FusionStorage 8.0 Technical White Paper

Issue	**02**
Date	**2020-03-01**

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

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://www.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. Besides 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 carriers. 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
- Huge system size and complex management, exerting high pressure on operation and maintenance (O&M) personnel
- 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

New challenges will inevitably raise new requirements. 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.
- Enriched access methods are provided for diverse types of data in data centers, such as structured and unstructured data.
- 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, Huawei FusionStorage may be the answer. As a data center-class product, Huawei FusionStorage provides intelligent distributed storage and supports elastic large-scale expansion. It leverages unique elastic erasure coding (EC) and load-sensitive dynamic deduplication and compression technologies to provide more available storage space while ensuring service performance. In addition, FusionStorage provides the ever-new storage capability to support smooth software upgrades and coexistence of hardware in different generations, ensuring non-disruptive services during software and hardware updates.

By using system software, FusionStorage organizes local storage resources of general-purpose hardware to build fully distributed storage pools and provide distributed block, object, big data, and file storage services for upper-layer applications. Each storage service supports a variety of service functions and value-added features. You can purchase and deploy one or more storage services to accommodate flexible and efficient data access requirements of ever-changing services.

2 Highlights

FusionStorage provides intelligent distributed storage with superb efficiency, high availability (HA), and outstanding scale-out. It delivers high performance, large capacity, and robust scalability for complex workloads in the cloud and Artificial Intelligence (AI) era and is adopted in a wide variety of scenarios, including cloud resource pools, critical workload databases, big data analysis, content storage, data backup, and data archiving of financial institutions, carriers, governments, and public utilities.

2.1 Distributed Storage for On-Demand Use

Using distributed technologies, FusionStorage organizes storage media, such as hard disk drives (HDDs) and solid-state drives (SSDs), into large-scale storage pools of different types and provide industry standard interfaces for upper-layer applications and clients. This eliminates resource islands and uneven hardware resource utilization encountered by siloed storage systems running in traditional data centers.

- Block storage: provides standard access interface protocols such as SCSI and iSCSI, supports a wide range of virtualization platforms and database applications, and delivers superb performance and high scalability to meet SAN storage requirements of virtualization, cloud resource pools, and databases.
- Object storage: provides application programming interfaces (APIs) compatible with Amazon S3 and supports mainstream cloud computing ecosystems to meet requirements of content storage, cloud backup, cloud archiving, and public cloud storage services.
- Big data storage: supports the Hadoop Distributed File System (HDFS) and provides cloud-enabled compute and storage separation solution for big data analysis scenarios, enabling you to efficiently process massive amounts of data, deploy and use resources on demand, and reduce the TCO.
- File storage: provides APIs that comply with NFS, CIFS, and FTP and delivers shared storage resources for unstructured data through ultra-large distributed file systems. File storage features outstanding performance and supports large-scale horizontal expansion,

ideal for video and audio storage, high-performance computing (HPC), and video surveillance.

This document focuses on block storage provided by FusionStorage.

2.2 Resilient and Efficient Storage for Critical Workloads

FusionStorage forms large-scale storage pools using distributed technologies and provides industry standard interfaces for upper-layer applications and clients. This enables on-demand storage resources while removing bottlenecks such as unbalanced hardware resource utilization encountered by siloed storage systems running in traditional data centers. FusionStorage can start small and scale out to thousands of nodes, enabling linear performance growth as capacity expands. To better support the cloud migration of critical workloads, FusionStorage uses unique FlashLink® performance acceleration technology, intelligent stripe aggregation, I/O priority scheduling, cache algorithms, data identification and processing, and NVMe SSD caches to deliver 1 millisecond stable latency, even when data reduction is enabled. FusionStorage satisfies your I/O-intensive, latency-sensitive, and capacity-hungry needs and is ready to supercharge your business of today as well as that of the future.

2.3 Comprehensive Enterprise-Grade Features, Powering HA Data Centers

FusionStorage provides highly available storage services for enterprise applications. At the I/O, system, and data center levels, it uses cutting-edge technologies like end-to-end Data Integrity Fields (DIFs), multi-type data redundancy protection modes, comprehensive system sub-health check and self-healing, distributed active-active storage, and asynchronous replication.

It supports multi-copy and dynamic EC data redundancy protection modes. Asingle FusionStorage system tolerates the simultaneous failure of up to four nodes or four cabinets. That is, system reliability remains uncompromised even if nodes are faulty. EC technology improves disk space utilization three-fold compared to the traditional three-copy mode, reducing hardware investments while offering a variety of EC schemes for flexible selection and on-demand deployment. FusionStorage leverages dynamic deduplication and compression on SSDs or HDDs used as main storage to save storage space. Based on front-end application loads, FusionStorage automatically chooses between inline deduplication and post-process deduplication to ensure a high data reduction ratio and provide stable storage performance.

FusionStorage provides the industry's only cross-cluster, gateway-free distributed active-active feature. This allows you to build active-active systems with zero RPO and close-to-zero RTO in Oracle RAC or VMware virtualization scenarios, and obtain six-nines solution-level availability to ensure always-on services. Moreover, by supporting asynchronous replication with an RPO of seconds, you can effortlessly build disaster recovery (DR) solutions of different protection levels.

2.4 Extensive Compatibility, Ideal for Next-Gen Cloud Infrastructure

FusionStorage is compatible with diverse software and hardware platforms. FusionStorage block storage with an open architecture works seamlessly with a variety of containers and computing virtualization platforms to provide a data storage layer to private, public, and hybrid cloud data centers with scale-out capabilities on demand. This allows you to effortlessly build an open cloud platform without worrying over vendor lock-in when selecting infrastructure. With abundant computing power provided by Huawei Kunpeng series processors, FusionStorage offloads some storage functions to the chip layer to accelerate software performance.

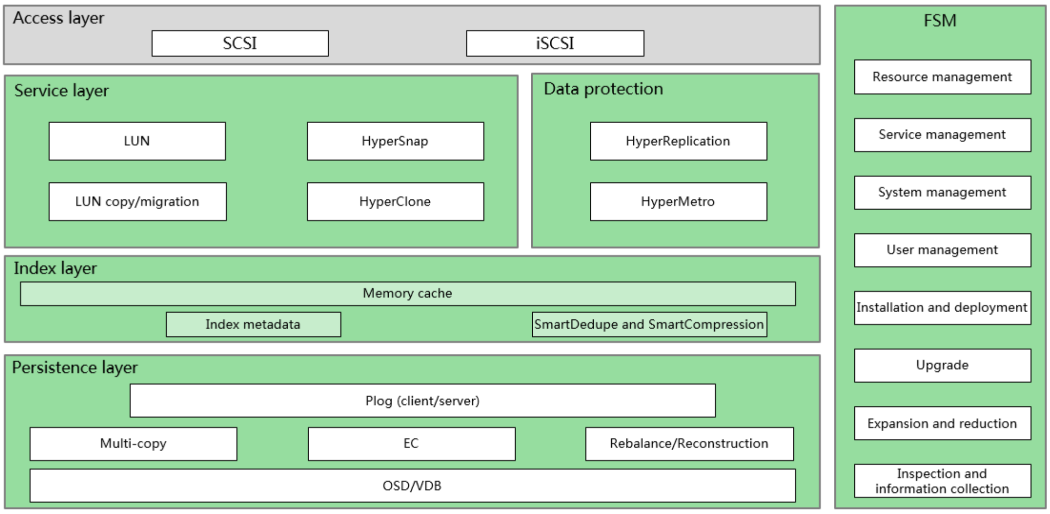
2.5 Intelligent Data Services and System O&M

FusionStorage provides a unified system management platform, which comprises the Data Service Subsystem (DSS) and Operations and Maintenance Subsystem (OMS). Its intelligent risk prediction shields storage resource service risks in advance, helping accurately implement capacity expansion, procurement, and service change decision-making. FusionStorage provides a comprehensive system sub-health check and processing mechanism to implement intelligent fault locating and one-click automatic service recovery.

3 Architecture

3.1 Software Architecture

Figure 3-1 FusionStorage software architecture



FusionStorage is a scale-out product that supports iSCSI and SCSI protocols for external systems. It leverages the Plog mechanism, an Append Only redirect-on-write (ROW) write mechanism, to store data. By using log caching and full-stripe writing, FusionStorage aggregates random I/Os into sequential I/Os and writes them onto back-end storage media. FusionStorage supports global foreground and background self-adaptive deduplication and compression. Data redundancy is ensured using the multi-copy or EC mechanism. To help build a complete data protection solution, FusionStorage supports enterprise storage features, including LUNs, HyperSnap (snapshot), HyperClone (clone), HyperReplication (asynchronous replication), and HyperMetro (active-active storage). The FusionStorage

Manager (FSM) module enables you to configure, manage, and monitor system resources, perform online upgrades, and expand capacity.

Table 3-1 describes FusionStorage subsystems.

Table 3-1 FusionStorage subsystems

Subsystem	Type		Function
Service subsystem	Access layer		Provides standard access interfaces for applications to access the storage system and supports SCSI and iSCSI protocols.
	Service layer		Implements enterprise-class volume features, including HyperSnap, HyperClone, HyperReplication, and HyperMetro.
	Index layer		Converts logical and physical data space and implements SmartDedupe (deduplication) and SmartCompression (compression).
	Persistence layer		Stores data including data generated by multi-copy, EC, data balancing, and reconstruction by using the Plog mechanism, manages disks, and reads from and writes to disks through Object Storage Device (OSD) and Virtual Data Block (VDB) components.
Management subsystem	FSM	Resource management	Manages and allocates storage pools and provides data redundancy protection using multi-copy and EC.
		Service management	Provisions block storage services by storage pool.
		System management	Initializes the system, configures service functions, manages device topology, and provides device topology diagrams to present and manage device topological relationships.
		User management	Adds, deletes, modifies, and queries users, including user levels and permissions.
		Installation and deployment	Initially installs and deploys the system.
		Upgrade	Upgrades the system, including software, operating systems, and firmware.
		Expansion and reduction	Expands or reduces system capacity in an online manner.
		Inspection and information collection	Manages device information and collects detailed device configurations and running status information, allowing you to check device configurations and health status.

3.2 Hardware Architecture

Designed based on general-purpose hardware, FusionStorage can run on Huawei and other mainstream servers in the industry. For details about the supported servers, see the compatibility list. When Huawei servers are used, integrated software and hardware solutions can be provided to enhance reliability, performance, and serviceability.

To maximum system reliability and performance, you are advised to adopt the Huawei hardware models listed in Table 3-2 (for detailed hardware configurations, consult your Huawei sales representative).

Table 3-2 Recommended FusionStorage hardware models

Hardware	Recommended Model
TaiShan nodes	Huawei TaiShan 200(Model 5280) series
	Huawei TaiShan 200(Model 2280) series
	Huawei TaiShan 5280 series
	Huawei TaiShan 2280 series
x86 nodes	Huawei FusionServer 5288 V5 series
	Huawei FusionServer 2288H V5 series

General-purpose servers are comprised of CPUs, system disks, caches, main storage (the number of disks can be determined flexibly, with a specified minimum number), RAID controller cards, memory, network interface cards (NICs), and power modules. For third-party servers and components, compatibility tests are usually performed. Servers and components that pass the compatibility tests are recorded in the compatibility list for reference.

Using integrated software and hardware solutions is recommended because they are comprehensively tested and verified in various abnormal scenarios and thus provide higher reliability.

3.3 Network Architecture

FusionStorage supports Ethernet, InfiniBand (IB), and Remote Direct Memory Access over Converged Ethernet (RoCE) networking modes. All the three networking modes support physical isolation for the management network, front-end storage network, back-end storage network, and service network. The management network transmits only management messages, such as service configuration and log collection messages. The front-end storage network transmits host I/O data and control messages and determines the storage performance. The back-end storage network transmits background I/Os, such as data reconstruction, deduplication, and cache flushing I/Os. The service network transmits front-end service I/Os such as iSCSI I/Os.

Upon a fault, background I/Os, such as reconstruction I/Os, will occupy bandwidth. To prevent impacts on host I/Os, it is recommended that the front- and back-end storage networks be physically separated. If such impacts are not considered, the back-end storage network can be converged with the front-end storage network. In this case, host I/Os and background I/Os both are transmitted over the front-end storage network.

3.3.1 Ethernet Networking

3.3.1.1 Deployment Schemes

FusionStorage connects to data center networks through top of rack (TOR) access switches. It supports both independent and converged deployment of compute and storage nodes. Independent deployment means that compute nodes (running applications) and storage nodes are deployed on different servers. Converged deployment means that compute nodes (running applications) and storage nodes are deployed on the same servers. The switch models listed in Table 3-3 are recommended for Ethernet networking. This document uses 10GE access switches as an example.

Table 3-3 Recommended switch models in Ethernet networking

Access Switch		Aggregation Switch	
Type	**Recommended Model**	**Type**	**Recommended Model**
10GE	CE6850 and CE7850 series	40GE (fixed switch)	CE7850 series
10GE	CE6850 and CE7850 series	40GE (modular switch)	CE12800 series
25GE	CE6860 and CE8800 series	100GE (fixed switch)	CE8800 series
25GE	CE6860 and CE8800 series	100GE (modular switch)	CE12800 series

3.3.1.2 Independent Deployment of Compute and Storage Nodes

FusionStorage supports independent deployment of compute and storage nodes. Cabinet-level security is recommended, which requires at least three cabinets. Independent deployment supports two networking schemes, depending on whether front- and back-end storage networks are converged. Figure 3-2 and Figure 3-3 show the recommended networking architecture.

Figure 3-2 Independent deployment, where front- and back-end storage networks are converged

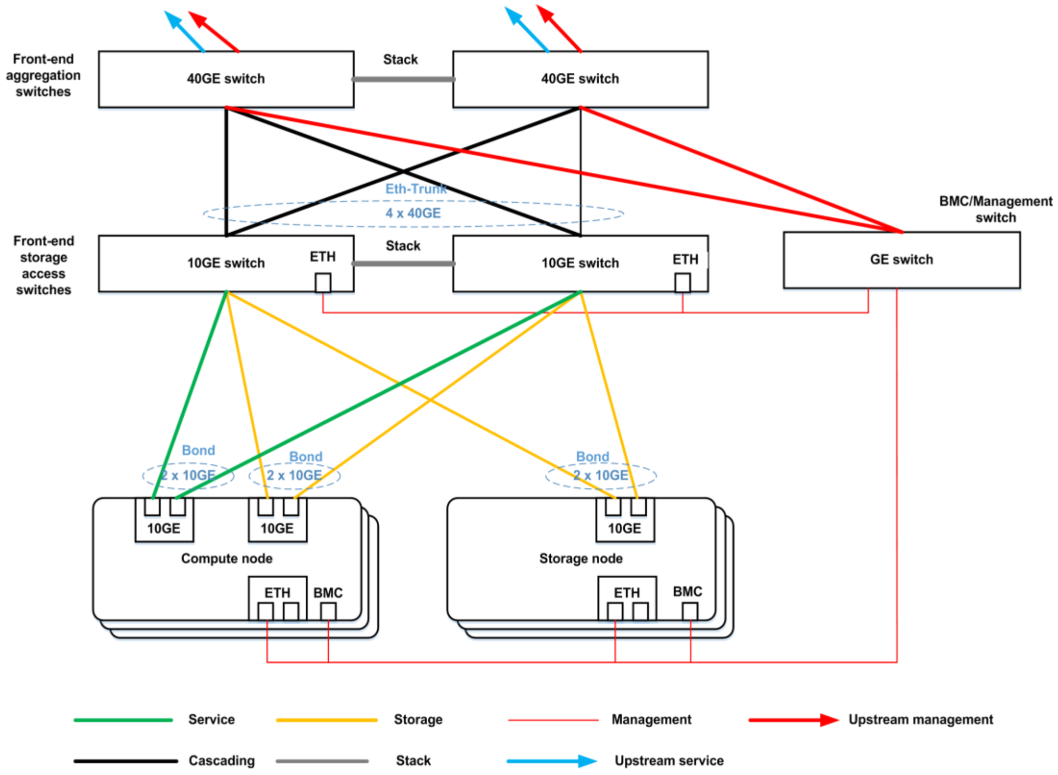
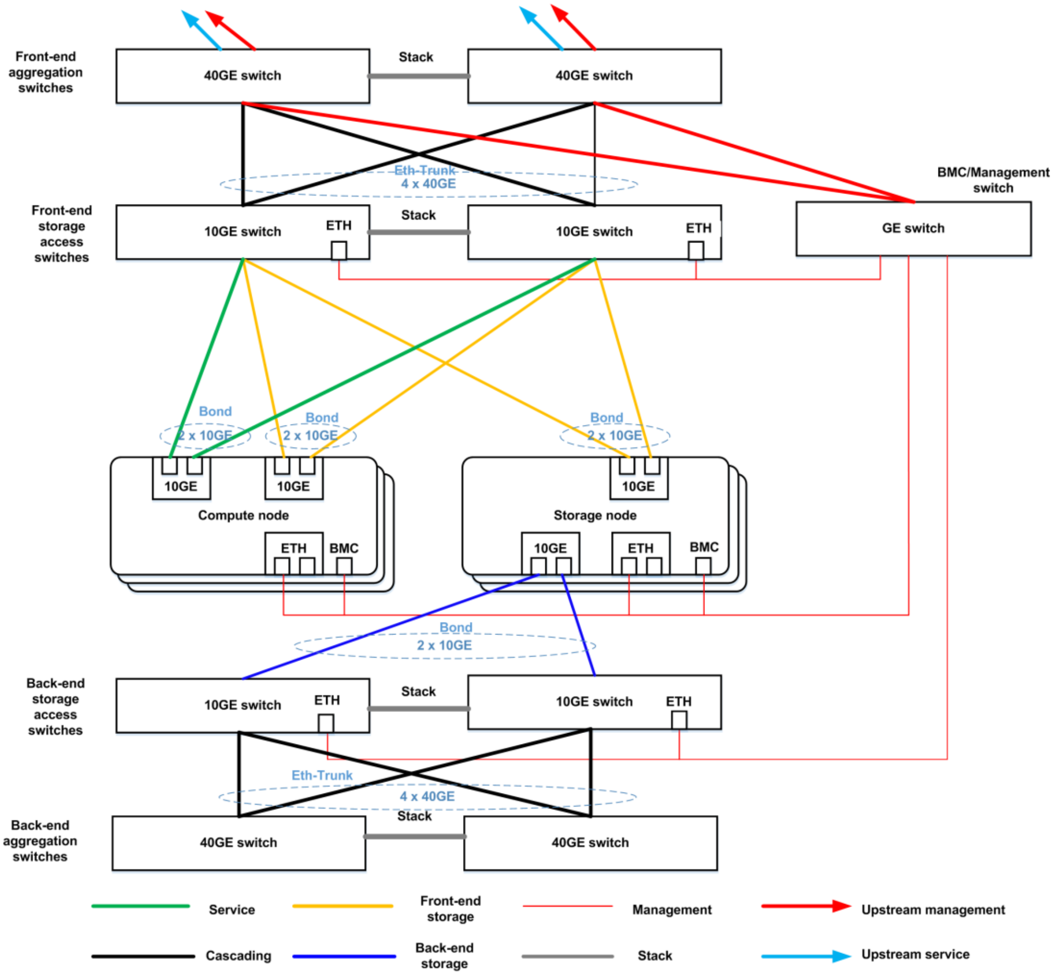


Figure 3-3 Independent deployment, where front- and back-end storage networks are separated



Application Scenarios

10GE networking applies to blade or rack server virtualization scenarios, such as public cloud and large-scale private cloud. When front- and back-end storage networks are separated, the back-end storage network carries background I/Os, including reconstruction and load balancing I/Os. This prevents host I/Os from being affected by background I/Os, ideal for scenarios that require high host I/O performance.

10GE Networking Structure

- If front- and back-end storage networks are converged, each storage node can be configured with one or two dual-port 10GE NICs. When one dual-port 10GE NIC is configured, the storage and management networks share the NIC. When two dual-port 10GE NICs are configured, one NIC is used by the storage network and the other by the management network.
- If front- and back-end storage networks are separated, each storage node can be configured with two or three dual-port 10GE NICs. When two dual-port 10GE NICs are configured, the front-end storage network and management network share one NIC. When three dual-port 10GE NICs are configured, the front-end storage network, back-end storage network, and management network each use one NIC.
- NICs of compute nodes can be shared by storage, service, and management networks:

  - When the NICs are not shared, the storage network must exclusively use one NIC, and the service and management networks can share an NIC or use independent NICs.
  - When the NICs are shared, the storage, service, and management networks share one NIC.
- The BMC port of each node is connected to the GE switch, and the management port of each switch managed by the GE switch is also connected to the GE switch.
- Stacking the 10GE front-end storage access switches is recommended. The front-end storage access switches connect to front-end aggregation switches, and the front-end aggregation switches connect to the customer network.
- Stacking the 10GE back-end storage access switches is recommended. The back-end storage access switches connect to back-end aggregation switches.
- Bonding mode 1 is recommended for 10GE network ports on nodes. If bonding mode 2 or 4 is used, set **Transmit Hash Policy** to **layer3+4** in the configuration file.
- If network ports on nodes are bonded in mode 2 or 4, configure Eth-Trunk for the ports on the switches connecting to a same node.
- When access switches are stacked and aggregation switches are also stacked, configure Eth-Trunk for ports that interconnect the access and aggregation switches.

3.3.1.3 Converged Deployment of Compute and Storage Nodes

FusionStorage supports converged deployment of compute and storage nodes. For large-scale storage pool scenarios (containing more than 64 storage nodes in the future), cabinet-level security is recommended, which requires at least three cabinets. For FusionCube scenarios, node-level security can be used, which requires at least three storage nodes. Converged deployment supports two networking schemes, depending on whether front- and back-end storage networks are converged. Figure 3-4 and Figure 3-5 show the recommended networking architecture.

