vinchin

Vinchin HyperProtect Backup Storage Technical White Paper

Date: 2024-06-04

Contents

1 Abstract

As the next-generation intelligent all-flash storage benchmark, Vinchin HyperProtect backup system (HyperProtect for short) is designed to back up data of mission-critical services in data centers of large enterprises in especially financial and manufacturing industries. In the digital era, IT construction encounters great challenges, against which customer requirement analysis is needed.

- Customer requirements and challenges:
  - Increasing data needs to be backed up within a limited time window.
  - Rapid recovery is required in case of a fault.
  - Predictable recovery duration is required to facilitate DR drills and time estimation.
  - Data volume grows rapidly but the budget is limited.
  - Ransomware becomes a major threat to data security. Therefore, ransomware protection is required.

Based on end-to-end acceleration and the active-active high-reliability architecture, Vinchin HyperProtect features rapid backup, rapid recovery, efficient reduction, and solid resilience. It ensures backup efficiency, recovery speed, and zero data loss. With the fastest recovery speed, it helps achieve efficient backup and recovery and greatly reduce the TCO. It is widely used in industries such as government, finance, carrier, healthcare, and manufacturing.

This document describes and highlights the unique advantages of HyperProtect in terms of product positioning, hardware and software architectures, and features.

2 Overview

This chapter describes HyperProtect products and unique benefits to customers.

2.1 Product Series

HyperProtect STR series products include:

- HyperProtect 5000
- HyperProtect 8000

HyperProtect 5000/8000 support all-flash and HDD forms. The all-flash form supports only dual-port SAS SSDs. User data and system metadata are stored on SSDs. The HDD form supports dual-port SAS SSDs and NL-SAS HDDs. SSDs store metadata to accelerate backup and recovery. HDDs store backup data.

Table 2-1 HyperProtect STR series hardware specifications

	5000 All-Flash	**5000 HDD**	**8000 All-Flash**	**8000 HDD**
Single-node specifications	2 U, 2 controllers	2 U, 2 controllers	2 U, 2 controllers	2 U, 2 controllers
Max. number of nodes	1	1	2	2
Data disk type	SAS SSD	NL-SAS HDD	SAS SSD	NL-SAS HDD
Capacity per data disk*	3.84 TB/7.68 TB	4 TB/8 TB/14 TB/20 TB	3.84 TB/7.68 TB	4 TB/8 TB/14 TB/20 TB
Back-end port type	SAS 3.0			
Front-end port type	8/16/32 Gbit/s Fibre Channel, 10GE/25GE/40GE/100GE			

NOTE

*: If you need disks of other capacity specifications, contact Vinchin sales personnel for evaluation.

* For more information, see the specifications.

2.2 Benefits

Based on end-to-end acceleration and the active-active high-reliability architecture, HyperProtect backup storage features rapid backup, rapid recovery, efficient reduction, and solid resilience. It helps achieve efficient backup and recovery and greatly reduce the TCO. It is widely used in industries such as government, finance, carrier, healthcare, and manufacturing.

Rapid Backup, Rapid Recovery

End-to-end acceleration: The front-end network protocol offload technology releases CPU resources. The back-end CPU multi-core parallel scheduling enables dedicated cores through grouping and task partitioning, improving processing capabilities of nodes.

High bandwidth: Based on backup service characteristics, multiple sequential data flows are aggregated for read and write operations to improve bandwidth performance. Source deduplication reduces the amount of data transmitted over the network and shortens the backup duration.

Efficient Reduction

Multi-layer deduplication: Precise segmentation of backup data flows, aggregation and preprocessing of backup data, and multi-layer inline variable-length deduplication improve logical capacity and reduce the TCO.

Feature-based reduction: Data flow features can be identified. Data combination before compression, high-performance predictive encoding, and byte-level compaction help improve the data reduction ratio.

End-to-end data reduction: Source deduplication and deduplicated replication reduce network bandwidth costs.

Solid Resilience

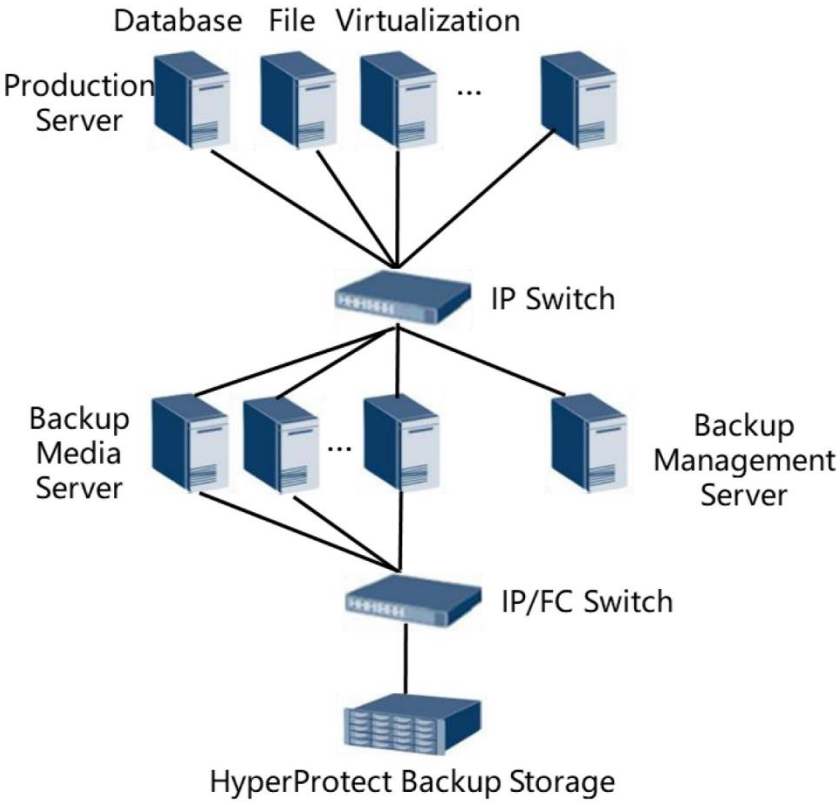
System resilience: The active-active redundant hardware architecture ensures failover within seconds without interrupting backup jobs.

Cyber resilience: Link encryption, array encryption, secure snapshot, Regulatory Compliance WORM, Enterprise WORM, and Air Gap technologies ensure security and availability of copies.

2.3 Backup System Networking

HyperProtect can integrate mainstream backup software including Veritas, Veeam, Commvault, TSM, Networker, and AISHU to form a centralized backup solution for efficient backup and recovery of key service data.

Figure 2-1 HyperProtect backup storage system networking



3 Hardware Architecture

HyperProtect employs a full-mesh architecture. All field replaceable units (FRUs), such as front-end interface modules, controllers, back-end interface modules, power modules, BBUs, fan modules, and disks, are redundant against single points of failure. All FRUs are hot-swappable.

The RDMA high-speed network implements shared access to the global cache at a low latency.

3.1 Hardware

HyperProtect products provide two forms, as described in Table 3-1.

Table 3-1 HyperProtect STR series

Item	HyperProtect 5000 All-Flash	HyperProtect 5000 HDD	HyperProtect 8000 All-Flash	HyperProtect 8000 HDD
Node	2 U enclosure with 25 disk slots	2 U enclosure with 25 disk slots 2 U enclosure with 12 disk slots	2 U enclosure with 25 disk slots	2 U enclosure with 25 disk slots 2 U enclosure with 12 disk slots
Architecture	Active-active	Active-active	Active-active	Active-active
Disk enclosure type*	2 U SAS disk enclosure with 25 x 2.5-inch disk slots	2 U SAS disk enclosure with 25 x 2.5-inch disk slots 4 U SAS disk enclosure with 24 x 3.5-inch disk slots 4 U SAS high-density enclosure with 75 x 3.5-inch disk slots	2 U SAS disk enclosure with 25 x 2.5-inch disk slots	2 U SAS disk enclosure with 25 x 2.5-inch disk slots 4 U SAS disk enclosure with 24 x 3.5-inch disk slots 4 U SAS high-density enclosure with 75 x 3.5-inch disk slots
Disk type	2.5-inch SAS SSD	2.5-inch SAS SSD 3.5-inch NL-SAS	2.5-inch SAS SSD	2.5-inch SAS SSD 3.5-inch NL-SAS

Item	HyperProtect 5000 All-Flash	HyperProtect 5000 HDD	HyperProtect 8000 All-Flash	HyperProtect 8000 HDD
		HDD		HDD
Interface module	• Front end: 4-port 8 Gbit/s, 16 Gbit/s, and 32 Gbit/s Fibre Channel interface modules, 4-port 10GE and 25GE interface modules, as well as 2-port 40GE and 100GE interface modules • Scale-out: Scale-out interface modules include 4-port 25 Gbit/s RDMA interface modules for a 4-controller direct-connection network. • Back end: 4-port 12 Gbit/s SAS interface modules (for connecting to SAS disk enclosures)			

NOTE

*: In HDD form, 2 U SAS disk enclosures (with 2.5-inch disks) contain cache disks.

3.1.1 HyperProtect 8000

HyperProtect 8000 uses a 2 U SAS controller enclosure (with 25 x 2.5-inch disk slots or 12 x 3.5-inch disk slots) integrating disks and two controllers. Key modules are FRUs in redundant mode and can be replaced online. It provides unified backup storage services for SAN and NAS. It supports front-end protocols such as Fibre Channel and iSCSI for SAN and NFS, CIFS, and NDMP for NAS. Each controller enclosure supports up to 12 hot-swappable interface modules. The HyperProtect mid-range device supports SAS disk enclosures.

The two controllers in a controller enclosure of the HyperProtect mid-range device are interconnected through RDMA mirror channels, and multiple controller enclosures can be directly connected through scale-out interface modules. Each controller has two GE management and maintenance ports and one serial port. Figure 3-1 and Figure 3-2 show the front and rear views of the HyperProtect mid-range device.

