the rollout periods of new services from one week to one hour, greatly reducing the time to market (TTM).

3 Product Architecture

3.1 Software Architecture

FusionStorage object storage is a software-defined object storage product that supports large-scale horizontal expansion. Its software architecture complies with industry-leading scale-out, service-oriented, and microservice-based design principles.

Figure 3-1 Software architecture



As shown in the preceding figure, FusionStorage object storage consists of three layers: persistence layer, index layer, and service layer. These layers are described as follows:

- The persistence layer is composed of general-purpose servers and storage media. It is responsible for data layout, load balancing, and data recovery, and provides EC data redundancy, striking a balance between performance and costs. The persistence layer is the foundation of FusionStorage object storage and determines the system scalability, performance, and reliability.
- The index layer is responsible for metadata distribution, indexing, and failover in the event of faults. It provides high-speed metadata access and query capabilities for the service layer. As shown in the preceding figure, the metadata of the index layer is eventually stored in the persistence layer. Therefore, the metadata enjoys the data storage capability of the persistence layer and is evenly stored on nodes, ensuring system reliability.
- The service layer provides S3 APIs. It provides access to the object storage service, a global namespace, and a variety of value-added features, such as deduplication, QoS, multi-tenant, and quota. In addition, FusionStorage object storage supports mainstream object storage protocols and implements on-demand storage resource allocation.

The software architecture of FusionStorage object storage has the following highlights:

- **Industry-leading distributed architecture**

  The fully distributed software architecture of FusionStorage object storage features distributed cluster management, a DHT routing algorithm, distributed stateless engines, and distributed intelligent caching. This eliminates single points of failure (SPOFs) across the whole storage system.
- **High reliability and performance**

  FusionStorage object storage balances loads among all disks and dispersedly stores data, thereby preventing data hotspots in the system. Effective routing algorithms and distributed caching ensure high performance.
- **Rapid concurrent data reconstruction**

  If disks become faulty, the system automatically, concurrently, and quickly reconstructs the disks using data fragments distributed across different nodes in the resource pool.
- **Easy expansion and ultra-large capacity**

  The distributed stateless engines of FusionStorage object storage support ultra-large scale-out expansion, ensuring smooth and concurrent increases in storage and computing resources.

3.2 Data Services

FusionStorage object storage provides APIs complying with S3, de-facto standards in the cloud storage fields. This enables FusionStorage object storage to be widely used and supported by multiple tools, development packages, and third-party software. Developed based on HTTP, HTTPS and S3 are mature Representational State Transfer (REST) protocols. Complying with HTTP design principles, REST protocols are simple, reliable, and stateless, natively ideal for network access.

FusionStorage object storage adopts an account-bucket-object model. Buckets can be regarded as directories and objects can be regarded as files. Users can locate and use their data through Uniform Resource Identifiers (URIs). FusionStorage object storage abandons the directory tree structure. It has simplified read and write semantics, and is suitable for storing huge amounts of unstructured data on which reads are performed more frequently than writes.

FusionStorage object storage has the following advantages:

- Adopts a cutting-edge scale-out distributed architecture and DHT routing algorithm to meet the requirements of mass data storage.
- Supports multiple services by providing external APIs compatible with Amazon S3.
- Provides EC-based data protection techniques, balancing reliability and space usage.
- Supports the multi-tenant mode, making the most of enterprise and private cloud storage resources.
- Features massive scalability, high security, robust reliability, high efficiency, and wide compatibility, applicable to mass data storage and centralized backup.

3.2.1 Global Namespace

FusionStorage object storage supports multiple regions and availability zones (AZs). To support multiple regions and AZs, storage resources in different regions need to be virtualized into a global namespace to implement domain name resolution, location services (LSs), and load balancing. This allows clients to access object storage space and resources using domain names.

Architecture

To use the object storage service on a client, a tenant needs to create buckets as well as create and manage objects in the buckets. When creating a bucket, the tenant specifies the region to which the bucket belongs and associates the bucket with a cluster in the specified region. An object belongs to a bucket, and a bucket belongs to a region. A bucket name must be globally unique. When the tenant initiates a request to access a bucket or an object in a bucket, the system first resolves domain names, queries bucket locations, and balances the load. Then, the system enables the tenant to establish a connection with the cluster that processes the access request.

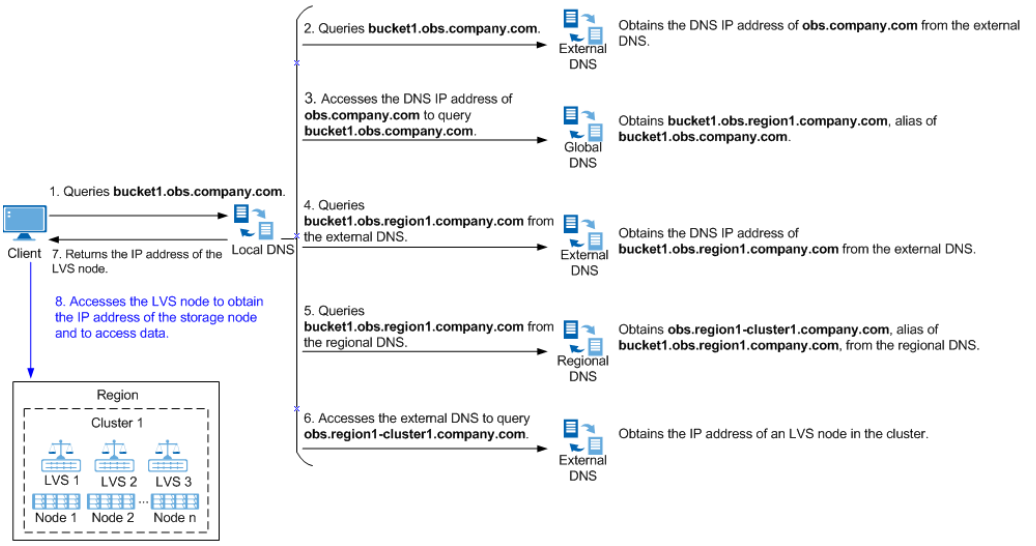
The global namespace is implemented as follows:

- Clients access service domain names of object storage through Domain Name Systems (DNSs).
- A global DNS is used to construct the global namespace. Based on bucket names, the system can provide global and regional domain names for buckets. The global domain name of a bucket is in the *Bucket name.System global domain name* format, and the regional domain name of the bucket is in the *Bucket name.Regional domain name* format. For example, if the bucket name is **bucket1**, the system global domain name is **obs.company.com**, and the regional domain name is **obs.region1.company.com**, then the global domain name of the bucket is **bucket1.obs.company.com** and the regional domain name of the bucket is **bucket1.obs.region1.company.com**.

Domain Name Resolution

Figure 3-1 shows the domain name resolution process.

Figure 3-2 Process of resolving domain names of the object storage service



1. A client sends a request to the local DNS to query bucket **bucket1** using bucket domain name **bucket1.obs.company.com**.
2. The local DNS cannot resolve domain name **bucket1.obs.company.com** and forwards the request to the external DNS. The external DNS cannot resolve the domain name but knows that **obs.company.com** can be queried in the global DNS. Therefore, the external DNS returns the DNS IP address of **obs.company.com** to the local DNS.
3. The local DNS accesses the DNS IP address of **obs.company.com** and obtains **bucket1.obs.region1.company.com**, alias of **bucket1.obs.company.com**, from the global DNS.
4. The local DNS accesses **bucket1.obs.region1.company.com** and obtains the DNS IP address of **bucket1.obs.region1.company.com** from the external DNS.
5. The local DNS accesses the DNS IP address of **bucket1.obs.region1.company.com** and obtains **obs.region1-cluster1.company.com**, alias of **bucket1.obs.region1.company.com**, from the regional DNS.
6. The local DNS accesses **obs.region1-cluster1.company.com** and obtains the IP address of a Linux Virtual Server (LVS) node (taking LVS 1 as an example) in the cluster where bucket **bucket1** is located from the external DNS.

NOTE

This example uses LVS nodes for load balancing. A cluster can contain multiple LVS nodes, with service requests distributed equally among them.

7. The local DNS returns the IP address of LVS 1 to the client.
8. The client sends a request to LVS 1. LVS 1 selects a suitable storage node based on load balancing policies and forwards the request to the storage node for processing.

3.2.2 Distributed Hash Routing

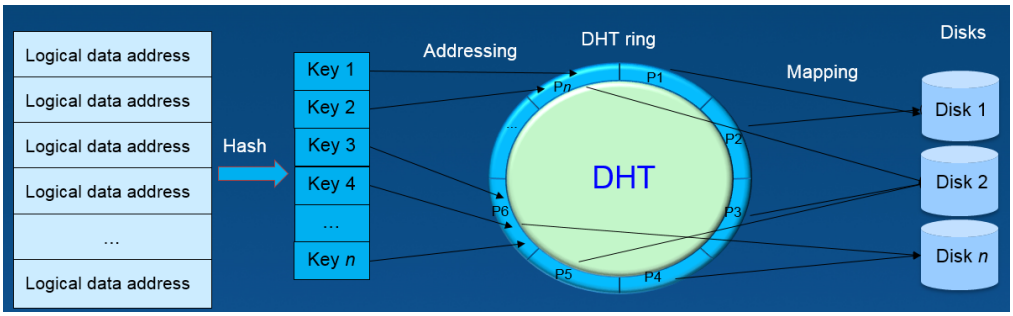
FusionStorage object storage adopts a DHT routing algorithm to address and store data. Each storage node stores a small proportion of data.

Instead of using DHT routing algorithms, traditional storage systems manage metadata centrally. On a traditional storage system, each I/O operation will initiate a query request to the metadata service. As the system scale grows, the metadata size also increases. The

concurrent operation capability of the system is subject to the capacity of the server running the metadata service. As a result, the metadata service eventually becomes a system performance bottleneck.

Unlike traditional storage systems, FusionStorage object storage employs a DHT routing algorithm for data addressing, as shown in Figure 3-2.

Figure 3-3 Data addressing of FusionStorage object storage



NOTE

- DHT ring (also called hash space): a ring space consisting of up to 2^{32} ultra-large logical space units
- P: short for partition. The DHT ring is evenly divided into *N* parts (*N* indicates a number), and each part is a partition.
- Disk: One or more partitions map to each disk.

The DHT ring of FusionStorage object storage contains a maximum of 2^{32} logical space units, and is evenly divided into *N* partitions. The *N* partitions are evenly allocated on all disks in the system. For example, if *N* is 3600 and the system has 36 disks, each disk is allocated 100 partitions. The system configures the partition-disk mapping during system initialization and will adjust the mapping accordingly after the number of disks in the system changes. Mapping tables occupy only a small space and are stored in the memory of the primary management node for fast routing.

The DHT ring technology adopted by FusionStorage object storage has the following advantages:

- **Outstanding performance**

  The DHT ring enables data to be evenly stored and processed on all disks, eliminating read and write performance bottlenecks incurred by frequent data access on certain disks. Unlike traditional storage systems, FusionStorage object storage does not manage metadata centrally. Therefore, the metadata service does not become a performance bottleneck of the system.

- **High reliability**

  The partition allocation algorithms are flexible. Identical data copies are not stored onto the same disk, server, or cabinet.

- **Rapid scale-out**

  When new physical nodes are added, only part of the data needs to be migrated for load balancing.

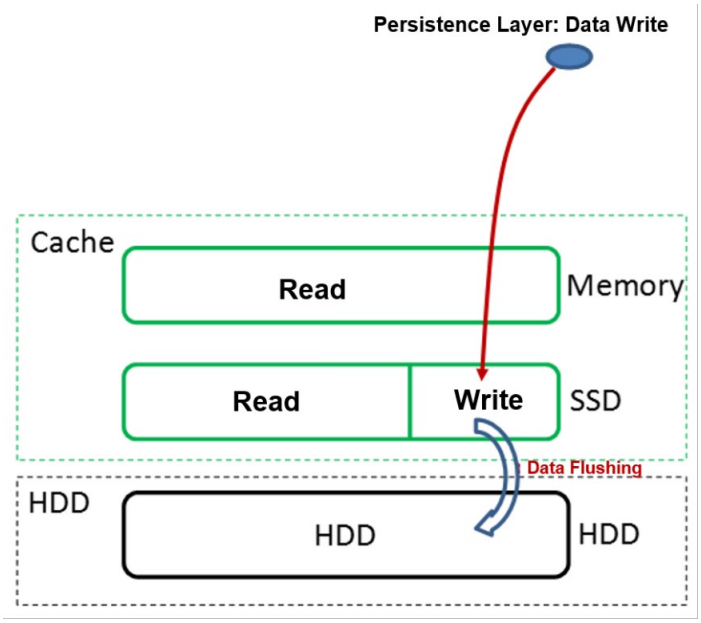
3.2.3 Cache Mechanisms

FusionStorage object storage employs multi-level cache mechanisms to improve storage I/O performance. The write and read cache mechanisms are different.

Write Cache Mechanism

During an I/O write on a node, the persistence layer stores write I/Os in the SSD cache and completes the write on the node. Then, the persistence layer periodically flushes write I/Os from the SSD cache onto HDDs in batches. A threshold is also set for the write cache. If the threshold is reached, data will also be automatically flushed to disks. The following figure shows the details.

Figure 3-4 Write cache mechanism



NOTE

FusionStorage object storage supports large I/O pass-through. By default, I/Os greater than 256 KB will be written directly to disks rather than to the cache. This configuration is modifiable.

Read Cache Mechanism

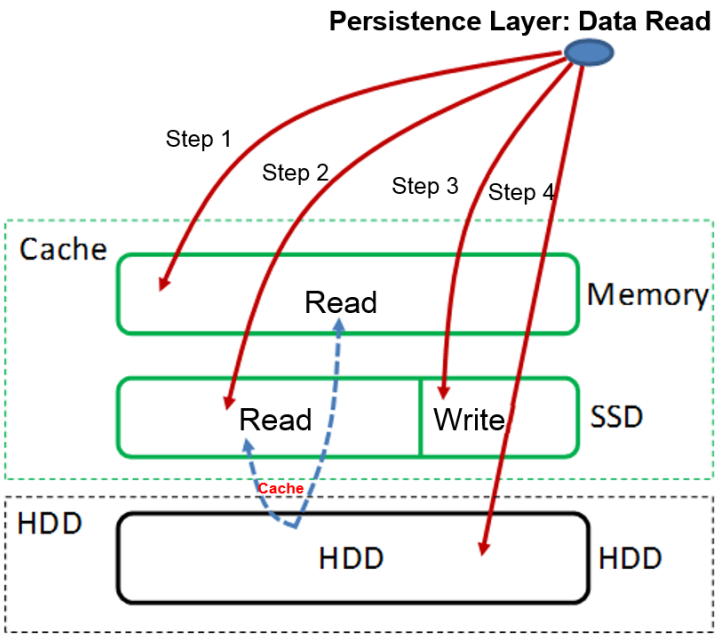
FusionStorage object storage uses SSDs as read cache media to speed up storage access. It employs a multi-level read cache mechanism. Level 1 (L1) is the memory cache and uses the least recently used (LRU) mechanism to cache data. Level 2 (L2) is the SSD cache and leverages the hotspot read mechanism to collect statistics of read data and record hotspot access factors. When the hotspot access factors of data reach a specific threshold, the system automatically caches the data onto the SSD cache and removes data that has not been accessed for a long time from the SSD cache. FusionStorage object storage also supports prefetching. During a data read, FusionStorage object storage will calculate correlation of read data and fetch highly correlated data blocks to the SSD cache.

When the persistence layer receives an I/O read operation from the upper layer:

2. The persistence layer checks whether required I/O data is in the memory read cache. If the data is in the memory read cache, the persistence layer returns the data and moves the data to the head of the LRU queue in the read cache. Otherwise, the persistence layer proceeds to step 2.

3. The persistence layer checks whether the required I/O data is in the SSD read cache. If the data is in the SSD read cache, the persistence layer returns the data and increases the hotspot access factor of the data. Otherwise, the persistence layer proceeds to step 3.
4. The persistence layer checks whether the required I/O data is in the SSD write cache. If the data is in the SSD write cache, the persistence layer returns the data and increases the hotspot access factor of the data. If the hotspot access factor reaches the threshold, the persistence layer fetches the data to the SSD read cache. If the data is not in the SSD write cache, the persistence layer proceeds to step 4.
5. The persistence layer locates the required I/O data on disks and returns the data. In addition, the persistence layer increases the hotspot access factor of the data and fetches the data to the SSD read cache if the hotspot access factor reaches the threshold.

Figure 3-5 Read cache mechanism

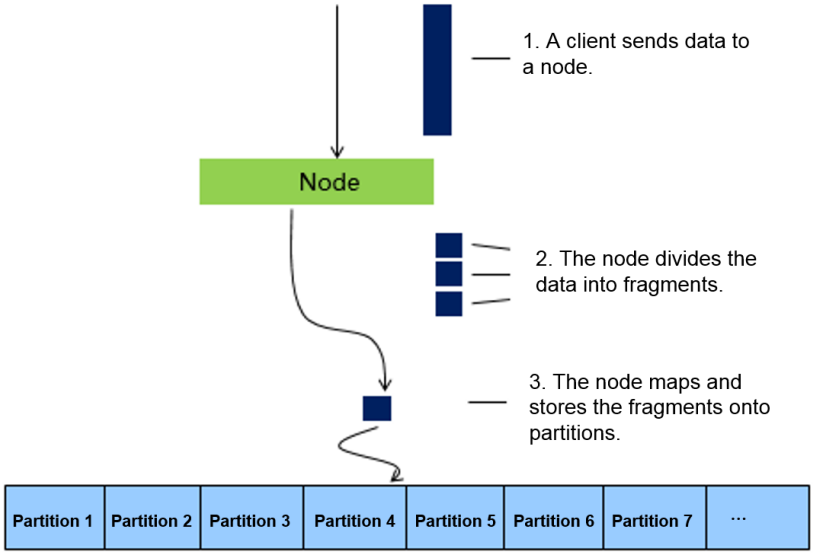


3.2.4 I/O Processes

Data Write Process

Figure 3-6 shows the data write process.