Figure 3-4 Converged deployment, where front- and back-end storage networks are converged

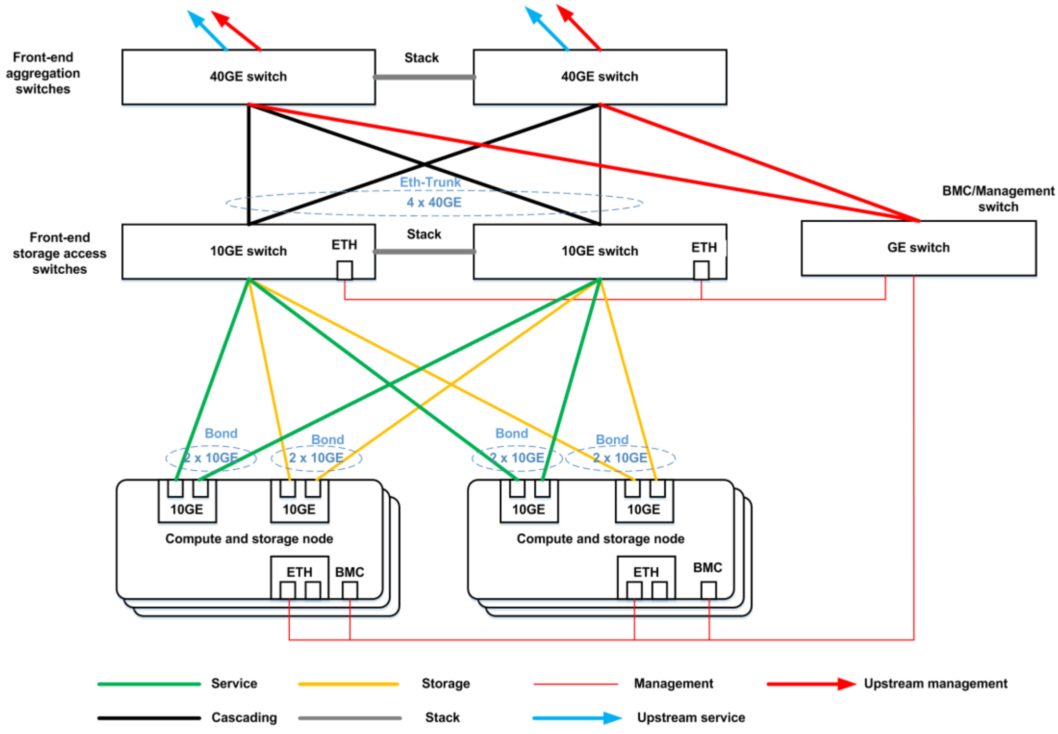
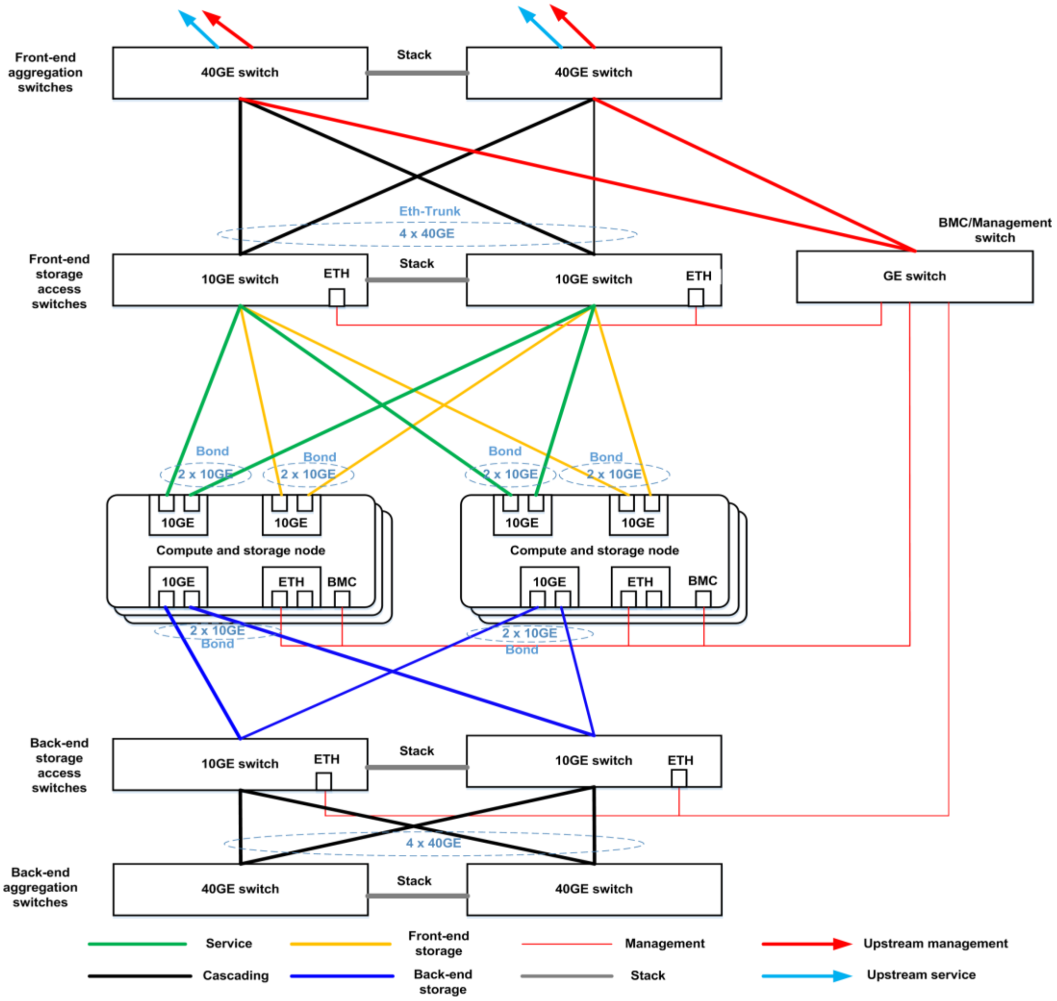


Figure 3-5 Converged deployment, where front- and back-end storage networks are separated



Application Scenarios

10GE networking applies to blade or rack server virtualization scenarios. When front- and back-end storage networks are separated, the back-end storage network carries background I/Os, including load balancing I/Os. This prevents host I/Os from being affected by background I/Os, ideal for scenarios that require stable host I/O performance.

10GE Networking Structure

- If front- and back-end storage networks are converged, each node is configured with two dual-port 10GE NICs by default. In this case, one NIC is used by the storage network, and the other by service and management networks.
- If front- and back-end storage networks are separated, each node is configured with three dual-port 10GE NICs by default. The front- and back-end storage networks each use an NIC, and the service and management networks share one NIC.
- The storage network must use independent physical network ports and cannot share physical network ports with other networks.
- The BMC port of each node is connected to the GE switch, and the management port of each switch managed by the GE switch is also connected to the GE switch.

- Stacking the 10GE front-end storage access switches is recommended. The front-end storage access switches connect to front-end aggregation switches, and the front-end aggregation switches connect to the customer network.
- Stacking the 10GE back-end storage access switches is recommended. The back-end storage access switches connect to back-end aggregation switches.
- Bonding mode 1 is recommended for 10GE network ports on nodes. If bonding mode 2 or 4 is used, set **Transmit Hash Policy** to **layer3+4** in the configuration file.
- If network ports on nodes are bonded in mode 2 or 4, configure Eth-Trunk for the ports on the switches connecting to a same node.
- When access switches are stacked and aggregation switches are also stacked, configure Eth-Trunk for ports that interconnect the access and aggregation switches.

3.3.2 IB Networking

3.3.2.1 Deployment Schemes

In database and data warehouse scenarios, databases and data warehouses are deployed on physical servers that function as compute nodes and FusionStorage is deployed on physical servers that function as storage nodes. Such deployment is called independent deployment of compute and storage nodes. FusionStorage employs IB for node interconnection and supports SSD caches and main storage, significantly improving system performance and reliability while retaining high scalability. This provides high-performance data access and storage for mission-critical enterprise applications, including databases, data warehouses, Enterprise Resource Planning (ERP), and Customer Relationship Management (CRM). FusionStorage supports 40, 56, and 100 Gbit/s IB networking.

Access switches are independent from each other. That is, the access switches are not stacked or in active/standby mode. Access switches and aggregation switches are interconnected through IB cables and do not require Eth-Trunk.

3.3.2.2 Independent Deployment of Compute and Storage Nodes

Independent deployment supports two networking schemes, depending on whether front- and back-end storage networks are converged. Figure 3-6 and Figure 3-7 show the recommended networking architecture.

Figure 3-6 Independent deployment, where front- and back-end storage networks are converged

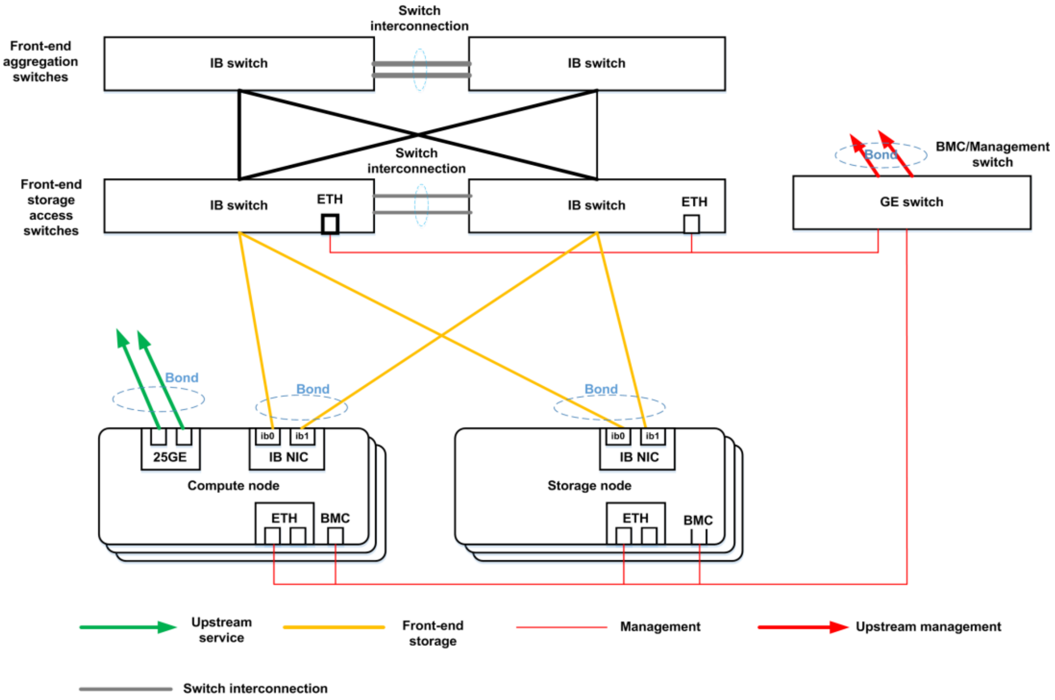
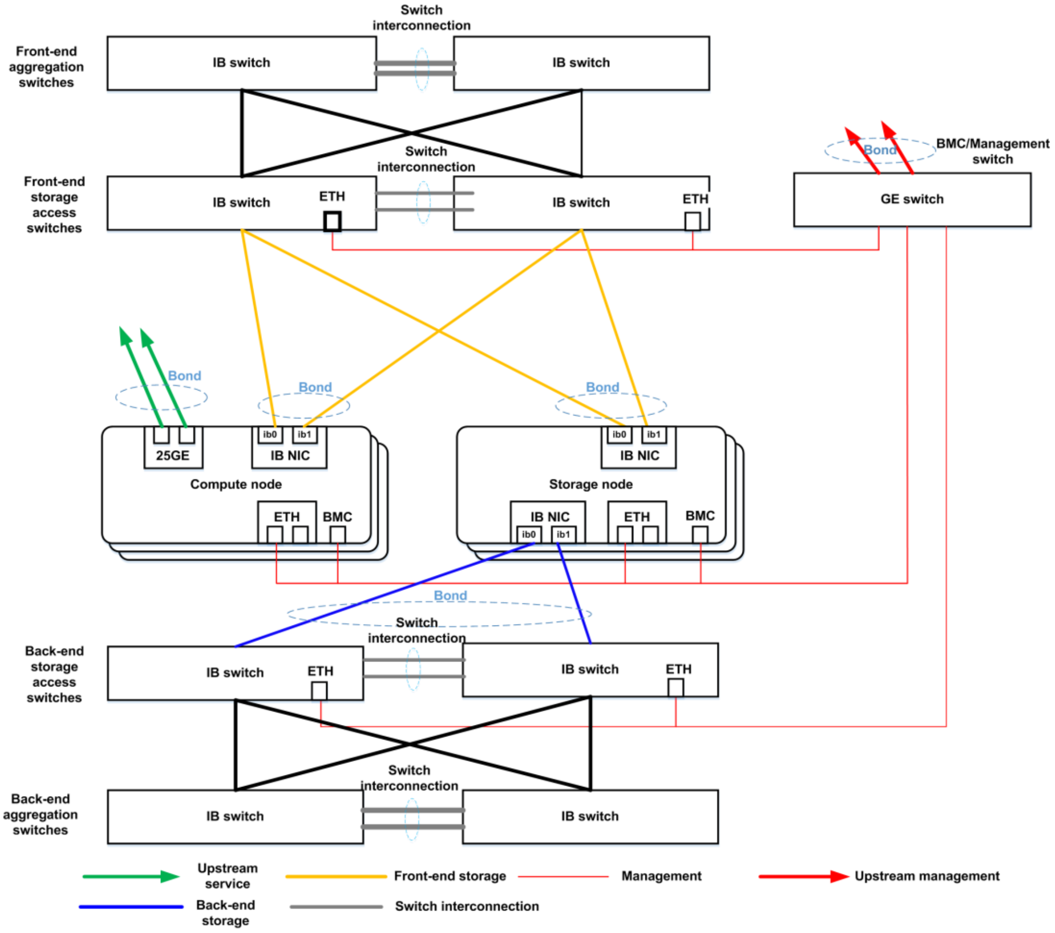


Figure 3-7 Independent deployment, where front- and back-end storage networks are separated



Application Scenarios

IB networking applies to high-performance data storage scenarios, providing high-performance data access and storage for mission-critical enterprise applications, including databases, data warehouses, ERP, and CRM. When front- and back-end storage networks are separated, the back-end storage network carries background I/Os, including reconstruction and load balancing I/Os. This prevents host I/Os from being affected by background I/Os, ideal for scenarios that require high host I/O performance.

IB Networking Structure

- If front- and back-end storage networks are converged, each storage node is configured with one dual-port IB NIC and one management GE NIC by default.
- If front- and back-end storage networks are separated, each storage node can be configured with two dual-port IB NICs and one management GE NIC. In this case, the front- and back-end storage networks each use one IB NIC.
- It is recommended that each compute node be configured with one IB NIC, one service NIC, and one management NIC. The service and management networks can share an NIC.
- The BMC port of each node is connected to the GE switch, and the management port of each switch managed by the GE switch is also connected to the GE switch.
- You are advised to configure active/standby bonding for IB NICs and assign IP addresses for the bond ports. The IP addresses are used for heartbeat communication between FusionStorage nodes and between applications and use the Internet Protocol over InfiniBand (IPoIB) mode.
- Storage links communicate with each other in remote direct memory access (RDMA) mode. IB ports transmit I/O data evenly and are mutually redundant.
- In normal cases, a port on one node communicates only with a port with the same port ID on another node. For example, port **ib0** or **ib1** on node A communicates only with port **ib0** or **ib1** on node B, and ports **ib0** and **ib1** of the two nodes do not communicate with each other. However, if port **ib0** on node A and port **ib1** on node B fail at the same time, the two nodes communicate through ports **ib1** and **ib0**.

3.3.2.3 Converged Deployment of Compute and Storage Nodes

Converged deployment supports two networking schemes, depending on whether front- and back-end storage networks are converged. Figure 3-8 and Figure 3-9 show the recommended networking architecture.

Figure 3-8 Converged deployment, where front- and back-end storage networks are converged

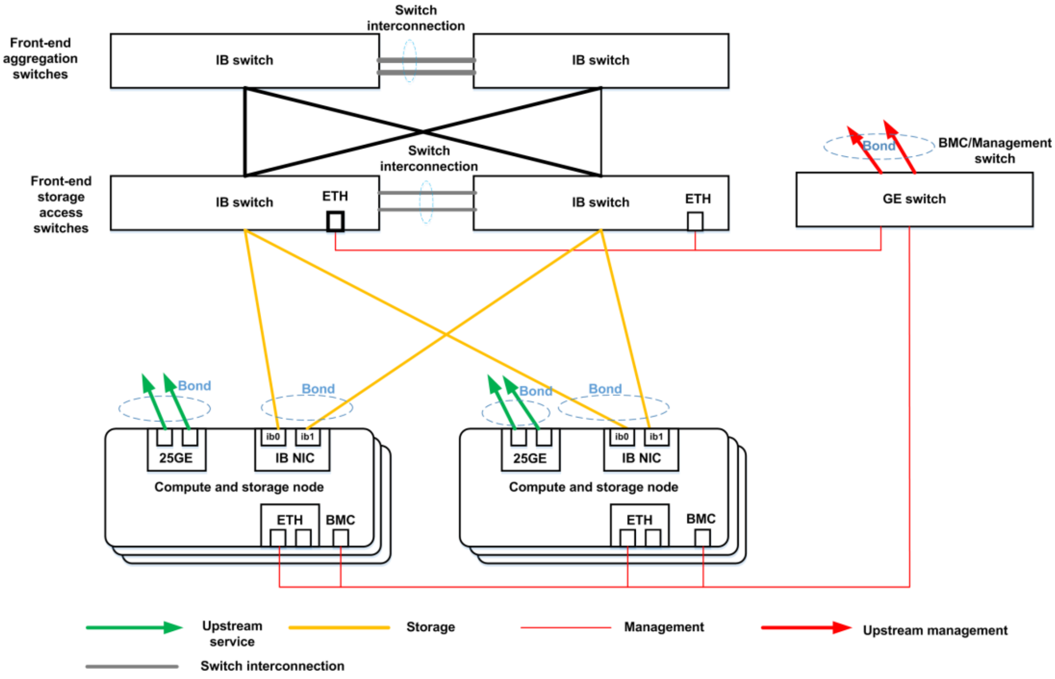
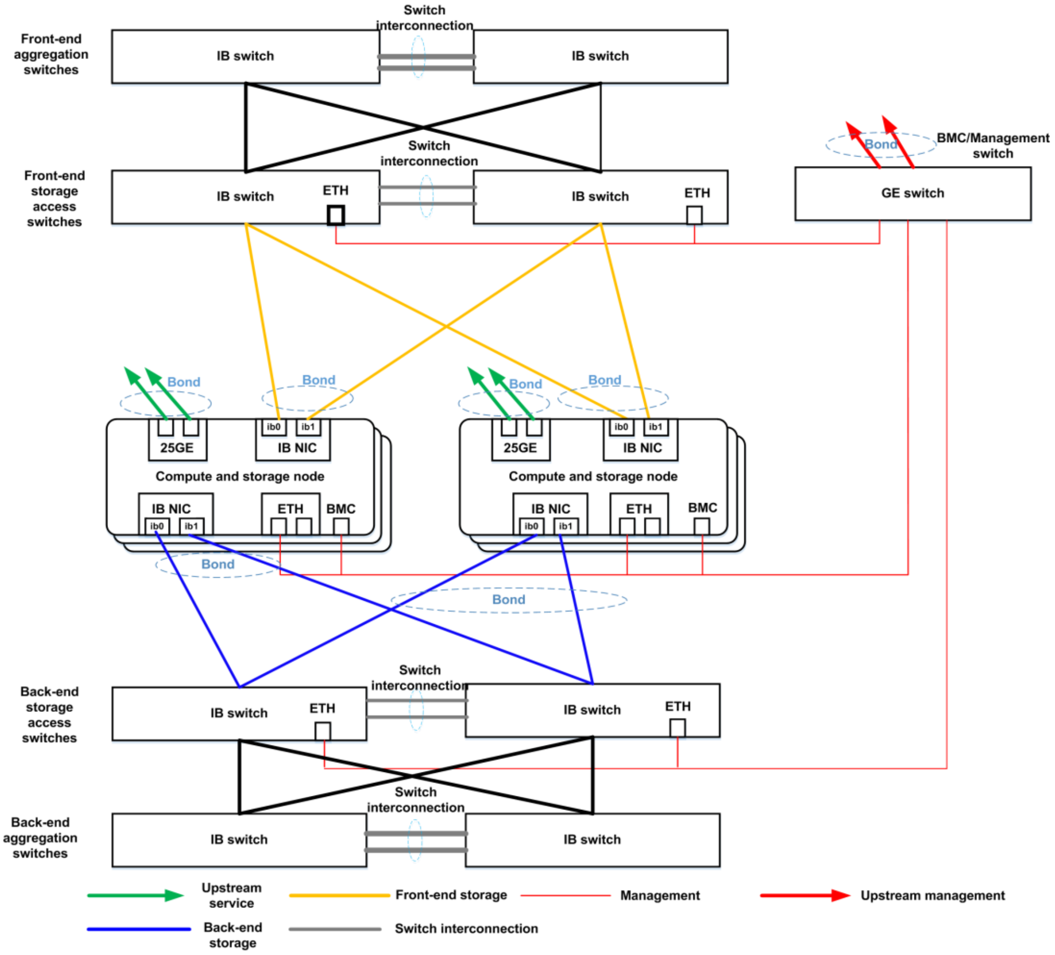


Figure 3-9 Converged deployment, where front- and back-end storage networks are separated



Application Scenarios

IB networking applies to high-performance data storage scenarios, providing high-performance data access and storage for mission-critical enterprise applications, including databases, data warehouses, ERP, and CRM. When front- and back-end storage networks are separated, the back-end storage network carries background I/Os, including reconstruction and load balancing I/Os. This prevents host I/Os from being affected by background I/Os, ideal for scenarios that require high host I/O performance.

IB Networking Structure

- If front- and back-end storage networks are converged, each node is configured with one dual-port IB NIC, one service NIC, and one management NIC by default. The service and management networks can share an NIC.
- If front- and back-end storage networks are separated, each node is configured with two dual-port IB NICs, one service NIC, and one management NIC by default. The service and management networks can share an NIC.
- The BMC port of each node is connected to the GE switch, and the management port of each switch managed by the GE switch is also connected to the GE switch.
- You are advised to configure active/standby bonding for IB NICs and assign IP addresses for the bond ports. The IP addresses are used for heartbeat communication between FusionStorage nodes and between applications and use the IPoIB mode.
- Storage links communicate with each other in RDMA mode. IB ports transmit I/O data evenly and are mutually redundant.
- In normal cases, a port on one node communicates only with a port with the same port ID on another node. For example, port **ib0** or **ib1** on node A communicates only with port **ib0** or **ib1** on node B, and ports **ib0** and **ib1** of the two nodes do not communicate with each other. However, if port **ib0** on node A and port **ib1** on node B fail at the same time, the two nodes communicate through ports **ib1** and **ib0**.

3.3.3 RoCE Networking

3.3.3.1 Deployment Schemes

In RoCE networking, FusionStorage connects to data center networks through TOR access switches. It supports both independent and converged deployment of compute and storage nodes. Independent deployment means that compute nodes (running applications) and storage nodes are deployed on different servers. Converged deployment means that compute nodes (running applications) and storage nodes are deployed on the same servers. The switches recommended for 25GE, 40GE, or 100GE RoCE networking are the same as those for Ethernet networking.

3.3.3.2 Independent Deployment of Compute and Storage Nodes

FusionStorage supports independent deployment of compute and storage nodes. Cabinet-level security is recommended, which requires at least three cabinets. Independent deployment supports two networking schemes, depending on whether front- and back-end storage networks are converged. Figure 3-10 and Figure 3-11 show the recommended networking architecture.