Figure 3-1 Front view of a mid-range 25-slot device

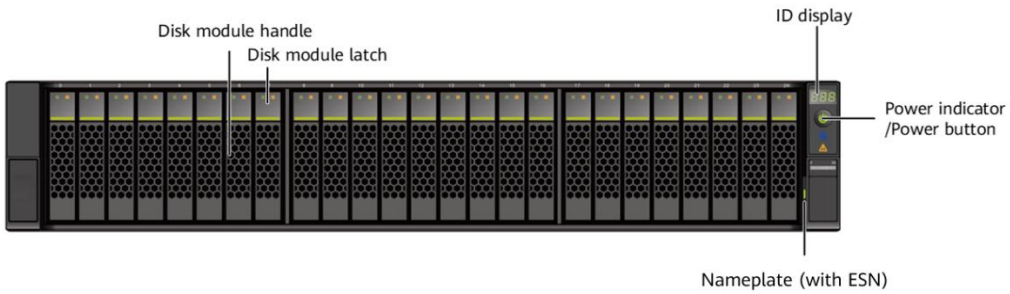
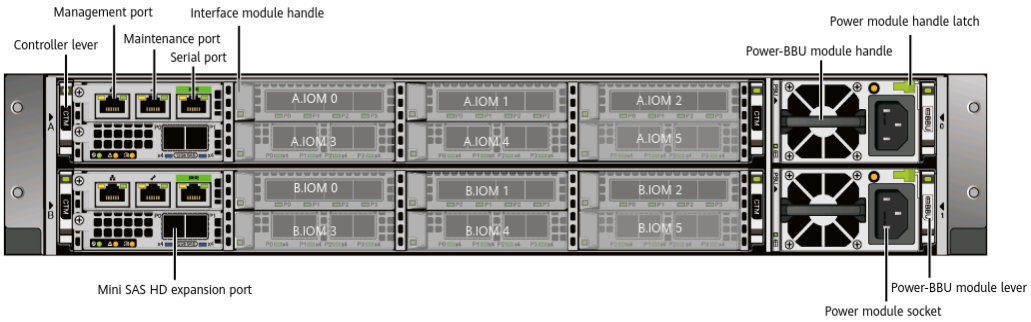
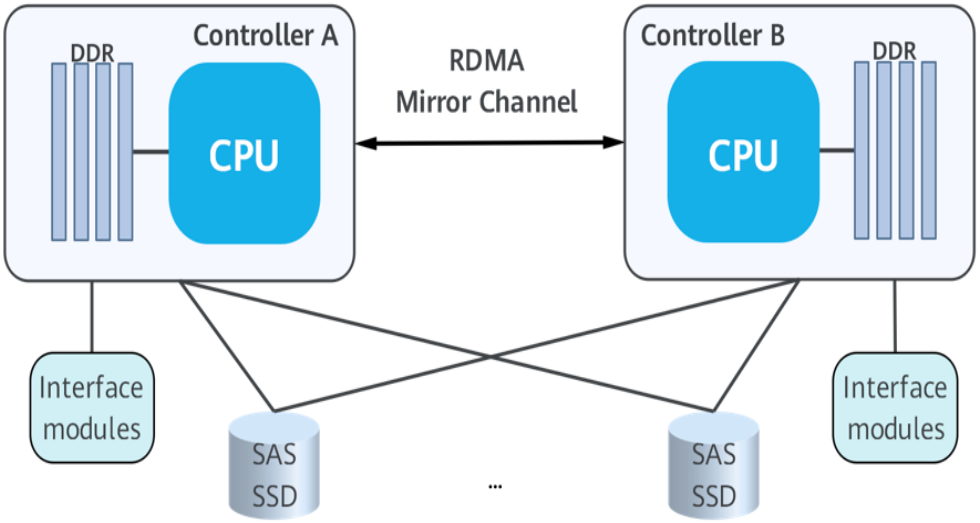


Figure 3-2 Rear view of a mid-range device



The HyperProtect mid-range device is of 2 U high, disks and the two controllers are integrated, and a symmetric active-active architecture is used. The two controllers support service takeover upon a fault and load balancing, and are mirrored through RDMA. Figure 3-3 shows the logical architecture.

Figure 3-3 Logical architecture of a mid-range device



3.1.2 HyperProtect 5000

HyperProtect 5000 uses a 2 U SAS enclosure (with 25 x 2.5-inch disk slots or 12 x 3.5-inch disk slots) integrating disks and the two controllers. A symmetric active-active architecture is adopted. Key modules are FRUs in redundant mode and can be replaced online. Each controller enclosure supports six hot-swappable I/O modules.

Each controller has two GE management/maintenance ports and one serial port. The two controllers in an enclosure are mirrored through RDMA channels. Four controllers can be directly connected without switches through scale-out interface modules. Onboard back-end SAS ports are provided, and SAS disk enclosures are supported.

Figure 3-4 Front view of a 25-slot SAS enclosure

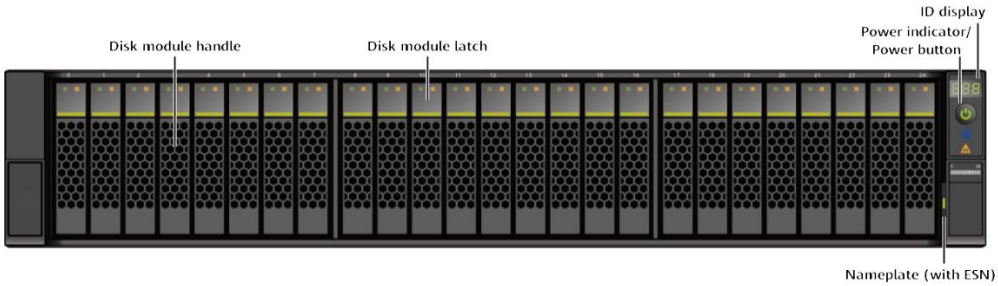
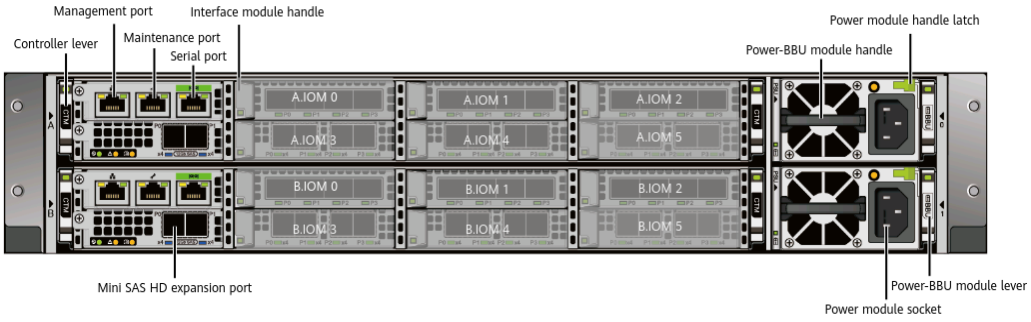
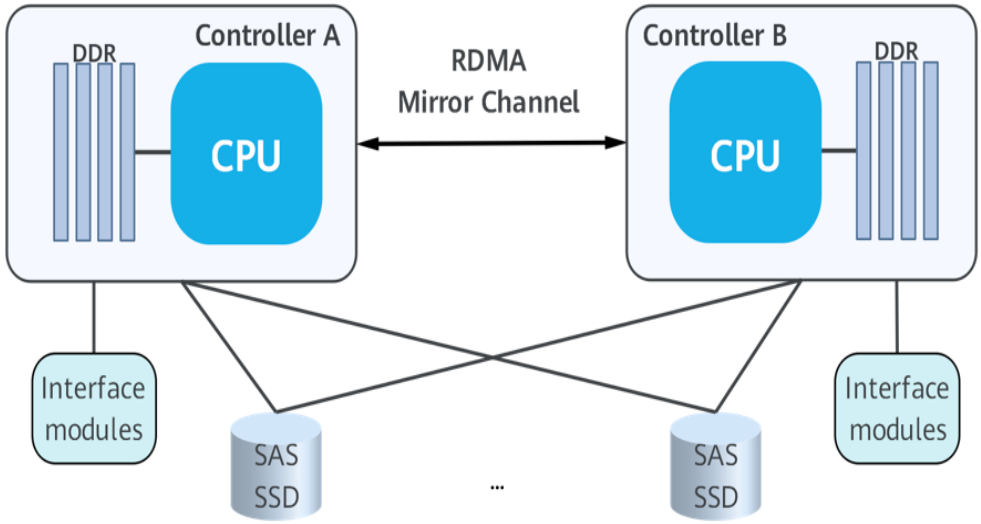


Figure 3-5 Rear view of a SAS enclosure



A symmetric active-active architecture is used. The two controllers support service takeover upon a fault and load balancing, and are mirrored through RDMA. Figure 3-6 shows the logical architecture.

Figure 3-6 Logical architecture



3.1.3 SAS Disk Enclosures

The backup storage supports 2 U SAS disk enclosures (using 2.5-inch disks). 2 U SAS disk enclosures (using 2.5-inch disks) support only 2.5-inch dual-port SAS SSDs. 3.5-inch NL-SAS HDDs are not supported.

The back-end SAS loop supports SAS disk enclosure cascading. A maximum of four SAS disk enclosures are supported. It is recommended that 2 U SAS disk enclosures (using 2.5-inch disks).

3.1.3.1 2 U SAS Disk Enclosures (Using 2.5-Inch Disks)

The 2 U SAS disk enclosure (using 2.5-inch disks) uses the SAS 3.0 protocol and supports 25 SAS SSDs. A controller enclosure connects to a SAS disk enclosure through onboard SAS ports or SAS interface modules.

Figure 3-7 Front view of a 2 U SAS disk enclosure (with 25 x 2.5-inch disk slots)

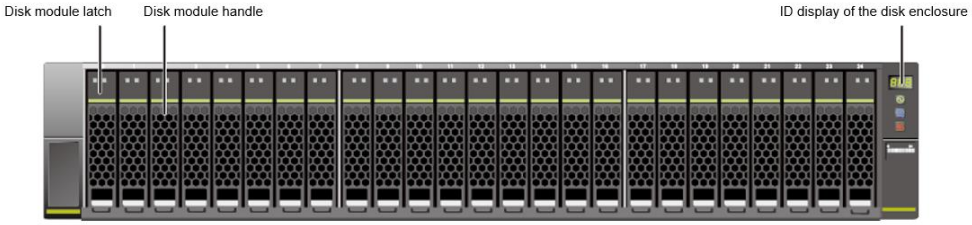
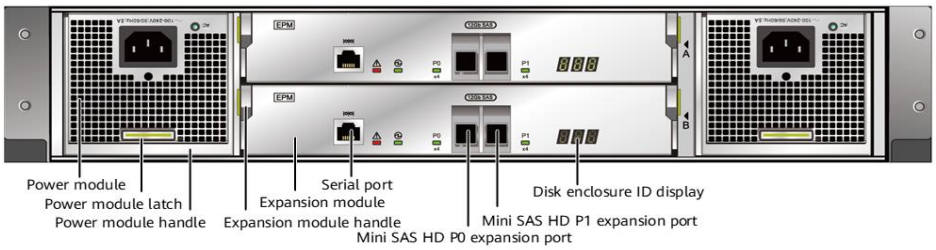


Figure 3-8 Rear view of a 2 U SAS disk enclosure (with 25 x 2.5-inch disk slots)



3.2 Full-Mesh Hardware Architecture

Fully Interconnected Controllers Within a Controller Enclosure

The controller enclosure of an HyperProtect high-end device contains four controllers, each of which is an independent hot-swappable service processing unit. Each controller connects to the passive backplane through three pairs of high-speed RDMA links, thereby fully interconnecting with the other three controllers.

Thanks to the full interconnection of controllers, data flows between controllers do not need to be forwarded by a third component, achieving balanced, fast, and efficient access. No external cables or switches are required for the controllers within a controller enclosure, which simplifies deployment. In addition, the passive backplane uses only passive components, further improving reliability.

For the HyperProtect mid-range model, a controller enclosure contains two controllers. Interface modules are not shared and are owned by specific controllers.

4 Software Architecture

HyperProtect uses a symmetric active-active architecture to implement load balancing. Optimization is made based on characteristics in backup scenarios, maximum performance of the backup storage system. HyperProtect products use a unified software architecture that supports interworking.