Figure 3-6 Data write process

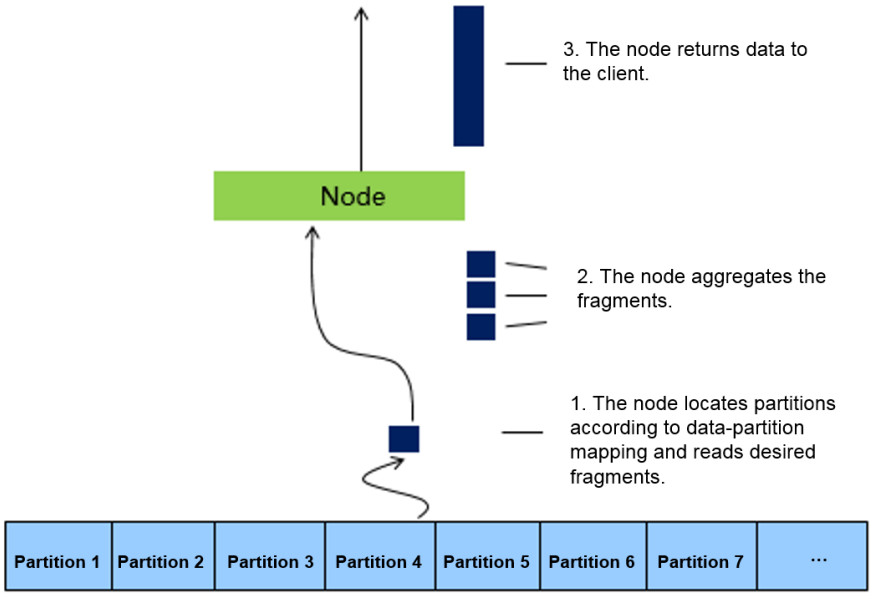


1. Access request: An object storage client sets up a connection with a node providing the object storage service and transmits data to the node.
2. Storage policy selection: The node determines the data storage policy based on user configurations.
3. Data fragmentation: The node calculates the fragment size based on the data storage policy and divides the data into fragments of the same size.
4. Data routing: The node disperses the fragments onto different disks by invoking storage APIs.

Data Read Process

Figure 3-7 shows the data read process, which is the reverse of the data write process.

Figure 3-7 Data read process



1. Access request: An object storage client sets up a connection with a node providing the object storage service and reads data from the node.
2. Data routing: The node locates partitions using the DHT routing algorithm and reads desired fragments.
3. Data restoration: If some fragments are damaged, the node restores them based on data storage policies.
4. Data aggregation: The node aggregates the fragments to generate a complete piece of data and sends the data to the client.

Buffers in Memories

FusionStorage object storage nodes reserve buffers in memories to fragment and aggregate data during data writes and reads. The buffers function as follows:

- During data writes, data and parity fragments are stored in buffers and then concurrently written to multiple nodes to achieve high write efficiency.
- During data reads, nodes will predict the data read scope, read continuous fragments from multiple nodes in advance, and store the fragments in the buffers to improve data read efficiency.

The access service of FusionStorage object storage dynamically adjusts buffer sizes and the number of nodes that concurrently respond to reads and writes according to data sizes and connection speed of clients. This achieves the highest data throughput using the least resources.

3.2.5 Features

3.2.5.1 Data Redundancy

FusionStorage object storage implements data redundancy using EC, ensuring data reliability and availability in the event of hardware failures.

The access service of FusionStorage object storage fragments data uploaded by users, divides *N* consecutive data fragments into an EC group, and calculates the EC group using EC to generate *M* parity fragments. The data and parity fragments in each EC group are stored in a group of consecutive partitions in the storage cluster. This ensures that fragments in the same EC group are stored on different physical nodes, improving reliability.

A maximum of *M* fragments can be damaged in an EC group without impacting the ability to recover an object. The access service of FusionStorage object storage can restore damaged fragments using other fragments in the EC group.

By using EC, FusionStorage object storage delivers high data reliability and provides higher storage space utilization than the multi-copy mode, striking an optimal balance between reliability and cost-effectiveness.

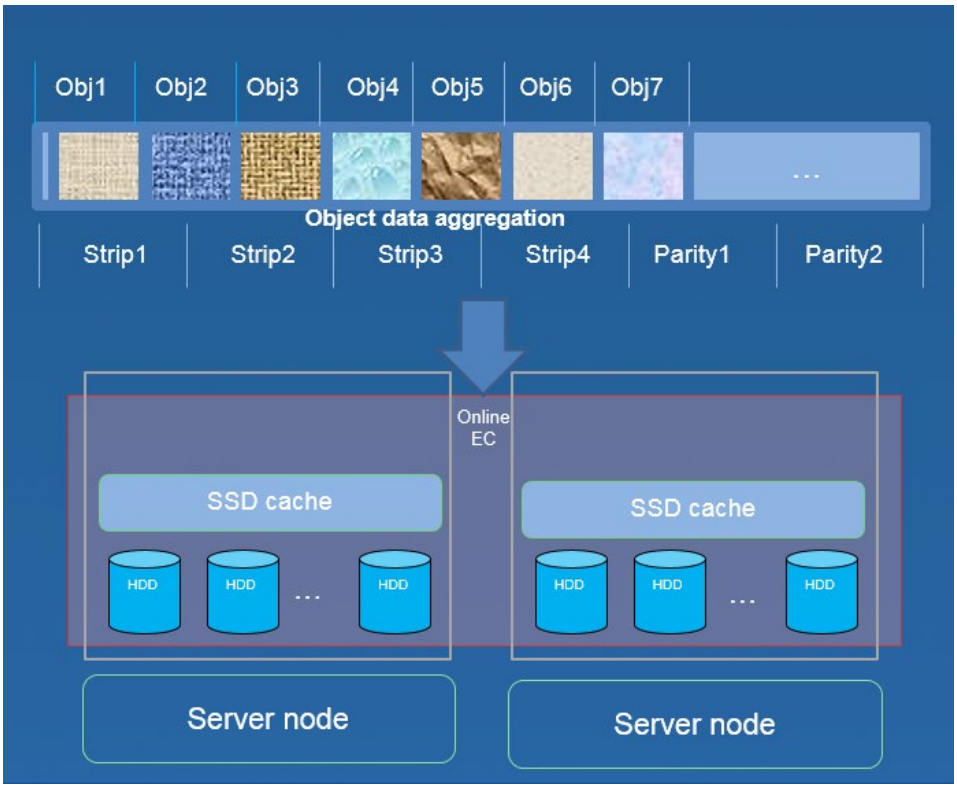
3.2.5.2 Online Aggregation of Small Objects

Traditional object storage systems face the following challenges incurred by small objects:

- Three copies are kept for each small object. The system space utilization is only about 33%.
- When encoding small objects using EC, the system must read these objects from HDDs, imposing high demands on performance.

To address these two challenges, FusionStorage object storage provides the capability of aggregating small objects online, significantly improving the space utilization. Figure 3-8 shows the aggregation process.

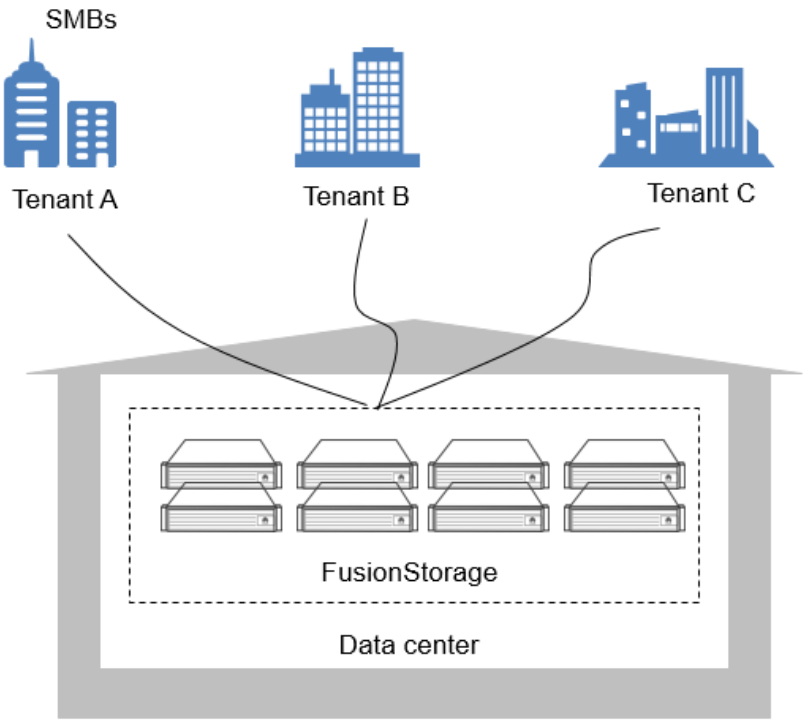
Figure 3-8 Online aggregation of small objects



As shown in the preceding figure, small objects (such as **Obj1**) uploaded by clients are written into the SSD cache first. After the total size of the small objects reaches the size of a stripe, the system calculates the objects using EC and stores generated data fragments (such as **Strip1**) and parity fragments (such as **Parity1**) onto HDDs. In this way, small objects are erasure coded, and the space utilization is significantly improved. For example, if the EC scheme is 12+3, the space utilization is about 80%, approximately 2.4 times higher than 33% of the traditional three-copy mode.

3.2.5.3 Multi-Tenant Management

Figure 3-9 Multi-Tenant

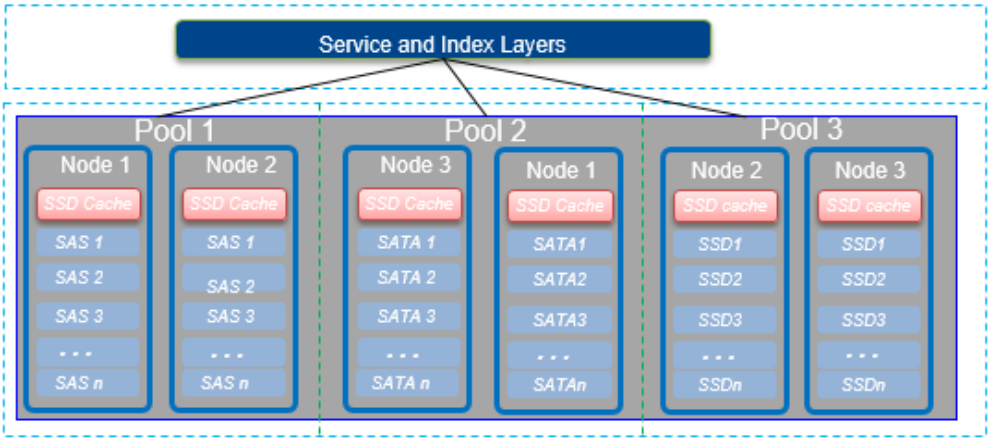


As shown in the preceding figure, FusionStorage object storage provides multi-tenant management. Data of different tenants is logically isolated to facilitate resource allocation. Multi-tenant management has the following benefits:

- A single system provides a variety of client services, reducing initial investments.
- The system is centrally managed, data is logically isolated, and online storage is supported.
- Encrypted HTTPS transmission and user authentication are supported to ensure data transmission security.

3.2.5.4 Multiple Storage Pools

Figure 3-10 Multiple storage pools



As shown in the preceding figure, FusionStorage object storage supports storage pools with different performance and reliability levels, meeting a variety of user requirements. The storage pools have the following characteristics:

- **Customized storage policy**

  You can customize primary storage, cache, and redundancy policies by storage pool to meet the performance and cost requirements of different services.

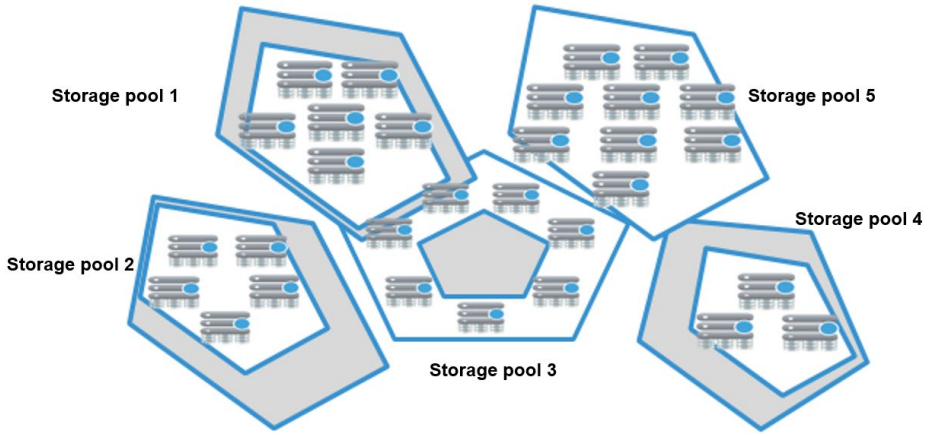
- **Ultimate scalability**

  One FusionStorage object storage cluster supports a maximum of 128 storage pools and 4096 servers, meeting future cloud service expansion requirements.

- **High reliability**

  Storage pools are isolated from each other, preventing one faulty storage pool from affecting others.

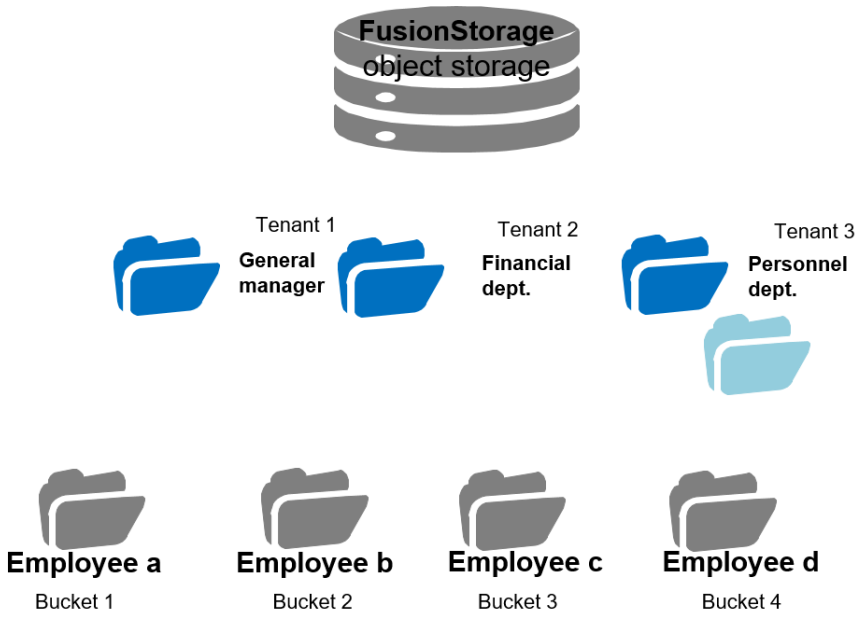
Figure 3-11 Isolated storage pools



3.2.5.5 Quota and Resource Statistics

FusionStorage object storage supports bucket and tenant capacity quotas as well as object resource statistics. The following figure shows a capacity quota example where company departments represent tenants and employees in the departments represent buckets. You can set a 40 TB quota for the financial department (tenant 2) and a 10 TB quota for employee b (bucket 2) in the department.

Figure 3-12 Quota



The capacity quota function of FusionStorage object storage has the following characteristics:

- **Bucket capacity quota**

  Specifies the maximum size of a bucket. When the bucket size reaches the specified upper limit, new data cannot be written into the bucket.

- **Tenant capacity quota**

  Specifies the maximum capacity assigned to a tenant. When the total size of buckets in a tenant reaches the specified upper limit, the tenant and all its users cannot write new data.

FusionStorage object storage can use REST APIs to obtain resource statistics of tenants and buckets, such as the number and capacity of objects.

- **Bucket resource statistics**

  Includes bucket sizes and the number of objects in buckets. Users can query their own bucket resources.

- **Tenant resource statistics**

  Includes the tenant quotas, number of buckets and objects in tenants, and the total capacity.

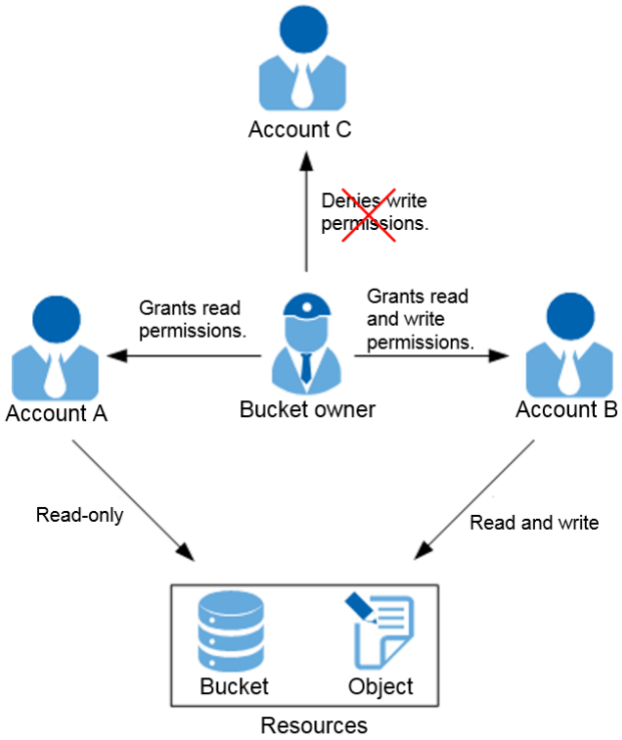
3.2.5.6 Access Permission Control

FusionStorage object storage implements access permission control for buckets and objects. You can only access resources for which you have permissions. ACLs and bucket policies are used to implement the access permission control.