Figure 3-10 Independent deployment, where front- and back-end storage networks are converged

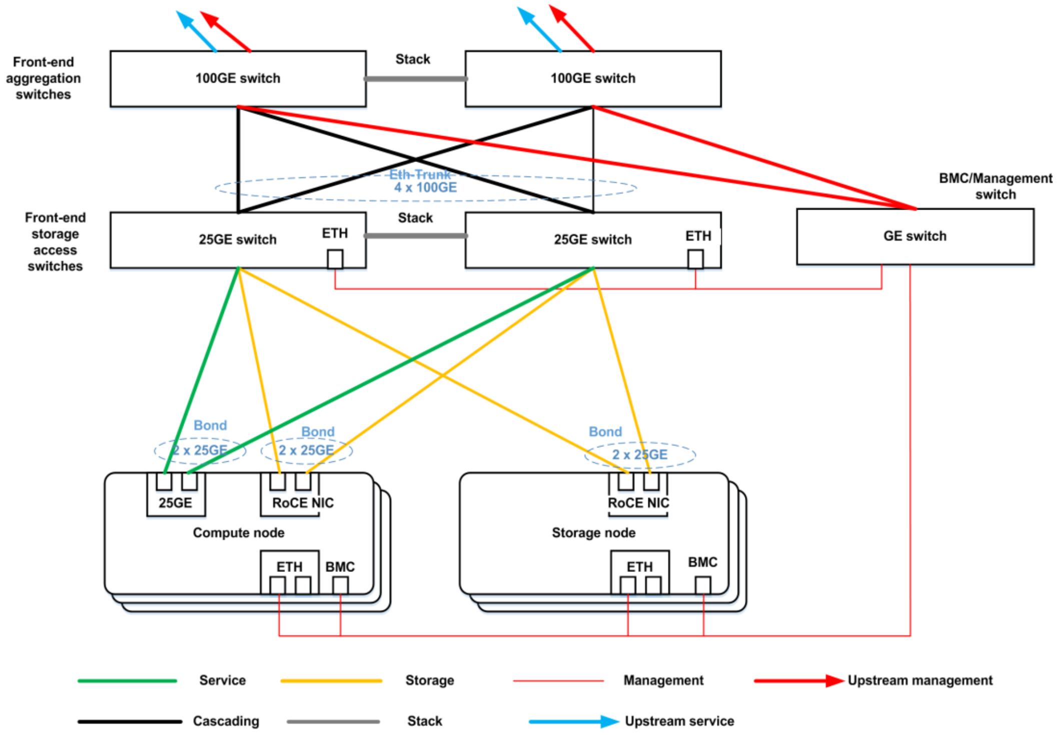
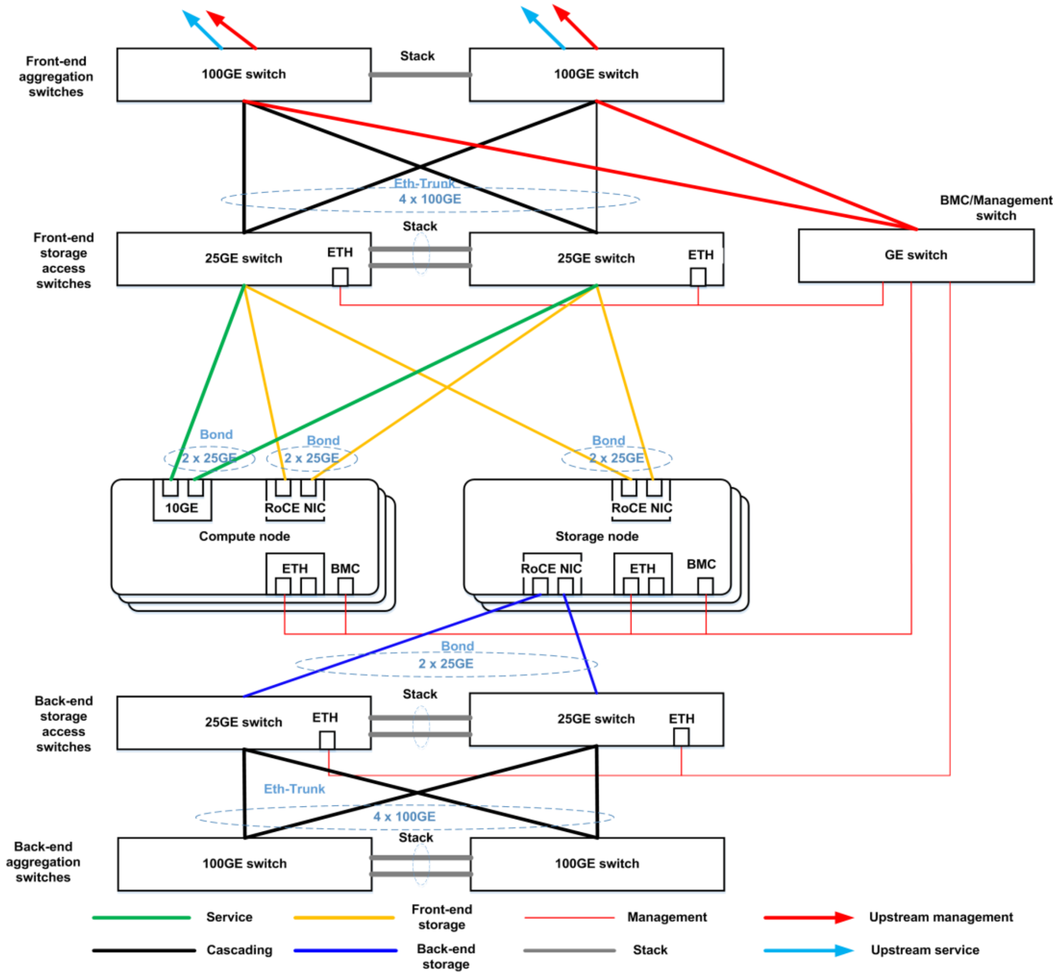


Figure 3-11 Independent deployment, where front- and back-end storage networks are separated



Application Scenarios

RoCE networking applies to low latency, low CPU load, and large bandwidth scenarios such as HPC and big data processing. When front- and back-end storage networks are separated, the back-end storage network carries background I/Os, including load balancing I/Os. This prevents host I/Os from being affected by background I/Os, ideal for scenarios that require stable host I/O performance.

RoCE Networking Structure

- If front- and back-end storage networks are converged, each storage node can be configured with one RoCE NIC and one management NIC.
- If front- and back-end storage networks are separated, each storage node can be configured with two RoCE NICs and one management NIC. In this case, the front- and back-end storage networks each use one RoCE NIC.
- Each compute node must be configured with one RoCE NIC for the storage network and can also be configured with one service NIC and one management NIC. The service and management networks can share an NIC.
- The BMC port of each node is connected to the GE switch, and the management port of each switch managed by the GE switch is also connected to the GE switch.

- Stacking the 25GE front-end storage access switches is recommended. The front-end storage access switches connect to front-end aggregation switches, and the front-end aggregation switches connect to the customer network.
- Stacking the 25GE back-end storage access switches is recommended. The back-end storage access switches connect to back-end aggregation switches.
- Bonding mode 1 is recommended for 25GE network ports on nodes. If bonding mode 2 or 4 is used, set **Transmit Hash Policy** to **layer3+4** in the configuration file.
- If network ports on nodes are bonded in mode 2 or 4, configure Eth-Trunk for the ports on the switches connecting to a same node.
- When access switches are stacked and aggregation switches are also stacked, configure Eth-Trunk for ports that interconnect the access and aggregation switches.

3.3.3.3 Converged Deployment of Compute and Storage Nodes

FusionStorage supports converged deployment of compute and storage nodes. For large-scale storage pool scenarios (containing more than 64 storage nodes in the future), cabinet-level security is recommended, which requires at least three cabinets. For FusionCube scenarios, node-level security can be used, which requires at least three storage nodes. Converged deployment supports two networking schemes, depending on whether front- and back-end storage networks are converged. Figure 3-12 and Figure 3-13 show the recommended networking architecture.

Figure 3-12 Converged deployment, where front- and back-end storage networks are converged

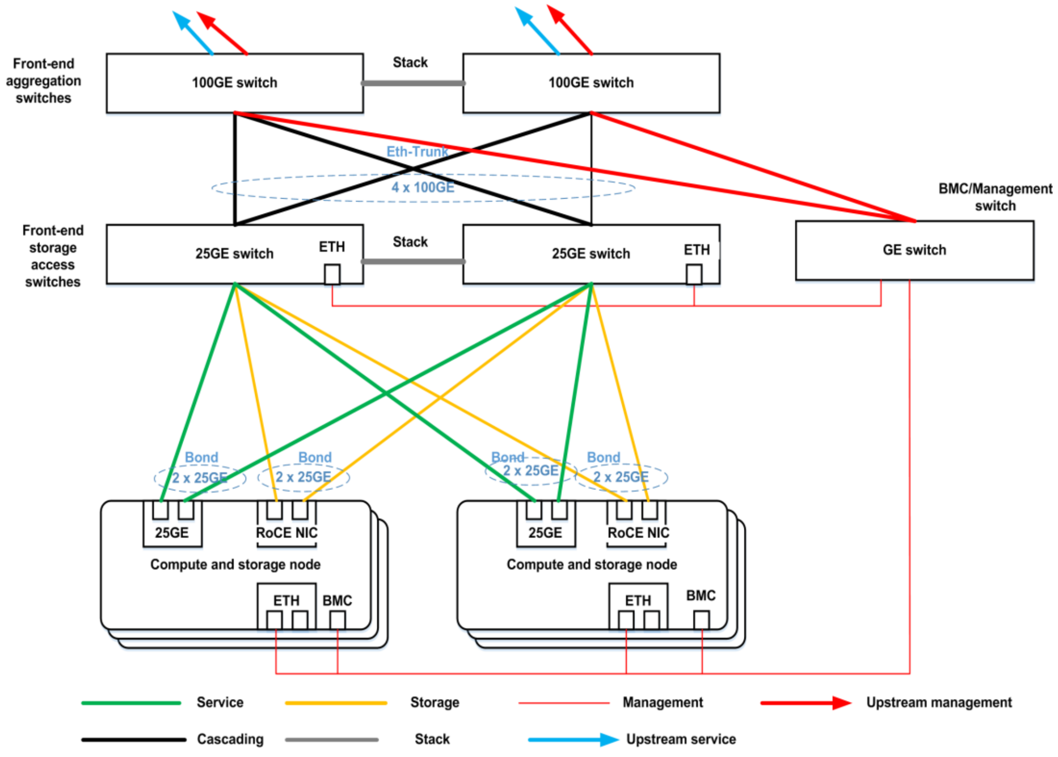
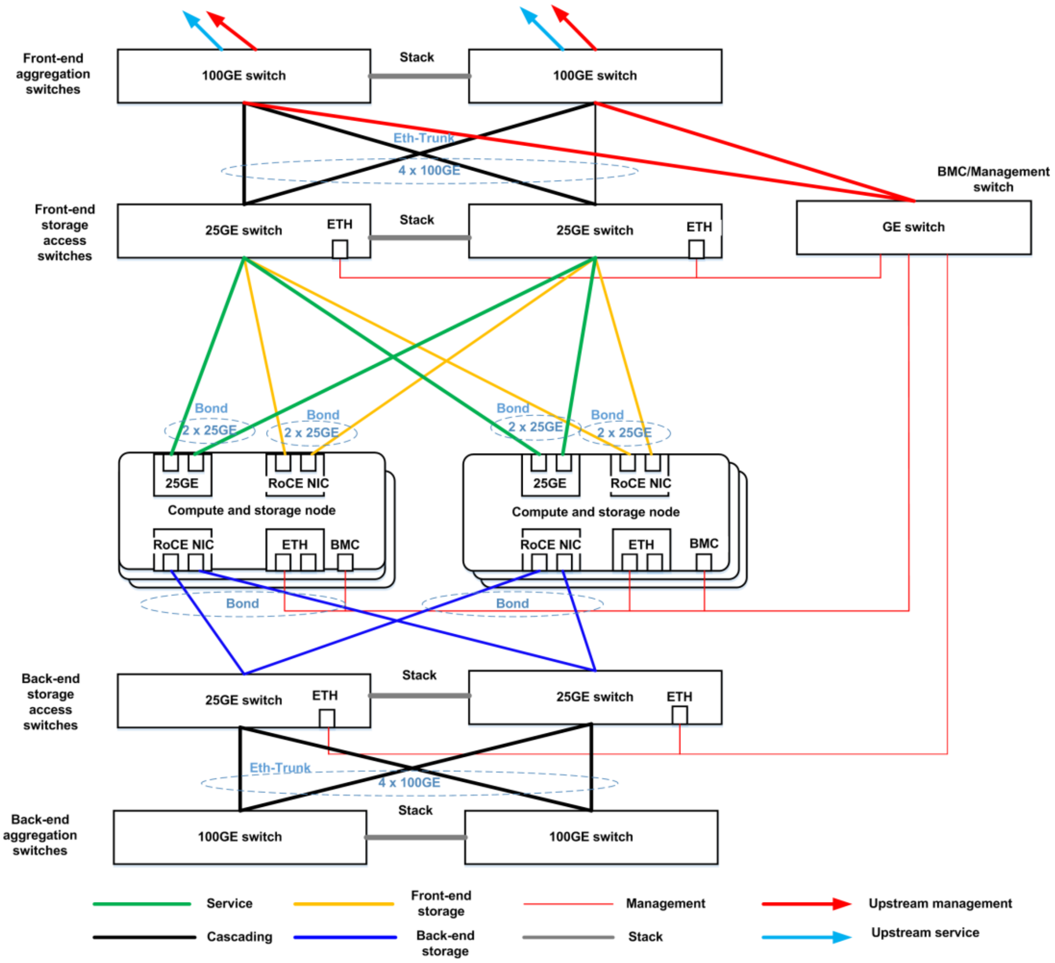


Figure 3-13 Converged deployment, where front- and back-end storage networks are separated



Application Scenarios

RoCE networking applies to low latency, low CPU load, and large bandwidth scenarios such as HPC and big data processing. When front- and back-end storage networks are separated, the back-end storage network carries background I/Os, including load balancing I/Os. This prevents host I/Os from being affected by background I/Os, ideal for scenarios that require stable host I/O performance.

RoCE Networking Structure

- If front- and back-end storage networks are converged, each node can be configured with one RoCE NIC, one service NIC, and one management NIC. The service and management networks can share an NIC.
- If front- and back-end storage networks are separated, each node can be configured with two RoCE NICs, one service NIC, and one management NIC. In this case, the front- and back-end storage networks each use one RoCE NIC. The service and management networks can share an NIC.
- The BMC port of each node is connected to the GE switch, and the management port of each switch managed by the GE switch is also connected to the GE switch.
- Stacking the 25GE front-end storage access switches is recommended. The front-end storage access switches connect to front-end aggregation switches, and the front-end aggregation switches connect to the customer network.

- Stacking the 25GE back-end storage access switches is recommended. The back-end storage access switches connect to back-end aggregation switches.
- Bonding mode 1 is recommended for 25GE network ports on nodes. If bonding mode 2 or 4 is used, set **Transmit Hash Policy** to **layer3+4** in the configuration file.
- If network ports on nodes are bonded in mode 2 or 4, configure Eth-Trunk for the ports on the switches connecting to a same node.
- When access switches are stacked and aggregation switches are also stacked, configure Eth-Trunk for ports that interconnect the access and aggregation switches.

3.4 Key Service Processes

3.4.1 Components

Figure 3-14 shows the logical structure of FusionStorage.

Figure 3-14 Logical structure of FusionStorage

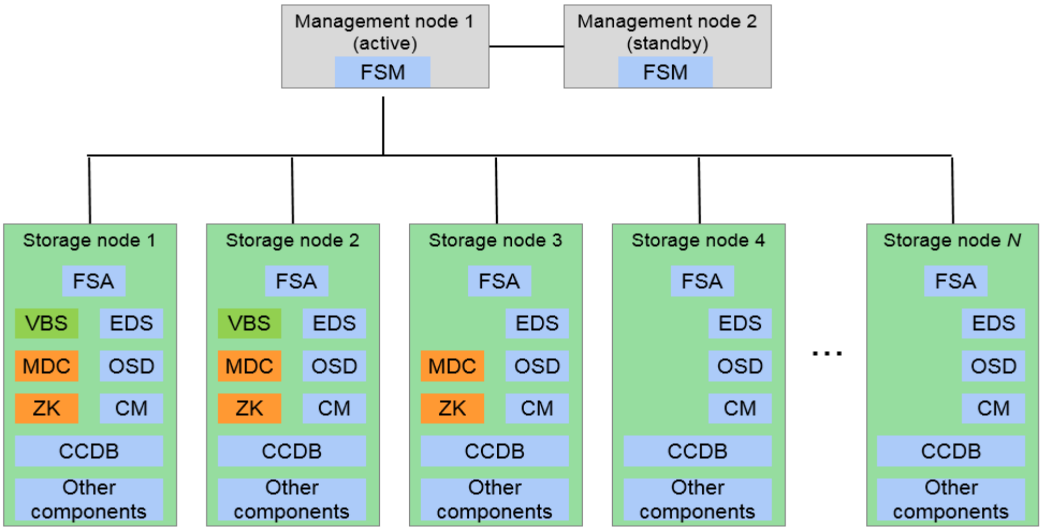


Table 3-4 Key components of FusionStorage

Component	Description
FSM	Management module of FusionStorage. FSM provides O&M functions such as alarming, monitoring, logging, and configuration. It is typically deployed on two management nodes in active/standby mode.
FusionStorage Agent (FSA)	Management agent component of FusionStorage. FSA is deployed on each node to enable communication between the node and FSM, collect node monitoring and alarm information, as well as receive upgrade packages and perform upgrades when software components on the node need to be upgraded.
Metadata Controller (MDC)	Metadata control component that controls the status of the distributed cluster as well as data distribution and rebuilding rules. At least three MDCs must be deployed in a system to form an MDC cluster. When the

Component	Description
	system is started, the Zookeeper (ZK) cluster elects a primary MDC to monitor other MDCs. If the primary MDC fails, the ZK cluster elects a new one. Each storage pool has a home MDC. If the home MDC of a storage pool fails, the primary MDC assigns another MDC to manage the pool. One MDC manages a maximum of two storage pools. MDC can be started on every storage node. When a storage pool is added, its home MDC automatically starts. A maximum of 96 MDCs can be started in one system.
ZK	An odd number (for example, 3, 5, or 7) of ZKs must be deployed in a FusionStorage system to form a ZK cluster. The ZK cluster provides primary arbitration for the MDC cluster. At least three ZKs must be deployed, and more than half of the deployed ZKs must be active and accessible.
Virtual Block Service (VBS)	Virtual block storage management component that manages volume metadata. VBS provides the distributed storage access point service through SCSI or iSCSI interfaces and enables compute resources to access distributed storage resources.
Enterprise Data Service (EDS)	Component that processes I/O services sent from VBS. EDS provides block storage features, such as HyperSnap and HyperClone. It manages storage space and establishes index relationships between data blocks and storage space to facilitate locating data storage locations. In addition, EDS can perform deduplication and compression before storing data in physical space.
OSD	Component that handles I/O messages from VBS, implements data redundancy protection, and stores data onto persistent storage media. One OSD is deployed on each node, each OSD has multiple instances, and each HDD corresponds to one instance by default. When an SSD or SSD card is used as the main storage, the SSD or SSD card also corresponds to one instance.
Cluster Manager (CM)	Cluster management component that manages the status of the entire storage cluster, including the status of each component, and monitors the status of each component in real time. If a component is faulty, CM will trigger appropriate measures based on the component status to rectify the fault.
Cluster Configuration Database (CCDB)	Cluster configuration database that stores user configurations, including EDS configurations.

3.4.2 Access Protocols

FusionStorage block storage uses the SCSI or iSCSI protocol to provide block access interfaces through VBS.

- SCSI

  A standard SCSI driver is used to map volumes to a host and provide volume space for the operating system and databases on the host. The host communicates with the storage system through VBS using a private communication protocol. The implementation in the two deployment schemes is as follows:

  - In independent deployment where compute and storage nodes are deployed on different servers, VBS is deployed on compute nodes. Hosts access the storage system through VBS using SCSI.
  - In converged deployment where compute and storage nodes are deployed on the same servers, VBS simulates a local disk to provide the SCSI target function and manages all local host access commands.

- iSCSI

  A standard iSCSI driver works with multipathing software to support iSCSI block access interfaces and provides volume space for operating systems and databases. The implementation in the two deployment schemes is as follows:

  - In independent deployment where compute and storage nodes are deployed on different servers and VBS is virtually deployed (such as interconnection with VMware), VBS is deployed in Control Virtual Machines (CVMs) on compute nodes and functions as an iSCSI target. Guest VMs functioning as iSCSI initiators access the VBS using iSCSI.
  - In converged deployment where compute and storage nodes are deployed on the same servers, VBS is deployed on storage nodes. The storage nodes directly provide iSCSI block access interfaces for host applications. In this deployment scheme, the storage access mode is the same as that of traditional storage systems.

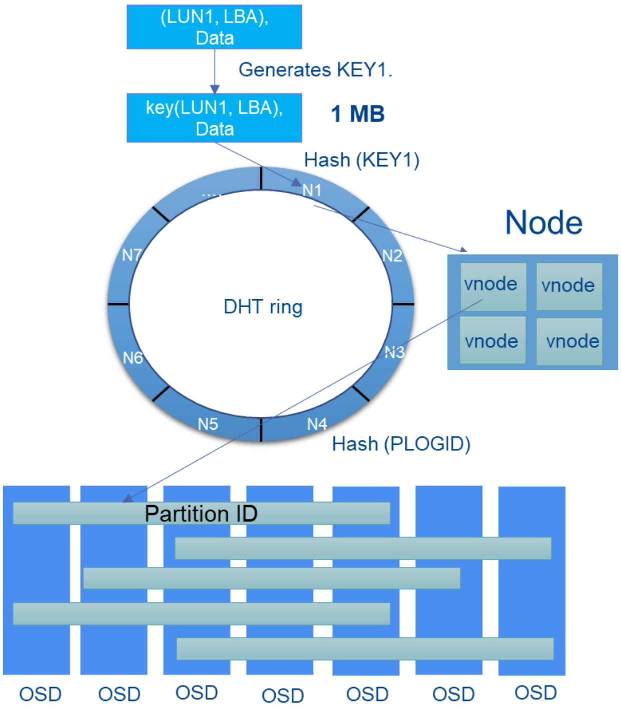
3.4.3 Data Routing

FusionStorage block storage supports large-capacity horizontal expansion and elastic scaling by leveraging a two-layer mapping mechanism:

1. VBS divides volume space by 1 MB, calculates hash values by using LUN IDs and logical block addresses (LBAs), and locates the nodes that process data.
2. Plog of OSD calculates hash values by using Plog IDs and offsets to determine specific data storage locations on disks.

Figure 3-15 shows the data routing process.

Figure 3-15 Data routing of FusionStorage



The distributed hash table (DHT) ring at layer 1 is to distribute data to storage nodes calculated using the hash algorithm for processing. This ensures that each piece of data will be processed by a storage node, thereby assuring balanced service processing. The system locates storage nodes according to LUN IDs and LBAs, and then locates the vnodes that actually process the data on the storage nodes. A vnode is a logical processing unit. A node is divided into four logical processing units, that is, four vnodes. Suppose that a storage cluster consists of six nodes. When one node becomes faulty, the services processed by the four vnodes on the faulty node can be taken over by four of the other five nodes in the cluster. In this case, four nodes in the cluster each run five vnodes, and one node runs four vnodes. The vnode mechanism ensures that services on a faulty node can be distributed to multiple nodes for takeover, preventing service processing bottlenecks when only one node takes over all services of a faulty node.