4.1 SAN+NAS Parallel Architecture

HyperProtect supports SAN and NAS. The space management software of SAN and that of NAS are at the same software layer. This solves the problem of performance differences and resource competition between LUNs and file systems when SAN is supported based on the NAS architecture or when NAS is supported based on the SAN architecture. CIFS and NFS are supported for file access and Fibre Channel and iSCSI are supported for block access. SAN and NAS share physical layer resources. Resources are allocated on demand and do not need to be reserved, simplifying space management and improving resource utilization. Both SAN and NAS support scale-out to multiple controllers. Hosts can access any LUN or file system from a front-end port on any controller.

4.1.1 Active-Active Logical Architecture for SAN

In a traditional SAN architecture, each LUN is owned by a specific controller. Customers need to plan the owning controllers of LUNs for load balancing. However, it is difficult for an asymmetrical logical unit access (ALUA) architecture to implement load balancing on live networks because service pressures vary with LUNs and vary in different periods for a same LUN.

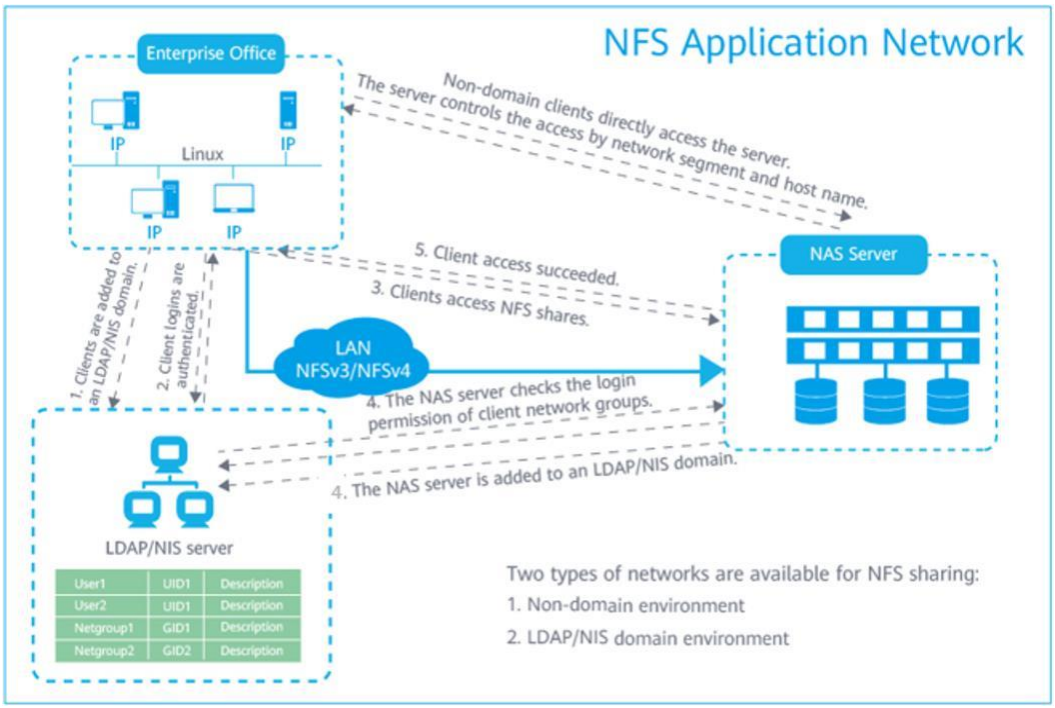
HyperProtect uses a symmetric active-active software architecture. The balancing algorithm is used to distribute the capacity of a single LUN to all controllers to balance the read and write requests received by each controller. In this way, system resources can be fully utilized. Global cache allows that LUNs have no ownership. Each controller processes received read and write requests. (In an ALUA architecture, read and write requests must be processed by the LUN's owning controller.) This balances loads among controllers and simplifies O&M. During O&M, you only need to create LUNs that meet capacity requirements. You do not need to consider the impact of new LUN ownership on controller load balancing and whether real-time service load changes of controllers will cause the controllers to become performance bottlenecks.

4.1.2 Distributed Logical Architecture for NAS

HyperProtect adopts the distributed NAS architecture. When a file system is created, it is evenly distributed to different controllers. During normal system running, all directories and files in the file system are distributed among all cores of a CPU. In this way, the cross-CPU access latency is reduced, and performance is improved in scenarios such as read and write, directory traversal query, attribute traversal query, and batch attribute setting.

4.1.2.1 NAS Protocols

4.1.2.1.1 NFS



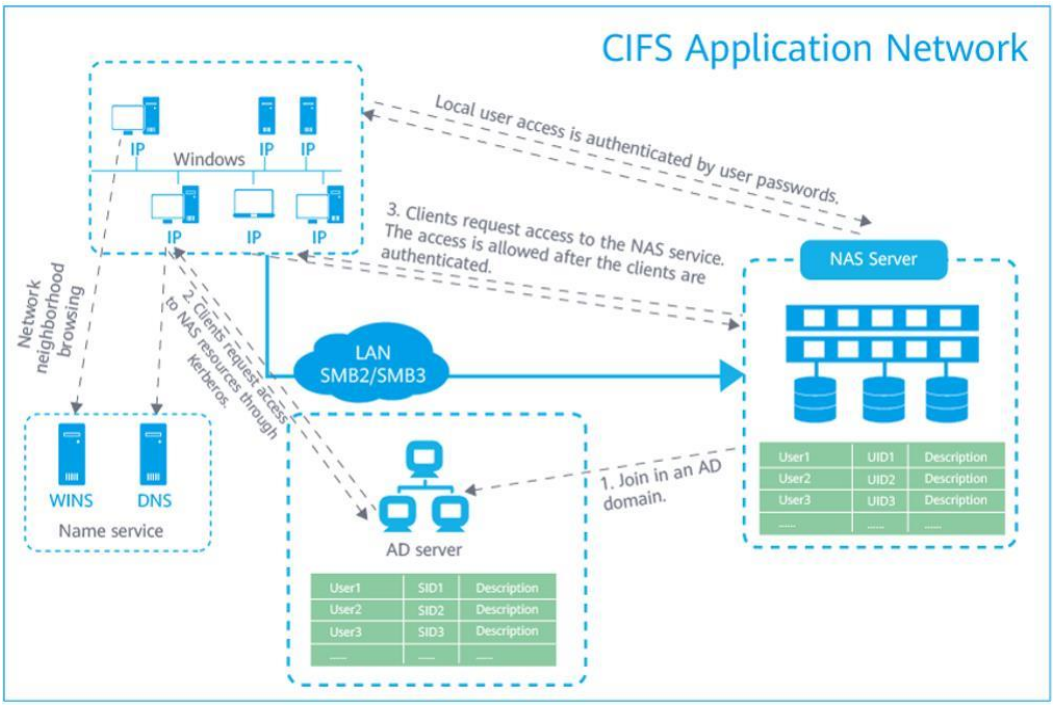
Network File System (NFS) is a common network file sharing protocol used in Linux and Unix environments.

NFS is mainly used for:

- Backup software installed on Unix and Unix-like OSs such as Linux, AIX, and Solaris

HyperProtect supports NFSv3 and NFSv4.1 as well as usage in local user environments (non-domain environments) and LDAP/NIS domain environments. Users can import LDAP certificates for secure domain transmission with LDAPS. In a multi-tenancy environment, the LDAP/NIS service can be configured separately for each vStore.

4.1.2.1.2 CIFS



Server Message Block (SMB), also called Common Internet File System (CIFS), is a network file sharing protocol widely used in Windows environments.

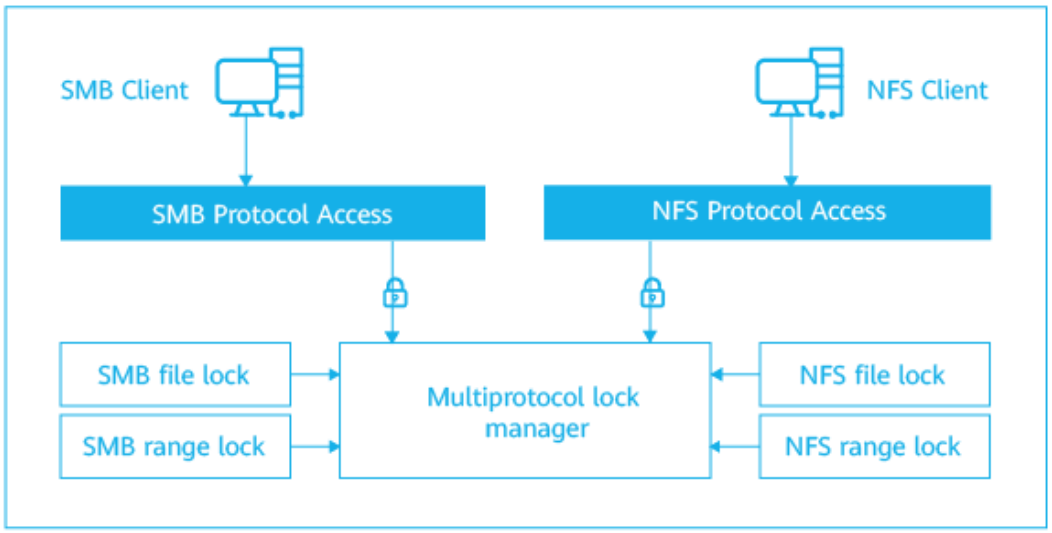
SMB is mainly used for:

- Backup software installed on Windows OSs

  HyperProtect supports SMB 2.0 and SMB 3.0 as well as usage in local user environments (non-domain environments) and AD domain environments. Kerberos and NTLM authentications by AD domains are also supported. The AD domain environments can be individual, parent-child, or trusted domains.

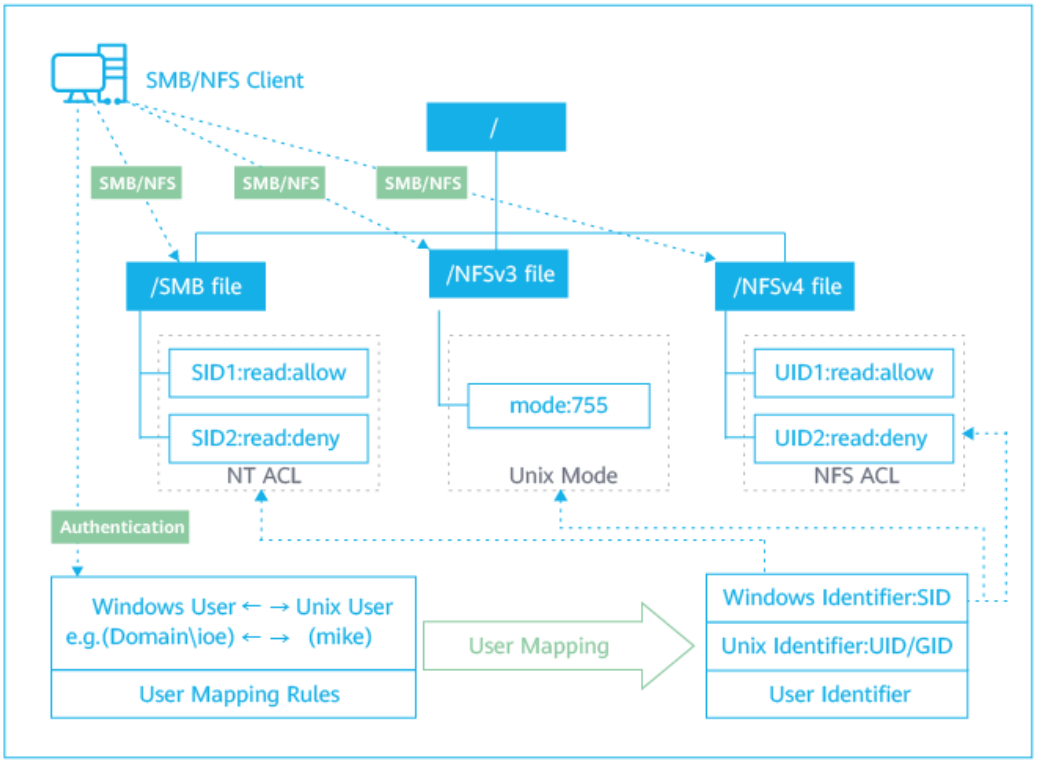
4.1.2.1.3 Multi-Protocol Access

HyperProtect supports access across NFS and SMB protocols. Both NFS and SMB shares are configurable for a file system. The system uses a multi-protocol lock manager with NFS and SMB for exclusive file access without data corruption or inconsistency.