ACL

ACLs grant accounts (also called tenants in FusionStorage object storage) the permission to access resources. Each entry in an ACL specifies permissions (read-only, write, or read and write) of specific accounts. ACLs can grant but cannot deny permissions. The following figure shows an example.

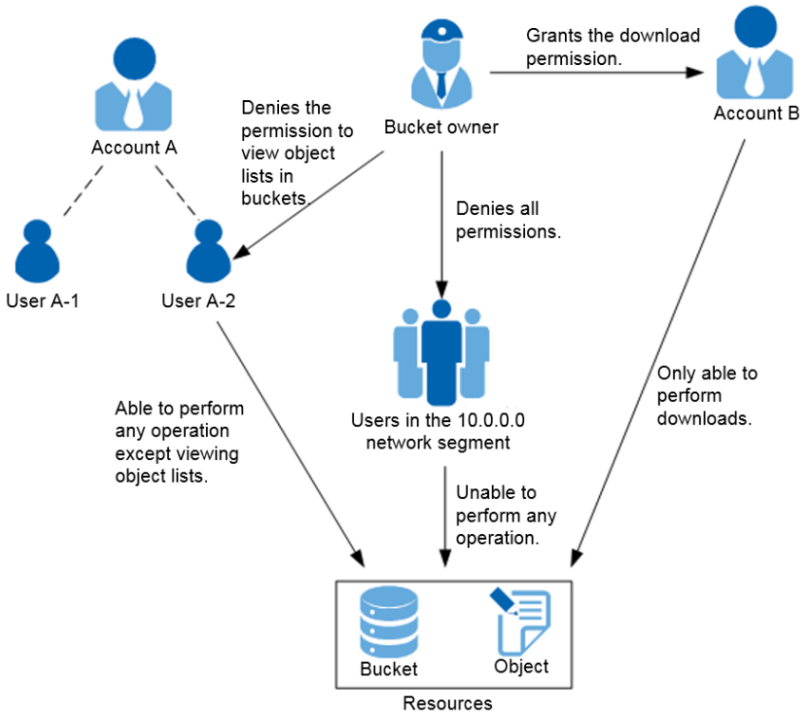
Figure 3-13 ACL



Bucket Policy

Bucket policies control access from accounts and users to buckets and objects. Bucket policies can both grant and deny permissions. Bucket policies provide more refined permission control than ACLs. For example, bucket policies can control specific operations (such as PUT, GET, and DELETE), forcibly enable HTTPS access, control access from specific IP address segments, allow access to objects with specific prefixes, and grant access permissions to specific clients. The following figure shows an example.

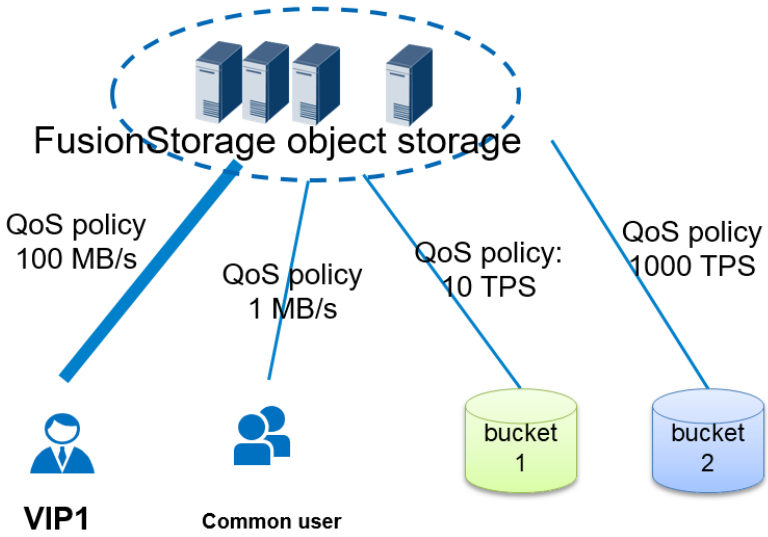
Figure 3-14 Bucket policy



3.2.5.7 QoS

FusionStorage object storage provides QoS for the object storage service to properly allocate system resources and deliver better service capabilities.

Figure 3-15 QoS



In multi-tenant scenarios such as private cloud, customers require that transactions per second (TPS) and bandwidth resources in storage pools be properly allocated to tenants or buckets with different priorities and that the TPS and bandwidth resources of mission-critical services

be sufficient. To meet customer requirements, FusionStorage object storage provides the following refined QoS capabilities:

- **Refined I/O control**

  Enables the system to provide differentiated services for tenants and buckets with different priorities.

- **TPS- and bandwidth-based QoS for tenants and buckets**

  Accurately controls operations, such as PUT, GET, DELETE, and LIST.

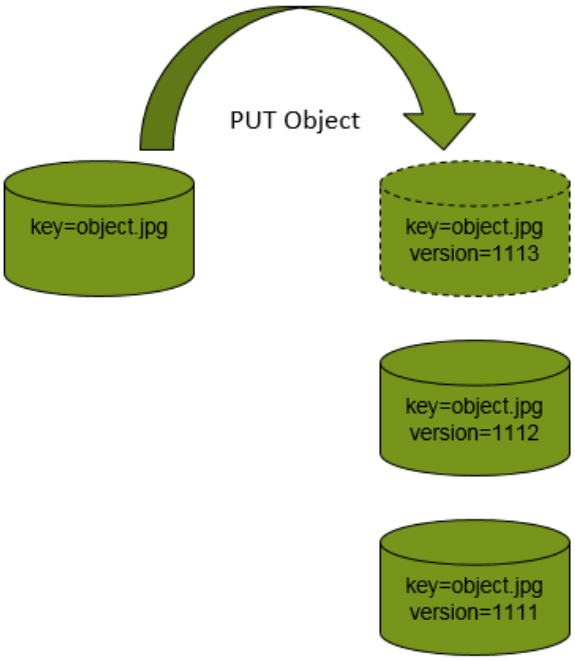
QoS allocates buckets with different TPS and bandwidth capabilities for applications of different priorities. This maximizes storage pool resource utilization and prevents mission-critical services from being affected by other services. Different QoS policies can be configured for VIP and common tenants in the same system to ensure service quality for high-priority tenants.

3.2.5.8 Object Versioning

FusionStorage object storage employs object versioning to keep multiple versions of an object in a bucket. Object versioning prevents the loss incurred by mistaken object deletions and overwrites.

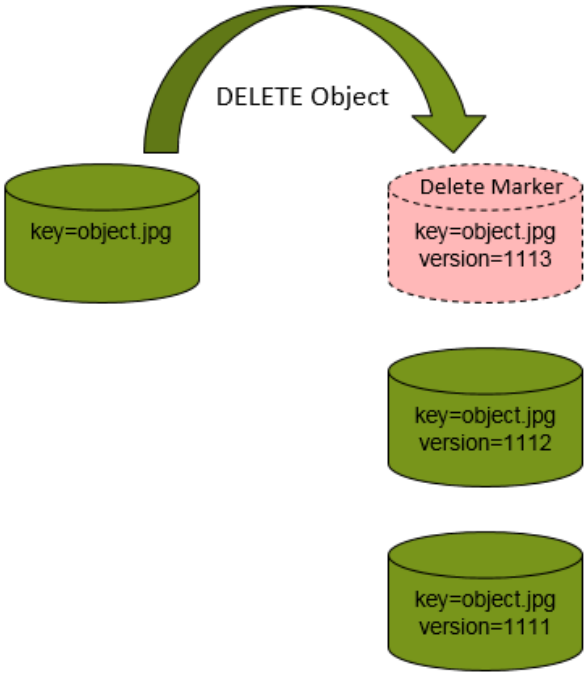
When you upload a new version of an object, the early versions of the object will not be overwritten. Figure 3-6 shows an example. When a new version of **object.jpg** is uploaded into a bucket that already contains objects of the same name, the original objects still remain in the bucket and can be downloaded by specifying their versions.

Figure 3-16 PUT object



When you delete an object without specifying a version ID, the system generates a delete marker, as shown in Figure 3-7. All versions remain in the bucket and are accessible.

Figure 3-17 DELETE object



By default, object versioning is disabled.

After object versioning is enabled:

- The system creates a unique version ID for each uploaded object. Namesake objects are not overwritten and are distinguished by version IDs.
- Objects can be downloaded by version ID. By default, the latest versions of objects are downloaded if no version ID is specified.
- If you delete an object without specifying its version ID, the system only generates a delete marker and remains all versions of the object in the bucket. To permanently delete an object, you must specify its version ID.
- When you list objects, the latest versions are returned by default. You can also list all versions of objects.

After object versioning is suspended:

- All versions of existing objects remain in buckets.
- The system sets version IDs of new objects to null. Such objects will be overwritten after newer objects of the same names are uploaded.
- Objects can be downloaded by version ID. By default, the latest versions of objects are downloaded if no version ID is specified.
- You can delete an object by specifying its version ID. If you do not specify a version ID, the latest version of the object is deleted and a delete marker whose version ID is null is generated.

3.2.5.9 Object Lifecycle Management

FusionStorage object storage provides object lifecycle management to store objects cost-effectively throughout their lifecycle. By using this function, you can define whether

objects expire after being stored for a specific period of time or at a specific point in time. This function is also available for noncurrent versions of objects when object versioning is enabled.

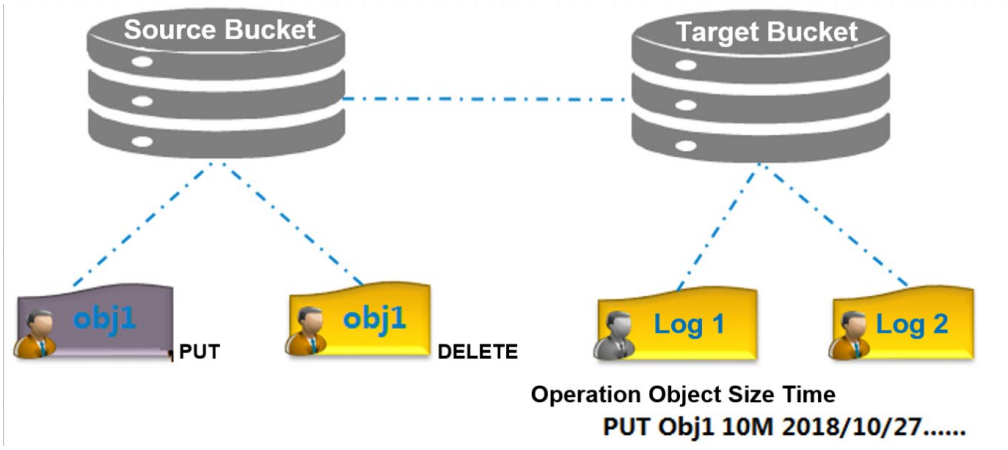
For example:

- You may need periodic log files for a week or month. After that, you might want to delete them.
- Some documents are frequently accessed for a period of time but then infrequently accessed. You can archive these documents and delete them later.

3.2.5.10 Bucket Access Logging

FusionStorage object storage implements bucket access logging for security audit and system operation tracing. After access logging is enabled for a bucket, the system records all operations (including PUT, GET, and DELETE) on the bucket in logs, consolidates the logs into log files, and saves the log files in a specified bucket. Besides the operations, requesters, bucket names, request time, response status, and error codes (if any) will also be recorded in the logs.

Figure 3-18 Bucket access logging

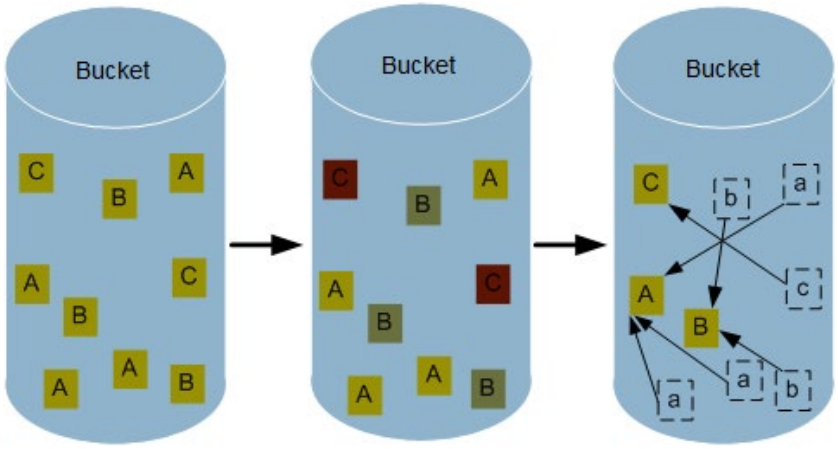


In the preceding figure, the source bucket is enabled with access logging and the target bucket is where access logs of the source bucket will be stored. Operations, such as PUT and DELETE, will be recorded in logs, and the logs will eventually be stored in the target bucket. You can list the logs in the target bucket and download desired logs.

3.2.5.11 Object-Level Deduplication

Object-level deduplication enables the system to automatically detect and delete duplicate objects. After object-level deduplication is enabled for accounts, the system automatically searches for duplicate objects belonging to the accounts, retains one copy of an object, and replaces its duplicates with pointers indicating the location of the one remaining copy. By doing so, redundant data is deleted and storage space is freed up.

Figure 3-19 Deduplication



The system identifies duplicate objects by comparing their MD5 values, data protection levels, and sizes. Object-level deduplication saves storage space and increases space utilization.

For example, when clients upload identical files (or images, videos, software) onto a web disk, the system with object-level deduplication enabled will save only one copy and replace all the other identical files with pointers indicating the location of that one copy.

Enabling and disabling object-level deduplication is done at the account level. Object-level deduplication is disabled by default.

3.2.5.12 WORM

Write Once Read Many (WORM) is a technology that allows data to be read-only once being written. Users can set protection periods for objects. During protection periods, objects can be read but cannot be modified or deleted. After protection periods expire, objects can be read or deleted but cannot be modified. WORM is mandatory for filing systems.

The WORM feature of FusionStorage object storage does not provide any privileged interfaces or methods to delete or modify object data that has the WORM feature enabled.

WORM policies can be configured for buckets. Different buckets can be configured with different WORM policies. In addition, you can specify different object name prefixes and protection periods in WORM policies. For example, you can set a 100-day protection period for objects whose names start with **prefix1** and a 365-day protection period for objects whose names start with **prefix2**.

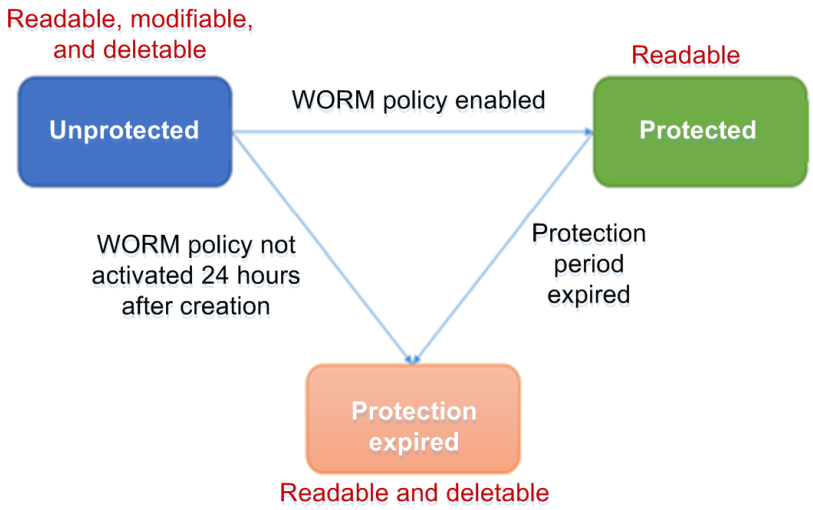
FusionStorage object storage uses built-in WORM clocks to time protection periods. After a WORM clock is set, the system times protection periods according to the clock. This ensures that objects are properly protected even if the local clock time is changed. Each object has creation time and expiration time measured by its WORM clock. After WORM properties are set for an object, the object uses a WORM clock to count the time, preventing its protection period from being changed due to local node time changes.

A WORM clock can automatically adjust its time according to the local node time:

- If the local node time is earlier than the WORM clock time, the WORM clock winds back its time 128 seconds or less every hour.
- If the local node time is later than the WORM clock time, the WORM clock adjusts its time to the local node time.

Objects enabled with WORM have three states: unprotected, protected, and protection expired, as shown in the following figure.

Figure 3-20 WORM



- Unprotected: Objects in the unprotected state can be read, modified, and deleted, same as common objects.
- Protected: After a WORM policy is enabled for a bucket, the objects that meet the WORM policy enter the protected state and can only be read.
- Protection expired: When the WORM protection period of objects expires, the objects enter the protection expired state. In this state, the objects can only be read or deleted.