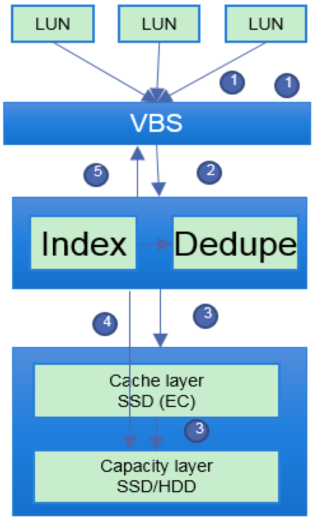
The DHT ring at layer 2 is to persistently store data to storage space using the hash algorithm. The hash algorithm ensures that data is evenly stored on disks. The system locates data in disks based on Plog IDs and offsets, improving data search efficiency. The DHT routing technology uses Huawei-developed algorithms to ensure that data is evenly distributed on disks. When hardware is increased or decreased (upon a capacity expansion or fault), the system can ensure data migration validity, implement automatic and quick self-healing, and achieve automatic resource balancing.

Cabinet-level, node-level, and disk-level security can be implemented for storage space, and node-level security is used by default.

3.4.4 Read I/O Process

Figure 3-16 shows the read I/O process of FusionStorage.

Figure 3-16 Read I/O process of FusionStorage (taking EC as an example)



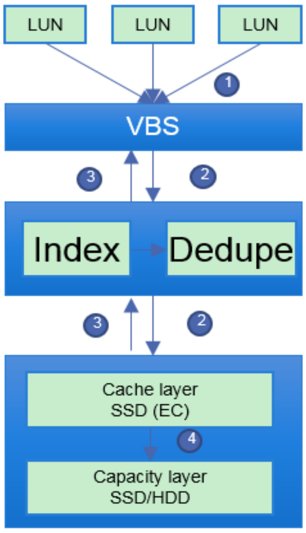
The read I/O process is described as follows:

1. An upper-layer application sends a read I/O request to the storage system. VBS of the storage system forwards the request to a specific node calculated using the DHT hash algorithm at layer 1.
2. After receiving the request, EDS (Index+Dedupe) on the node first searches for the required data in the memory write cache. If the required data is found, EDS returns the data to VBS.
3. If the data is not found in the memory write cache, EDS searches for it in the memory read cache. If the data is still not found, EDS searches for the data in the SSD cache. If the data is not hit, EDS reads the data from storage media. For details, see 6.2.2 Read Cache.

3.4.5 Write I/O Process

Figure 3-17 shows the write I/O process of FusionStorage.

Figure 3-17 Write I/O process of FusionStorage (taking EC as an example)



The write I/O process is described as follows:

1. An upper-layer application sends a write I/O request to the storage system. VBS of the storage system forwards the request to a specific node calculated using the DHT hash algorithm at layer 1.
2. After receiving the request, EDS (Index+Dedupe) on the node writes the data to SSD cache disks at the cache layer in EC mode. In addition, EDS stores the data in the memory on the node. After that, EDS returns a write success message to VBS, and VBS forwards the message to the upper-layer application. After the data in the memory is aggregated into a large block, the system flushes the data to storage media at the capacity layer. For details, see 6.2.1 Write Cache.

4 Block Storage Features

4.1 SmartThin

FusionStorage block storage provides the SmartThin feature to implement thin provisioning, offering applications much more virtual storage resources than those actually available on physical storage devices. This delivers much higher storage space utilization than traditional provisioning.

FusionStorage adopts the DHT routing technology. No centralized metadata is required for recording volume thin provisioning information. This eliminates performance deterioration encountered by traditional SAN storage devices.

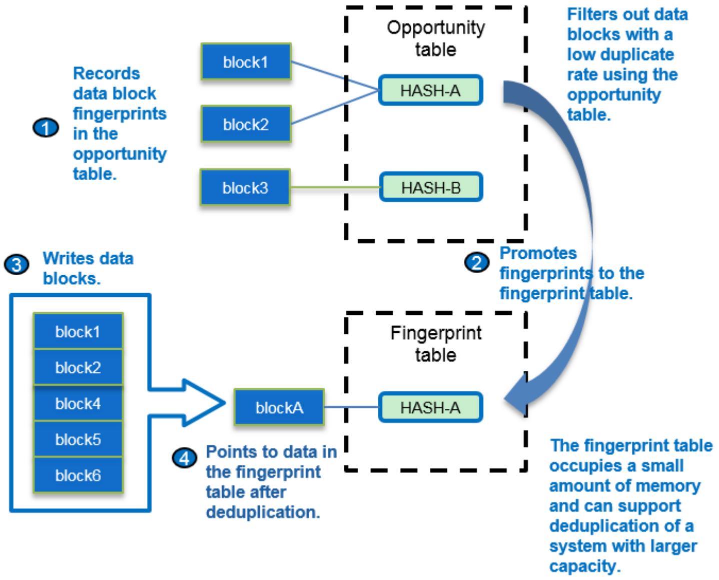
4.2 SmartDedupe and SmartCompression

To cope with the increase in operation costs incurred by management data growth, storage device vendors introduce data reduction techniques, including data deduplication and compression, to reduce the amount of data that is actually stored. SmartDedupe and SmartCompression are Huawei-developed features that respectively implement deduplication and compression for FusionStorage block storage.

FusionStorage adopts intelligent self-adaptive deduplication technology, oriented to user requirements. When the service load is heavy, inline deduplication is disabled automatically to ensure performance and post-process deduplication is implemented to delete duplicate data. When the service load is light, inline deduplication is enabled automatically to prevent read/write amplification of post-process deduplication. The intelligent self-adaptive deduplication technology automatically switches between inline and post-process deduplication based on service loads without user awareness, making the most of the two deduplication modes.

To obtain better deduplication and compression effects, FusionStorage adopts global deduplication. The distributed storage space is huge. To reduce the memory space consumed by the fingerprint table, an opportunity table is introduced, as shown in Figure 4-1. The fingerprints of data blocks are first recorded in the opportunity table for counting. When the number of data blocks with a same fingerprint reaches a specific threshold (3 by default and modifiable), the system promotes the fingerprint to the fingerprint table and implements data deduplication accordingly.

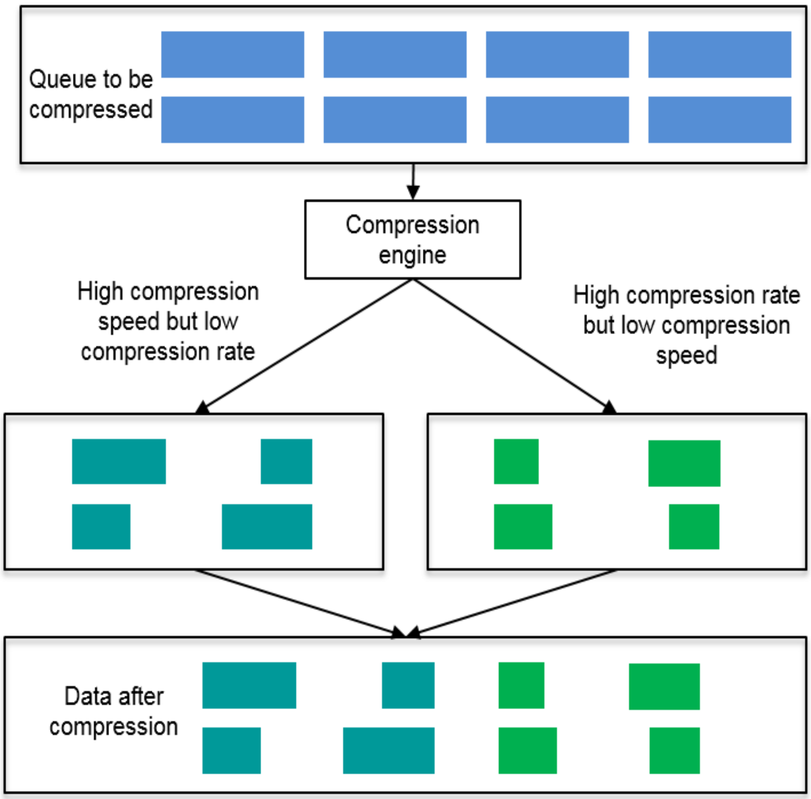
Figure 4-1 Deduplication of FusionStorage


Opportunity table
block1
block2
block3
HASH-A
HASH-B
1 Records data block fingerprints in the opportunity table.
Filters out data blocks with a low duplicate rate using the opportunity table.
2 Promotes fingerprints to the fingerprint table.
3 Writes data blocks.
block1
block2
block4
block5
block6
blockA
Fingerprint table
HASH-A
4 Points to data in the fingerprint table after deduplication.
The fingerprint table occupies a small amount of memory and can support deduplication of a system with larger capacity.

If compression is disabled, the system directly applies for storage space to store data blocks. If compression is enabled, compression will be performed for the data blocks before storage. The data blocks will be compressed by the compression engine at the granularity of 512 bytes and then saved in the system.

The compression engine of FusionStorage runs in a combination of two different compression algorithms. One is the algorithm with high compression speed but low compression rate and the other one is the algorithm with high compression rate but low compression speed. By configuring different execution ratios of the two compression algorithms, you can obtain different performance and data reduction rates. Only one compression algorithm can be selected for a storage pool. Changing the compression algorithm of a storage pool does not affect compressed data. During data reads, compressed data will be decompressed using the same compression algorithm when the data was compressed.

Figure 4-2 Multi-level compression of FusionStorage


Queue to be compressed
Compression engine
High compression speed but low compression rate
High compression rate but low compression speed
Data after compression

4.3 Multiple Storage Pools

To provide storage media of different performance levels and implement fault isolation, FusionStorage block storage supports multiple storage pools. One FSM module can manage multiple storage pools. Multiple storage pools share one FusionStorage block storage cluster, including ZKs and the primary MDC. Each storage pool has a home MDC. When a storage pool is created, an MDC automatically starts as the home MDC of the pool. The maximum number of storage pools is 128 and that of MDCs is 96. If there are more than 96 storage pools, existing MDCs will be appointed as the home MDCs for the excessive storage pools. One MDC manages a maximum of two storage pools. The home MDC of a storage pool is responsible for initializing the pool. At the initialization stage, the storage resources are partitioned and the views of the partitions and OSDs are stored in ZK disks. If the home MDC of a storage pool is faulty, the primary MDC appoints another MDC to manage the pool.

Plan storage pools in accordance with the following rules:

- A storage pool contains only one type of disks. Plan different storage pools for different types of disks.
- It is recommended that disks in a storage pool have the same capacity. Otherwise, the system uses all disks as they have the capacity same as the disk with the smallest capacity in the storage pool.
- A storage pool contains only one type of cache media. Plan different storage pools for different types of cache media.

- For a new storage pool, it is recommended that each storage node in the storage pool have the same number of disks. The difference between the numbers of disks on storage nodes in the same storage pool must not be greater than 2.

4.4 Data Encryption

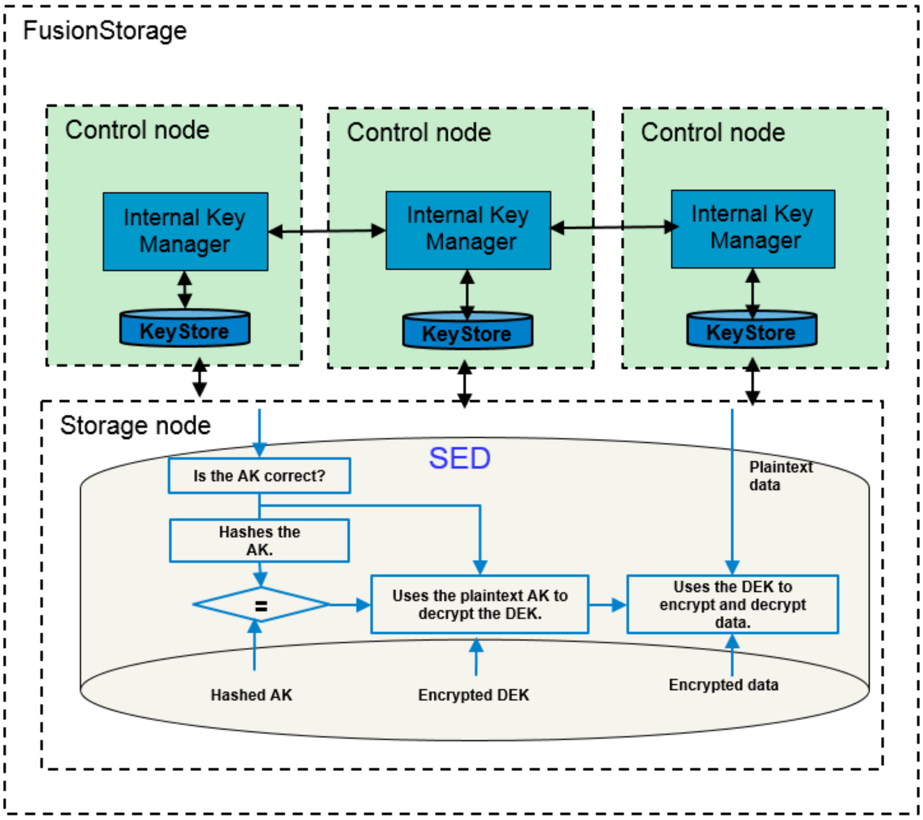
FusionStorage block storage supports data encryption by using self-encrypting disks (SEDs) and Internal Key Manager (an internal key management system of FusionStorage), implementing static data encryption and ensuring data security.

Data encryption of FusionStorage brings the following benefits:

- By using SEDs, data is encrypted and decrypted on disks, eliminating impacts on service application processing.
- Data is encrypted after being flushed to disks without affecting upper-layer value-added features, such as data deduplication and compression.
- Fast data destruction is supported by destroying keys.
- Internal Key Manager is easy to deploy, configure, and manage.

FusionStorage adopts a distributed architecture to manage Internal Key Manager. Internal Key Managers deployed on multiple nodes work together to provide secure key services for SEDs. Figure 4-3 shows the implementation.

Figure 4-3 Data encryption of FusionStorage



As shown in Figure 4-3, data encryption is implemented using the following two mechanisms:

- Authentication key (AK) mechanism: After data encryption is enabled, the storage system enables the AutoLock function of SEDs and uses AKs assigned by Internal Key Manager to authenticate the access to the SEDs. SED access is protected by AutoLock and only the storage system itself can access its SEDs. When the storage system accesses an SED, it acquires an AK from Internal Key Manager. If the AK is consistent with that of the SED, the SED decrypts the data encryption key (DEK) for data encryption/decryption. If the AKs are inconsistent, read and write operations will fail.
- DEK mechanism: After AutoLock authentication is successful, SEDs use built-in encryption chips and internal DEKs to encrypt or decrypt the data that is written or read. When you write data, the system encrypts the plaintext data using the DEK of the AES encryption engine and then writes the data into storage media. When you read data, the system decrypts the requested data into plaintext using the DEK. The DEK cannot be acquired separately, which means that the original information on an SED cannot be read directly after it is removed from the storage system.

4.5 SmartQoS

SmartQoS is the QoS feature of FusionStorage that enables you to set upper limits on IOPS or bandwidth for certain applications. Based on the upper limits, SmartQoS can accurately limit performance of these applications, preventing them from contending for storage resources with critical applications.

SmartQoS extends the information lifecycle management (ILM) strategy to implement application performance tiering within a storage system. When multiple applications run on one storage system, proper QoS configurations ensure the performance of critical services:

- SmartQoS controls storage resource usage by limiting the performance upper limits of non-critical applications so that critical applications have sufficient storage resources to achieve performance objectives.
- Some services are prone to traffic bursts or storms in specified time periods, for example, daily backup, database sorting, monthly salary distribution, and periodic bill settlement. The traffic bursts or storms will consume a large amount of system resources. If the traffic bursts or storms occur at production time, interactive services will be affected. To avoid this, you can limit the maximum IOPS or bandwidth of these services during traffic burst occurrence time to control array resources consumed by the services, preventing production or interactive services from being affected.

SmartQoS of FusionStorage has the following characteristics:

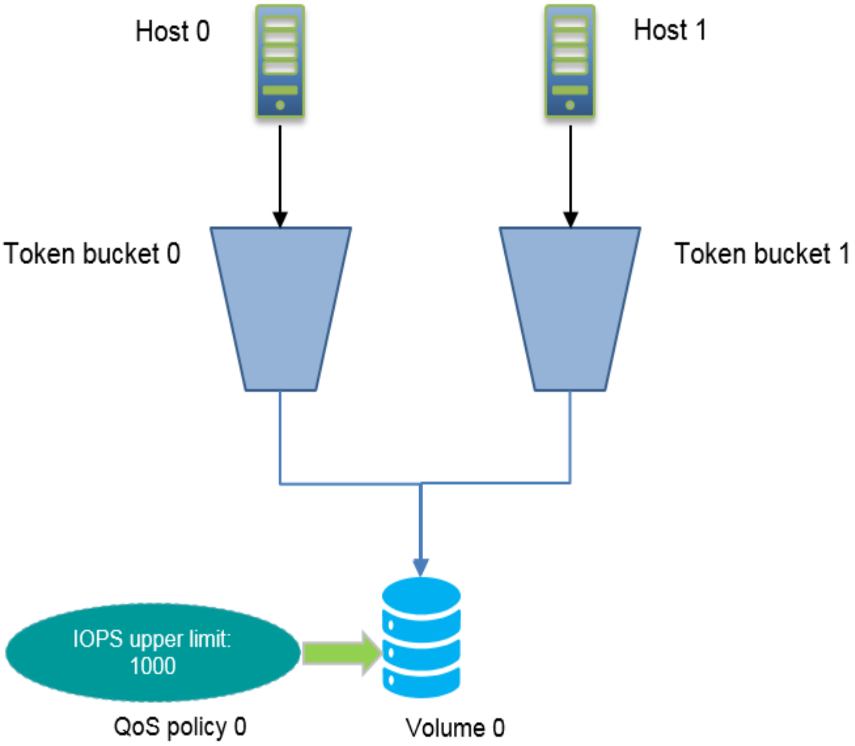
SmartQoS enables you to set performance objectives for volumes and storage pools by specifying bandwidth and IOPS upper limits. The total read and write performance, read performance, or write performance can be limited. QoS policies can take effect in specified time ranges based on service loads to prevent I/O storms from affecting production services.

SmartQoS leverages a self-adaptive adjustment algorithm based on negative feedback and a volume-based I/O traffic control management algorithm to limit traffic based on the performance control objectives (such as IOPS and bandwidth) specified by users. The I/O traffic control mechanism prevents certain services from affecting other services due to heavy traffic and supports burst traffic functions within specified time ranges. QoS traffic control of FusionStorage is implemented as follows:

- Self-adaptive adjustment algorithm based on negative feedback

  When a volume is mounted to multiple VBS nodes, the system resources consumed by services on the volume need to be controlled. That is, the overall performance of the volume needs to be limited. This requires coordination of distributed traffic control parameters.

Figure 4-4 Self-adaptive adjustment algorithm based on negative feedback

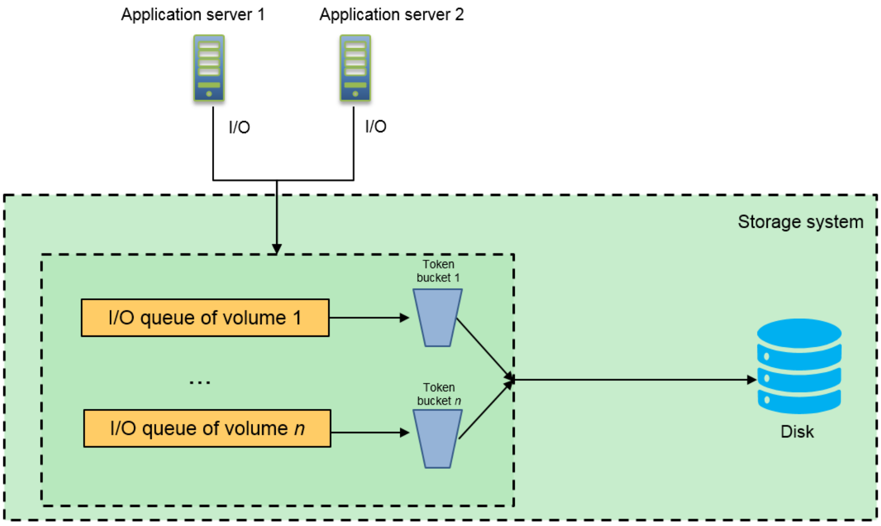


Suppose that the system is in the initial state and the IOPS upper limit of **Volume 0** is 1000, as shown in Figure 4-4. The system will detect the service pressure from **Host 0** and **Host 1** on **Volume 0**, and adaptively adjust the number of tokens in **Token bucket 0** and **Token bucket 1** to limit the maximum IOPS to 1000.

- Volume-based I/O traffic control management algorithm

  QoS traffic control management is implemented by volume I/O queue management, token allocation, and dequeuing control. After you set a performance upper limit objective for a QoS policy, the system determines the performance upper limit of each VBS node by coordinating the distributed traffic control parameters and then converts the performance upper limit into a specified number of tokens. If the traffic to be restricted is IOPS, one I/O consumes one token. If the traffic to be restricted is bandwidth, one byte consumes one token. Volume-based I/O queue management uses the token mechanism to allocate storage resources. The larger the number of tokens in the I/O queue of a volume is, the more the I/O resources allocated to the volume are.

Figure 4-5 Volume-based I/O traffic control management algorithm



As shown in Figure 4-5, I/Os from application servers first enter I/O queues of volumes. SmartQoS periodically processes I/Os waiting in the queues. It dequeues the head element in a queue, and attempts to obtain tokens from a token bucket. If the number of remaining tokens in the token bucket meets the token requirement of the head element, the system delivers the element to another module for processing and continues to process the next head element. If the number of remaining tokens in the token bucket does not meet the token requirement of the head element, the system puts the head element back in the queue and stops I/O dequeuing.

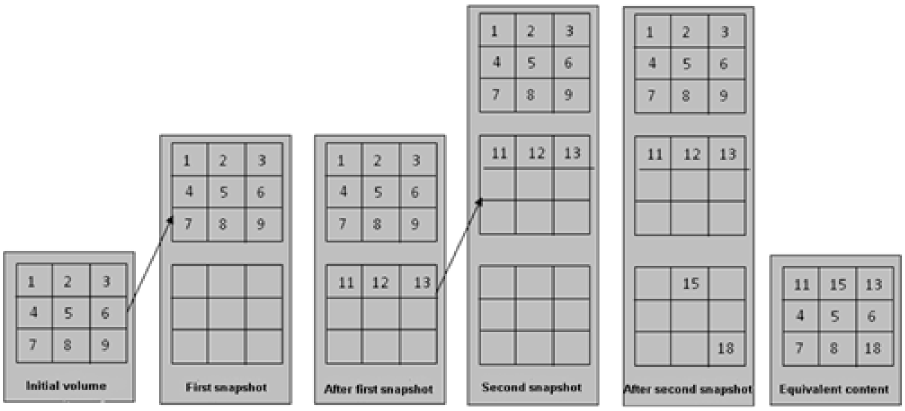
For more details about SmartQoS, see *FusionStorage 8.0.0 SmartQoS Technical White Paper*.