HyperProtect supports the following security modes for multi-protocol access:

- NT mode: File attributes and ACL permissions can only be set on Windows clients with SMB. The system automatically maps file permissions of SMB users to NFS users on Linux clients based on the user mapping relationship for successful authentication of NFS users during file access. NFS users are prohibited from setting the mode or ACL permissions for files.
- Unix mode: File permissions can only be set on Linux/Unix clients with NFS. The system automatically maps file permissions of NFS users to SMB users based on the user mapping relationship for successful authentication of SMB users during file access. SMB users are prohibited from setting the ACL permissions for files.



5 System Performance Design

To meet the requirements of high throughput and data reduction ratio, the number of CPU cores of an HyperProtect backup storage controller is the largest in the industry, providing powerful computing capabilities.

HyperProtect implements software optimization on the whole I/O path covering the front-end network, controller, and back-end network, and fully utilizes its powerful hardware capabilities to provide customers with ultra-high bandwidth. Table 5-1 describes the key performance design based on the I/O delivery process from hosts to SSDs/HDDs for addressing the current problems and pain points.

Table 5-1 Key performance design of HyperProtect

I/O Path	Challenges	Key Design	Principles
Front end	The native Ethernet protocol involves multiple layers and consumes a large amount of CPU resources, resulting in limited maximum performance.	Direct TCP Offloading Engine, saves CPU resources for other tasks, improving the maximum system throughput.	• I/Os bypass the kernel mode to reduce cross-mode overheads. • Protocols are offloaded to hardware, reducing CPU usage.
Controller	Computing capabilities of multiple CPUs and cores must be fully utilized.	The intelligent multi-core technology reduces the resource conflict probability and CPU scheduling overheads, and improves I/O processing efficiency delivered by CPU resources, thereby improving the maximum system throughput.	• I/Os are distributed among CPUs by CPU group to reduce the latency caused by cross-CPU scheduling. • A CPU is divided into different partitions based on services to reduce mutual interference. • No lock is used in the service partitions to avoid lock conflicts.
Back end	Write amplification causes short SSD service life	The multistreaming technology reduces write amplification of garbage collection	Hot and cold data is separated to reduce write penalty within disks.

I/O Path	Challenges	Key Design	Principles
	and low write performance.	and improves SSD write bandwidth.	
		ROW-based full-stripe write reduces write amplification of EC verification and improves system write bandwidth.	ROW-based full-stripe write reduces random write amplification.
Optimization for backup scenarios	Deduplication is not user-friendly for sequential read/write of HDDs.	Fingerprint duplication check is separated from address indexing to ensure that the data layout on disks after deduplication and compression is consistent with the file LBA.	The ID-centric three-layer metadata architecture solves the problem of poor disk read performance caused by disordered data distribution on HDDs caused by deduplication.
	After full backup is performed for multiple times, the latest full backup data is scattered, affecting recovery performance of the next backup job.	Data locality rearrangement is performed for large data blocks scattered due to segmentation and deduplication.	Data locality rearrangement against data dispersion

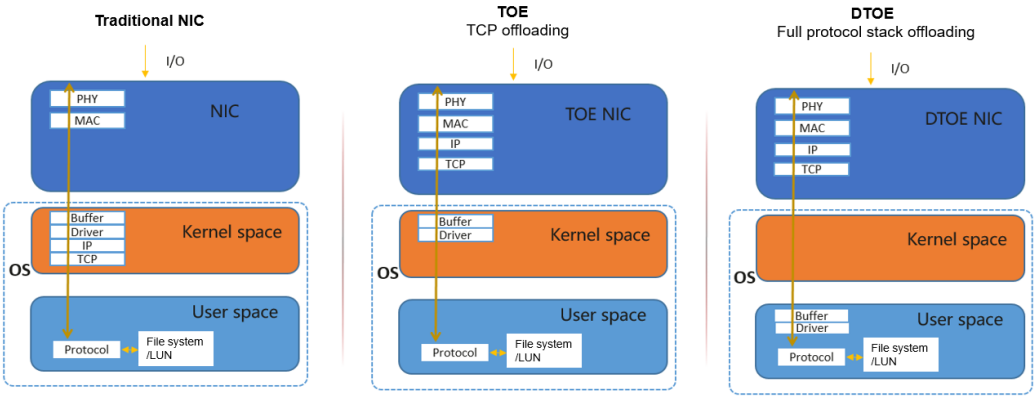
5.1 Front-end Network Optimization

Front-end network optimization mainly refers to reduction of latency between applications and storage devices, including protocol offloading optimization in NAS and iSCSI scenarios and scheduling optimization in common scenarios.

Protocol Offloading

The performance bottleneck of NICs (NAS/iSCSI) lies in the long I/O path. The overhead of TCP and IP protocols is extremely high. Vinchin uses the deeply optimized user-mode TCP/IP and iSCSI protocol stacks and TOE-enabled NIC passthrough to offload protocol computing to interface modules. This releases compute resources of CPUs for other tasks and improves maximum system performance, as shown in Figure 5-2.

Figure 5-1 DTOE technology



When controllers use traditional NICs, network protocol stack processing involves deep layers. Every time a data packet is processed, an interruption is triggered, causing high CPU overhead.

By using the TOE technology, NICs offload the TCP/IP protocol. An interruption is triggered only after an application completes a data processing procedure, which significantly reduces the overhead. However, in this case, some drivers are running in kernel mode, resulting in latency caused by system calls and thread switchover of the user mode and kernel mode.

5.2 Intra-Controller Optimization

5.2.1 Intelligent Multi-Core Technology

HyperProtect products contain more CPUs and cores than any other product in the industry to provide powerful computing capabilities. The intelligent multi-core technology reduces performance deterioration caused by cross-CPU access in multi-core and multi-CPU architectures, fully utilizes powerful computing capabilities of CPUs, and implements linear increase of system performance as the number of CPUs increases. The intelligent multi-core technology includes CPU grouping, service grouping, and the lock-free design between cores.

5.2.1.1 vNode Processing Domain

vNode Domain

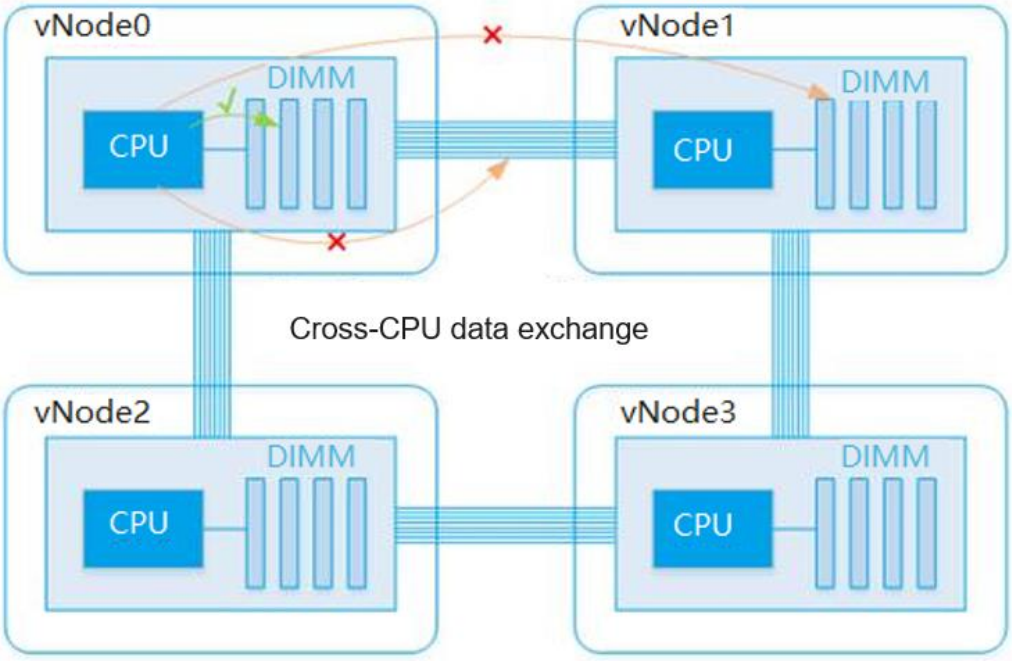
For better scalability, a storage system is divided into multiple vNodes. Each vNode is a logical resource object and corresponds to a CPU as well as memory resources directly accessed by the CPU.

A vNode is also a service processing unit. A service should be processed within one vNode, and the relationship between cache mirroring copies is also established between vNodes.

No Cross-CPU Access

A vNode contains physical resources such as the CPU and memory. The memory is the local memory that can be directly accessed by the CPU. This means that the CPU does not need to access remote memory (which would involve higher latency). I/O requests from hosts are

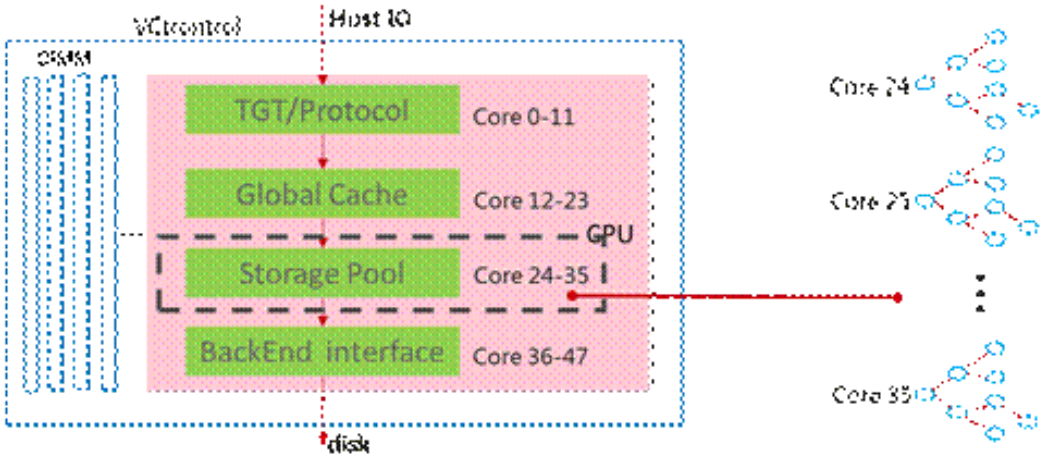
distributed to vNodes based on the intelligent distribution algorithm and are processed in the vNodes from end to end. This eliminates the overhead caused by communication across CPUs, accessing the remote memory, and conflicts between CPUs, allowing performance to increase linearly with the number of CPUs. In the following figure, vNode0 minimizes its access to the memory of other vNodes, and requests of vNode0 are rarely forwarded to other vNodes through the channels between CPUs.