3.3 Storage Management

3.3.1 Storage as a Service

As the management graphical user interface (GUI) of FusionStorage object storage, DeviceManager provides the following functions:

- Initialization wizard

  You can create a resource pool, configure and manage authentication modes and regions, and provision services by following wizards.
- Storage pool management

  You can create and delete storage pools, query resource statistics and disk topologies of storage pools, as well as expand or reduce capacities of storage pools.
- Object service configuration

  1. Authentication configuration

     You can select an authentication mode among Provisioning Orchestration Engine (POE), Identity and Access Management (IAM), and Keystone, and complete interconnection. When POE authentication is selected, you can manage service accounts in the default cluster of the default region.

2. Region configuration
   - LS configuration

     Allows you to configure public and private addresses for the global LS, address for the regional LS, and a global domain name.
   - Region management
     - Allows you to add non-default regions to global management.
     - Allows you to add clusters to regions for centralized management.
3. Security configuration

   Security configuration includes setting access policies, Transport Layer Security (TLS) policies, and time verification.
   - Access policies
     - Access policy based on IP addresses and access key IDs (AKs)

       Within a statistical period, if the number of access failures of an AK through an IP address is greater than or equal to the preset access failure threshold and the percentage of access failures in total access attempts is greater than or equal to the preset failure rate threshold, the access from the AK through this IP address will be denied.
     - AK-based access policy

       Within a statistical period, if the number of access failures of an AK is greater than or equal to the preset access failure threshold and the percentage of access failures in total access attempts is greater than or equal to the preset failure rate threshold, the access from the AK will be denied.
     - IP address-based access policy

       Within a statistical period, if the number of access failures of an IP address is greater than or equal to the preset access failure threshold and the percentage of access failures in total access attempts is greater than or equal to the preset failure rate threshold, the access from the IP address will be denied.
   - TLS policies

     You can configure TLS policies as required.
   - Time verification

     When time verification is enabled, access requests will be rejected if the time difference between clients and servers exceeds 15 minutes.
4. Billing service configuration
   - You can change the password of the account that interconnects FusionStorage object storage with a billing center.
   - You can enable or disable the automatic generation of billing files.
5. Other configuration
   - You can configure namespaces for responses of the object storage service.
   - You can configure website redirection to redirect to desired websites if access to static websites hosted by the object storage service fails.

3.3.2 Cluster Management

FusionStorage object storage uses cluster management software to provide the following functions:

- **Basic cluster information monitoring**

You can query basic cluster information, including cluster names, health status, running status, versions, cluster capacities, and node quantity.

- **Performance monitoring**

  You can view the bandwidth, number of requests per 10 seconds, number of request failures per 10 seconds, and service access failure rates per 10 seconds of the object storage service.

- **Account management**

  You can create, modify, and delete object storage service accounts when POE authentication is used.

- **Alarm management**

  You can configure alarm notification, as well as view, handle, mask, and dump alarms.

- **User management**

  You can manage users, configure security policies, manage IP address whitelists, and configure Simple Mail Transfer Protocol (SMTP) servers.

- **License management**

  You can view active licenses and import new ones.

- **Cluster management**

  You can start or stop the system, and set system time, external DNSs, and configuration file import and export rules.

- **Node management**

  You can stop and freeze nodes.

3.3.3 Cluster Expansion

FusionStorage object storage delivers superb scalability thanks to its distributed architecture. A single FusionStorage object storage cluster can contain 3 to 4096 nodes. As the number of nodes increases, the storage and computing capabilities increase linearly. This delivers a linear growth in bandwidth and concurrent request processing capability.

Capacity expansion of FusionStorage object storage has the following characteristics:

- Online capacity expansion is supported. Services are not adversely affected during capacity expansion.
- Flexible capacity expansion is delivered. Nodes can be added to existing or new storage pools.
- When nodes are added to existing storage pools, FusionStorage object storage implements rapid load balancing without migrating a large amount of data.

3.4 Recommended Hardware

FusionStorage object storage is designed based on general-purpose hardware. To ensure system reliability and optimal performance, you are advised to use the hardware models listed in the following table. For more details about hardware configurations, consult your Huawei sales representative.

Table 3-1 Recommended hardware

Hardware Type	Recommended Model	Description
Cabinet	Standard IT cabinet	Providing 42 U space for device installation
General-purpose hardware nodes	Huawei FusionServer 5288 V5	Storage node with 36 disk slots Typical configurations: 112 GB memory, Intel Skylake 4114 V5 CPUs, and 800 GB, 1.6 TB, or 3.2 TB NVMe SSDs as cache
	Huawei FusionServer 2288H V5	Storage node with 12 disk slots Typical configurations: 80 GB memory, Intel Skylake 4114 V5 CPUs, and 800 GB, 1.6 TB, or 3.2 TB NVMe SSDs as cache
	Huawei TaiShan 5280 V2	Storage node with 36 disk slots Typical configurations: 112 GB memory, Huawei-developed Kunpeng 920 CPUs, and 1.6 TB, or 3.2 TB NVMe SSDs as cache
	Huawei TaiShan 2280 V2	Storage node with 12 disk slots Typical configurations: 80 GB memory, Huawei-developed Kunpeng 920 CPUs, and 1.6 TB, or 3.2 TB NVMe SSDs as cache
Network devices	Huawei CE6855-48S6Q-HI	10GE switch
	Huawei CE6865-48S8CQ-EI	10GE or 25GE switch
	Huawei CE5855-48T4S2Q-EI	GE switch
Keyboard, Video, and Mouse (KVM) controller		Providing eight KVM ports

3.5 System Networking

The network planes of FusionStorage object storage are as follows:

- **Service plane**

  Interconnects with the customer's service network and supports multiple subnets.
- **Storage plane**

  Enables communication among FusionStorage object storage nodes and supports multiple subnets. The storage plane supports only IPv4 networking.
- **Management plane**

Interconnects with the customer's management network, enabling maintenance terminals to access FusionStorage object storage.

- **BMC plane**

  Interconnects with Mgmt ports of FusionStorage object storage nodes to enable remote device management.

- **Control plane**

  Manages internal cluster information of FusionStorage object storage.

Figure 3-8 shows a networking diagram.

Figure 3-21 Networking diagram

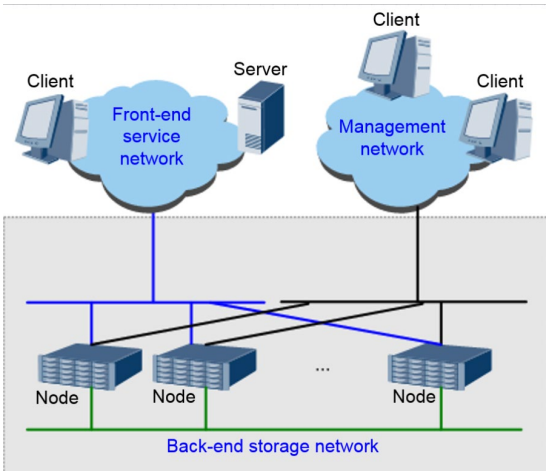


Table 3-1 lists the three networking solutions provided by FusionStorage object storage.

Table 3-2 Networking solutions

Solution	Front-End Service Network	Back-End Storage Network
10GE networking	10GE	10GE
25GE networking	25GE	25GE
GE networking	GE	10GE

In addition, FusionStorage object storage can be used in the Huawei FusionCloud private cloud solution to provide object storage services. When used in the Huawei FusionCloud private cloud solution, FusionStorage object storage follows the networking principles of this solution.

3.5.1 Networking Within a Cluster

FusionStorage object storage supports two networking setups within a cluster, depending on whether service and storage planes share switches.

Figure 3-22 Networking setup in which service and storage planes use separate switches

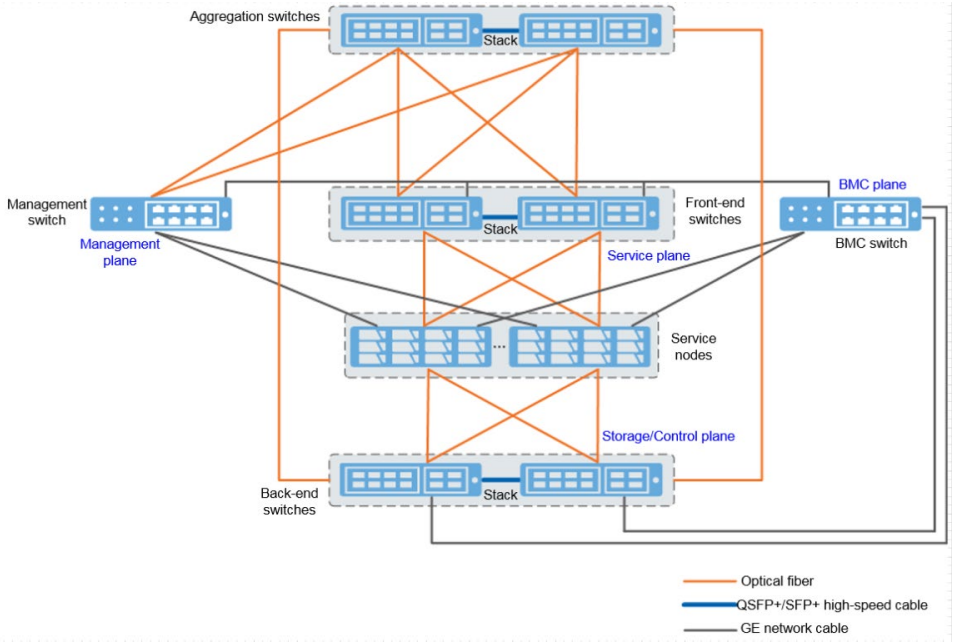
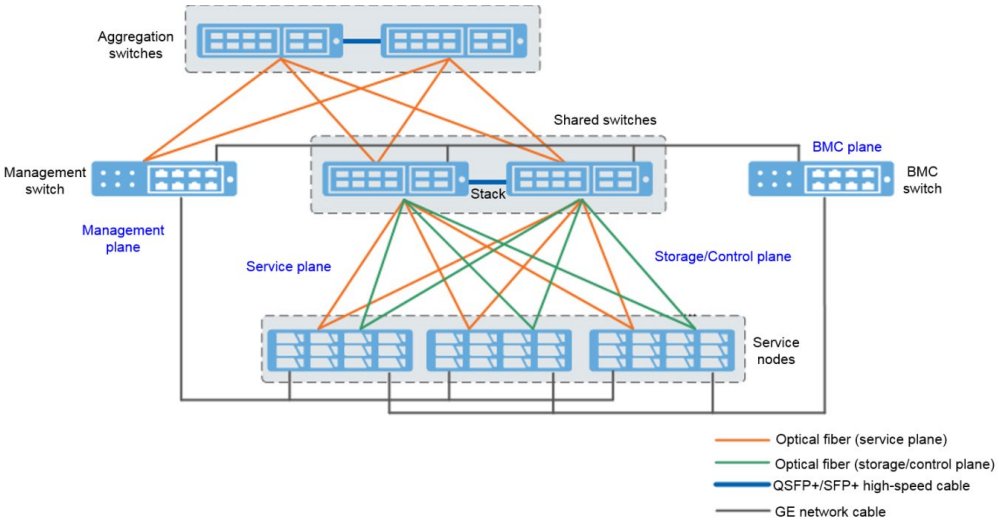


Figure 3-23 Networking setup in which service and storage planes share switches



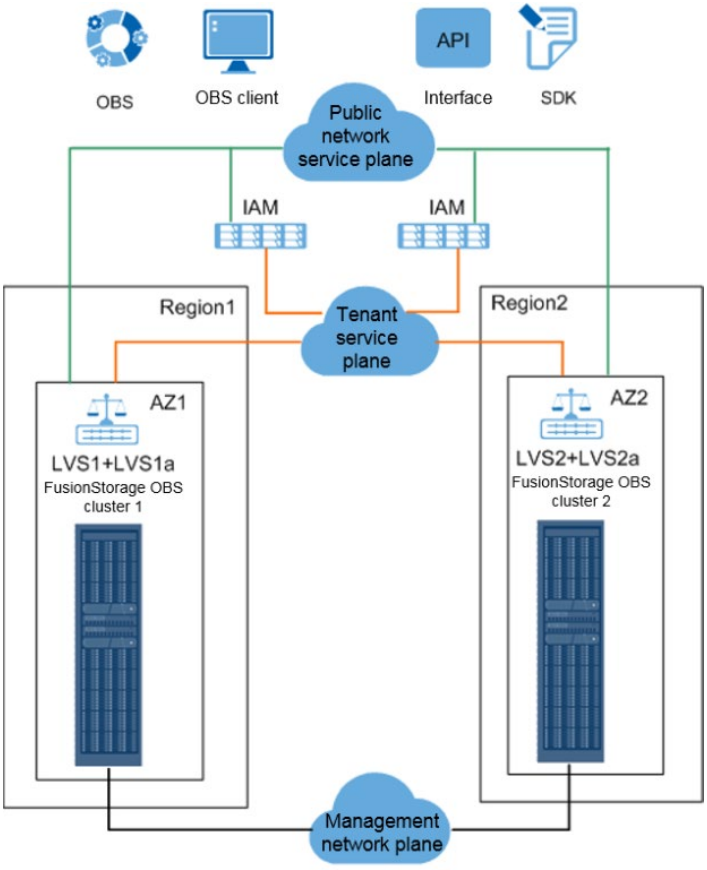
The preceding figures show the connections between nodes and switches in a single subnet. A single cluster consists of several such subnets interconnected through aggregation switches.

3.5.2 Networking Among Regions

FusionStorage object storage supports multi-region networking to provide the object storage service with a global namespace. To meet network and data reliability requirements, FusionStorage object storage supports multi-region multi-AZ deployment. The networking of multi-region multi-AZ deployment is similar to that of multi-region single-AZ deployment. This section uses multi-region single-AZ deployment as an example.

As shown in the following figure, two regions are deployed, each containing one AZ and one cluster. IAM indicates an authentication server. FusionStorage object storage also supports Keystone and built-in POE authentication. This example uses IAM for authentication.

Figure 3-24 Networking Among Regions



NOTE

The preceding figure uses LVS nodes for load balancing.

Load balancing and IAM authentication services are provided by servers independent from the FusionStorage object storage nodes. FusionStorage object storage needs to connect to load balancing clusters and IAM according to the following principles:

- Each FusionStorage object storage cluster connects to a load balancing cluster.
- Load balancing clusters connect to the public network service plane of FusionStorage object storage to implement load balancing.
- FusionStorage object storage connects to IAM to implement unified authentication.

The storage plane is an internal network of FusionStorage object storage. Nodes in each FusionStorage object storage cluster are interconnected through switches on the storage plane. Traffic of the storage plane is not transmitted across clusters.

3.6 DNS and Load Balancing Deployment

3.6.1 LAN-based DNS

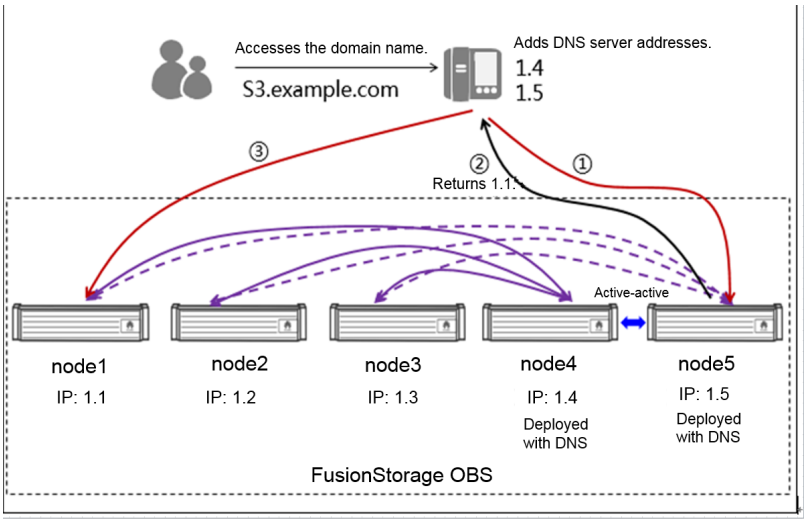
The LAN-based DNS solution is simple and convenient. This section uses a five-node cluster shown in Figure 3-11 as an example. DNS services are deployed on **node4** and **node5** and run in active-active mode.

For clarity, network switches on the storage plane are not illustrated in Figure 3-11, and **1.1**, **1.2**, **1.3**, **1.4**, and **1.5** represent external IP addresses. On clients (PCs or servers), the DNS server addresses are set to the IP addresses (**1.4** and **1.5** in this example) of the nodes that run the DNS service of FusionStorage object storage.

When a user attempts to access domain name **S3.example.com**:

1. The local client selects one (**1.5** in this example) of the two DNS server addresses to resolve domain name **S3.example.com**.
2. The DNS service of FusionStorage object storage on **node5** resolves domain name **S3.example.com** into an IP address (**1.1** in this example) and returns the IP address to the client.
3. The client caches the IP address and then accesses the associated node (**node1** in this example). As long as the IP address is cached, the client does not need to request domain name resolution again. Instead, the client directly accesses the IP address in the cache.

Figure 3-25 LAN-based DNS solution



This solution is easy to deploy but does not support access across network segments. If the node associated with the IP address stored in the cache becomes faulty, the client continues to access the faulty node until the IP address in the cache expires.