4.6 HyperSnap

HyperSnap is the snapshot feature of FusionStorage that captures the state of volume data at a specific point in time. The snapshots created using HyperSnap can be exported and used for restoring volume data.

FusionStorage uses the ROW mechanism to create snapshots, which imposes no adverse impact on volume performance.

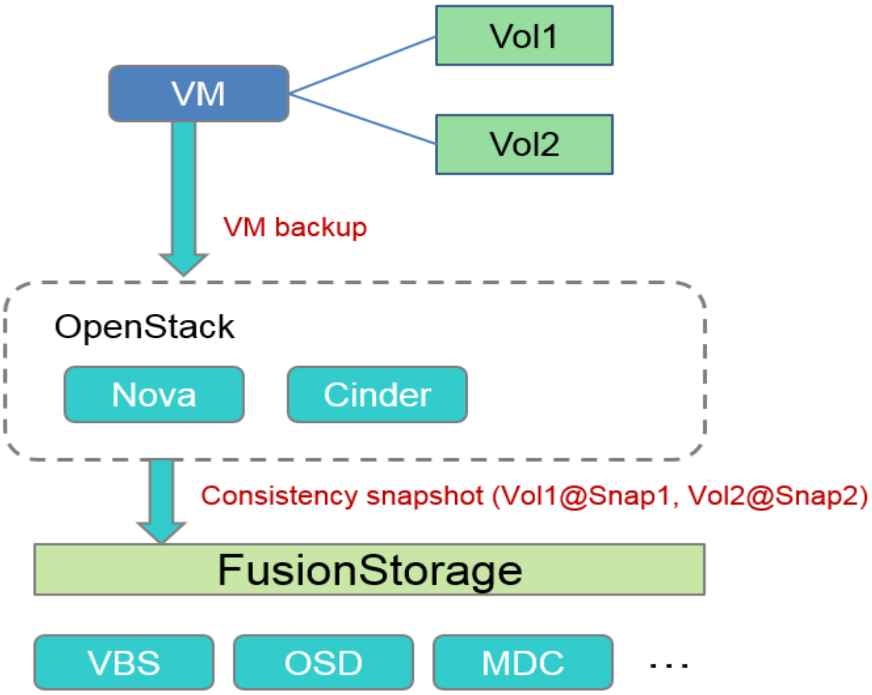
Figure 4-6 HyperSnap of FusionStorage



SCSI volumes with multiple mount points are called shared volumes. All iSCSI volumes are shared volumes. To back up shared volumes, FusionStorage supports snapshots for the shared volumes. The procedure for creating snapshots for shared volumes is the same as that for common volumes.

FusionStorage supports the consistency snapshot capability. Specifically, FusionStorage can ensure that the snapshots of multiple volumes used by an upper-layer application are at the same point of time. Consistency snapshots are used for VM backup. A VM is usually mounted with multiple volumes. When a VM is backed up, all volume snapshots must be at the same time point to ensure data restoration reliability.

Figure 4-7 Consistency snapshot function of FusionStorage



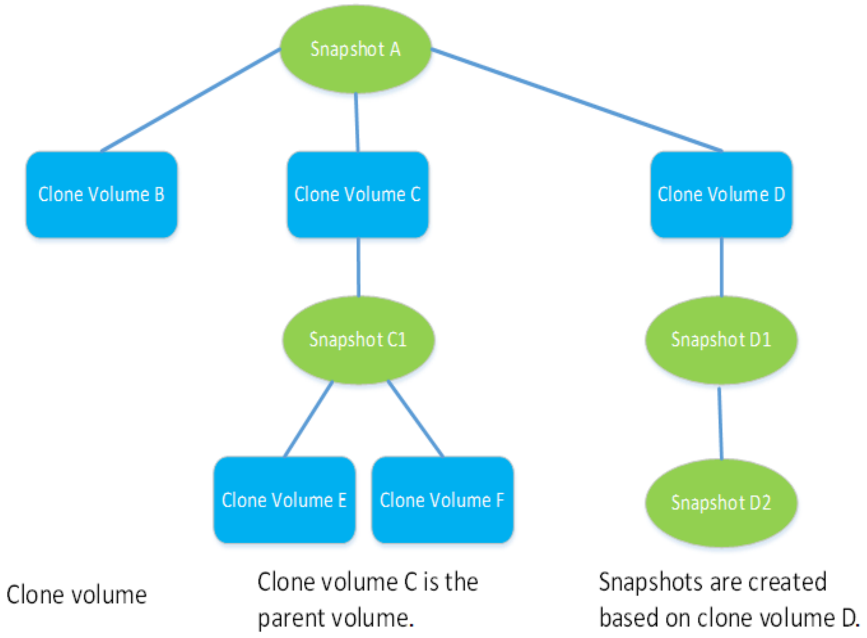
4.7 HyperClone

HyperClone is the clone feature provided by FusionStorage. FusionStorage block storage provides the linked clone function to create multiple clone volumes from one snapshot. Data on each clone volume is consistent with that of the snapshot. Data writes and reads on a clone volume have no impact on the source snapshot or other clone volumes.

FusionStorage supports a linked clone rate of 1:2048, effectively improving storage space utilization.

A clone volume has all functions of a common volume. You can create snapshots for a clone volume, use the snapshots to restore the clone volume, and clone the clone volume.

Figure 4-8 Linked clone of FusionStorage



4.8 HyperReplication

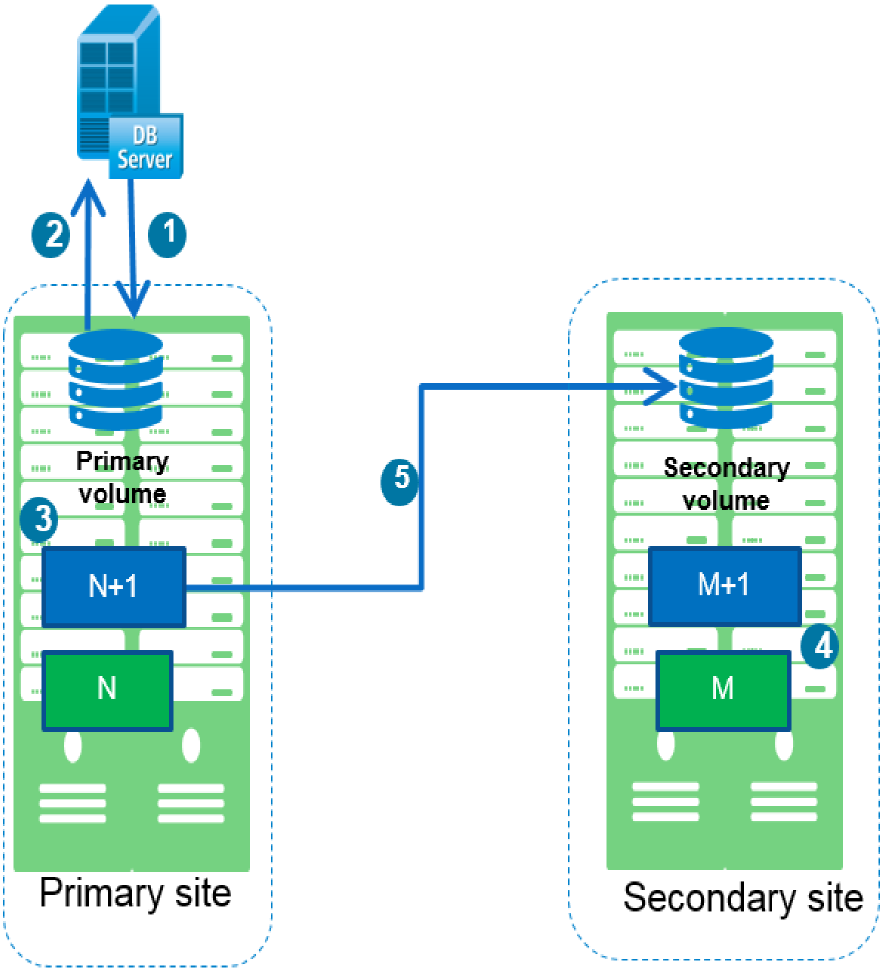
HyperReplication is the remote replication feature that periodically synchronizes differential data on primary and secondary volumes of two FusionStorage block storage systems. All the data generated on primary volumes after the last synchronization will be synchronized to the secondary volumes.

You can deploy DR clusters as required. A DR cluster provides replication services and manages DR nodes, cluster metadata, replication pairs, and replication consistency groups. DR nodes can be deployed on the same servers as storage nodes or on independent servers. DR clusters have excellent scalability. A single DR cluster contains three to 64 nodes. One FusionStorage block storage system supports a maximum of eight DR clusters. A single DR cluster supports 64000 volumes and 16000 consistency groups, meeting future DR requirements.

After an asynchronous remote replication relationship is established between a primary volume at the primary site and a secondary volume at the secondary site, initial synchronization is implemented. After initial synchronization, the data status of the secondary volume becomes consistent. Then, I/Os are processed as follows:

1. The primary volume receives a write request from a production host.
2. The system writes the data to the primary volume, and returns a write completion response to the host.
3. The system automatically synchronizes incremental data from the primary volume to the secondary volume at a user-defined interval, which ranges from 10 seconds to 1440 minutes. If the synchronization mode is manual, you need to trigger synchronization manually. Before synchronization, a snapshot is generated for the primary and secondary volumes respectively. The snapshot of the primary volume ensures that the data read from the primary volume during the synchronization remains unchanged. The snapshot of the secondary volume backs up the secondary volume's data in case that the data becomes unavailable if an exception occurs during the synchronization.
4. During synchronization, data is read from the snapshot of the primary volume and copied to the secondary volume. After synchronization, the system automatically deletes the snapshots of the primary and secondary volumes.

Figure 4-9 Asynchronous replication of FusionStorage



FusionStorage supports only asynchronous replication.

4.9 HyperMetro

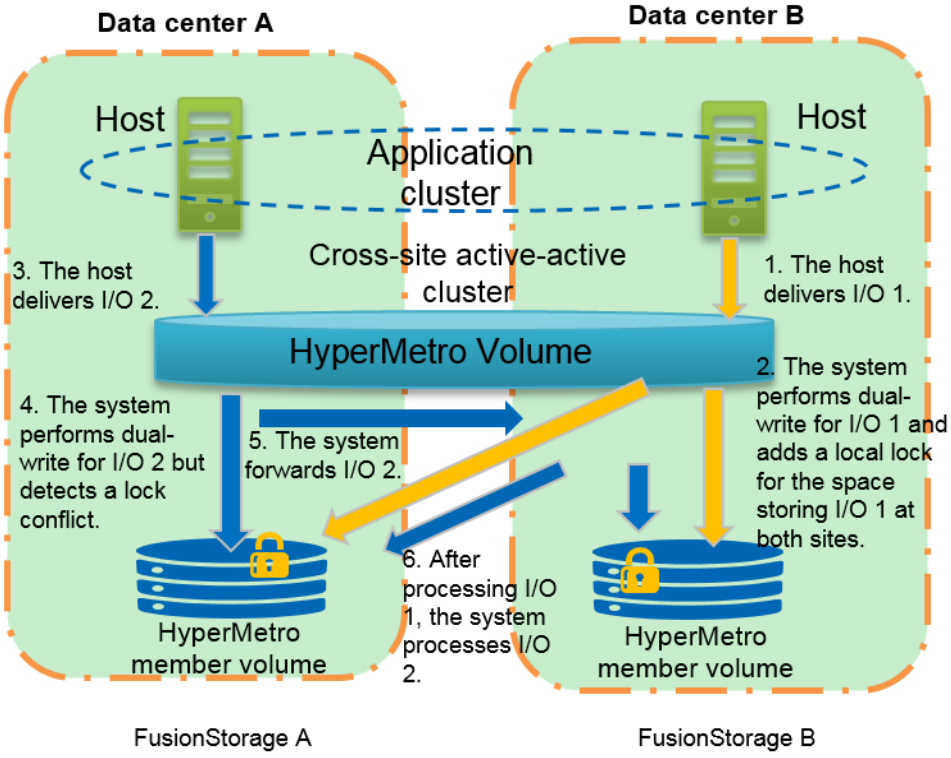
HyperMetro is the active-active storage feature that establishes active-active DR relationships between two FusionStorage block storage systems in two data centers. It provides HyperMetro volumes based on volumes of the two FusionStorage block storage systems and enables the HyperMetro volumes to be read and written by hosts in the two data centers at the same time. If one data center fails, the other automatically takes over services without data loss and service interruption.

HyperMetro supports incremental synchronization. If a site fails, the site winning arbitration continues to provide services. I/O requests change from the dual-write state to the single-write state. After the faulty site recovers, incremental data can be synchronized to it to quickly restore the system.

FusionStorage supports logical write error handling. If the system is running properly but one site fails to process a write I/O, the system will redirect the write I/O to a normal site for processing. After the fault is rectified, incremental data can be synchronized from the normal site to the one that fails to process the I/O. By doing so, upper-layer applications do not need to switch sites for I/O processing upon logical write errors.

HyperMetro supports a wide range of upper-layer applications, including Oracle RAC and VMware. It is recommended that the distance between two FusionStorage block storage systems be less than 100 km in database scenarios and be less than 300 km in VMware scenarios. For details about supported upper-layer applications, see Huawei Storage Interoperability Navigator.

Figure 4-10 HyperMetro of FusionStorage


Data center A
Data center B
Host
Host
Application cluster
Cross-site active-active cluster
3. The host delivers I/O 2.
1. The host delivers I/O 1.
HyperMetro Volume
2. The system performs dual-write for I/O 1 and adds a local lock for the space storing I/O 1 at both sites.
4. The system performs dual-write for I/O 2 but detects a lock conflict.
5. The system forwards I/O 2.
6. After processing I/O 1, the system processes I/O 2.
HyperMetro member volume
HyperMetro member volume
FusionStorage A
FusionStorage B

For more details about HyperMetro, see *FusionStorage 8.0.0 HyperMetro Technical White Paper*.

5 Elastic Scaling

FusionStorage block storage uses a fully distributed architecture and does not have centralized access components or modules, eliminating scalability bottlenecks caused by a single component or module. FusionStorage leverages DHT routing algorithms to distribute service data evenly among nodes and disks in storage pools, achieving quasi-linear elastic expansion with a linear degree of up to 90%. Multiple fault domains are divided for distributed clusters by using storage pools, and each fault domain is independent of each other. A maximum of 4096 storage nodes is supported.

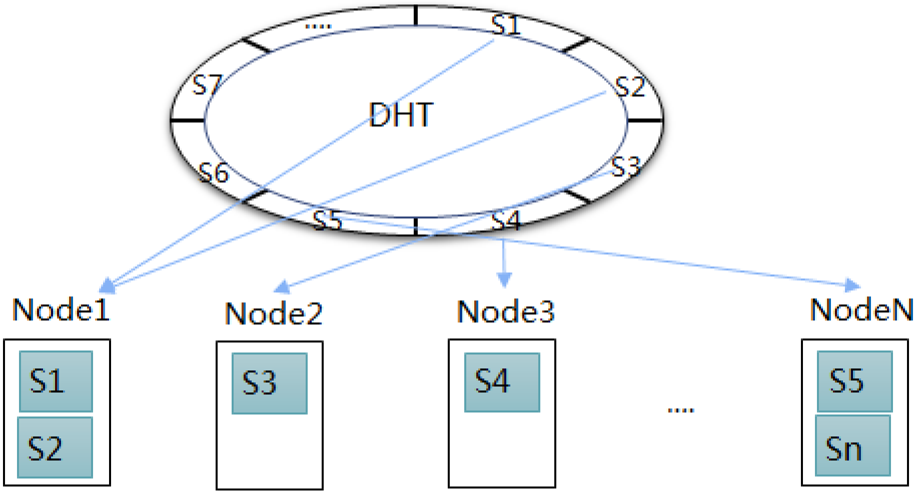
FusionStorage adopts a Share Nothing architecture. Each node processes and stores only the data allocated to it and stores only a small amount of cluster data. Therefore, the system can be linearly expanded.

5.1 DHT Routing Algorithms

Shard DHT Routing Algorithm

FusionStorage block storage uses the shard DHT routing algorithm to balance data processing. When the storage system receives an I/O request from an application, the system uses the shard DHT routing algorithm to determine the node that processes the request. Each LUN in the storage system is divided into multiple shards at the granularity of 1 MB. Each shard is calculated using the hash factor of the LUN (that is, LUN ID) and start LBA of the shard to obtain the hash value of the shard. Then, the system locates each shard in the DHT ring according to hash values. Figure 5-1 shows the implementation.

Figure 5-1 Shard DHT routing



Each storage pool has a shard DHT ring. Each shard DHT ring has 4096 shards, and the 4096 shards are evenly allocated to the nodes in a storage pool. Suppose that a storage pool has 16 nodes. The 4096 shards will be evenly allocated to the 16 nodes. Each node processes 256 shards (that is, 4096/16).

Shard DHT routing is implemented as follows:

Suppose that an upper-layer application delivers a request to write data whose LUN ID is 1, start LBA is 64 MB, and length is 2 MB. The system first divides the data into two shards: **S2** (64 MB to 65 MB) and **S3** (65 MB to 66 MB). By using the shard DHT routing algorithm, the system forwards **S2** and **S3** to **Node1** and **Node2** respectively for processing.

The shard DHT routing algorithm has the following characteristics:

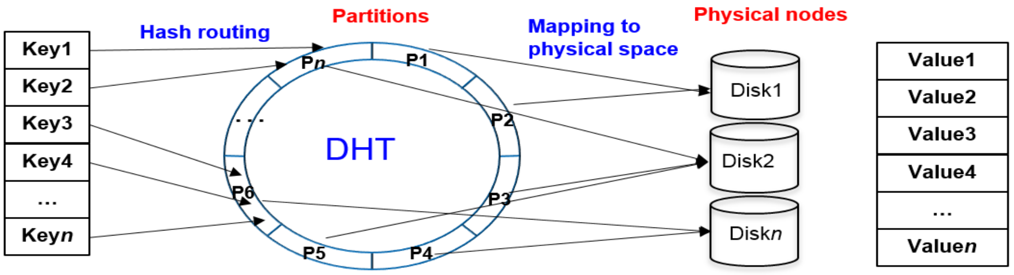
- Balance: Data is distributed to all nodes as evenly as possible, thereby balancing loads among nodes.
- Monotonicity: When new nodes are added to the system, the system redistributes data among nodes. Only a small proportion of shards will be allocated to the new nodes for processing, and the data on the existing nodes is not significantly adjusted.

STORE DHT Routing Algorithm

FusionStorage block storage uses the STORE DHT routing algorithm to balance data storage. Each storage node stores a small proportion of data, and the data is routed and stored using the STORE DHT routing algorithm.

Traditional storage systems typically employ the centralized metadata management mechanism, which allows metadata to record the disk distribution of the LUN data with different offsets. For example, the metadata may record that the first 4 KB of data in LUN1+LBA1 is distributed on LBA2 of the 32nd disk. Each I/O operation initiates a query request for the metadata service. As the system scale grows, the metadata size also increases. However, the concurrent operation capability of the system is subject to the capability of the server accommodating the metadata service. In this case, the metadata service may become a performance bottleneck of the system. Unlike traditional storage systems, FusionStorage block storage uses the STORE DHT routing algorithm for data addressing. Figure 5-2 shows the implementation.

Figure 5-2 STORE DHT routing



The DHT ring of FusionStorage block storage contains 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. The partition-disk mapping table occupies only a small space, and FusionStorage block storage nodes store the mapping table in the memory for rapid routing. FusionStorage block storage does not employ the centralized metadata management mechanism and therefore does not have performance bottlenecks incurred by the metadata service.

The STORE DHT routing algorithm has the following characteristics:

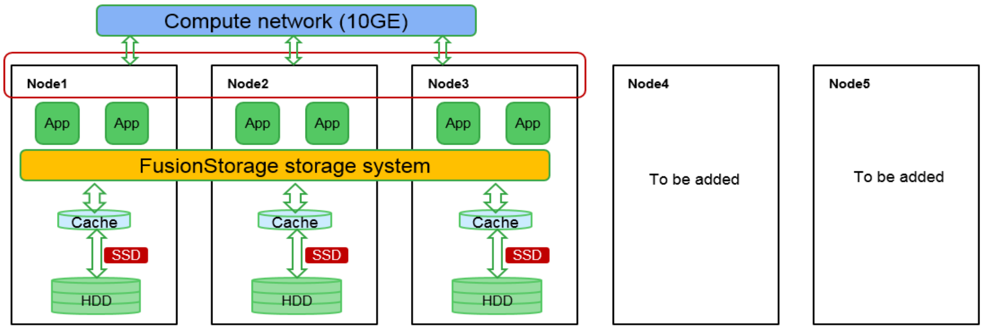
- Balance: Data is distributed to all nodes as evenly as possible, thereby balancing loads among nodes.
- Monotonicity: When new nodes are added to the system, the system redistributes data among nodes. Data migration is implemented only on the new nodes, and the data on the existing nodes is not significantly adjusted.

5.2 Smooth Expansion

FusionStorage block storage uses a distributed architecture and supports easy capacity expansion and ultra-large storage:

- After new nodes are added, the system implements fast node balancing, avoiding migration of a large amount of data.
- Disks, compute nodes, and storage nodes can be added separately or together to expand capacity.
- There is no independent storage controller. Storage access requests, storage bandwidth, and cache resources are evenly distributed on each node. The system IOPS, throughput, and cache linearly increase as nodes expand.

Figure 5-3 Capacity expansion of FusionStorage



5.3 Performance Acceleration

FusionStorage block storage organizes decentralized SSDs or HDDs into efficient SAN-like storage pools, providing higher IOPS than SAN devices and maximizing the performance.

Distributed Storage Service Component

FusionStorage block storage adopts stateless distributed software component VBS to process storage services. VBS is deployed on each node, addressing the performance bottleneck of centralized storage services. VBS on a single node occupies a small amount of CPU resources and provides higher IOPS and throughput than centralized storage controllers. Suppose that a system contains 20 nodes that need to access the storage resources provided by FusionStorage, and the bandwidth that each node provides for the storage plane is 2 x 10 Gbit/s. One VBS is deployed on each node (that is, each node has one storage controller), so that the total throughput can reach 400 Gbit/s (20 x 2 x 10 Gbit/s). With the growth of the cluster scale, storage controllers can be linearly added, eliminating the performance bottlenecks caused by centralized storage controllers in conventional dual-controller or multi-controller storage systems.

Distributed Cache

FusionStorage block storage distributes caches and bandwidth to each node.

In a FusionStorage block storage cluster, independent I/O bandwidth is allocated to disks of each node, resolving the problem of independent storage systems that a large number of disks share the limited bandwidth between compute and storage devices.