5.2.1.2 Lock-free Design Between Cores

In a service group, each core uses an independent data organization structure to process service logic. This prevents CPUs in a service group from accessing the same memory structure, and implements the lock-free design between CPU cores.

The following figure shows an example. CPU cores 24 to 35 match the storage pool service group and run only the storage pool service logic. In the storage pool service group, services are allocated to different cores, which use independent data organizations to prevent lock conflicts between the cores.



5.2.1.3 Intelligent Dynamic Load Balancing of CPUs

The traditional CPU grouping technology solves the collision domain problem of each service, but brings the problem of unbalanced resource usage between CPU groups in different scenarios. The dynamic load balancing technology provided by HyperProtect defines differentiated scheduling policies based on the computing overheads of tasks. In this way, tasks are balanced among CPU core groups. High-density computing tasks are distinguished from common density computing tasks and are used as scheduling units for load balancing among CPU groups. This prevents scheduled tasks from interfering with each other, improving task execution efficiency.

Figure 5-2 CPU grouping technology evolution

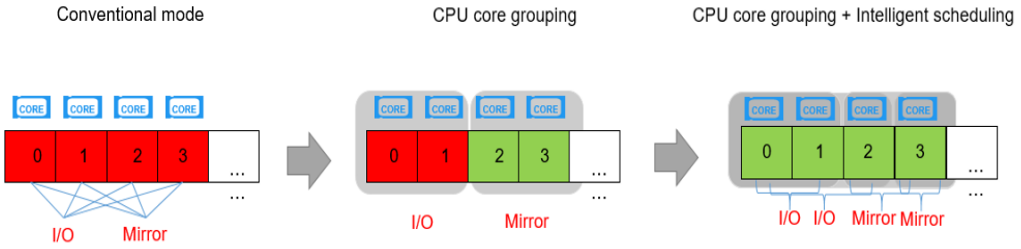


Table 5-2 Scheduling mode comparison

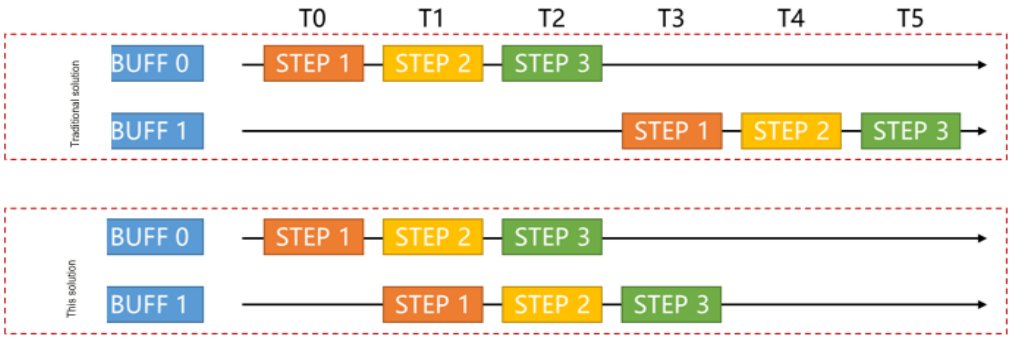
	Traditional Mode	Fixed CPU Core Grouping	CPU Core Grouping + Intelligent Scheduling
Advantages	All CPU cores can be fully scheduled and utilized, making this mode suitable for scenarios where services change frequently.	Task grouping prevents interference between different tasks, reduces frequent resource switching across CPU cores, and improves CPU processing efficiency.	Tasks are allocated to proper CPU cores based on the load status of CPU cores and group attributes of the tasks. In this way, load balancing among CPU cores is implemented and CPU resources are rarely idle and wasted.
Disadvantages	Different tasks compete for time slices of different CPU cores, and high lock conflicts cause frequent forwarding of data I/Os among cores, leading to low CPU computing efficiency, high latency, and unstable performance.	CPU resource groups are bound to tasks, which cannot adapt to scenarios where services change frequently. As a result, some CPU resources are idle and wasted.	Development is difficult.

The vNode, service grouping, lock-free, and intelligent dynamic CPU load balancing technologies enable system performance to increase in quasi-linear mode with the number of controllers, CPUs, and CPU cores.

5.2.2 Efficient Fingerprint Algorithm

The fingerprint NEON instruction is used, the code logic is optimized, and multiple instruction streams are mixed through software and hardware to calculate fingerprints concurrently.

Multi-buffer technology: Two types of hash calculations are performed at the same time and two buffers are provided. Two instruction streams are mixed to perform the two calculations concurrently. In this case, arithmetic logic units (ALUs) in CPUs can be fully utilized, improving performance. The multi-buffer technology achieves nearly twice computing bandwidth with the same compute resources.



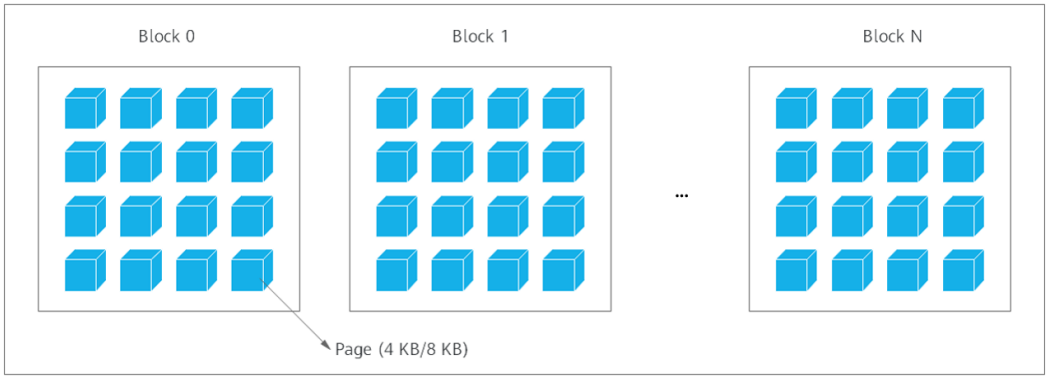
5.3 Back-end Network Optimization

5.3.1 Multistreaming

SSDs use NAND flash. The following figure shows the logical structure. Each SSD consists of multiple NAND flash chips, each of which contains multiple blocks. Each block further contains multiple pages (4 KB or 8 KB). The blocks in NAND flash chips must be erased before data is written. Before erasing data from a block, the system moves valid data in the block, which causes write amplification in the disk.

The multistreaming technology classifies data and stores different types of data in different blocks. There is a high probability that data of the same type is all valid or garbage data. Therefore, this technology reduces the amount of data to be moved during block erasure (one read and one write operations required for movement each time) and minimizes write amplification, improving write performance and prolonging the service life of SSDs.

Figure 5-3 Logical structure of an SSD



In an ideal situation, during garbage collection, all data in a block is invalid so that the whole block can be erased without data movement to minimize write amplification. But this is rarely the case. The update frequency of data in a storage system is different. This defines cold and hot data. For example, metadata (hot data) is updated frequently and is more likely to cause garbage than user data (cold data). The multistreaming technology enables SSD drivers and controllers to work together to store hot and cold data in different blocks. This increases the possibility that all data in a block is invalid, reducing valid data to be moved during garbage collection and improving SSD performance and reliability. Figure 5-4 shows data movement during garbage collection without separation of hot and cold data, in which a large amount of data needs to be moved. Figure 5-5 shows data movement during garbage collection with separation of hot and cold data, in which less data needs to be moved.

Figure 5-4 Data movement during garbage collection without separation of hot and cold data

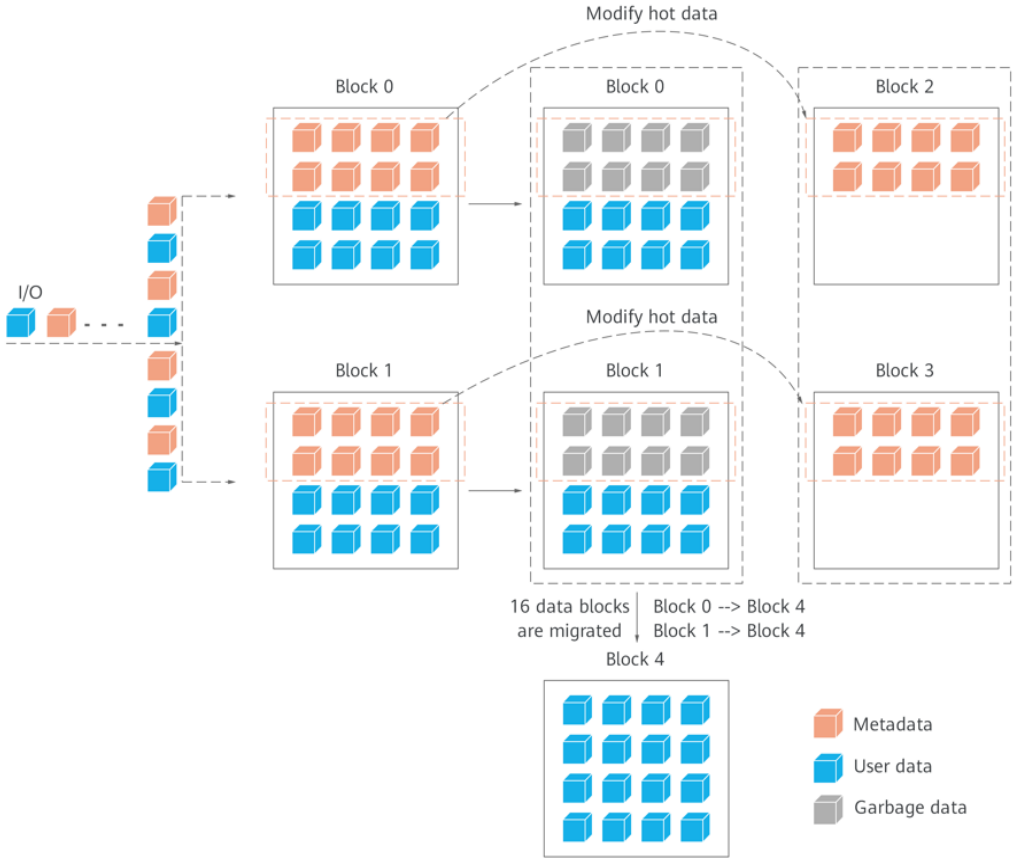
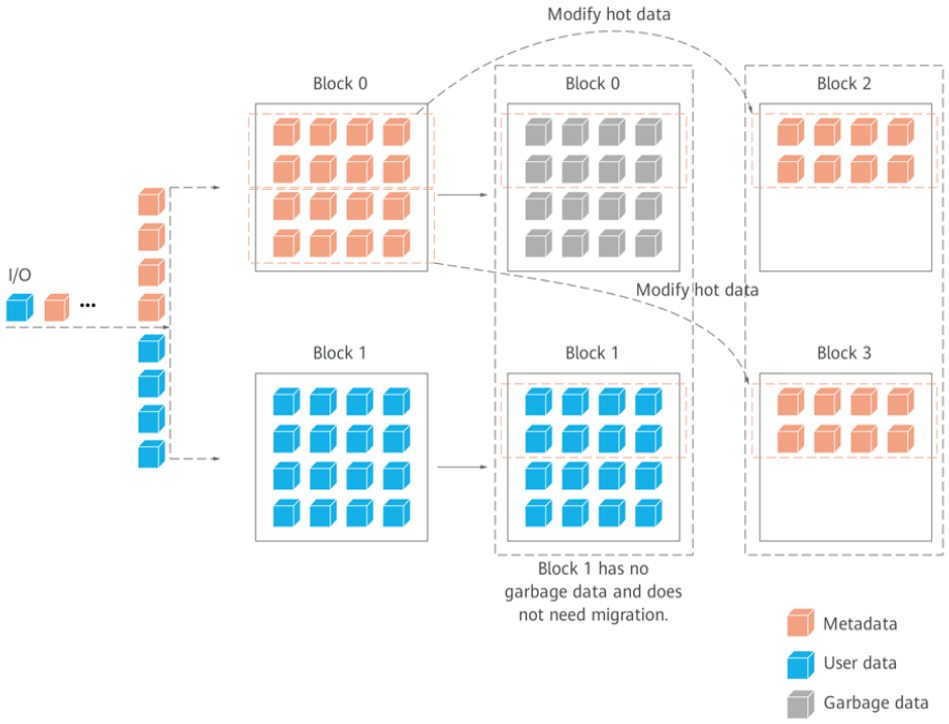