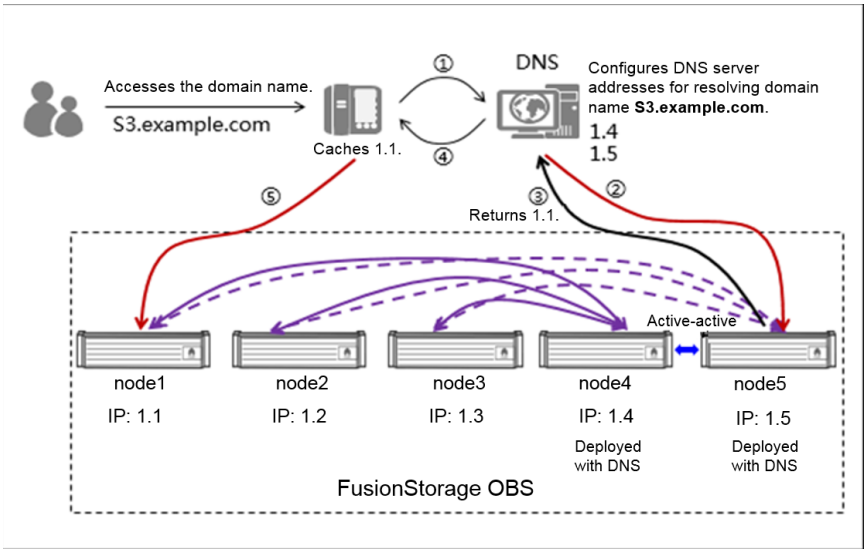
3.6.2 WAN-based DNS

Unlike the LAN-based DNS solution, the WAN-based DNS solution employs DNS servers. In the example shown in Figure 3-12, IP addresses **1.4** and **1.5** of the nodes running the DNS service of FusionStorage object storage are configured to resolve domain name **S3.example.com** on a DNS server.

When a user attempts to access domain name **S3.example.com**:

1. The client requests the DNS server to resolve domain name **S3.example.com** through the WAN.
2. The DNS server selects one DNS server address (**1.5** in this example) from the two addresses and forwards the resolution request to the associated node (**node5** in this example).
3. **node5** resolves **S3.example.com** into IP address **1.1** (associated with **node1**) and returns the IP address to the DNS server.
4. The DNS server caches the resolved IP address locally and forwards it to the client.
5. The client caches the resolved IP address. As long as the IP address is cached, the client does not need to request domain name resolution again. Instead, the client directly accesses the IP address in the cache.

Figure 3-26 WAN-based DNS solution



The advantage of this solution is that you do not need to configure DNS information on clients. Instead, you can directly use clients for data access. The disadvantage is that the DNS server may not have the load balancing function. If the storage node associated with an IP address cached in the DNS server becomes faulty, clients may still access the faulty node. Services running on such clients may be interrupted until the IP address of the faulty node expires in the cache (default IP address expiration time: 30 seconds).

3.6.3 WAN-based Load Balancing

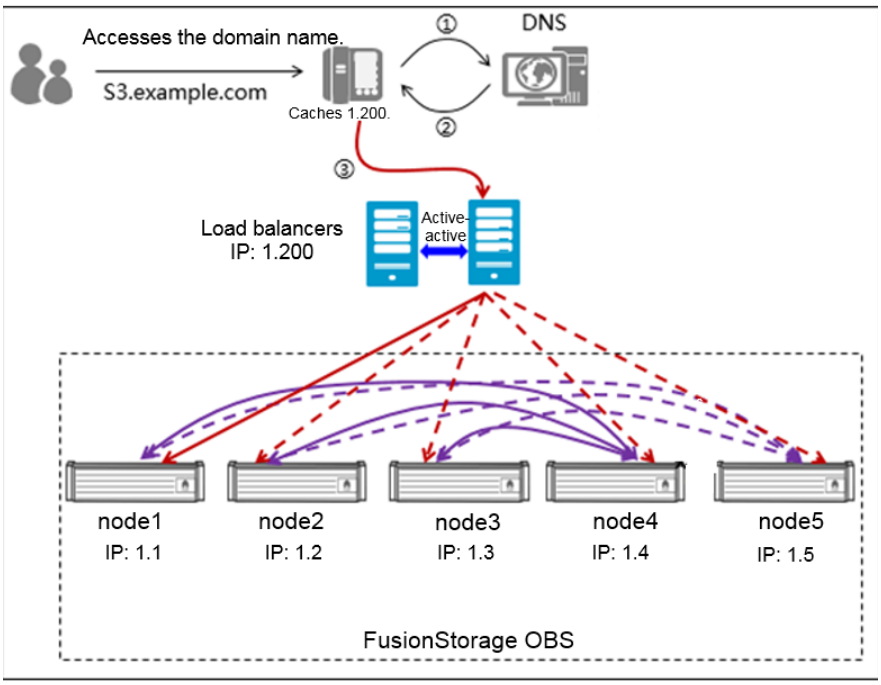
Mandatory for websites, load balancers provide a unified entrance for network access and improve network service availability. Usually, multiple load balancers are deployed to form a cluster. For clarity, Figure 3-13 uses only two load balancers in active-active mode as an example. Assume that the IP address of the load balancers is **1.200** and the IP address has been added to a DNS server.

When a user attempts to access domain name **S3.example.com**:

1. The client requests the DNS server to resolve the domain name.

2. The DNS server returns the IP address (**1.200** in this example) of the load balancers to the client.
3. The client caches the IP address and sends an access request to the IP address.

Figure 3-27 WAN-based load balancing solution



After a load balancer receives an access request, it selects a node (**node1** in this example) based on the health status and workload of each node in the FusionStorage object storage cluster and then forwards the access request to the node. If the access request is a read request, **node1** delivers data to the load balancer. The load balancer forwards the data to the client to complete the access. If the access request is a write request, **node1** writes data into the cluster and returns an operation status code to the load balancer. Then the load balancer forwards the operation status code to the client.

The WAN-based load balancing solution ensures high availability. Even if some nodes are faulty, services will not be interrupted. For example, if **node1** fails, the load balancers will quickly detect the fault (default node status check interval: 1 second), label the node as faulty, and stop assigning tasks to the node.

However, this solution has a higher cost than the LAN- and WAN-based DNS solutions. If a FusionStorage object storage cluster processes only a small amount of data or a small number of access requests, deploying load balancers to prevent service interruption during short peak hours incurs certain costs.

Select a solution based on your requirements to strike a balance between costs and reliability.

4 Outstanding Performance and Scalability

4.1 Superb Single-Bucket Performance

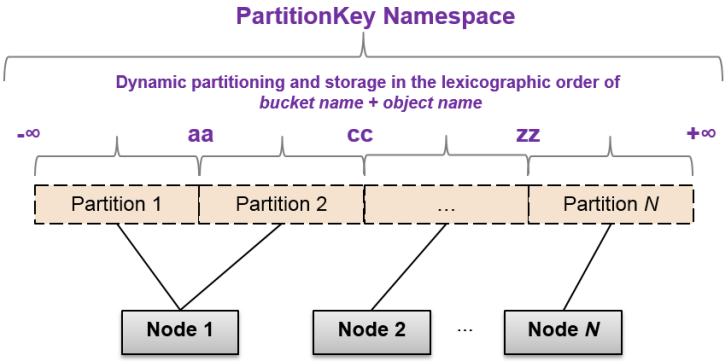
Traditional object storage systems face the following two challenges:

- System scalability is limited, unable to meet 100 PB scalability requirements.
- Metadata access hotspots exist. The concurrent write performance of a bucket is low (300 TPS at most), and the number of objects a bucket can contain is limited (50 million at most).

The two challenges limit the capacity and performance of a single bucket, and the system capability cannot be fully utilized. Multiple buckets are required to meet capacity and performance demands, increasing object storage management complexity. To address these two challenges, FusionStorage object storage improves the performance of a single bucket in the following ways:

- The service, index, and persistence layers of FusionStorage object storage are decoupled from each other and can be expanded independently. A single cluster supports up to 4096 nodes and EB-level scalability, enabling you to store, use, and manage huge volumes of data in a single resource pool. This eliminates the scalability bottleneck of a single bucket.
- Metadata is distributed using dynamic range partitioning technology. Each server only manages a group of fragmented object metadata and supports failover and dynamic load balancing.

Figure 4-1 Dynamic load balancing



As shown in the preceding figure, FusionStorage object storage sorts object names in the lexicographic order of *bucket name+object name* to form a metadata collection. The metadata collection is dynamically stored in multiple partitions based on the metadata size and access frequency. The partitions storing metadata are located in different physical nodes. In this way, metadata is dispersed on all nodes, eliminating the bottleneck of metadata management on a single bucket.

The persistence layer routes data using a DHT routing algorithm and ensures that data is evenly distributed to all nodes and disks in the system, resolving the data distribution bottleneck of a single bucket.

A single bucket supports up to 100 billion objects, fully able to meet the read and write requirements of your applications.

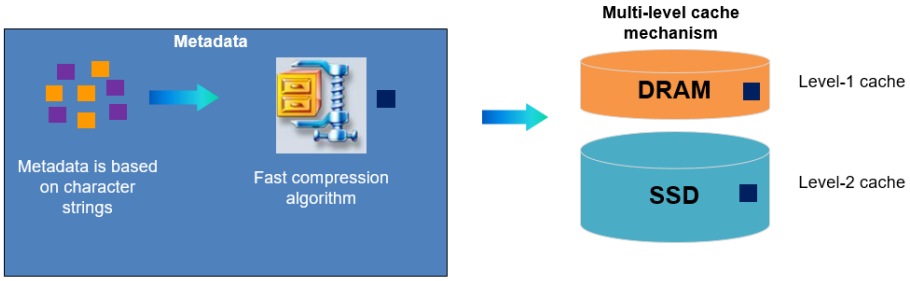
4.2 Dispersed Metadata Storage

FusionStorage object storage does not provide a centralized metadata management service. Instead, all nodes of FusionStorage object storage can provide metadata services. Metadata is fragmented and evenly distributed to nodes using the same DHT routing algorithm as data, eliminating bottlenecks. Requests for metadata services are evenly assigned to nodes. You can increase the number of nodes to improve data request processing capabilities as required.

4.3 Multi-Level Metadata Cache

FusionStorage object storage supports multi-level metadata cache to improve the read performance of objects and ensure quick access to hot data.

Figure 4-2 Multi-Level metadata cache



As shown in the preceding figure, metadata of FusionStorage object storage is compressed before being stored, significantly reducing the metadata volume:

- Metadata is mainly character strings, and the compression rate is high.
- Fast compression algorithms adopted by FusionStorage object storage deliver an excellent compression effect with low CPU usage.

After compression, metadata is first stored on the DRAM (L1 cache), which provides microsecond-level metadata read performance. SSDs function as L2 cache and provide millisecond-level metadata read performance.

4.4 Global Load Balancing

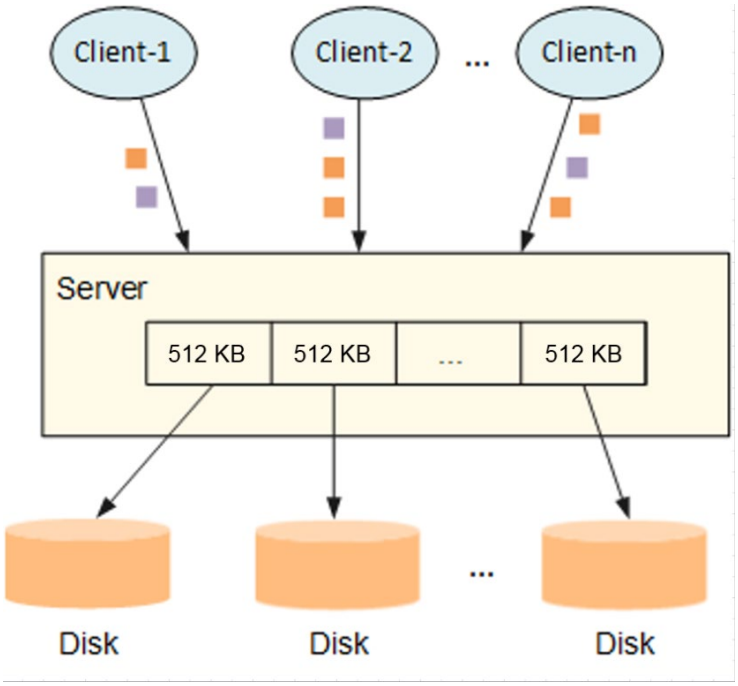
The DHT routing algorithm adopted by FusionStorage object storage evenly allocates data I/Os from upper-layer applications to disks on different servers, globally balancing load and avoiding access hotspots.

- The system automatically scatters data of each object onto different disks of multiple servers. Frequently accessed data and rarely accessed data are evenly distributed on each server, preventing hotspots in the system.
- The fragment distribution algorithm used by FusionStorage object storage ensures that data and parity fragments are evenly distributed on the disks of all servers.
- If nodes are removed due to a failure, or if new nodes are added, FusionStorage object storage employs data restoration and reconstruction algorithms to balance the load among nodes.

4.5 Online Data Aggregation

FusionStorage object storage can aggregate objects of different sizes into a full stripe, divide the stripe into 512 KB fragments, and write the fragments onto HDDs. This maximizes the large I/O advantages of HDDs and avoids HDDs' disadvantages in input/output operations per second (IOPS).

Figure 4-3 Online data aggregation



As shown in the preceding figure, objects uploaded by different clients are aggregated into 512 KB I/Os on the same server. Every *N* 512 KB I/Os are concurrently written onto *N* HDDs (EC scheme: *N*+*M*). A single HDD supports about 200 IOPS and 100 MB/s bandwidth. Assume that clients need to write 200 I/Os, each with 100 KB. If aggregation is not performed, the IOPS bottleneck of the HDD is reached but the required bandwidth is only about 20 MB/s (200 x 100 KB). If a server aggregates the 200 I/Os into forty 512 KB I/Os, the HDD only needs to provide 40 IOPS and 20 MB/s bandwidth. Neither the IOPS nor the bandwidth will constitute bottlenecks. This enables the HDD to process more I/Os, maximizing its bandwidth advantage.

4.6 Stateless Cluster

By using the object storage technology and a one-time addressing DHT routing algorithm, the access service of FusionStorage object storage is loosely coupled with the storage service and nodes are stateless. Based on load balancing, any node can process service requests. The number of nodes is not limited by status synchronization or locking mechanisms, and theoretically can be infinitely increased to support linear capacity expansion.

4.7 Elastic Expansion

Elastic expansion of FusionStorage object storage has the following characteristics:

- **Fast load balancing**

  After new nodes are added, FusionStorage object storage implements fast load balancing and avoids migration of a large amount of data.

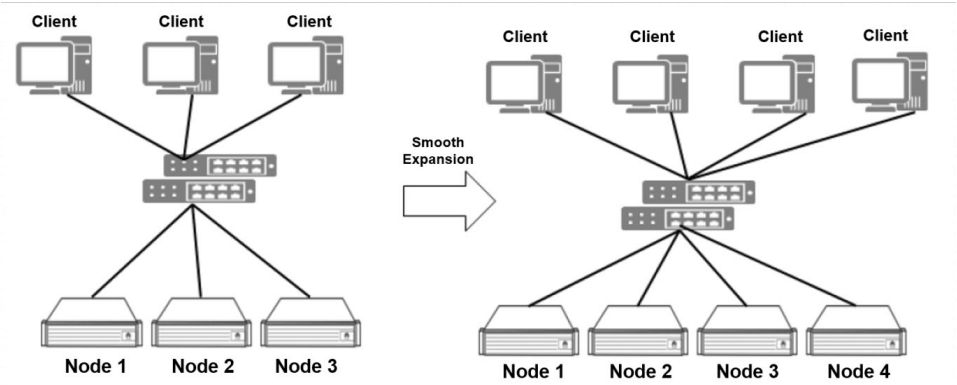
- **Flexible expansion**

  You can add disks and nodes to expand capacity.

- **Linear performance growth**

  Computing, storage, and cache resources are evenly distributed on each node. The system TPS, throughput, and cache linearly increase as more nodes are added.

Figure 4-4 Elastic expansion



FusionStorage object storage supports dynamic node increase. It is recommended that a single FusionStorage object storage cluster contain 3 to 4096 nodes. As the number of nodes increases, the storage and computing capabilities increase linearly. This delivers a linear growth in bandwidth and concurrent request processing capability.

FusionStorage object storage provides a global cache whose capacity expands linearly as the number of nodes increases. In addition, more nodes bring a higher cache hit ratio of hot data, which greatly reduces random disk I/Os and improves the overall system performance.

Increasing capacity or performance of traditional storage systems requires horizontal expansion and reconfiguration of applications, interrupting user services. Unlike traditional storage systems, FusionStorage object storage functions as a single object storage system using a global namespace and is accessible through a unified domain name. These features enable FusionStorage object storage to support minute-level capacity expansion and automatic load balancing. Modification on servers, clients, and applications is not required, and user services are not interrupted.

5 Solid Reliability

5.1 Data Redundancy Protection

FusionStorage object storage uses EC to implement data redundancy protection.

5.1.1 Data Fragmentation

To implement data protection and high read/write performance, the system performs data fragmentation. When an object is being fragmented, the system selects suitable nodes according to the default protection level. After fragmentation, the system evenly distributes fragments on each selected node. During a data read, the system concurrently reads fragments from the nodes.