FusionStorage block storage can use some node memory as the read cache and SSDs as the write cache. Data cache resources are evenly distributed on each node. The total cache on all nodes is far greater than that provided by external storage devices. Even when using large-capacity, low-cost SATA disks, FusionStorage block storage can still provide high I/O performance.

FusionStorage block storage can use SSDs for caching data. In addition to providing the write cache, the SSDs can collect statistics on and cache hotspot data, further improving system performance.

Global Load Balancing

The 5.1 DHT Routing Algorithms adopted by FusionStorage block storage ensures that the I/O operations performed by upper-layer applications are evenly distributed on different disks of different nodes, globally balancing loads.

- The system automatically scatters data blocks of each volume onto different disks of different nodes. Frequently accessed data and rarely accessed data are evenly distributed on each node, preventing hotspots in the system.
- The data fragment distribution algorithm ensures that primary and secondary copies are evenly distributed to different nodes and different disks. In this way, each disk contains the same number of primary and secondary copies.
- If nodes are removed due to a failure or if new nodes are added, loads are automatically balanced among all nodes after system rebuilding.

SSDs for Distributed Storage

FusionStorage block storage can use SSDs to provide distributed storage for high-performance applications. SSDs deliver higher read and write performance than SATA or SAS disks.

FusionStorage can virtualize the PCIe SSD cards configured on storage nodes into a virtual storage pool to provide high-performance read and write for applications.

FusionStorage block storage supports both Huawei-developed SSD cards and mainstream PCIe SSD cards of other vendors.

High-Speed IB Networking

FusionStorage block storage supports IB networks designed for high-bandwidth and low-latency applications. IB networks provide the following benefits:

- Nodes are interconnected at a high speed using the 56 Gbit/s FDR IB networking.
- Standard multi-layer fat-tree networking achieves smooth capacity expansion.
- A communication network where congestion hardly occurs eliminates data switching bottlenecks.
- Communication latency within nanoseconds is provided to transmit compute and storage information promptly.
- Lossless network QoS ensures data integrity during transmission.
- Multi-plane communication between active and standby ports improves transmission reliability.

6 Superb Performance

FusionStorage block storage leverages the dynamic intelligent partitioning, static disk selection, and Turbo EC algorithms to implement efficient distributed storage featuring balanced service distribution and superb reliability and performance. While using HDDs as the main storage, FusionStorage adopts the distributed SSD cache acceleration solution to build SSDs on each storage node into a shared distributed cache storage pool for all services, accelerating the performance of HDDs.

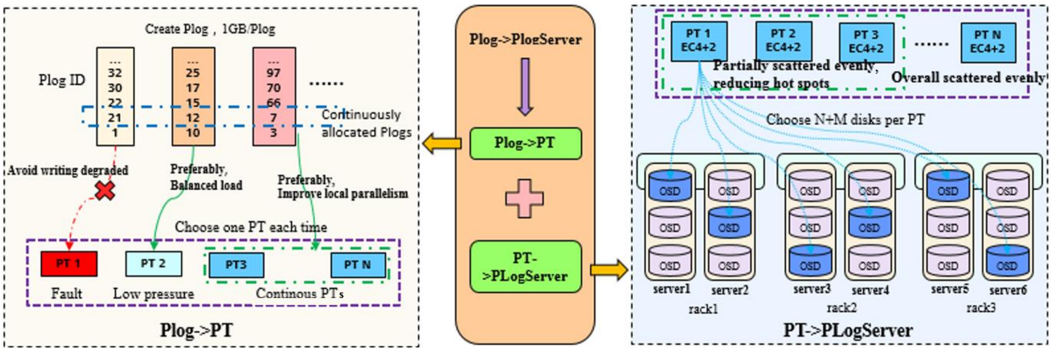
6.1 Distributed Storage Optimization Algorithms

6.1.1 Dynamic Intelligent Partitioning and Static Disk Selection Algorithms

FusionStorage block storage uses the DHT routing technology to implement smooth system capacity expansion and performance acceleration. During data persistence, FusionStorage uses two algorithms to optimize the performance and reliability of distributed storage:

1. A user creates a Plog and selects the dynamic intelligent partitioning algorithm of partitions (PTs in Figure 6-1) for the Plog.
2. Partitions select the local static disk selection algorithm of OSDs.

Figure 6-1 DHT dynamic intelligent partitioning and static disk selection algorithms



The dynamic intelligent partitioning algorithm introduces a self-adaptive negative feedback mechanism to achieve superb reliability and performance. Its major improvements and objectives are as follows:

- Write reliability is not degraded. If the partition corresponding to a Plog falls into a faulty disk, the Plog is discarded and a new Plog is selected to write data.
- Loads are balanced and hotspots are eliminated. In random access scenarios, polling or distributed hash algorithms cannot fully ensure balanced data layouts and disk access performance. For example, in some storage systems based on the CRUSH hash algorithm, the utilization difference between OSDs reaches 20%, causing continuous generation of hotspot disks. In addition, disk fault recovery, slow disk hotspots, and QoS traffic control affect system performance. Plog Manager of FusionStorage periodically collects statistics on the available capacity and I/O workloads of disks, nodes, and partitions to intelligently identify hotspot and slow disks. In actual tests, the balance difference of FusionStorage can be controlled at 5%, greatly improving the overall system performance.

The local static disk selection algorithm aims to optimize the local balancing of the mapping between a partition and an OSD. Optimization is achieved in the following scenarios:

- Data balancing performance is optimized without affecting reliability when nodes are added, reducing invalid data migration. In the algorithm, cyclic selection of partitions by partition group rather than conventional disk-based balancing is used. Intra-group balancing does not degrade reliability and greatly reduces invalid data migration. The invalid data migration ratio is only 1% to 3% of that of the CRUSH algorithm, and the actual invalid data migration ratio is only 0.05% to 0.8% of that of the CRUSH algorithm.
- In multi-copy scenarios, the balancing of primary partitions is optimized. Common algorithms only focus on the balancing of initial primary copies. Once a node or a disk becomes faulty, primary copies are selected again for many data blocks. Whether the new primary copies are evenly distributed affects the current service performance and the performance of subsequent incremental data synchronization. The existing solution is static fixed selection of primary copies, that is, selecting secondary copy 1 first and then secondary copy 2 after a primary copy becomes faulty. FusionStorage uses dynamic selection of primary copies. Specifically, the system selects primary copies based on the busy degree of the disks where secondary copies reside. This prevents some disks from becoming hotspots after a fault occurs. The improved algorithm in actual tests enhances the data balancing speed by up to 28.8%.
- In EC scenarios, the number of supported concurrent data rebuilding tasks is improved and performance is augmented. The partition grouping algorithm is used to scatter data on multiple disks. Data rebuilding is concurrently performed on multiple disks, improving the performance. After the improvement, the disk read latency decreases by up to 45% during system restoration.

6.1.2 Turbo EC Algorithm Featuring Fast Encoding and Rebuilding

Distributed storage is highly elastic and scalable, and is oriented to cloud computing storage environments with massive amounts of data. However, The multi-copy redundancy solution used by distributed storage for a long time restricts the deployment of environments with massive amounts of data in enterprises due to costs. The EC technology can improve disk usage and reduce costs. However, performance and reliability problems due to its technical complexity restricts the promotion of the technology in key business scenarios of enterprises.

This technology is only used for object storage systems and some distributed NAS systems in backup and archiving scenarios.

The core of the EC technology is to build a complete technical solution by constructing an encoding matrix and a fast encoding/decoding algorithm. The EC technology brings the following benefits for end-to-end storage systems:

- Improving the reliability and availability of storage systems
- Reducing CPU usage
- Minimizing write amplification of disk I/Os
- Lowering bandwidth consumption

The preceding benefits are brought about by using the following key features:

- Improved encoding efficiency
- Reduced read I/Os during EC array rebuilding

In actual tests, Turbo EC of FusionStorage improves the encoding efficiency by 22% and the rebuilding efficiency by 35% compared with the conventional encoding method.

6.2 SSD Cache Acceleration

Due to mechanical limitations, the performance of HDDs is basically unchanged for decades although the capacity increases greatly. The random I/O latency, ranging from several milliseconds to tens of milliseconds, severely affects user experience and system performance. Compared with HDDs, SSDs offer higher performance but also higher costs. Nowadays, SSDs are used as the system cache or tier to strike a balance between performance and costs.

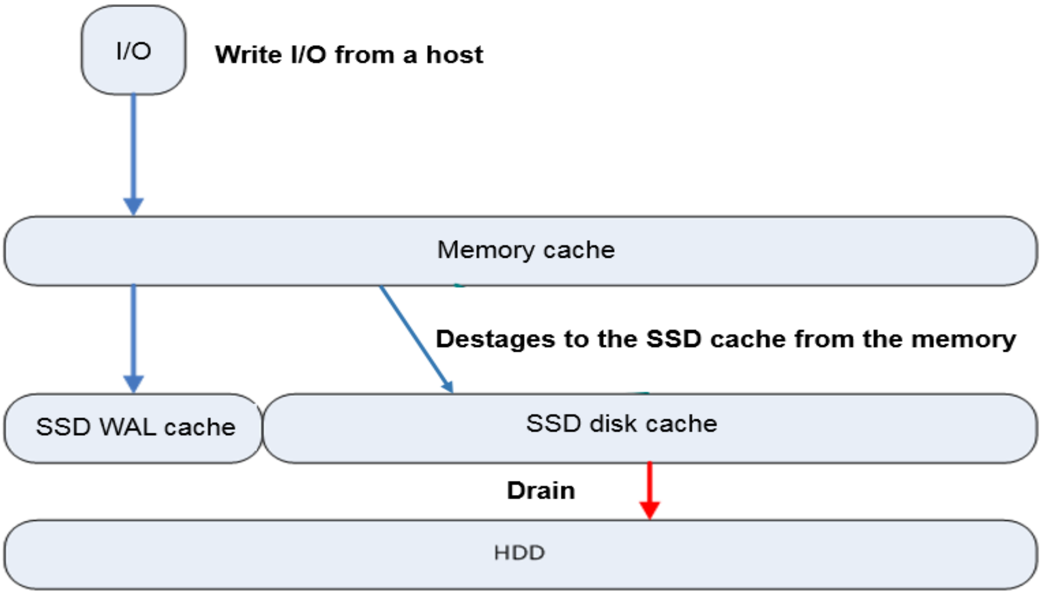
FusionStorage optimizes the I/O processing path obviously using the Append Log technology compared with previous versions.

The SSD Cache function is divided into two parts:

- WAL cache: caches write I/Os from hosts to ensure temporary persistency of data and prevent data loss upon power failures.
- Disk cache: caches data before it is written into HDDs. In this way, I/Os are not directly written to HDDs, reducing the latency.

Figure 6-2 shows the logical architecture of the distributed cache.

Figure 6-2 Logical architecture of the distributed cache



Generally, the WAL cache and disk cache are two partitions logically divided on an SSD or an SSD card.

6.2.1 Write Cache

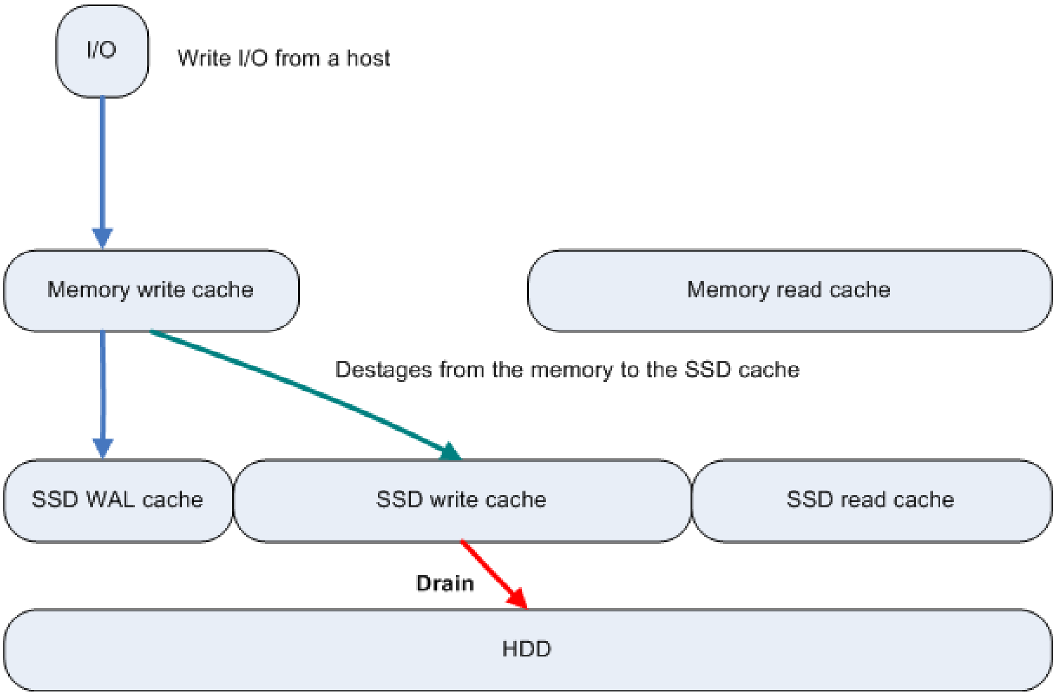
When VBS sends a write I/O (**Write I/O from a host** in Figure 6-3), the write I/O is stored in the memory write cache. In addition, the write I/O is synchronously recorded into the SSD WAL cache in the form of logs using the 2+2 EC scheme and a write success message is returned. This process is called the host write I/O process.

Generally, the SSD disk cache is divided into two parts: SSD write cache and SSD read cache. Data in the memory write cache is aggregated into full stripes and then written to the SSD write cache in copy or EC mode, and a write success message is returned. Instead of being stored in the SSD write cache, large block I/Os are directly written from the memory write cache to HDDs.

When the data storage watermark in the SSD write cache reaches threshold, the data is migrated to HDDs from the SSD write cache.

As data in the memory write cache is gradually flushed to the SSD write cache, data in the SSD WAL cache will be eliminated. Asynchronous garbage collection is usually performed on such data.

Figure 6-3 Write cache



In conventional copy modes, data is written to the SSD cache in the form of copies. Then full stripes are asynchronously read from the SSD cache and written to HDDs. Unlike the conventional copy modes, the SSD WAL cache solution of FusionStorage has the following advantages:

- The write amplification of the FusionStorage SSD WAL cache is insignificant. The overhead of the 2+2 EC scheme is 2. For the SSD cache in copy mode, the minimum overhead must be 2.
- The bandwidth consumed by FusionStorage is low.
- The FusionStorage SSD WAL cache is of high reliability and uses the redundancy protection of *N*+2.
- Flushing FusionStorage data to the main storage is usually triggered and completed by the random access memory (RAM). This is more efficient than reading data from the SSD cache and then writing the data to the main storage.

6.2.2 Read Cache

FusionStorage block storage employs a multi-level read cache mechanism. Level 1 is the memory cache and uses the least recently used (LRU) mechanism to cache data. Level 2 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 factor of data reaches a specific threshold, the system automatically fetches the data onto the SSD cache and removes data that has not been accessed for a long time from the SSD cache.

Upon receiving a read I/O sent by VBS, the EDS performs the following operations:

Step 1 The OSD checks whether required I/O data is in the memory write cache. If the data is in the memory write cache, the OSD returns the data and moves the data to the head of the LRU queue in the read cache. Otherwise, the OSD proceeds to Step 2.

Step 2 The OSD checks whether the required I/O data is in the memory read cache. If the data is in the memory read cache, the OSD returns the data and increases the hotspot access factor of the data. Otherwise, the OSD proceeds to Step 3.

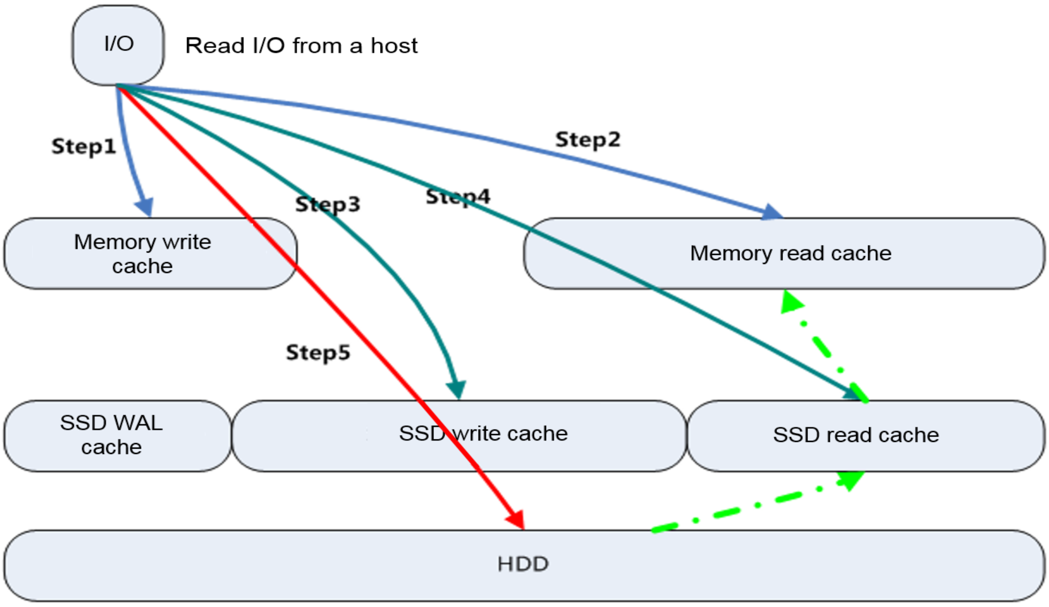
Step 3 The OSD checks whether the required I/O data is in the SSD write cache. If the data is in the SSD write cache, the OSD returns the data. Otherwise, the OSD proceeds to Step 4.

Step 4 The OSD checks whether the required I/O data is in the SSD read cache. If the data is in the SSD read cache, the OSD returns the data and increases the hotspot access factor of the data. If the hotspot access factor reaches the threshold, the OSD fetches the data to the memory read cache. If the data is not in the SSD read cache, the OSD proceeds to Step 5.

Step 5 The OSD locates the required I/O data on disks and returns the data. In addition, the OSD increases the hotspot access factor of the data. The OSD fetches the data to the SSD read cache if the hotspot access factor reaches the threshold.

----End

Figure 6-4 Read cache



6.2.3 Large IO Pass-through

Table 6-1 compares the performance of different media. Regarding random small I/Os, SSDs are superior to HDDs in performance by tens or hundreds of times. However, regarding sequential large I/Os, SSDs present no obvious advantage.

Table 6-1 Performance comparison between different media

Medium	4 KB Random Write IOPS	4 KB Random Read IOPS	1 MB Write Bandwidth	1 MB Read Bandwidth	Average Latency
SAS	180	200	150 MB/s	150 MB/s	3 to 5 ms
NL-SAS	100	100	100 MB/s	100 MB/s	7 to 8 ms

Medium	4 KB Random Write IOPS	4 KB Random Read IOPS	1 MB Write Bandwidth	1 MB Read Bandwidth	Average Latency
SATA	100	100	80 MB/s	80 MB/s	8 to 10 ms
SSD	70,000	40,000	500 MB/s	500 MB/s	< 1 ms
SSD card	600,000	800,000	2 GB/s	3 GB/s	< 1 ms

NOTE

The performance data of HDDs is obtained with the write cache disabled. The write cache of HDDs constituting a storage system must be disabled to ensure reliability.

Operating principles of HDDs provided by different vendors are similar, and their performance has slight difference within 10%.

SSDs and SSD cards differ significantly in performance. Table 6-1 merely uses one type of SSDs and one type of SSD cards as an example. The bandwidth performance of SSDs is limited by the bandwidth of the SAS/SATA ports (6 Gbit/s SATA ports are most commonly used in tests).

The above performance data shows that SSDs are significantly superior to HDDs in performance regarding small random IOPS. In large sequential I/O scenarios, SSD cards are superior than HDDs in bandwidth, but SSDs have no obvious advantage over HDDs due to bandwidth limitation of the SAS/SATA ports. One SSD may be used as the cache for multiple disks. If one SSD is used as the cache for more than five HDDs, direct operations on the HDDs provide higher performance. Nevertheless, FusionStorage enables large I/Os to directly bypass the SSD cache. Direct operations on the HDDs provide the following benefits:

- Large I/O performance is improved.
- The cache space originally occupied by large I/Os is released so that more random small I/Os can be cached, increasing the cache hit ratio of random small I/Os and improving the overall system performance.
- The number of write I/O operations is reduced and the service life of SSD cards is increased.

7 Security

As the Internet and cloud technologies develop, IT systems face increasing security threats, including conventional and emerging security threats. In terms of network security, security threats can be classified into those from external networks (such as IP attacks, software vulnerabilities, viruses, Trojan horses, SQL injection attacks, and phishing attacks) and those from internal networks (such as ARP spoofing, malicious plug-ins, unauthorized Internet access, mobile device access, and lack of application monitoring).

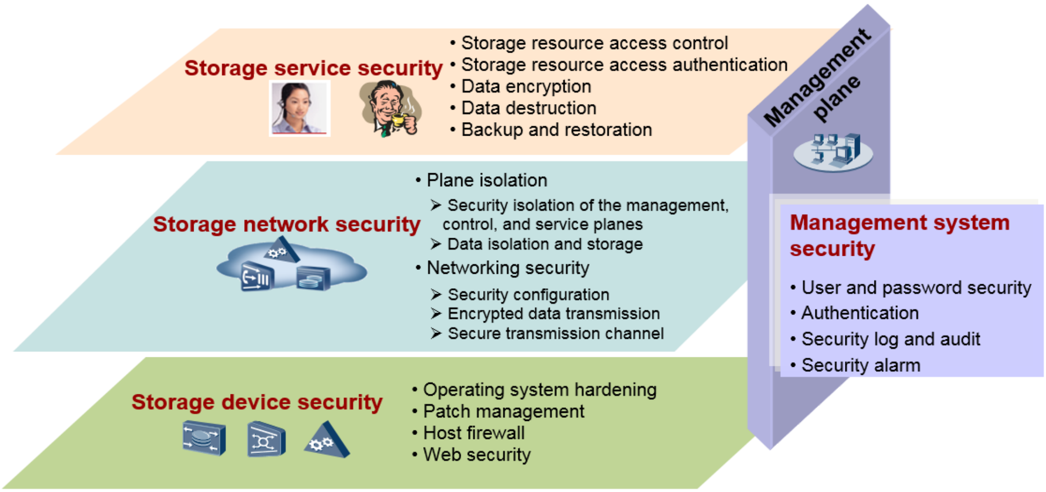
Besides the preceding threats, storage systems are subject to data leakage, data damage, and temporary or permanent loss of accessibility and availability. Technical control measures are required to monitor data integrity, confidentiality, and availability, to prevent unauthorized use of storage resources and data.

For details about FusionStorage storage system security, see *FusionStorage 8.0.0 Security Technical White Paper*.

7.1 Security Framework

To address the security threats and risks faced by storage products, Huawei provides a storage product security solution, as shown in Figure 7-1. The FusionStorage security framework ensures storage product security at multiple dimensions and protects inter-layer security based on the security design of the management plane.