Figure 5-5 Data movement during garbage collection with separation of hot and cold data



HyperProtect separates hot and cold data into three types: metadata, newly written data, and valid data that must be moved during garbage collection. Metadata is the hottest. Newly written data is moved if it is not modified for a long time. The moved data has the lowest probability of being modified and is thus considered the coldest. Data separation reduces SSD write amplification, greatly improves garbage collection efficiency, and reduces the amount of data written to SSDs to improve system write performance and prolong the SSD service life.

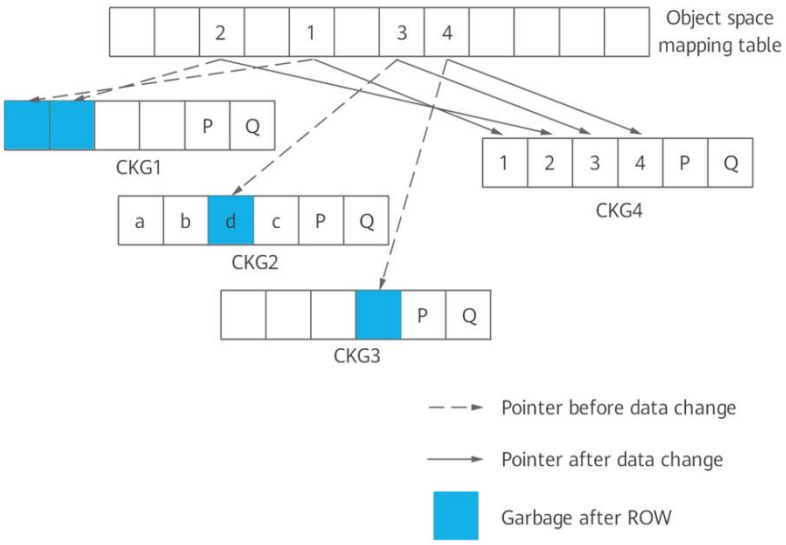
5.3.2 Large-Block Sequential Write

Most I/Os of backup jobs are sequential write I/Os. However, in scenarios where backup data is used to directly starts services, most I/Os of applications are random write I/Os. For the random write model, traditional RAID write processes generate several times of write amplification penalty, causing write performance to deteriorate greatly.

HyperProtect uses ROW-based large-block sequential write for all writes including new data writes and data modifications, avoiding RAID write penalty caused by data reads and modification verification required in traditional RAID. This greatly reduces the overhead on controller CPUs and read/write loads on SSDs in write processes. In addition, ROW allocates a new flash chip for each write to balance the number of erasure times. In this way, flash chips do not wear out due to repeated erasures, prolonging the service life.

Compared with the overwrite mode in traditional RAID, the ROW large-block sequential write mode eliminates the RAID write penalty and enables different RAID levels to provide high performance. When using HyperProtect, users only need to consider data reliability requirements of services without the need to consider the impact of different RAID levels on performance.

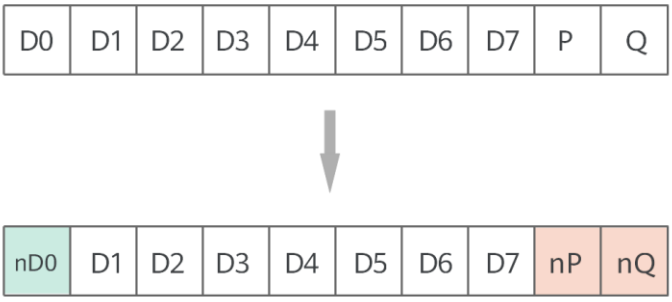
Figure 5-6 ROW-based large-block sequential write



In the figure above, the system uses RAID 6 (4+2) and writes new data blocks 1, 2, 3, and 4 to modify existing data. In traditional overwrite mode, a storage system must modify every CKG where these blocks reside. For example, when writing data block 3 to CKG 2, the system must first read the original data block d and the parity data P and Q. Then it calculates new parity data P' and Q', and writes P', Q', and data block 3 to CKG 2. In ROW-based full-stripe write, the system uses the data blocks 1, 2, 3, and 4 to calculate P and Q and writes them to a new CKG. Then it modifies the LBA pointer to point to the new CKG. During this process, there is no need to read any existing data.

For traditional RAID, for example, RAID 6, when D0 is changed, the system must first read D0, P, and Q before writing write new nD0, nP, and nQ. Therefore, read and write amplifications are both 3. Generally, read and write amplifications of random I/O writes in traditional RAID (xD+yP) are y+1.

Figure 5-7 Write amplification in traditional RAID 6



The following table lists the write amplification statistics of different traditional RAID levels.

Table 5-3 Write amplification of traditional RAID levels

RAID Level	Write Amplification of Random Write I/Os	Read Amplification of Random Write I/Os	Write Amplification of Sequential Write I/Os
RAID 6 (14D+2P)	3	3	1.14 (16/14)

Typically, RAID 5 uses 11D+1P, RAID 6 uses 22D+2P, where D indicates data columns and P, Q, and R indicate parity columns. The following table compares write amplifications on HyperProtect using these RAID levels.

Table 5-4 Write amplification in ROW-based large-block sequential write

	Write Amplification of Random Write I/Os	Read Amplification of Random Write I/Os	Write Amplification of Sequential Write I/Os
RAID 6 (22D+2P)	1.09 (24/22)	0	1.09

On HyperProtect, performance difference between RAID 6 and RAID 5 is within 5%.

HyperProtect uses larger RAID stripes to improve RAID space utilization while ensuring data reliability. The following figure compares RAID space utilization of traditional backup storage and HyperProtect.

Table 5-5 RAID space utilization comparison

	Traditional RAID		HyperProtect RAID	
RAID Level	Largest RAID Stripe	RAID Space Utilization	Largest RAID Stripe	RAID Space Utilization
RAID 6	12+2	85.7%	23+2	92%

HyperProtect can provide higher data reliability and space utilization simultaneously.

5.4 Design for the Specific Backup Service Model

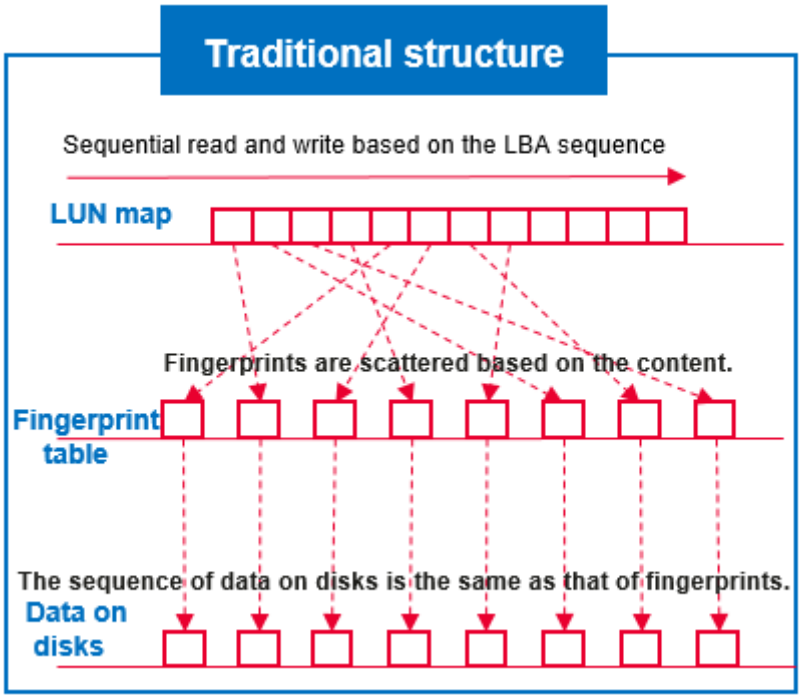
Based on characteristics of the concurrent large-block sequential write + large-block sequential read I/O model of backup services, intra-controller end-to-end optimization has been made for HyperProtect backup storage. Large data segments help reduce the amount of metadata managed per unit space, improve metadata processing efficiency of large-bandwidth sequential reads and writes, and save CPU resources required for metadata processing to facilitate other services, improving performance. In this way, the amount of metadata and storage space occupied by metadata are reduced. With the high data reduction ratio in backup scenarios, the proportion of space occupied by metadata is significantly reduced.

For variable-length segmentation, the larger the input continuous data blocks, the higher the deduplication ratio of data segments after segmentation. However, the value does not increase linearly. Based on analysis of a large amount of experimental data, 4 MB data blocks are used for variable-length segmentation of HyperProtect backup storage. Based on sequential write characteristics of backup applications, buffering is used to aggregate data of sequential write service flows to 4 MB data blocks in the cache, and then these blocks are transmitted to the variable-length segmentation module at a time. In this way, the variable-length module delivers a higher data deduplication ratio. For non-sequential write requests, data is not aggregated into 4 MB data blocks. At the same time, writing large blocks to small blocks reduces the number of writes within unit time, thereby reducing processing overheads and improving back-end and write performance.