FusionStorage object storage stores data using EC and supports multiple data protection schemes for tenants by using different data fragmentation mechanisms. Data written into a FusionStorage object storage cluster is divided into fixed-size (for example, 512 KB) data fragments. Every *N* data fragments are calculated to generate *M* parity fragments. The *N* and *M* fragments form a stripe and are written into the system. As long as the number of lost fragments in a stripe does not exceed *M*, data can be read and written. Lost fragments can be restored from the remaining fragments using a data restoration algorithm. The space utilization of the system is about *N/(N+M)*. *M* determines the data reliability and can be 2, 3, or 4. A larger value of *M* brings a higher reliability.

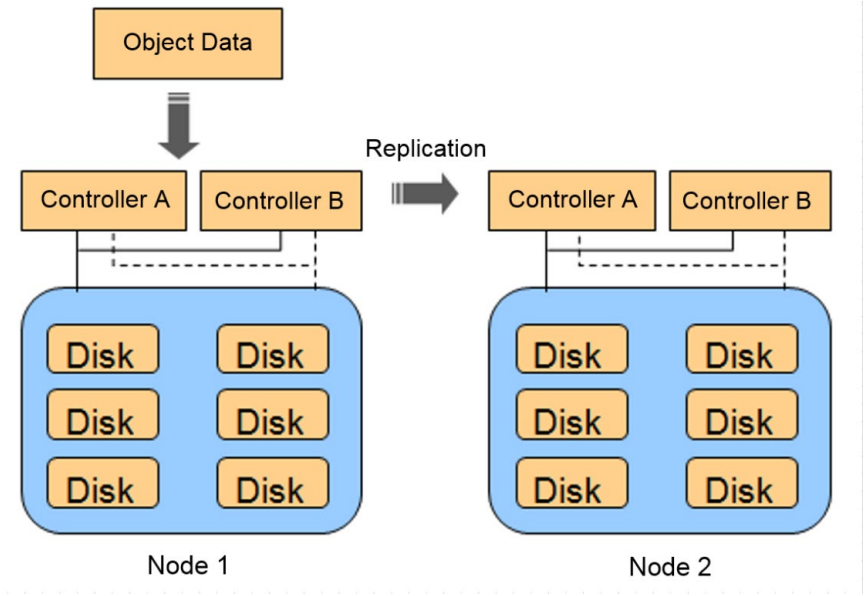
5.1.2 N+M Data Protection

FusionStorage object storage delivers higher reliability and disk utilization than storage systems that use traditional RAID technology.

The traditional RAID technology stores data on different disks in the same RAID group. If a disk fails, RAID reconstruction is implemented to restore data stored on faulty disk. RAID levels commonly used by storage systems are RAID 0, 1, 5, and 6. RAID 6, which offers the highest reliability among all RAID levels, merely tolerates a concurrent failure of two disks at most.

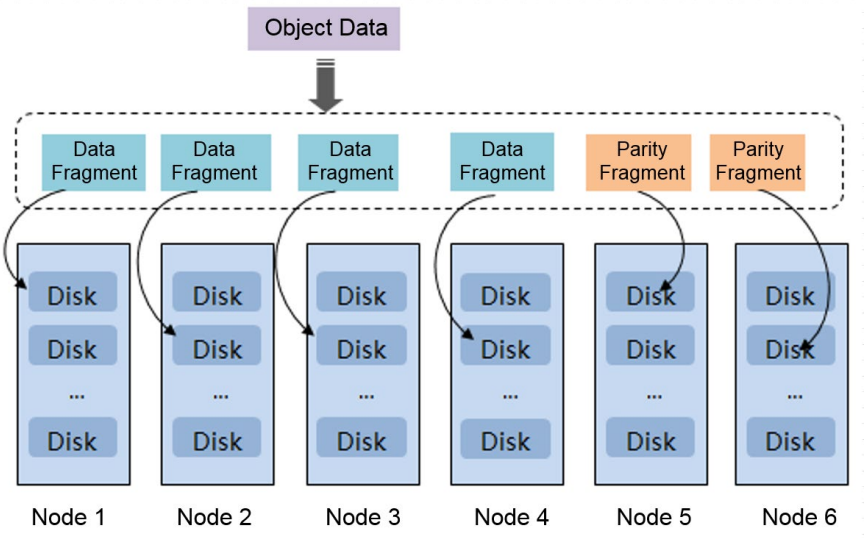
Furthermore, such storage systems use controllers to execute RAID-based data storage. To prevent controller failures, one storage system is typically equipped with two controllers to ensure service availability. However, if both controllers fail, service interruption is still inevitable. Such storage systems can further improve system reliability by implementing inter-node synchronous or asynchronous data replication, but this will decrease disk utilization, driving up TCO. The following figure shows the inter-node replication of such storage systems.

Figure 5-1 Inter-node replication of such storage systems



Unlike traditional RAID, the data protection technology employed by FusionStorage object storage delivers distributed and inter-node redundancy. Data written into FusionStorage object storage is divided into *N* data fragments, and *M* parity fragments are generated for the *N* data fragments using EC. The *N*+*M* fragments are then stored on *N*+*M* nodes. The following figure shows a 4+2 protection scheme where four data fragments and two parity fragments are stored on six nodes.

Figure 5-2 N+M redundancy



Because fragments of each stripe are saved on multiple nodes, FusionStorage object storage can tolerate disk- and node-level failures. The system functions properly as long as the number of concurrently failed nodes does not exceed *M*. Through data reconstruction, the system can restore damaged data to ensure data reliability.

The data protection schemes provided by FusionStorage object storage achieve high reliability similar to that provided by traditional RAID based on data replication among multiple nodes. Furthermore, the data protection schemes of FusionStorage object storage maintain a disk utilization of up to *N/(N+M)*. Unlike traditional RAID that requires hot spare disks to be allocated in advance, FusionStorage object storage allows any available space to serve as hot spare space, further improving storage system utilization.

FusionStorage object storage provides multiple *N+M* data protection schemes. You can flexibly configure data redundancy to obtain your desired reliability levels.

5.1.3 Node- and Cabinet-Level Security

FusionStorage object storage uses distributed architecture. Object data and metadata are distributed on each node after fragmentation and EC. When the number of nodes is greater than or equal to (*N/M*)+1, the system supports node-level security (if *N/M* is not an integer, round it up to the nearest integer).

For example, if *N+M* is 4+2, only three nodes are required to implement node-level security. Each object is divided into six fragments, and each node stores only two fragments. Data can still be read if a node becomes faulty for a period of time. You can use four nodes to maintain read/write performance and reliability of the 4+2 protection scheme if one node becomes permanently faulty.

When all the nodes in a storage pool are located in different cabinets, the system supports cabinet-level security. The formula for calculating the number of required cabinets for cabinet-level security is the same as that for calculating the number of required nodes for node-level security.

Figure 5-1 shows an example of node-level security layout. If any of the nodes becomes faulty, the system will read the four fragments on the other two nodes and restore damaged fragments

using EC. To maintain the write performance of the 4+2 protection scheme when one node is down for a long period of time, configure one more node.

Figure 5-3 Node-level security layout

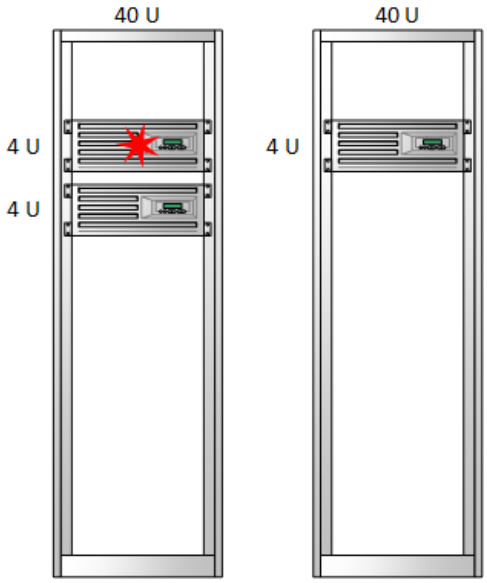
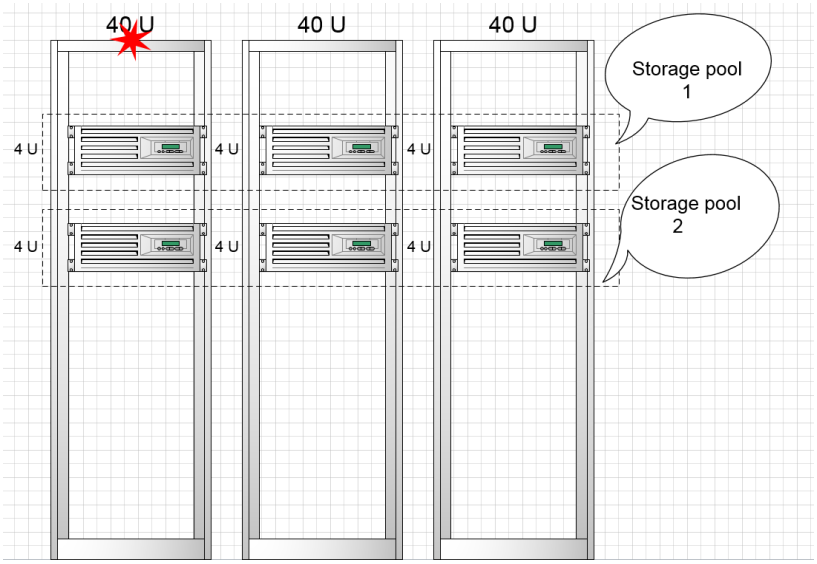


Figure 5-2 shows an example of cabinet-level security layout. To implement cabinet-level security, nodes from multiple cabinets are selected to form a storage pool. Nodes in each storage pool are horizontally located in different cabinets. If one cabinet is faulty, only one node in the storage pool is faulty. Data can still be read according to EC data fragmentation principles. To maintain EC write performance in the event of a cabinet failure, configure one more cabinet for redundancy.

Figure 5-4 Cabinet-level security layout



5.1.4 Cross-Site EC

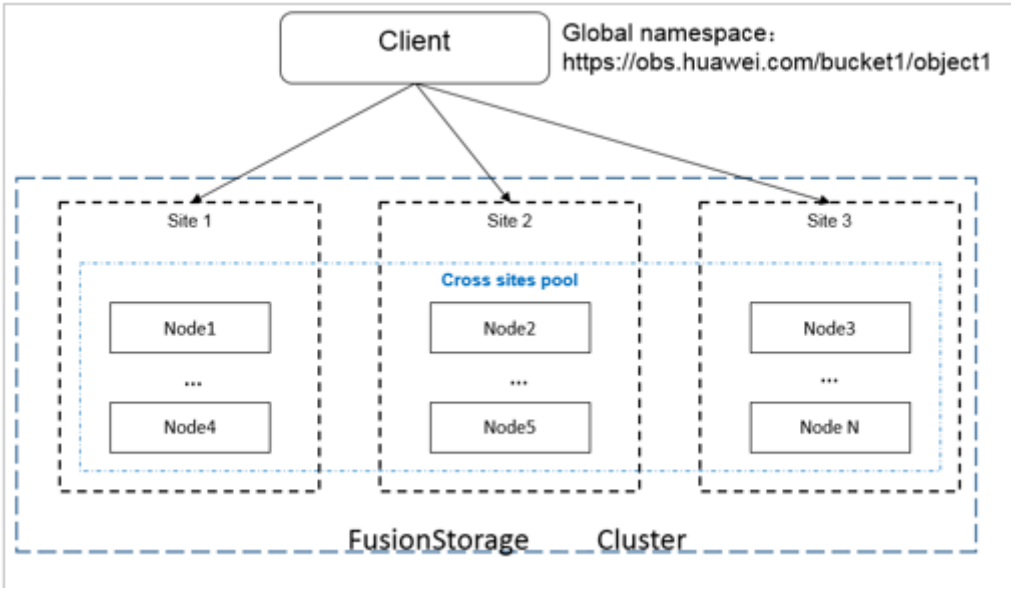
For the purpose of data reliability, data redundancy is required in a site and disaster recovery is required among sites to prevent the fault of a single site.

In general, a site contains one or more physical data centers. It has independent cooling, fire extinguishing, moisture-proof, and electricity facilities. Within a site, computing, network, storage, and other resources are logically divided into multiple clusters. The distance between sites in a region is generally not less than 6 km and not more than 100 km. Sites are connected through high-speed optical fibers and the latency generally ranges from 1 to 5 ms, meeting the requirements of building a cross-site HA system.

FusionStorage object storage uses cross-site Erasure Coding (EC) to provide data redundancy protection among sites. Compared with the traditional two-copy redundancy mode, cross-site EC improves space utilization by more than 12% without changing data durability.

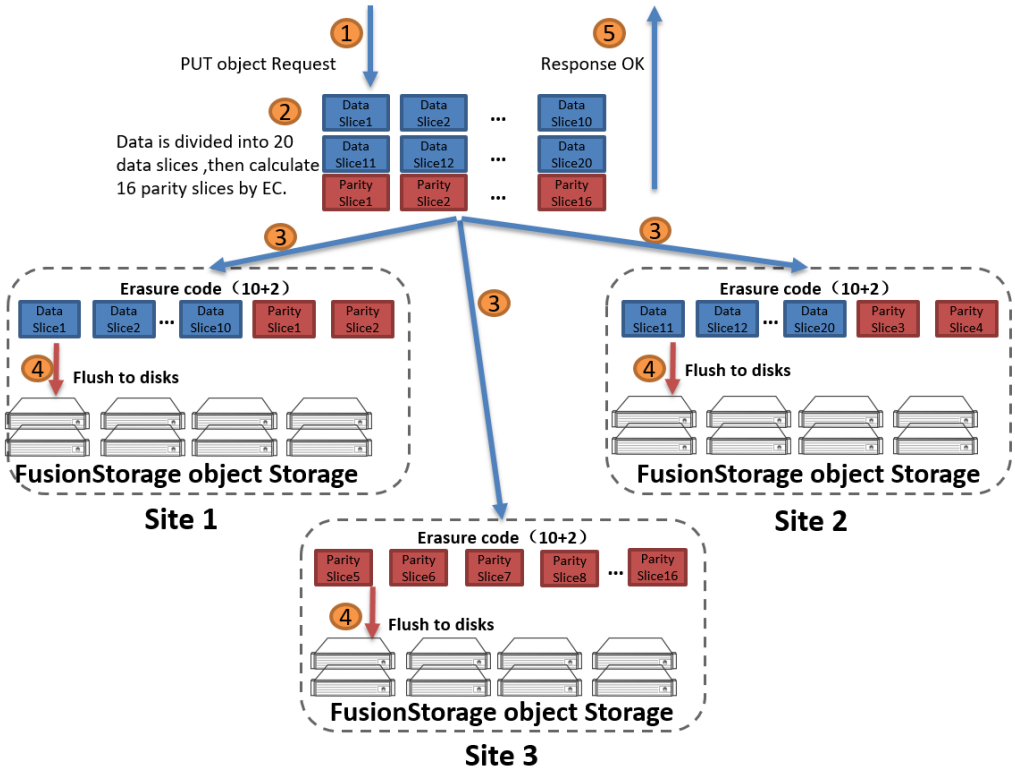
As shown in Figure 5-5, physical devices of a FusionStorage object storage cluster can be deployed across three sites. User object data is evenly distributed in the three sites and externally presented as a complete object storage service cluster with a unified namespace (unified domain name). When a site is completely faulty due to an extreme disaster, FusionStorage object storage ensures service continuity, RPO of 0, and zero data loss, and provides data durability of up to 99.9999999999% (twelve nines).

Figure 5-5 Logical architecture of the three-site FusionStorage object storage cluster



As shown in Figure 5-6, the data utilization of the 20+16 cross-site EC scheme is 55.5% and that of the two-site copy mode (EC configuration of a single site is 10+2) is 10/(12 + 12) = 41.6%. Cross-site EC enables more efficient use of storage space.

Figure 5-6 Data distribution of the three-site FusionStorage object storage cluster



After receiving the PUT Object request, FusionStorage object storage divides original object data into fragments of a fixed length. A certain number of original data fragments (for example, 20 in the preceding figure) are calculated according to the EC algorithm to obtain the number of parity fragments. Original data fragments and parity fragments are evenly distributed on different nodes of three sites based on the algorithm.

With regard to reliability:

- If one site is faulty, two faulty nodes or disks in each of the other two sites can be tolerated. For example, if site 1 is faulty, two nodes or disks in site 2 are faulty, and two nodes or disks in site 3 are faulty, read and write services are not affected.
- Data is synchronous among the three sites and the RPO is 0.

5.2 Data Consistency

FusionStorage object storage provides the following two data consistency check methods:

- **Online data consistency check**

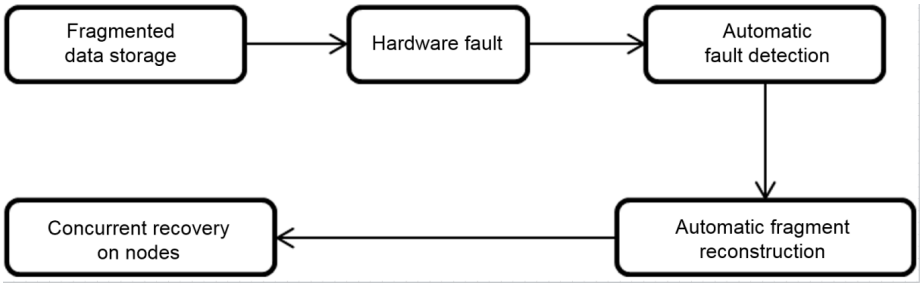
  A client can insert a Content-MD5 header when uploading an object (PUT operation). After receiving the object, FusionStorage object storage calculates the MD5 value of the object, and compares the MD5 value with that in the header. If the two values are identical, data received is consistent with the data transmitted by the client. During a GET operation, FusionStorage object storage inserts the MD5 value of the object into the HTTP response returned to the client. The client can check data consistency using this MD5 value.
- **Background asynchronous data consistency check**

FusionStorage object storage periodically checks all objects in the system as follows: FusionStorage object storage reads data of an object, calculates its MD5 value, and checks whether the MD5 value is identical with that in the metadata recorded when the object was uploaded. If not identical, an alarm is generated. In addition, FusionStorage object storage will dynamically adjust the check speed according to service loads.