Figure 7-1 Overall security framework of FusionStorage



The overall security architecture of FusionStorage is described as follows:

- Storage service security: storage resource access control, storage resource access authentication, data encryption, data destruction, and backup and restoration
- Storage network security: plane isolation and networking security
- Storage device security: operating system hardening, patch management, host firewall, and web security
- Storage management security: user and password security, authentication, security log and audit, and security alarm

7.2 Device Security

FusionStorage uses the following measures to ensure device security: operating system hardening, security patches, and web security.

Operating system security is hardened by enhancing the security of the following aspects of Huawei EulerOS Linux operating system: memory isolation, kernel parameters, broadcast response, directory permission, account passwords, and audit logs.

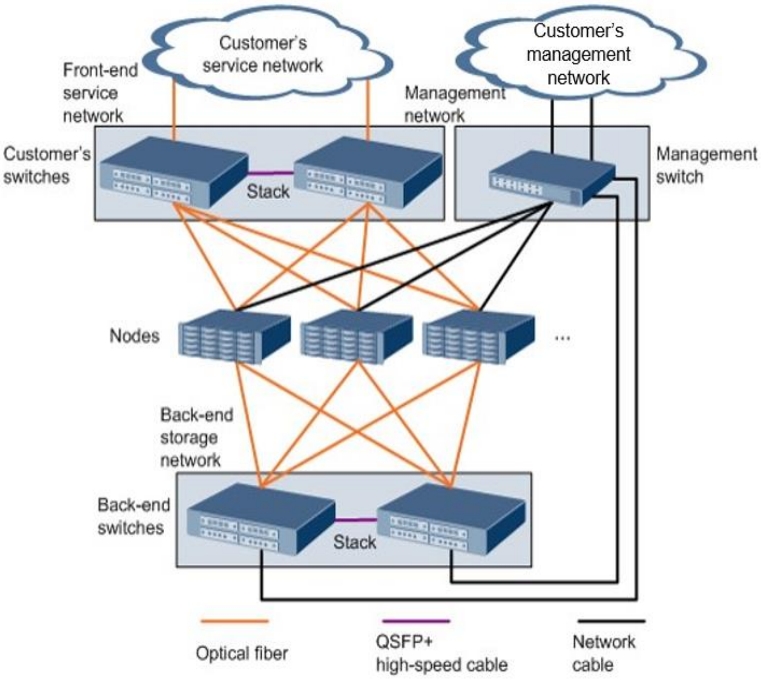
Security patches are used to repair operating system design vulnerabilities. FusionStorage periodically provides security patches for users based on the application requirements and the official release of operating system security patches and open-source software security patches.

Web security is used for software access control, including encrypted access over HTTPS, protection for sensitive information, access authentication control, and protection against cross-site scripting (XSS) attacks, SQL injection attacks, and cross-site request forgery (CSRF).

7.3 Network Security

Network security is ensured mainly by network isolation, which reduces mutual interference among the service network, storage network, and management network.

Figure 7-2 FusionStorage network isolation



As shown in Figure 7-2, FusionStorage can be divided into the front-end service network, back-end storage network, and management network based on functions.

- Front-end service network: used by clients to access FusionStorage and by FusionStorage to communicate with external devices such as domain control servers and domain name system (DNS) servers.
- Back-end storage network: used for communication between storage nodes. It is an enclosed network isolated from external networks.
- Management network: used by management personnel to maintain and manage FusionStorage clusters.

In addition, FusionStorage uses multiple secure transmission protocols, such as SSH, SFTP, and HTTPS, for remote system management.

7.4 Service Security

Service security covers access control, authentication, and antivirus. Access control is implemented mainly by the access control list (ACL) authentication mechanism. Authentication is implemented mainly by domain name encryption. Antivirus is implemented by antivirus agents and third-party antivirus software.

7.5 Management Security

Management security is a combination of user role, password combination, session expiration, logging, alarming, and auditing functions.

8 Reliability

FusionStorage adopts a distributed cluster architecture where the system is deployed in full redundancy mode, eliminating single points of failure (SPOFs). Hardware and software redundancy, network reliability, and sub-health management are designed and data protection and DR solutions are provided to offer 99.9999% system reliability. FusionStorage implements flexible security layout and redundancy of data fault domains. In addition, end-to-end data integrity protection and data protection in various fault scenarios are designed to implement high-reliability storage and service processing of data information.

8.1 Hardware Reliability

FusionStorage is designed based on general-purpose hardware. To ensure system reliability and optimal performance, it is recommended that Huawei-developed high-reliability hardware nodes oriented to distributed storage be used. FusionStorage hardware has the following features:

- Kunpeng series processors integrate CPUs, bridge chips, SAS controllers, and high-speed RoCE NICs. This reduces hardware system complexity and improves hardware reliability.
- More than 120 dedicated design items are made for issues such as the operating temperature, vibration, dust, and application read/write mode in terms of hardware reliability, structure reliability, and application reliability. This ensures high reliability of disks.
- High-reliability tests and filtering, such as tests in extreme environments (extreme temperature, extreme humidity, and extreme temperature and humidity), anti-seismic tests, anti-corrosion tests, wet dust tests, multi-condition accelerated life tests, and aging tests, have been completed for key components such as CPUs, disks, memories, and HBAs. This ensures reliable use of components and entire systems throughout their lifecycles.

- The requirements for the electrical stress, temperature, environment, and service life are subject to Huawei enterprise-level *Component Derating Specifications*. This reduces the overall system failure rate and improves the long-term system reliability.
- The redundancy design is used for system power supplies. This prevents the failure in a power supply from adversely affecting the system. Power supply modules (PSUs), also called power modules, support undervoltage and overvoltage protection of input power. When the input voltage is too low or too high, the output voltage of PSUs cannot be higher than the normal operating voltage required by next circuits (for example, cutting off the output of PSUs) to avoid abnormal next circuits and component damage due to abnormal output voltage.
- The redundancy design is used for system fans. This prevents the failure in a fan from adversely affecting the system. In addition, spare parts can be replaced online under the maximum configurations and temperature specifications.
- Hardware devices whose server platforms are mature and which have been batch delivered and run properly on live networks are used as system components. In addition, customized production processes and dedicated strict filtering are provided based on typical storage application scenarios. This ensures the quality of delivered hardware.
- Fast-Fail technology: FusionStorage monitors the sub-health status of hardware, implements anticipatory switchover of services, guarantees service continuity and performance, and ensures that latency-sensitive services are not interrupted based on the advantages of Huawei-developed hardware.
- The integrated iBMC module continuously monitors system parameters, triggers alarms, and performs recovery actions to minimize system downtime.
- Dedicated hot-swappable SAS system disks are used. RAID 1 protection is supported.

8.2 Software Reliability

8.2.1 Node Redundancy Design

FusionStorage is managed in distributed clustered mode, preventing SPOFs. When a node or a disk becomes faulty, it is automatically isolated from the cluster without affecting system services.

- ZK: provides arbitration for electing the primary MDC and stores metadata generated during system initialization. The metadata includes data routing information, such as the mapping between partitions and disks. An odd number of ZKs must be deployed in a system to form a ZK cluster. At least three ZKs must be deployed, and more than half of the deployed ZKs must be active and accessible.
- MDC: controls the status of the distributed clusters. A system must have a minimum of three MDCs. When a storage pool is added, an MDC will be automatically started or specified for the storage pool. ZK elects a primary MDC among multiple MDCs. The primary MDC monitors other MDCs. If the primary MDC detects the fault of an MDC, it restarts the MDC or specifies another MDC to manage the storage pool for the faulty MDC. When the primary MDC is faulty, a new primary MDC will be elected.
- OSD: performs I/O operations. OSDs work in active/standby mode. MDCs monitor OSD status in real time. When the active OSD where a specified partition resides is faulty, services will be automatically switched over to the standby OSD to ensure service continuity.

Thanks to the distributed architecture, FusionStorage 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 faults on the whole system.

8.2.2 Network Link Aggregation

FusionStorage uses the link aggregation technology to implement transmission link redundancy and performs link switchover (or isolation) when a link becomes faulty or sub-healthy to ensure service continuity. Link aggregation combines multiple physical links into a single logical link for applications. When a physical link becomes faulty, communication is switched to other links. Each node provides two ports for connecting to front-end network switches and two ports for connecting to back-end network switches. If a single network port or switch is faulty, the node or system is still available.

To prevent links from alternating up and down due to intermittent disconnections, packet loss, packet errors, or long latency, a link that is just recovered is suppressed by decreasing its priority. If the packet loss ratio is within the acceptable range in a specified period, the priority of the link is restored. If the packet loss ratio is unacceptable, the link remains in the degraded state until the packet loss ratio is within the acceptable range.

After a link is set to the fault state, the link will be suppressed when receiving a heartbeat packet again, that is, the path is in the normal state, but its priority is decreased. If the packet loss ratio is lower than the threshold within the specified check period, the system regards the link as usable and restores its priority. Otherwise, the system resets the check period and checks whether the packet loss ratio is lower than the threshold in the next check period.

8.2.3 Sub-health Management

Sub-health indicates that performance is severely degraded. Networks or nodes in a cluster may become sub-healthy, which affects user services. Sub-health management monitors the status of cluster resources to determine whether a resource is sub-healthy. If a resource is sub-healthy, an alarm is reported and the resource is isolated. The system isolates sub-healthy resources for which redundancy is provided to restore performance.

8.2.3.1 Disk Sub-health Management

A storage system contains a large number of disks that carry mission-critical data for users. Therefore, promptly detecting and efficiently handling disk faults is a challenge for all storage systems. To improve the fault tolerance capability of disks, FusionStorage block storage provides a series of fault tolerance designs to address this challenge.

- Intelligent scanning of bad sectors: Bad sectors are a common disk fault. A disk having bad sectors does not automatically report information about the bad sectors. Bad sectors can be detected only when data is read from or written to them. FusionStorage block storage allows users to intelligently set scanning policies without affecting services and disk reliability so that bad sectors can be detected and repaired within a short time, thereby reducing data loss risks.
- Disk fault prediction: After disks run for a long time, their components may age, and the fault rate will increase over time. FusionStorage block storage uses a disk fault model to monitor key disk indicators, and uses intelligent algorithms to evaluate disk health and predict disk faults.
- Slow disk detection and isolation: After disks run for a long time, their components may age. As a result, the disks respond slowly to I/Os, and the host applications that use the storage system freeze. FusionStorage block storage uses a disk latency model to monitor, analyze, and diagnose disk access latency in real time so as to evaluate whether a disk is

a slow disk and automatically isolate slow disks. This reduces the risks of service freezing or interruption caused by slow disks.

- Slow I/O processing: When an exception occurs on a disk or a link, the latency of certain I/Os on the disk may be abnormal. As a result, the response time of host I/Os is affected and host service performance fluctuates. FusionStorage block storage monitors the response time of each I/O. If the response time exceeds the preset threshold, FusionStorage block storage obtains data in other copies or in EC degraded read mode within a period of time to quickly respond to hosts.
- Comprehensive disk error handling: During I/O processing, FusionStorage block storage proactively identifies disk write protection (WP), command abortion (ABRT), disk faults, and triggers fault alarms and automatic data rebuilding to reduce data loss risks.

8.2.3.2 Network Sub-health Management

Cluster networks become sub-healthy when network performance deteriorates due to decreased NIC speed or increased packet loss rate or packet error rate. The system detects the changes in network resource status to locate the nodes affected by network sub-health, and performs active/standby switchover for bond network ports or isolates the nodes. The following management mechanisms are implemented for network sub-health:

- Multi-level detection: The local network of a node quickly detects exceptions such as intermittent disconnections, packet errors, and negotiated rates. In addition, nodes are intelligently selected to send detection packets in an adaptive manner to identify link latency exceptions and packet loss.
- Smart diagnosis: Smart diagnosis is performed on network ports, NICs, and links based on networking models and error messages.
- Level-by-level isolation and warning: Network ports, links, and nodes are isolated based on the diagnosis results and alarms are reported.

8.2.3.3 Service Sub-health Management

It is common that software or hardware faults occur during the running of the nodes in a distributed cluster. These faults render the nodes sub-healthy, causing problems such as access rate decrease due to repeated memory error correction and CPU underclocking. In this case, the system service latency increases. FusionStorage locates the sub-healthy nodes by collecting latency statistics and isolates problematic nodes or resources.

- Cross-process/service detection: The system collects statistics about the I/O latency of cross-process/service access. If the latency exceeds the threshold, an alarm is reported and comprehensive diagnosis is performed.
- Smart diagnosis: FusionStorage diagnoses processes or services with abnormal latency using the majority voting or clustering algorithm based on the reported abnormal I/O latency of each process or service.
- Isolation and warning: Abnormal processes or services are isolated (by allocating services to other processes in the cluster) and alarms are reported.

8.2.3.4 Fast-Fail

FusionStorage provides dedicated fast retry of path switching, that is, the Fast-Fail feature, to ensure that the I/O latency of a single sub-healthy node is controllable. The following mechanisms are used to implement the feature:

- I/O-level latency detection: checks whether the response time of each I/O exceeds the threshold and whether a response is returned for the I/O. If no response is returned, Fast-Fail is started.

- Fast-Fail: For read I/Os, other copies are read or degraded read is performed. For write I/Os, space is allocated to other disks.

8.3 Data Reliability

8.3.1 Data Protection

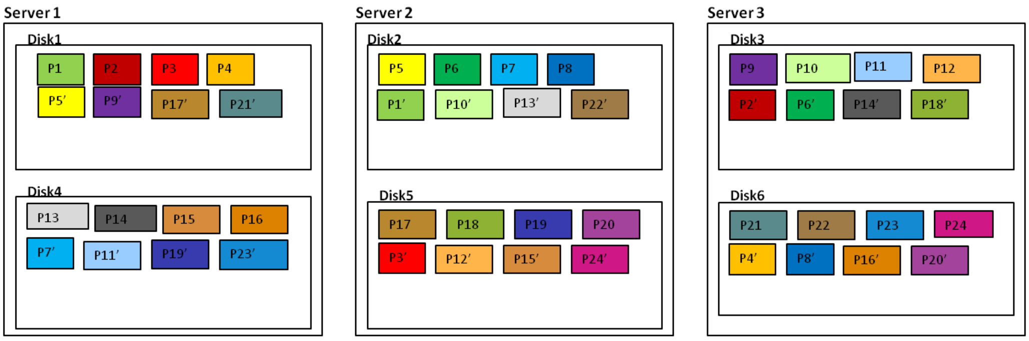
FusionStorage uses multi-copy, erasure coding (EC), and multi-level fault domains to ensure data availability. Multi-level hardware fault domains enable users to distribute data and copies in different hardware fault domains. Hardware faults in a single fault domain do not interrupt system services.

8.3.1.1 Multi-Copy

FusionStorage block storage uses the multi-copy backup mechanism to ensure data reliability. That is, one piece of data can be replicated and saved as two or three copies. Each volume in the system is fragmented based on 1 MB by default. The fragmented data is stored on nodes based on the DHT routing algorithm.

Figure 8-1 shows an example of multiple data copies. For data block P1 on Disk1 of Server1, its data backup is P1' on Disk2 of Server2. P1 and P1' are two copies of the same data block. If Disk1 becomes faulty, P1' can take the place of P1 to provide storage services.

Figure 8-1 Multi-copy redundant storage

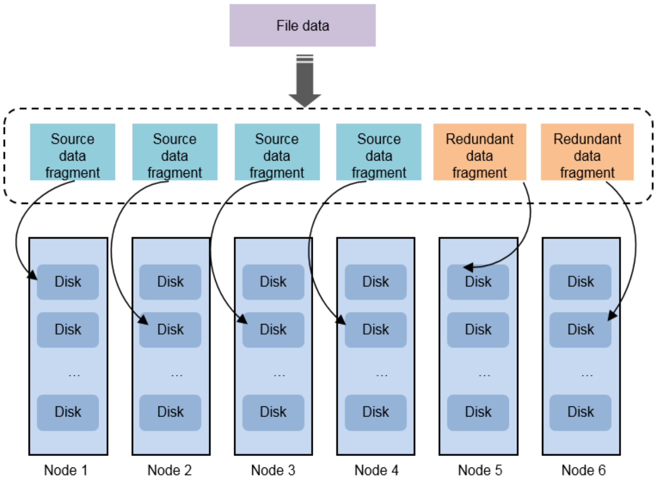


8.3.1.2 EC

FusionStorage block storage can also use erasure coding (EC) to ensure data reliability. Compared with the three-copy mode, EC improves disk utilization as well as storage reliability.

The EC-based data protection technology is based on distributed and inter-node redundancy. FusionStorage block storage uses Huawei-developed Low Density Erasure Coding (LDEC) algorithm. It is an MDS array code based on the XOR and Galois field multiplication. The algorithm has the minimum granularity of 512 bytes and supports CPU instruction acceleration and various mainstream EC schemes. Data written into FusionStorage is divided into *N* data fragments, and *M* parity fragments are generated for the *N* data fragments using EC. The *N*+*M* fragments are stored on *N*+*M* nodes, as shown in Figure 8-2.

Figure 8-2 EC



This figure illustrates how to store four data fragments and two redundant data fragments on six nodes.

Turbo EC is an enhanced data redundancy protection mechanism widely used in distributed storage. Data written into FusionStorage is divided into *N* data fragments, and *M* parity fragments are generated for the *N* data fragments using EC. If *M* fragments are damaged in an EC group, the system implements data recovery from the *N* fragments.

Compared with the multi-copy mechanism, EC improves disk utilization as well as storage reliability, thereby reducing costs. For example, a 4 MB I/O occupies 12 MB disk space in three-copy storage mode. However, if the 4+2 EC scheme is used, only 1 MB is required on each of the four data nodes and each of the two parity nodes occupies 1 MB space. That is, only 6 MB space is required in total. While providing the same reliability, EC saves 6 MB disk space compared with the three-copy storage mode.

When nodes are faulty and the number of normal nodes does not meet the minimum quantity required by an EC scheme, users are allowed to reduce data fragments to ensure that the reliability does not deteriorate. The reduction rule is ($N/2 + M$). For example, EC scheme 4+2 is shrunk to 2+2, EC scheme 8+2 to 4+2, and EC scheme 10+2 to 4+2. Note: If $N/2$ is an odd number, ($N/2 – 1$) is used as *N* in the EC scheme after reduction.

The performance of the EC mode is usually about 15% higher than that of the multi-copy mode. FusionStorage supports EC schemes up to 22+2, 20+3, and 20+4.

The EC data protection mode provided by FusionStorage block storage achieves high reliability similar to that provided by the conventional RAID mode based on data replication among multiple nodes. Furthermore, the EC data protection mode maintains high disk utilization of $N/(N + M)$. Different from the conventional RAID mode where independent hot spare disks are allocated in advance, FusionStorage enables any available space in the system to function as the hot spare space, thereby further improving the storage utilization.

8.3.1.3 Multi-Level Fault Domains

FusionStorage uses multi-level fault domains and multi-level security design, and supports flexible data layout policies. Node- and cabinet-level security can be implemented in an equipment room.

- Node-level security

In the multi-copy mode, different copies are distributed on different nodes. For example, if three copies are configured for a storage pool consisting of eight nodes, the storage pool can still provide services even when two nodes become faulty.

In the EC mode, different fragments are distributed on different nodes. For example, if EC scheme 4+2 is configured for a storage pool consisting of eight nodes, the storage pool can still provide services even when two nodes become faulty.

- Cabinet-level security

  In the multi-copy mode, different copies are distributed on nodes in different cabinets. For example, if three copies are configured for a storage pool whose nodes reside in eight cabinets, the storage pool can still provide services even when two cabinets become faulty.

  In the EC mode, different fragments are distributed on nodes in different cabinets. For example, if EC scheme 4+2 is configured for a storage pool whose nodes reside in eight cabinets, the storage pool can still provide services even when two cabinets become faulty.

8.3.1.4 Power Failure Protection

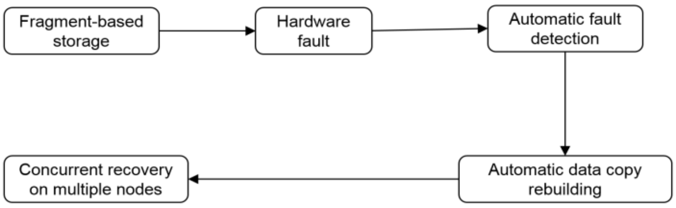
Nodes may be powered off while the system is running. FusionStorage uses non-volatile media to prevent metadata and cached data from being lost.

FusionStorage allows users to use SSDs as non-volatile media. During program running, metadata and cached data are written into non-volatile media. When a node is powered off and then restarted, the system will automatically restore metadata and cached data from the non-volatile media.

8.3.1.5 Fast Data Rebuilding

Each disk in the FusionStorage system stores multiple data blocks, whose copies are scattered on the nodes (except the nodes where the data blocks reside) in the system based on certain distribution rules. When detecting a disk or node fault, FusionStorage automatically recovers data in the background. Because data copies are scattered on different storage nodes, data rebuilding is performed on different nodes at the same time and each node has only a small amount of data to be rebuilt. This mechanism prevents performance deterioration caused by the rebuilding of a large amount of data on a single node, and minimizes the adverse impact on upper-layer services. Figure 8-3 shows the automatic data rebuilding process.

Figure 8-3 Automatic data rebuilding process


Fragment-based storage
Hardware fault
Automatic fault detection
Automatic data copy rebuilding
Concurrent recovery on multiple nodes

FusionStorage supports concurrent and fast troubleshooting and data rebuilding.

- Data blocks and their copies are distributed in an entire resource pool. If a disk fails, its data can be automatically and concurrently rebuilt across the whole resource pool.
- Data can be distributed across nodes. The failure of a single node does not affect data accessibility and rebuilding.

Loads can be automatically balanced in the event of faults or capacity expansion. Larger capacity and higher performance can be achieved without modifying application configurations.

8.3.2 Data Consistency

FusionStorage uses multiple technologies to ensure data consistency and integrity, including the strong-consistency replication protocol, read repair technology, and data integrity protection technology.

8.3.2.1 Strong-Consistency Replication Protocol

FusionStorage uses the strong-consistency replication protocol to ensure the consistency of multiple data copies. Only after all copies are successfully written to disks, the system prompts for the data write success. In most cases, FusionStorage ensures that data read from any copy is the same. If the disk where a copy resides is faulty temporarily, FusionStorage does not write data into the copy. FusionStorage restores data in the copy after the disk recovers. If a disk keeps faulty permanently or for a long time, FusionStorage removes it from the cluster and finds another available disk for the copy. Using the data rebuilding mechanism, FusionStorage can evenly distribute data among all disks.