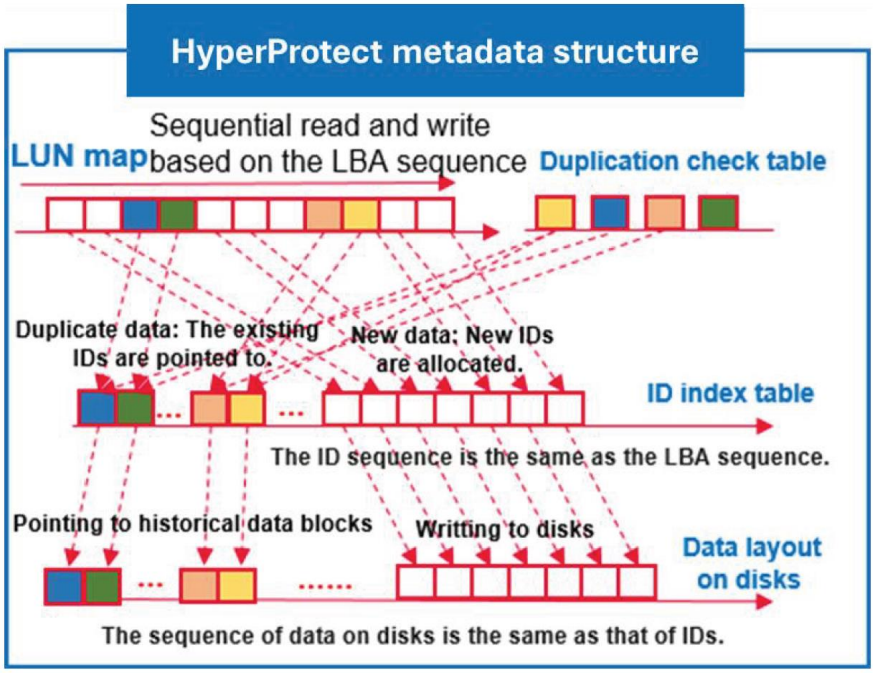
5.4.1 Optimization for Backup and Recovery Jobs

In backup jobs, I/O writes are generally based on logical addresses of files generated by the jobs. During backup and recovery, most I/O reads are based on these logical addresses. The sequence of data reads is basically the same as that of data writes. HyperProtect backup storage uses a special data layout architecture to match the backup software features.

In the traditional deduplication architecture, data is arranged based on the fingerprint table of data blocks. After being deduplicated, sequential data is scattered and stored based on fingerprints. Data on disks is stored based on fingerprints, which is irrelevant to the LBA sequence. In this case, the sequence of service data is affected. This mode is suitable for the all-flash mode. However, if HDDs are used, random access performance is low, which affects deduplication and recovery performance.



When a backup job is executed, data is written based on the logical address sequence of generated backup files. During recovery, data is read based on the logical address sequence of the backup files. In this way, the high sequential read/write performance of HDDs is utilized, with the low random read/write performance avoided. The ID-centric three-layer metadata structure is designed to solve the problem of low HDD read performance caused by disordered data due to deduplication.



Backup jobs based on backup software involve sequential writes, and recovery jobs involve sequential reads. ID metadata is applied for and allocated in batches to ensure that the ID allocation sequence is consistent with the LBA address sequence. Fingerprint duplication check is separated from address indexing. The duplication check table and ID index table are provided. The ID and LBA are in the same sequence, and the data layout on disks is consistent with IDs. This ensures that the metadata and data stored on HDDs are in the same sequence as those accessed by hosts.

As shown in the preceding figure, after variable-length segmentation is performed for file-level sequential write data, fingerprint calculation and duplication check are performed. For duplicate data, the LBA directly points to an existing ID. New data blocks are sorted based on LBA addresses and then allocated with new IDs. The new IDs are in ascending order and never reused. New data is written to disks after being compressed, and is arranged in the sequence of new IDs on storage media. From the end-to-end aspect, sequential data written by backup software is deduplicated and then stored in sequence on storage media. Benefits: 1. Backup performance is improved. I/Os are sequentially written by the storage system to storage media. The high bandwidth of HDDs improves performance of the entire system. 2. Recovery performance is improved. During new data recovery by backup software, large data blocks on storage media are sequentially read after ID conversion. A large segment of sequential data is read at a time because the data on the storage media is arranged based on the write sequence of the backup software. Excessive data that is not required for the current I/O read is stored in the cache. As long as the recovery job is not terminated, there is a high probability that the excessive data can be hit in subsequent sequential read requests within a short period of time, thereby improving recovery performance.

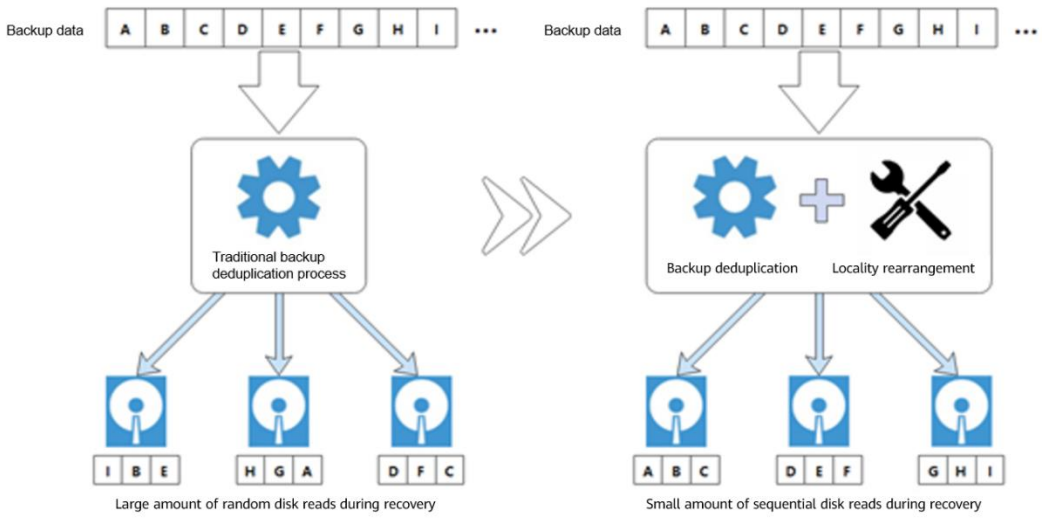
5.4.2 Recovery Performance Improvement

As the number of backup times increases and due to the deduplication relationships during incremental backup and full backup, deduplicated data of backup data that is written later is more scattered physically. As a result, recovery read performance deteriorates gradually. The latest backup data is most likely to be used for recovery, but recovery performance is low due to scattered data on HDDs. Therefore, the recovery duration is unacceptable in the case of data loss or unexpected situations. The locality rearrangement algorithm used by

HyperProtect implements user data instant recovery and improves backup data recovery performance.

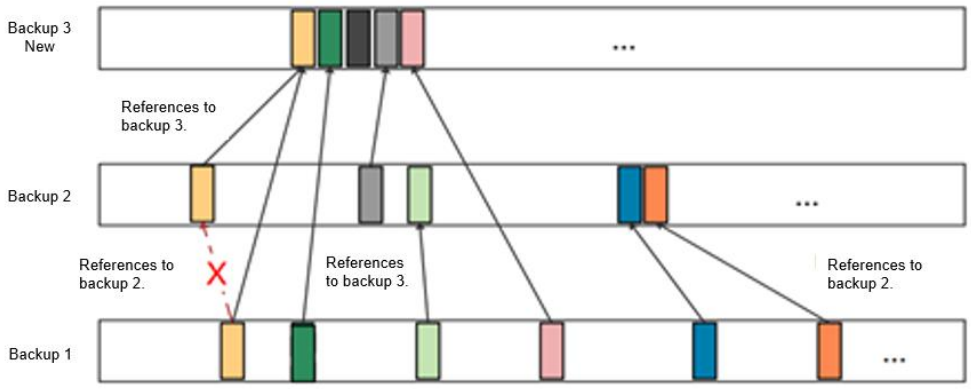
The latest backup data is usually deduplicated based on earlier backup data. Therefore, during recovery based on the latest backup, data of earlier backup copies is read randomly for multiple times. As a result, read performance of HDDs is low. Locality rearrangement aims to reduce the number of random read operations during backup and recovery, as shown in the following figure.

Figure 5-8 Data locality rearrangement



When backup data is written, the system determines whether to adjust the data layout on disks based on how scattered the data is and how many random operations on disks are involved during recovery. Different from the traditional backup data reference mode, locality rearrangement references earlier backup data to new backup data and deletes duplicate data blocks of the earlier backup data, as shown in Figure 5-9. In this way, during recovery based on the latest backup, only data of the latest backup copy and some data of earlier backup copies are sequentially read, reducing the impact of random reads on recovery performance.

Figure 5-9 Data reference relationships after locality rearrangement



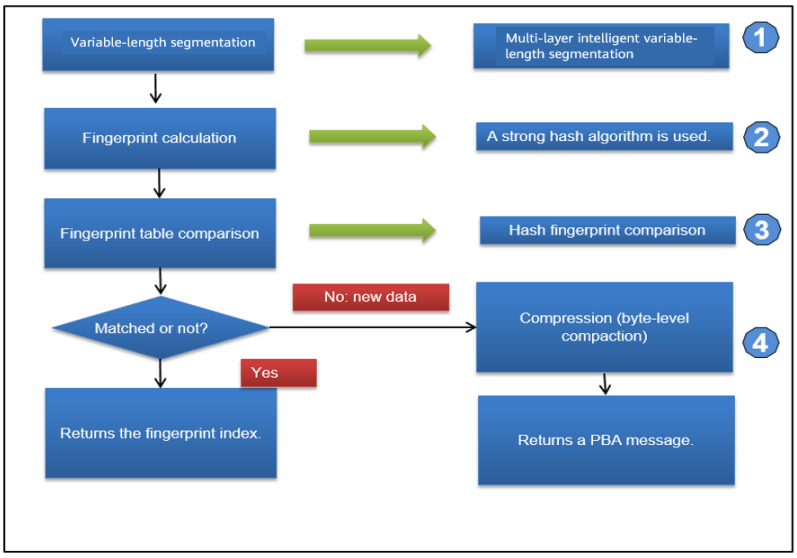
6 Data Reduction

Dedupe retains only unique data blocks. Duplicate data directly references these blocks to save storage space.

Compression compresses data to be written to storage media using the lossless compression algorithm to save storage space.

HyperProtect applies Dedupe for deduplication and then Compression for compression. HyperProtect uses inline variable-length deduplication (chunk size: 4 KB to 32 KB). This chapter describes principles of deduplication and compression.

Figure 6-1 Deduplication and compression



1. When data is written, variable-length segmentation is first performed. The system uses the multi-layer intelligent variable-length segmentation algorithm to calculate fingerprint characteristics of the data, based on which the data is then divided into variable-length blocks.
2. Then, the system uses the SHA1 algorithm to generate fingerprints for these blocks.
3. The system compares the new fingerprints with those in the fingerprint database. If the same fingerprint is found, the written data block is considered duplicate. In this case, the system updates the data count and returns the existing fingerprint index.