5.3 Fast Data Reconstruction

Each disk in FusionStorage object storage stores multiple data fragments, whose parity fragments are scattered on other nodes in the system based on specified distribution policies. When detecting a disk or node failure, FusionStorage object storage automatically starts data recovery in the background. Because parity fragments are distributed on different nodes, data reconstruction will start on these nodes at the same time to recover data. Each node reconstructs only a small amount of data, and multiple nodes reconstruct data concurrently. This eliminates the performance bottleneck caused by the reconstruction of a large amount of data on a single node and minimizes the impact on upper-layer services. The following figure shows the automatic data reconstruction process.

Figure 5-7 Fast data reconstruction



Fast data reconstruction supported by FusionStorage object storage has the following characteristics:

- Data and parity fragments are scattered in the entire resource pool. If a disk fails, its data can be automatically and concurrently reconstructed across the whole resource pool.
- Data is distributed onto different nodes. The failure of a single node does not affect data availability and reconstruction.
- Load balancing can be automatically implemented in the event of a fault or during capacity expansion. Capacity expansion enables applications to obtain larger capacity and better performance without requiring any adjustment. Unlike traditional RAID in which disks are restored one by one, the restoration of each disk in FusionStorage object storage is independent so that disks can be restored concurrently. The restoration speed is up to 2 TB/hour.

5.4 Cluster Reliability

FusionStorage object storage leverages a fully symmetric architecture. From the perspective of the physical structure, same system software is deployed on all nodes. From the perspective of user experience, all nodes are identical and can process user requests.

The service layer provides object storage services, including Amazon S3 services. Load balancers are configured at the front end of FusionStorage object storage clusters. If the node processing an access request is down, the access request will be smoothly switched to another node. When multiple clients access a cluster, the cluster automatically and evenly allocates the access requests to multiple nodes, achieving load balancing.

To ensure service reliability, the system starts monitoring processes on certain nodes. The cluster formed by the monitoring processes is called the Paxos control subsystem. The Paxos control subsystem provides node status monitoring and master selection functions. When new nodes are added or nodes become faulty, the Paxos control subsystem will report events to notify the subsystems or modules that pay attention to cluster status changes. As long as the number of faulty nodes in a FusionStorage object storage cluster does not exceed half of the nodes in the Paxos control subsystem, the cluster can work properly.

The index layer manages object metadata as well as stores and reads object metadata by interacting with the persistence layer.

The OAM management subsystem is responsible for configuring services and monitoring service and device status, and is accessible to clients through browsers. It is deployed on two nodes that also run the storage subsystem. The two nodes work in active/standby mode. In normal cases, the OAM management subsystem runs on one node. If the node fails, the OAM management subsystem switches over to the other node. The switchover is transparent to clients and does not change the IP address of the OAM management subsystem.

Thanks to the distributed architecture, FusionStorage object storage is able to maintain system availability in the event of any node fault (either man-caused or mechanical). Node overload control further helps minimize the impact of node failures on the whole system.

5.5 Hardware Reliability

Nodes of FusionStorage object storage leverage the following designs to ensure high reliability:

- Dedicated hot-swappable SAS system disks are used, supporting RAID 1 protection.
- Power and fan modules are redundant.
- Swappable mainboards and cable-free design are adopted, significantly increasing node reliability while reducing 80% of replacement time.
- Triple anti-vibration designs for disks (including using vibration damping screws on fans, enhancing enclosure rigidity, and adding spring washers and damping pads) reduce the disk failure rate and improve reliability of storage nodes.
- The heat dissipation directions of storage nodes are the same. The end-to-end heat dissipation design improves the heat dissipation efficiency, prolongs the service life of electronic components, and ensures stable running of storage nodes in case air conditioners in equipment rooms are abnormal. The cellular porosity rate of front-end panels is 75%. Counter-rotating fans improve wind speed. Flow-dividing air ducts and independent air channels for back-end I/O modules further improve heat dissipation efficiency.

5.6 Link Reliability

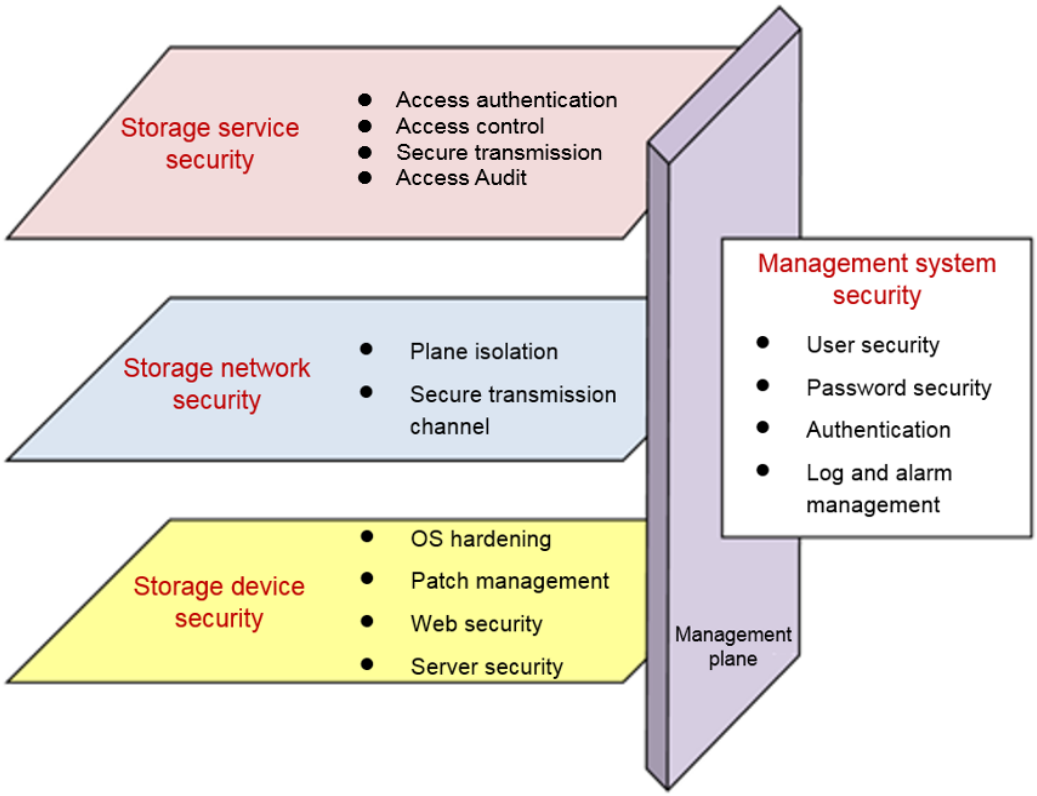
Each node uses two bonded ports to interconnect with two stacked service network switches and uses another two bonded ports to interconnect with two stacked storage network switches.

The failure of one port or switch will not affect node or system availability. In addition, port bonding implements disaster recovery and makes the most of the port bandwidth.

6 System Security

6.1 Security Architecture

Figure 6-1 Security architecture



6.2 Storage Device Security

6.2.1 Server Security

FusionStorage object storage supports Huawei rack servers. The rack servers provide secure management APIs, adopt secure firmware, and guarantee hardware security in the following ways:

- Authentication management

  iBMC (a Huawei proprietary intelligent management system that remotely manages rack servers) supports local, Lightweight Directory Access Protocol (LDAP), and dual-factor authentication modes. The local authentication mode includes user name + password authentication and SSH public key authentication. The dual-factor authentication mode includes USB key certificate authentication. In addition, iBMC also supports secondary authentication for important operations.

- Session management

  Session generation: 31-bit session IDs are generated using secure random numbers. Setting up multiple simultaneous sessions for one user is prohibited.

  Session termination: A session can be terminated in either of the following ways:

  Termination upon timeout: If a CLI, web, or SFTP session is inactive until the timeout period expires, the session is automatically disconnected.

  Manual termination: A user initiates a request to terminate a session. Administrators can terminate sessions of other users.

- Secure protocols

  External access uses SFTP, SSH, HTTPS, SNMPv3, and RMCP+ (IPMI LAN) by default. Transmission channels are encrypted using secure protocols. Insecure protocols HTTP, SNMPv1, SNMPv2c, and RMCP (IPMI LAN) are disabled by default.

- Data protection

  All sensitive data related to passwords and keys on iBMC is encrypted to prevent sensitive information from being disclosed.

  To prevent the content of upgrade packages from being cracked and tampered with, iBMC supports encryption and signature protection for the upgrade packages, ensuring their confidentiality and integrity.

  In addition to encryption protection, iBMC encapsulates the Linux shell. Users cannot directly access files in the file system after logging in through SSH or serial ports. This prevents file damage and information leakage.

  iBMC supports backup of key data files and calculation and storage of file checksums. It also provides a backup and restoration mechanism for file verification failures to prevent data file damage caused by abnormal system power-off and ensure the availability and integrity of data files.

- Access policies

  iBMC ensures secure web access by using login rules. The login rules specify the time, IP addresses, and MAC addresses allowed to access iBMC.

- Key management

iBMC key management uses the two-layer key management structure of root key + working key. A root key is used to encrypt a working key, and the working key encrypts the protected data.

- Security hardening

  System installation is minimized. On iBMC, the embedded Linux system is tailored and only necessary components are installed. Unused components and commands are deleted.

  For security purposes, the Linux shell command line is encapsulated so that only the whitelist commands can be executed.

  Security hardening has been performed on the SSH and Apache servers in the system. Only secure algorithms are used. Insecure protocols and ports are disabled by default.

- Log audit

  iBMC supports log audit. The log information includes user names, user IP addresses, operation time, and operation content. iBMC records SEL logs, operation logs, run logs, and security logs. You can query and audit the logs using APIs provided by iBMC.

  iBMC logs are saved in the flash file system of iBMC in real time. The system notifies you when the size of the logs is about to reach the maximum log storage capacity. When a log file reaches a specified size, the log file is automatically backed up. iBMC uses the principle of least privilege for logs. Unauthorized users cannot view or download log files.

  iBMC supports remote syslog dump. Logs are stored in remote syslog servers to prevent loss caused by log overwriting. The syslog servers can be verified.

6.2.2 Operating System Hardening

FusionStorage object storage uses EulerOS. The following measures are taken to protect the operating system:

- Operating system tailoring

  Unnecessary services and components are deleted or disabled to reduce the size of the operating system. This improves the startup speed and security of the operating system without affecting support for desired services and existing features.

- System service security hardening

  Insecure services, such as Telnet, SNMPv1, SNMPv2c, and FTP, as well as unnecessary or risky background processes and services are disabled. Secure communications and transmission protocols are adopted. For example, SSH v2 is used instead of Telnet.

- Kernel security hardening

  Execution stacks are protected against buffer overflow attacks. Functions, such as IP address forwarding, response to broadcast requests, and Internet Control Message Protocol (ICMP) redirects receiving, are disabled. TCP-SYN cookie protection is enabled to prevent SYN attacks (DoS attacks).

- Account and password protection

  Unnecessary users and user groups have been deleted. The password complexity check function has been enabled, and password validity periods and the number of login attempts have been configured.

- File and directory permission control

  Permissions on files and directories are minimized in accordance with security hardening specifications and application requirements in the industry.

- Logging and audit

Run logs of services and kernel processes are recorded and can be sent to log servers for audit.

6.2.3 Security Patch Management

Software design defects result in system vulnerabilities. System security patches must be installed periodically to fix these vulnerabilities and protect the system against attacks by viruses, worms, and hackers. Huawei adopts the following security patch management measures to enhance system security:

- Periodically provides on-demand security patches for users as operating system security patches and open-source software security patches are released.
- Releases patches to fix security vulnerabilities in FusionStorage object storage based on vulnerability severities.

6.2.4 Web Security

DeviceManager, the management GUI of FusionStorage object storage, provides the following web security enhancements:

- Secure access over HTTPS

  DeviceManager only supports secure access channels over HTTPS, enhancing access security.
- Prevention against XSS attacks

  XSS attacks occur when attackers use a vulnerable web application to send malicious code to users.
- Prevention against SQL injection

  SQL injection is a code injection technique. Malicious SQL statements are inserted into an entry field of a web form or into a query string of a page request for execution.
- Prevention against CSRF

  If a user logs in to website A and then to website B (containing attack programs) before the session on website A expires, an attacker can obtain the session ID of website A and log in to website A to intercept critical information of the user.
- Protection for sensitive information

  DeviceManager hides sensitive information to prevent interception by attackers.
- Restriction on file upload and download

  DeviceManager restricts file upload and download to prevent mission-critical files from being disclosed and insecure files from being uploaded.
- Prevention against unauthorized uniform resource locator (URL) access

  Each user type is granted specific permissions and users cannot access data beyond their permissions.

6.3 Storage Network Security

6.3.1 Plane Isolation

Depending on whether front and back ends share Top of Rack (TOR) switches, FusionStorage object storage supports two typical networking modes: networking mode in which front and

back ends use separate switches, and networking mode in which front and back ends share switches.

Figure 6-2 shows the networking mode in which front and back ends use separate switches.

Figure 6-2 Networking mode in which front and back ends use separate switches

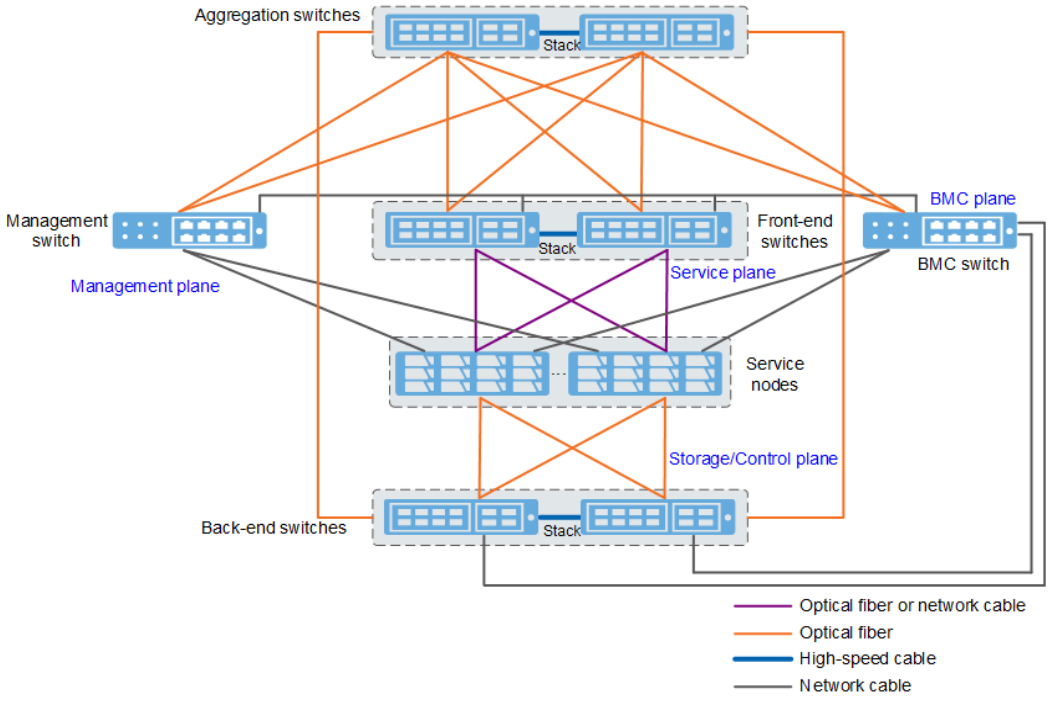
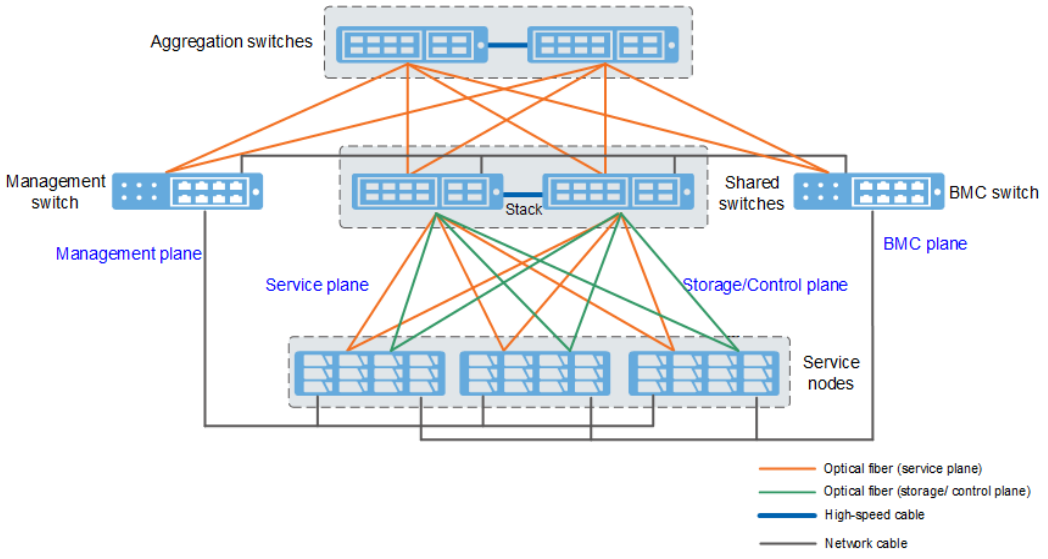


Figure 6-3 shows the networking mode in which front and back ends share switches.