8.3.2.2 Read Repair

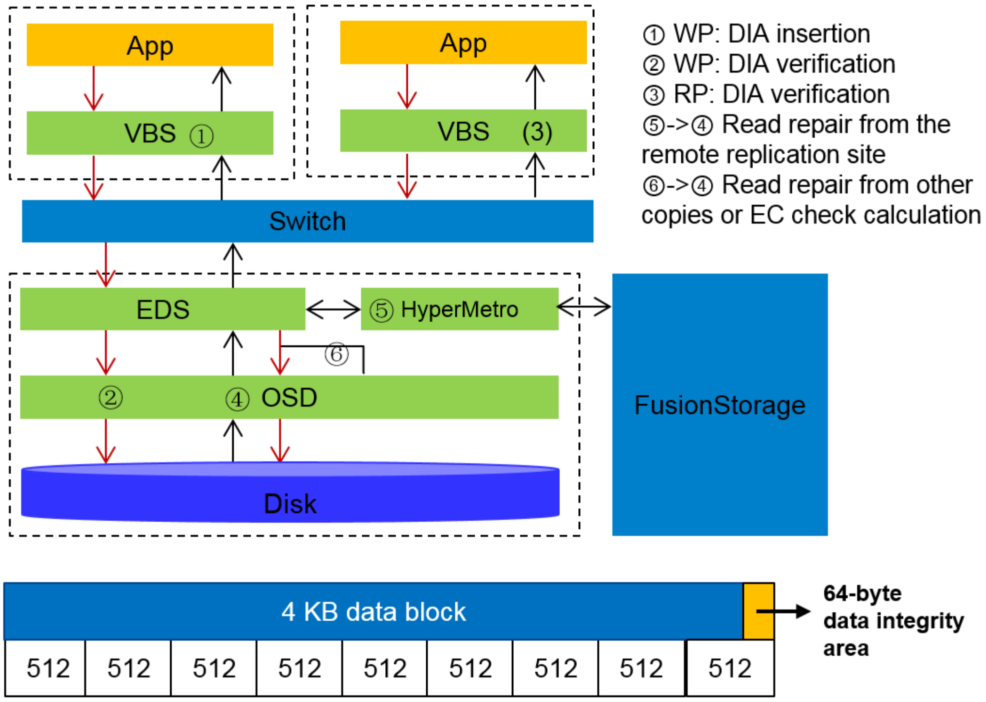
When a data read operation fails, the system determines the failure type. If the read failure occurs on a disk sector, the system automatically reads data from the copies stored on other nodes and writes the data to the faulty node of the disk sector. This ensures that the total number of data copies keeps unchanged and data consistency between copies.

8.3.2.3 Data Integrity Protection

FusionStorage block storage resolves the silent data corruption problem using mechanisms including I/O-level real-time end-to-end data integrity check, background periodic data check, and real-time self-healing and error correction of corrupted data.

FusionStorage provides an I/O-level real-time end-to-end data integrity protection solution, which can detect various silent data corruption scenarios, such as bit changes and incorrect positions of read/write data. When detecting silent data corruption, the system performs self-healing and error correction on data in real time to prevent data corruption from spreading. Figure 8-4 shows the error detection positions of silent data on the I/O path. The CRC32 algorithm is used to protect 4 KB data blocks. In addition, host LBA verification and disk LBA verification are supported.

Figure 8-4 FusionStorage data check



The silent data may be corrupted due to the component aging, electromagnetic interference, signal interference, or process defects of storage media. Periodic data check helps identify risks in advance and handle them, which prevents data loss caused by accumulation of silent data corruption.

No matter whether silent data corruption is detected on host I/Os or background periodic I/Os, automatic user-unaware error correction and self-healing for corrupted data will be triggered. Redundant data in storage media on other nodes in the system can be used for error correction and self-healing. Alternatively, redundant data in storage media on the nodes of HyperMetro remote systems can also be used for error correction and self-healing.

In addition, the system provides background periodic data check. The system periodically reads data from storage media and checks whether silent data corruption occurs. If silent data corruption occurs, a background repair process is triggered to perform error correction and self-healing. The background repair process can automatically adapt to the host service load. When the host service load is heavy, the process runs at a low speed. When the host service load is light, the process runs at a high speed.

8.4 Solution Reliability

8.4.1 Local Data Protection

FusionStorage uses local data protection to implement data backup in a cluster at a site to ensure service continuity. Local data backup is implemented using storage snapshots. A snapshot is an available copy of the specified data set. The copy contains the static image of the source data at the point in time when the source data is copied. A snapshot serves as a data backup and is accessible to hosts. The advantages of FusionStorage snapshots are as follows:

- Snapshots and their source data share the same storage space, without the need to plan exclusive storage space for snapshots.

- You can use a snapshot to create multiple data copies that are independent of each other and accessible to application servers for various purposes, such as data testing, archiving, and analysis. In this way, source data is protected, while backup data serves new purposes, meeting enterprises' various data needs.

8.4.2 Service Continuity Protection

FusionStorage enables users to build a DR solution based on distributed active-active (HyperMetro) or asynchronous replication (HyperReplication), achieving high availability of 99.9999% and ensuring service continuity.

8.4.2.1 HyperMetro

FusionStorage provides active-active feature HyperMetro. Two data centers back up each other and both are carrying services. In the event of a device fault or data center failure, the other functioning data center automatically takes over services. This ensures robust reliability, enhanced service continuity, and higher storage resource utilization.

HyperMetro has the following characteristics:

- Two HyperMetro-enabled FusionStorage clusters provide active-active read/write access capabilities. If any cluster fails, the system automatically switches services to the other cluster without data loss. Service continuity is guaranteed with zero recovery point objective (RPO) and near zero recovery time objective (RTO).
- A cross-site virtual volume is created from the volumes of two storage clusters. The data on the virtual volume is synchronized between the storage clusters in real time. The two storage clusters simultaneously process the read and write I/O requests from compute nodes.
- HyperMetro provides elastic scalability. For large-scale application systems, each storage cluster can have multiple nodes to share the load of data synchronization, meeting business growth requirements.
- HyperMetro arbitration can be implemented by setting static priorities or using a dedicated quorum server. If the quorum server becomes faulty, the system automatically switches to static priority arbitration to guarantee service continuity.

For more details about HyperMetro, see *FusionStorage 8.0.0 HyperMetro Technical White Paper*.

8.4.2.2 HyperReplication

FusionStorage offers asynchronous replication feature HyperReplication. It provides flexible and powerful data replication functions to achieve data backup and restoration, continuous support for service data, and disaster recovery (DR).

HyperReplication implements asynchronous remote replication to periodically synchronize data between the primary and secondary storage systems to support system DR. This minimizes service performance deterioration caused by the latency of long-distance data transmission.

HyperReplication has the following characteristics:

- Elastic scalability: HyperReplication can provide asynchronous replication services. Replication clusters are deployed only when asynchronous replication services need to be enabled. Replication clusters provide linear scalability for the performance and HyperReplication pairs. When the performance or HyperReplication pairs provided by a replication cluster cannot meet service requirements due to service scale growth, nodes can be added to the replication cluster. A single replication cluster can manage a maximum of 64 nodes.

- Multiple synchronization modes: HyperReplication supports four data synchronization modes, including manual synchronization, timed wait when synchronization begins, timed wait when synchronization ends, and synchronization by specifying a time policy.
- Primary/secondary switchover: HyperReplication allows users to perform primary/secondary switchover to meet service takeover requirements when a fault occurs.
- Replication consistency group: HyperReplication provides the consistency group function to ensure data consistency among multiple remote replication volumes. When a consistency group is split, data on the secondary site can still maintain time consistency of data to ensure the integrity and availability of the data used for backup and DR.

For details about the scenarios supported by HyperReplication, see *FusionStorage 8.0.0 HyperReplication Technical White Paper*.

9 Openness and Compatibility

FusionStorage supports various interfaces, such as SCSI, iSCSI, NFS, CIFS, NDMP, FTP, HDFS, S3, and Swift. In addition, FusionStorage is compatible with VMware, OpenStack, and Hadoop platforms, and can be smoothly integrated with cloud data centers and application platforms.

9.1 Compatibility with Storage Protocols

All storage services provided by FusionStorage comply with mainstream protocols and standards in the industry.

Block storage interfaces comply with the standard SCSI and iSCSI protocols and provide distributed block storage services.

9.2 Compatibility with Virtualization Platforms

FusionStorage provides distributed block storage services, supports standard SCSI and iSCSI storage interfaces, and can interconnect with mainstream virtualization platforms, including VMware, Xen, KVM, Hyper-V, and Huawei FusionSphere. Huawei also provides detailed best practices for VMware and Hyper-V.

For details about supported virtualization platforms, visit the following website of Huawei Storage Interoperability Navigator:

http://support-open.huawei.com/en/pages/user/compatibility/support-matrix.jsf

9.3 Compatibility with Cloud Management Platforms

9.3.1 OpenStack Cloud Management Platform

OpenStack aims to provide open-source software to facilitate the construction and management of public and private clouds. It is an open cloud management platform in the industry, and covers computing, storage, networks, and the like. OpenStack uses the Cinder, Manila, and Swift modules to manage block, file, and object storage services respectively.

FusionStorage provides distributed block storage services, provides a standard Cinder driver for the interconnection with OpenStack, and uses RESTful APIs to manage storage resources. Huawei has released providers for each OpenStack storage module to interconnect with mainstream release versions and commercial versions of OpenStack. FusionStorage is compatible with OpenStack R and will be compatible with later OpenStack versions.

9.3.2 Non-OpenStack Cloud Management Platforms

FusionStorage uses open RESTful APIs to integrate with the non-OpenStack cloud management platforms. Upper-layer cloud management platforms can use RESTful APIs to monitor and manage storage system resources, such as storage clusters, storage pools, and volumes.

All the APIs comply with HTTP 1.1 (RFC 2616) and use the RESTful architectural style. Each request contains the POST and GET methods of HTTP. Each response to a request contains the standard HTTP status code. Major management APIs are as follows:

- Storage pool management APIs: monitor storage pool information, including the capacity (total capacity, allocated capacity, and used capacity), status, security level, and redundancy policy.
- Volume management APIs: manage storage volumes, including creating volumes, mapping volumes to hosts, mounting volumes, deleting volumes, expanding volume capacity, unmounting volumes, and querying volumes.
- Snapshot management APIs: manage storage snapshots, including creating snapshots, deleting snapshots, querying snapshots, creating volumes using snapshots, and creating consistency snapshots for multiple volumes.
- System user management APIs: manage system users, including creating accounts, deleting accounts, and changing or resetting account passwords.
- Performance monitoring APIs: monitor performance of storage pools, hosts, and disks, such as storage space utilization, read/write IOPS, read/write bandwidth, and latency.
- Installation and deployment management APIs: provide automatic deployment, including adding hosts, querying hosts, and installing software.
- Operation log query APIs: query operation logs by operation name, operation result, or operation time.

9.4 Compatibility with Integrated Network Management Platforms

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.

FusionStorage supports the standard SNMP v2 and v3 management protocols, and provides open RESTful APIs to implement out-of-band centralized access management and unified maintenance.

9.5 Compatibility with Software

9.5.1 Operating Systems

In addition to providing storage services for virtualization platforms, FusionStorage provides SCSI storage services by deploying FusionStorage storage drivers in operating systems of physical servers.

For mainstream operating systems, FusionStorage provides complete compatibility verification and releases compatibility certification descriptions. The compatibility will be updated when each version is released. For details about supported operating systems, visit the following website of Huawei Storage Interoperability Navigator:

http://support-open.huawei.com/en/pages/user/compatibility/support-matrix.jsf

FusionStorage also supports operating systems that are not included in the compatibility list on the website. However, compatibility certification tests are required.

9.5.2 Databases

FusionStorage supports mainstream databases, including Oracle, IBM Db2, Sybase IQ, and Dameng. Huawei also provides active-active best practices of block storage services for Oracle RAC.

For details about supported databases, visit the following website of Huawei Storage Interoperability Navigator:

http://support-open.huawei.com/en/pages/user/compatibility/support-matrix.jsf

9.5.3 Big Data Applications

FusionStorage provides HDFS plug-ins to interconnect with mainstream big data application platforms of Huawei and other vendors, and provides standard NFS and NAS interfaces to access and analyze structured and semi-structured data.

For details about supported big data applications, visit the following website of Huawei Storage Interoperability Navigator:

http://support-open.huawei.com/en/pages/user/compatibility/support-matrix.jsf

9.6 Compatibility with Hardware

FusionStorage is a converged, elastic, and open distributed storage system that is compatible with mainstream servers, storage media, and I/O board cards (including NICs and RAID controller cards on servers).

For details about supported hardware, visit the following website of Huawei Storage Interoperability Navigator:

http://support-open.huawei.com/en/pages/user/compatibility/support-matrix.jsf

9.6.1 Servers

FusionStorage supports x86 servers and Huawei-developed TaiShan servers, and is compatible with x86-based servers from mainstream vendors.

FusionStorage also supports servers that are not included in the compatibility list on the website of Huawei Storage Interoperability Navigator. However, compatibility certification tests are required.

9.6.2 Storage Media

FusionStorage supports various main storage media, including SAS disks, NL-SAS disks, SATA disks, SATA SSDs, SAS SSDs, NVMe SSDs, PCIe SSD cards (not based on the NVMe protocol), and NVMe SSD cards.

FusionStorage also supports various cache media, including SSDs, SSD cards, PCIe SSD cards (not based on the NVMe protocol), and NVMe SSD cards.

FusionStorage provides plug-ins to support new types of PCIe SSD cards. PCIe SSD cards are compatible with FusionStorage if a proper plug-in is installed. The plug-ins define the mappings between PCIe numbers and ESNs of the PCIe SSD cards, and therefore allow different PCIe SSD cards to be easily supported by FusionStorage and allow the system to detect PCIe SSD card faults, such as capacitor faults and wear-out issues.

9.6.3 I/O Board Cards

FusionStorage provides plug-ins to support new types of RAID controller cards. Servers using different RAID controller cards are compatible with FusionStorage if a proper plug-in is installed. The plug-ins define the mappings among the PHY numbers of RAID controller cards, disk slot numbers, and disk equipment serial numbers (ENSs), and deal with special non-standard I/O error codes of some RAID controller cards.

FusionStorage allows all TCP/IP-based Ethernet NICs (including optical ports and electrical ports) to work on storage planes. It is recommended that NICs on the storage planes of different servers be the same to ensure compatibility among NICs.

10 Ever-New Storage

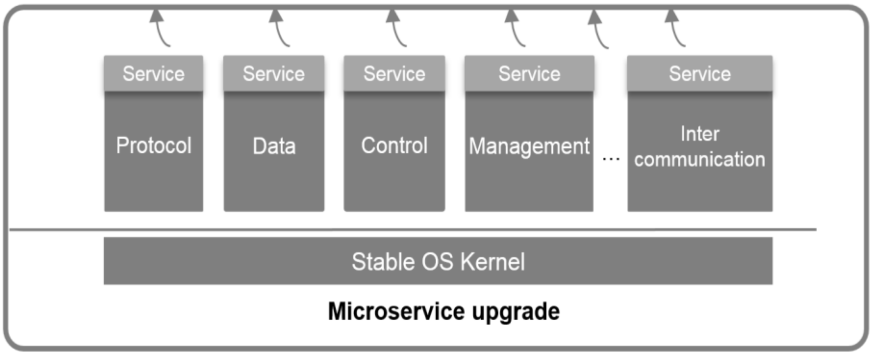
FusionStorage adopts a cloud-oriented design and a distributed architecture to provide long-term reliable storage services. During the upgrade of software and hardware platforms, the system can be continuously updated and provide stable storage services to achieve ever-new storage.

10.1 Storage Service Update

The modular design of FusionStorage enables storage services to be upgraded or updated by upgrading or updating microservices.

- Storage service upgrade: Old microservices are deactivated and new microservices are activated to upgrade storage services, for example, to optimize performance and enhance reliability.

Figure 10-1 FusionStorage microservice upgrade



10.2 System Rolling Update

The distributed architecture of FusionStorage offers high scalability to meet requirements for deploying large-scale clusters. During a long period of running, the storage system may require upgrade due to various reasons, such as introducing new features and upgrading kernels of underlying operating systems. FusionStorage supports system rolling update. Nodes and storage pools can be rolling updated in batches. This reduces upgrade risks and impacts on services, and improves upgrade efficiency.

System rolling update can be performed by only one click, which is simple and efficient. It takes only 5 minutes to upgrade one node in a storage pool. This ensures smooth and stable service performance.

10.3 Hardware Replacement

FusionStorage adopts x86 servers or TaiShan servers. After services run for a long time, system software and hardware enter the lifecycle maintenance phase. Old software and hardware need to be replaced by new software and hardware. In general, new hardware has higher performance and capacity, and new software has features that are different from those of old software. During capacity expansion, replacement due to faults, and software upgrade, only new hardware may be used. In this case, the coexistence and evolution of new and old hardware must be considered.

Based on the requirement analysis on different scenarios, Huawei adopts the ever-new design. First, services on old platforms are taken over to ensure service continuity. Then, new hardware is used to replace old hardware to achieve smooth evolution.

10.3.1 Adding New Hardware

FusionStorage allows users to add new hardware as a member of the storage cluster. In general, the processing capability of the new hardware cannot be fully utilized. It can be fully utilized only when the processing capability of the new hardware is comparable to that of existing hardware.

11 Storage Management

11.1 Block Storage Service

Users of the FusionStorage block storage management portal can be classified into system administrators, system operators, and system viewers by their roles. The functions of the management portal can be divided into the following categories: resource addition and configuration, resource management and maintenance, and system management and maintenance. Resource management and maintenance include system overview, storage pool management, block storage client management, volume management, virtual file system management, and hardware management.

- Resource addition and initial configuration

  Resource addition refers to adding servers to FusionStorage. Servers can be added one by one or in a batch. Before performing batch import, download a batch import template from DeviceManager. Before adding a server, you need to install the operating system, and then enter the management IP address of the server to be added on DeviceManager. After the server is added, you can install an agent for it on DeviceManager.

  DeviceManager provides a wizard for initial service configuration, including adding cluster servers, configuring the network, installing the software, and configuring basic services for the control cluster.

- Storage pool management

  You can view the statistics and disk topology of the selected storage pool, expand or reduce the capacity of the selected storage pool, and delete a storage pool. You can also create a storage pool.

- Block storage client management

  You can create and delete clients. You can also view the mounting information of a block storage client as well as the CPU and memory monitoring statistics to mount or unmount volumes on the client.

- Volume management

  You can create and delete volumes. When creating a volume, you need to specify a resource pool to which the volume belongs and set the volume name and size. If the created volume is used based on the SCSI protocol, the volume needs to be mounted to a host. If the volume is used based on the iSCSI protocol, you need to map the volume to a host or host group using the iSCSI protocol. On the page for mapping volumes, you can create hosts or host groups, configure initiators, configure CHAP authentication, and map volumes to or unmap volumes from hosts or host groups.

NOTE

By default, the iSCSI service is disabled. To use the iSCSI service, enable it and add the IP addresses and ports used for iSCSI listening.

- QoS policy management

  You can create or delete QoS policies, and check the QoS policy information on multiple pages.

- Snapshot management

  You can create linked clone volumes, set QoS policies, delete snapshots, and view the snapshot list on multiple pages. The list information includes the snapshot name, capacity, owning storage pool, and creation time.

11.2 Block Storage Cluster Management

FusionStorage block storage uses cluster management software to manage clusters. Cluster management software performs basic cluster information monitoring, performance monitoring, alarm management, user management, license management, and hardware management.

- Basic cluster information monitoring: allows users to view basic cluster information, including the cluster name, health status, running status, node information, and node process information.
- Performance monitoring: allows users to view the CPU usage, memory usage, bandwidth, IOPS, latency, disk usage, and storage pool usage.
- Alarm management: allows users to view alarm information, clear alarms, and shield alarms.
- User management: allows the system administrator to create new administrators and grant them management permissions for managing the system or resources. Administrators can query, delete, create, unlock, and freeze users. Password policies can be configured to enhance system security.
- License management: allows users to view activated licenses and import new licenses.
- Hardware management: includes server management and disk management. Server management allows users to view server software installation status, software version, whether a server is added to a cluster, and status and topology of storage pools created on servers, set the maintenance mode to facilitate fault handling, and monitor CPU and memory performance of servers. Disk management allows users to view the disk status, slot number, SN, usage, and type, and collect statistics on the IOPS, latency, bandwidth, and usage of disks.

11.3 eSight – Management Platform in a Data Center

Huawei eSight is used to centrally manage storage devices of different manufacturers in an enterprise data center. It provides diversified management functions such as a global topology view, capacity analysis, performance analysis, health analysis, fault diagnosis, and end-to-end visual management, greatly improving the storage operation & maintenance (O&M) efficiency. It has the following highlights:

- Manages storage devices of different models from different manufacturers, simplifying storage management.
- Displays storage hardware resources and logical resources on a unified device management view.
- Monitors devices in real-time, displays system information indicators in graphics, and automatically sends alarms, requiring no administrator attendance.
- Outputs intelligent analysis reports to monitor key services, laying a foundation for proper space planning.

11.4 eService Cloud-based Management

eService consists of a front-end client and a cloud system. It supports remote fault monitoring, inspection, and log collection to provide all-around protection and quick troubleshooting services for customers' Huawei IT devices, reducing accident risks and O&M costs. It has the following highlights:

- Checks software and hardware status periodically.
- Supports remote inspection without onsite attendance, saving 80% of the time.
- Generates one report for multiple devices at each site.
- When a fault occurs at a site, Huawei engineers can remotely collect system logs to help recover services rapidly, instead of going to the customer's site. They can collect software and hardware logs (cloud computing, storage, and server) by time period with one click.

11.5 SmartKit Intelligent Inspection

The SmartKit inspection tool contains knowledge bases of different product fields, making operations easier for maintenance personnel. Concurrent multi-gateway or multi-process inspection greatly improves the inspection efficiency. It has the following highlights:

- Inspection instance development is synchronous with network element (NE) version development.
- The lifecycle management team (LMT) continuously updates inspection instances based on maintenance experience and precaution notices, and releases corresponding scripts in a timely manner. SmartKit can then be automatically updated.
- Automatically outputs analysis reports and statistics reports.
- Automatically obtains an NE list from the network management system (NMS) to perform remote and batch inspection with high efficiency.

12 Acronyms and Abbreviations

Acronym/Abbreviation	Full Spelling
AD	active directory
CA	client agent
CLI	command-line interface
CMS	cluster management service
DAS	direct-attached storage
DHT	distributed hash table
DNS	domain name system
DS	data service
EC	erasure coding
FTP	File Transfer Protocol
GID	group ID
GUI	graphical user interface
HTTP	Hypertext Transfer Protocol
IAM	Identity and Access Management
IB	InfiniBand
IPMI	Intelligent Platform Management Interface
KV	key value
LDAP	Lightweight Directory Access Protocol
LUN	logical unit number
LVS	Linux Virtual Server
MDC	MetaData Controller
NAS	network-attached storage

Acronym/Abbreviation	Full Spelling
NIS	network information service
NVDIMM	non-volatile dual in-line memory module
OLAP	online analytical processing
OLTP	online transaction processing
OSD	Object Storage Device
RAID	redundant array of independent disks
RDMA	remote direct memory access
SAN	storage area network
SAS	serial attached SCSI
SATA	Serial Advanced Technology Attachment
SCSI	Small Computer System Interface
SSD	solid-state drive
TCP	Transmission Control Protocol
UID	user identity
VBS	Virtual Block Service
ZK	ZooKeeper