Figure 6-3 Networking mode in which front and back ends share switches



The network planes of FusionStorage object storage can be classified into the following types based on the service type:

- Service plane: communicates with the user service network and provides S3 APIs externally.
- Storage plane: enables communications among nodes of FusionStorage object storage.
- Management plane: communicates with the user management network for management and maintenance of FusionStorage object storage.
- BMC plane: connects to Mgmt ports of FusionStorage object storage nodes to enable remote device management.
- Control plane: maintains internal cluster information of FusionStorage object storage.

6.3.2 Secure Transmission Channel

FusionStorage object storage uses multiple secure transmission protocols, such as SSH, SFTP, and HTTPS, for remote system management. These transmission protocols use secure encryption algorithms for data protection. In addition, the system disables insecure protocols such as Telnet, FTP, and HTTP.

Table 6-1 Secure encryption algorithms

Application	Protocol	Port	Algorithm
Object storage service	TLS 1.2 (enabled by default), TLS 1.1, and TLS 1.0	5443 or 443	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 TLS_ECDHE_RSA_WITH_AES_128_

Application	Protocol	Port	Algorithm
			CBC_SHA TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA TLS_RSA_WITH_AES_128_GCM_SHA256 TLS_RSA_WITH_AES_256_GCM_SHA384 TLS_RSA_WITH_AES_128_CBC_SHA TLS_RSA_WITH_AES_256_CBC_SHA TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 TLS_RSA_WITH_AES_128_CBC_SHA256 TLS_RSA_WITH_AES_256_CBC_SHA256
Account management service	TLS 1.2 (enabled by default), TLS 1.1, and TLS 1.0	9443	TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 TLS_DHE_DSS_WITH_AES_128_CBC_SHA256 TLS_DHE_DSS_WITH_AES_256_GCM_SHA384 TLS_DHE_DSS_WITH_AES_256_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_AES_25

Application	Protocol	Port	Algorithm
			6_GCM_SHA384 TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
DeviceManager (system management GUI)	TLS 1.2	8088	ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES256-SHA384 ECDHE-RSA-AES128-GCM-SHA256
DeployManager (GUI for deployment, upgrade, and capacity expansion)	TLS 1.2	6098	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
Login to the CLI of a server using SSH	SSH V2	22	Ciphers: aes128-ctr, aes192-ctr, aes256-ctr MACs: hmac-sha2-256, hmac-sha2-512 KexAlgorithms: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521, diffie-hellman-group-exchange-sha256, diffie-hellman-group-exchange-sha1, diffie-hellman-group14-sha1

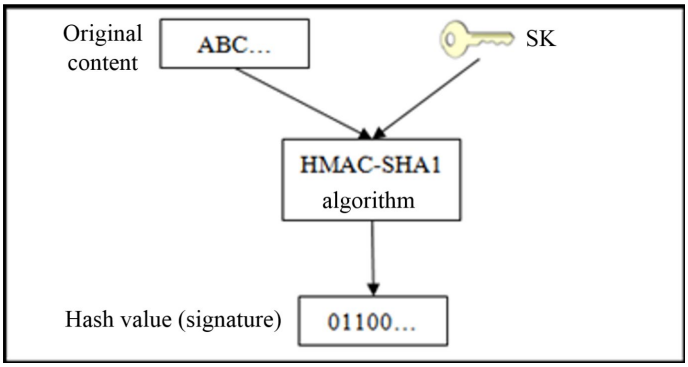
6.4 Storage Service Security

6.4.1 Access Authentication

FusionStorage object storage uses Access Key IDs (AKs) and Secret Access Keys (SKs) to authenticate user identities. During the authentication, keyed-hash message authentication code (HMAC) calculation is performed. The HMAC calculation uses hash algorithms to generate a message digest after inputting an SK and a message.

Each client user has an AK and an SK. The AK is used to uniquely identify the user, and the SK is used to calculate the signature. The SK is encrypted before being stored. Users must properly keep their SKs and prevent SK disclosure. An operation request sent by a client contains a user's AK and a signature calculated based on the user's SK (HMAC uses HMAC-SHA1 or HMAC-SHA256 for signature calculation). After receiving the request, FusionStorage object storage searches for the SK of the received AK in the system and uses the SK to calculate the signature. Then FusionStorage object storage compares the calculated signature with that in the user's request. If the two signatures are consistent, the authentication is successful. Figure 6-4 shows signature calculation using an SK.

Figure 6-4 Signature calculation using an SK



6.4.2 Object and Bucket Access Control

FusionStorage object storage provides flexible and secure access control for data stored in the cloud. You can set access control policies for buckets and objects as required. The access permissions include READ and WRITE. You can also grant other users permissions to access and set control policies for your buckets and objects. Such permissions include READ_ACP (reading current access control policies of buckets or objects), WRITE_ACP (setting access control policies of buckets or objects), FULL_CONTROL (having full control permissions, including READ, WRITE, READ_ACP, and WRITE_ACP).

6.4.3 Secure Data Transmission

FusionStorage object storage provides APIs compatible with Amazon S3. You can upload and download data securely by using terminal tools provided by Huawei or a third party. During data transmission, TLS 1.2 is used by default. TLS 1.1 and TLS 1.0 are also supported. Data security is ensured during transmission among regions of FusionStorage object storage and between FusionStorage object storage and external networks.

6.4.4 Object Access Audit

FusionStorage object storage records all non-query user activities and background operation instructions in logs. The logs can be used for auditing and operation tracing.

6.5 Management System Security

6.5.1 User Security

To prevent mis-operations from compromising storage system stability and service data security, you can define user roles to control user permissions. You can specify user permissions when creating a user and modify the permissions after creating the user. After a specified period of idle time, your DeviceManager session automatically times out. In this case, you need to log in again if you still want to access DeviceManager. The session timeout period is modifiable.

Table 6-2 User roles

Role	Permissions
Super administrator	Has full control over the storage system and can create users with different roles. The default super administrator is **admin**, whose default password is **Admin@storage**.
Administrator	Has the permission to configure the Call Home service and to view the Call Home service, users, user security policies, and alarms. **NOTE** Call Home is used to quickly discover faults and rectify the faults in a timely manner to ensure the normal running of the storage system. In this way, alarms and logs of storage devices can be sent back to the technical support center.
System viewer	Has the permission to view users and alarms.
Security administrator	Has the permission to view users, configure system security, and manage security rules, security policies, certificates, and Key Management CBB (KMC).

6.5.2 Password Security

FusionStorage object storage supports password complexity policies. A user password must contain special characters and at least two of the following character types: uppercase letters, lowercase letters, and digits. FusionStorage object storage supports the user login locking mechanism. The locking mechanism is configurable to prevent brute force cracking. Passwords are encrypted using secure encryption algorithms before being stored or transmitted. A password can be changed only after user authentication has been completed. Except for super administrators, users can change their own passwords only.

Table 6-3 Password security policies

Parameter	Description	Value
Min. Length	Minimum length of a password, preventing you from setting overly short passwords.	[Value range] Integer from 8 to 32 [Default value] 8
Max. Length	Maximum length of a password, preventing you from setting overly long passwords.	[Value range] Integer from 8 to 32 [Default value] 16
Complexity	Complexity of a password, preventing you from setting overly simple passwords.	[Value range] **A password must contain special characters and at least two of the following types: uppercase letters, lowercase letters, and digits** or **A password must contain special characters, uppercase letters, lowercase letters, and digits** [Default value] **A password must contain special characters and at least two of the following types: uppercase letters, lowercase letters, and digits** Special characters include !"#$%&'()*+,-./:;<=>?@[\]^`{_|}~ and spaces.
Number of Duplicate Characters	Maximum number of times a character can appear consecutively in a password. Value **0** indicates no limit.	[Value range] Integer from 0 to 9 [Default value] 3
Number of Retained Historical Passwords	Number of historical passwords retained for a user. A new password must be different from any of the historical passwords. Value **0** indicates no limit.	[Value range] Integer from 0 to 30 [Default value] 3
Password Validity Period (Days)	Password validity period. This parameter is mandatory when **Password Validity** is enabled.	[Value range] Integer from 1 to 999 [Default value]

Parameter	Description	Value
	You are advised to enable **Password Validity**. After the validity period of a password expires, the system prompts you to change the password.	90
Password Expiration Warning Period (Days)	Number of days prior to password expiration that a warning about password expiration is displayed.	[Value range] Integer from 1 to 99 [Default value] 7
Password Change Interval (Minutes)	Minimum interval required for changing a new password after the password is set.	[Value range] Integer from 1 to 9999 [Default value] 5

6.5.3 Authentication

The system supports local authentication and provides the automatic session timeout function for all authenticated users.

6.5.4 Log and Alarm Management

- Log management

  All operations performed on management interfaces (such as DeviceManager) are recorded in logs. Logs detail event generation time, user IDs (including associated terminals, ports, network addresses, or communication devices), event types, names of accessed resources, and event results. Logs can be queried. When the log storage space is used up, the system automatically dumps and deletes logs. Log time is synchronized with a unified time source.

- Alarm management

  System exceptions and faults are displayed on DeviceManager in real time, reminding users to handle them. DeviceManager also supports alarm notification by email.

7 Openness and Compatibility

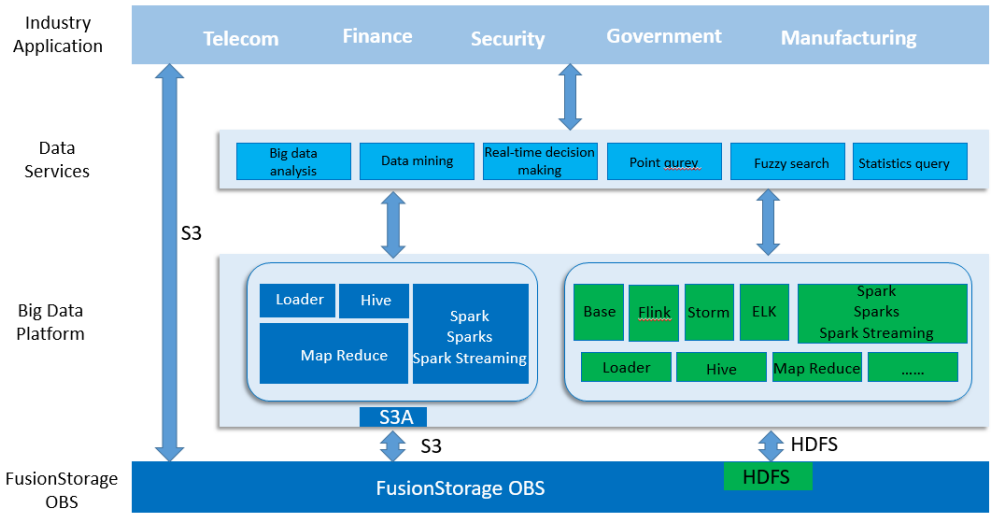
7.1 Mainstream Protocols

FusionStorage object storage is equipped with mainstream protocols and standards to provide customers with industry-leading object storage services. The object APIs are compatible with Amazon S3.

7.2 Big Data Platform

FusionStorage object storage uses open-source S3A or native HDFS（Hadoop Distributed File System）to provide HDFS semantic APIs for mainstream big data platforms in the industry. S3A simulates file system semantics on the object storage, but may require extra data copying and result in performance loss. Unlike S3A, the native HDFS of FusionStorage object storage supports complete directory tree semantics such as rename and move operations, making it a better fit with upper-layer application ecosystems of Hadoop and Spark.

Figure 7-1 Fit with Big Data Platform



The native HDFS of FusionStorage object storage supports complete HDFS semantics and provides the functions that are not supported by S3A. The following table compares the main functions of the native HDFS and S3A.

Table 7-1 Main Functions Compare

Description	The native HDFS of FusionStorage object storage	S3A	Disadvantage of S3A
Rename	Renames a file or a directory.	Copy + Delete	Copying requires extra cache space and results in large system overhead and low performance. Data consistency cannot be guaranteed.
Move	Moves a file or a directory to another parent directory.	Copy + Delete	
List Directory	Lists all contents in a directory.	Scans and filters all objects whose name prefixes contain the directory name.	Sub-directories and files are also scanned and filtered, resulting in low performance.
Delete Directory	Deletes a directory and all contents in the directory.	Scans all objects whose name prefixes contain the directory name and deletes them one by one or in a batch.	Deletion takes a long time, and data consistency cannot be guaranteed.

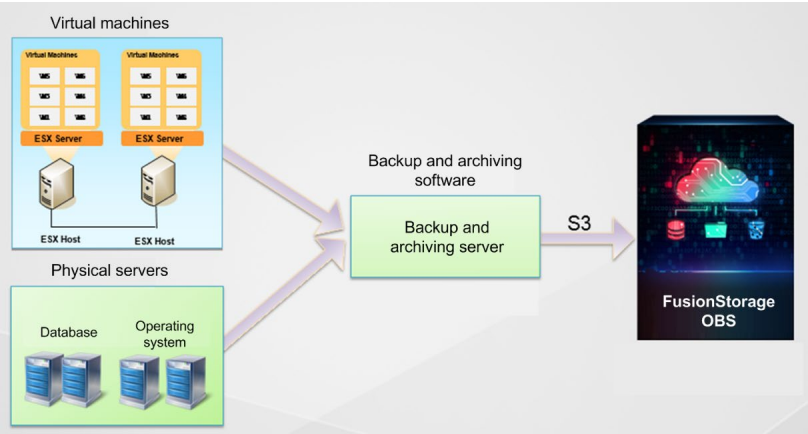
Description	The native HDFS of FusionStorage object storage	S3A	Disadvantage of S3A
Append	Appends data to a closed file.	Not supported	The application layer can simulate this operation by creating new objects. However, too many small objects may be generated.
Flush	Flushes data to caches.	Not supported	Data security cannot be ensured before objects are written to disks. Data cannot be recovered to a specific point in time.
Sync	Flushes data to disks.	Not supported	
Truncate	Truncates files from a certain location.	Not supported	
ACK	Implements error tolerance if a fault, such as network timeout, occurs during a write operation.	Not supported	If a network error occurs when a large file is being written, data written in the file cannot be restored using HTTP.

7.3 Backup and Archiving Software Platform

FusionStorage object storage provides backup storage space for backup and archiving software. The backup and archiving software backs up and archives data from files and applications such as VMs, databases, and operating systems.

Currently, mainstream backup and archiving software supported by FusionStorage object storage include Veritas NetBackup and Commvault Simpana. For more details about supported backup and archiving software, access Huawei Storage Interoperability Navigator.

Figure 7-2 Backup and Archiving

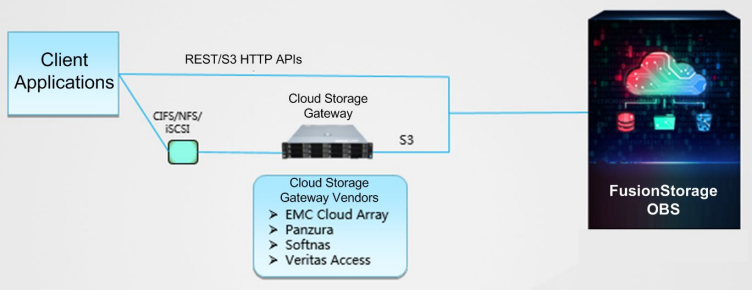


7.4 Mainstream Cloud Storage Gateway

A cloud storage gateway enables client applications to seamlessly use cloud storage space provided by FusionStorage object storage. A cloud storage gateway can be used for backup, archiving, disaster recovery, big data processing, storage tiering, and migration. Applications can use standard storage protocols such as NFS, CIFS, and iSCSI to connect to a cloud storage gateway. Cloud storage gateways use Amazon S3 APIs to connect to FusionStorage object storage, and FusionStorage object storage provides data storage services for files and volumes.

Currently, the cloud storage gateways supported by FusionStorage object storage include EMC Cloud Array, Panzura, SoftNAS, and Veritas Access. For more details about supported cloud storage gateways, access Huawei Storage Interoperability Navigator.

Figure 7-3 Mainstream Cloud Storage Gateway



7.5 Centralized Management Platform

IT O&M management platforms play a crucial role in data centers. They manage IT data centers in a unified manner and enable convenient device status management, monitoring, and configuration.

IT O&M management platform software generally accesses IT infrastructure using the SNMP protocol. FusionStorage object storage complies with SNMP and REST protocols and provides open APIs to support different features.

A Acronyms and Abbreviations

AZ	availability zone
CLI	command-line interface
DHT	distributed hash table
DNS	Domain Name System
EC	erasure coding
GUI	graphical user interface
HTTP	Hypertext Transport Protocol
IPMI	Intelligent Platform Management Interface
LVS	Linux Virtual Server
RAID	redundant array of independent disks
SAS	serial attached SCSI
SSD	solid state disk
TCP	Transmission Control Protocol
OBS	Object Storage Service