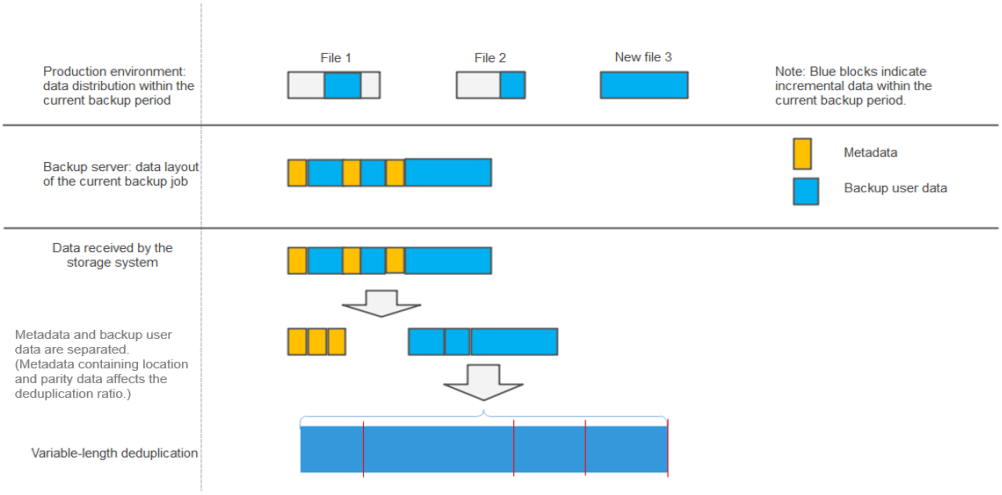
4. If the hash fingerprint is unique, the written data block is considered new. In this case, the system compresses the data and then writes the data to disks.

6.1 Preprocessing of Data Written by Backup Software

Generally, data written by backup software includes metadata and backup data. Some backup software combines metadata and backup data into one file and distributes the metadata to the entire file. The deduplication ratio of metadata is low because metadata contains random data such as timestamp and verification data. If metadata and backup data are deduplicated together, the overall data deduplication ratio decreases.

HyperProtect separates metadata and backup data inserted by backup software based on the data layout through feature-based processing. User data is aggregated and then variable-length segmentation and deduplication are performed. Metadata is compressed separately. In this way, the overall data reduction ratio is improved, as shown in Figure 6-2.

Figure 6-2 Preprocessing of data written by backup software

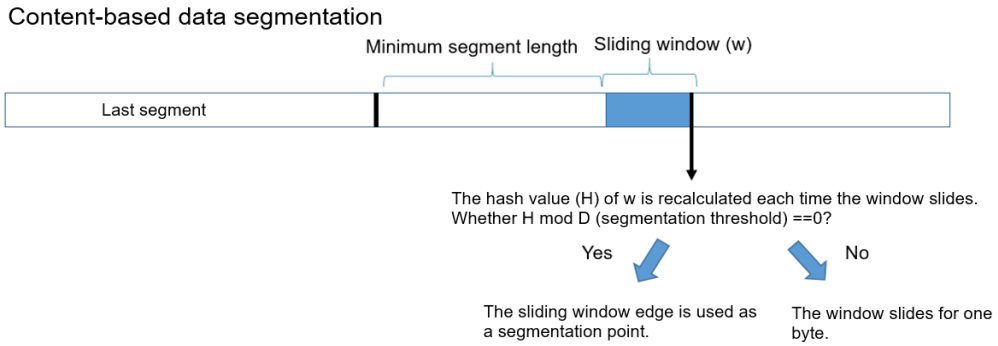


6.2 Variable-Length Deduplication

Conventionally, a storage system divides a user file in to fixed-length segments before storing it. In backup scenarios, this will cause serious problems. Once some data is inserted into or deleted from a user file, fixed-length segments that are slightly different from the original ones will be generated. As a result, a large proportion of data before and after the modification cannot be deduplicated. HyperProtect uses variable-length segmentation to solve this problem. The system calculates the features of data flows and slices data based on the features, thereby obtaining variable-length data blocks. After some data is inserted or deleted, the data flow features change only in a small range near the insertion or deletion position. In this way, the feature difference between the data before and after the modification is related only to the amount of inserted or deleted data. For data that does not change, the same variable-length segments are generated and deduplicated by the storage system, delivering a higher deduplication ratio.

On HyperProtect backup storage, data blocks are deduplicated using the content-based segmentation algorithm, which is mainly used to deal with low deduplication ratio caused by data offset. The following figure shows the basic principle of content-based segmentation. A sliding window is used to determine segmentation points based on the content.

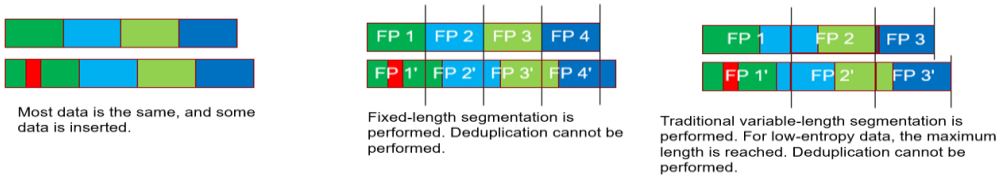
Figure 6-3 Sliding window calculation



6.2.1 Intelligent Multi-Layer Variable-Length Segmentation

Traditional variable-length segmentation does not deliver a good effect in slicing low-entropy data. It is likely that no proper segmentation point can be found even when the sliding window slides to the maximum extent. Consequently, the data block deduplication ratio is low because the maximum segment length is used for segmentation, as shown in Figure 6-4.

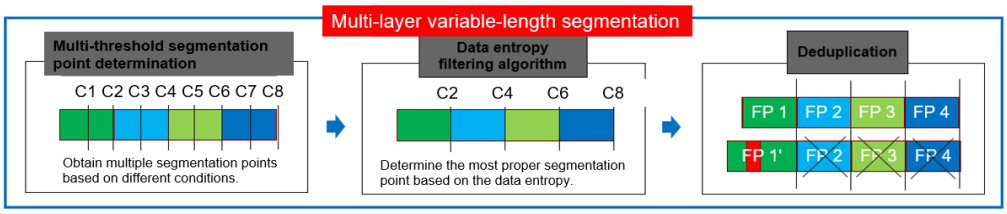
Figure 6-4 Deduplication effects when traditional fixed-length and variable-length segmentation methods are used in case of data insertion



To overcome the disadvantages of traditional variable-length segmentation in slicing low-entropy (highly duplicated) data, Vinchin uses the multi-layer variable-length segmentation algorithm. As shown in Figure 6-5, multiple thresholds are used for obtaining different segmentation points. Then, the data entropy filtering algorithm is used to select proper segmentation points that meet requirements. Variable-length segmentation is performed based on the selected segmentation points.

Through data simulation, the multi-layer variable-length segmentation increases the deduplication ratio by more than 10% compared with the traditional segmentation algorithm.

Figure 6-5 Multi-layer variable-length segmentation



6.2.2 Deduplication Principles

The deduplication ratio of backup data is high in application scenarios where backup storage systems are used. The traditional mode in which a storage system reads duplicate data from disks, decompresses the data, and then compares the data byte by byte consumes a large amount of CPU compute resources. HyperProtect uses weak-hash fingerprint indexes and strong-hash fingerprint comparison to eliminate data inconsistency caused by single hash conflicts. This saves a large amount of CPU resources of controllers and relieves back-end disk pressure, improving system performance.

The deduplication process on HyperProtect is as follows:

1. After user data is sliced into variable-length segments, the system calculates weak and strong hash fingerprints for each data block.

2. The system sends each data block and these fingerprints to corresponding file system owning controllers and compares the weak-hash fingerprint with those in the fingerprint database.

3. If the system finds an identical weak hash fingerprint, it reads the corresponding strong hash fingerprint according to location saved in the fingerprint table and compares it with the strong hash fingerprint of the new data block. If the two fingerprints are the same, the system performs step 7. If the two fingerprints are different, a hash conflict occurs and the system writes the data to disks instead of performing deduplication. If compression is enabled, the data is compressed before being written.

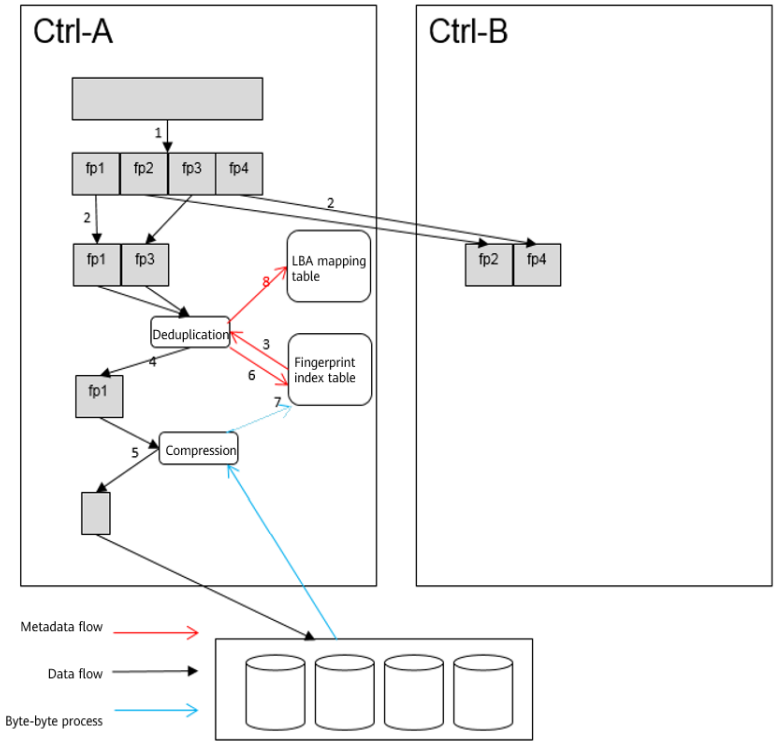
4. If no same fingerprint is found in the fingerprint table, the data to be deduplicated is new data, and the new weak hash fingerprint is recorded in the fingerprint index table.

5. If compression is enabled, the system compresses the new data block. If compression is disabled, the system directly applies for storage space for the data block. The storage location will be recorded.

6. The mapping relationship between the weak hash fingerprint and the data storage location is saved in the fingerprint table.

7. If the strong hash fingerprints are the same, the data block is duplicate and will not be saved. The system only increases the reference count of the fingerprint and uses an existing index for the data block.

Figure 6-6 Deduplication process



Inline deduplication on HyperProtect deletes duplicate data before the data is written to storage media. The inline deduplication process is as follows:

The storage system performs variable-length segmentation, calculates fingerprints of data blocks after segmentation, and then compares the fingerprints with those in the system for duplication check. If an identical fingerprint is found, the file system LBA directly points to the ID of that data block, and the duplicate data block is not written. If no identical fingerprint is found, the system allocates a new ID to the new data block and records the mappings between the file system LBA and new ID, between the fingerprint and new ID, and between the new ID and write location on storage media.

The ID table is used as the conversion center of LBA, fingerprint table, and storage media address metadata. This design is oriented to the read/write service model of backup software and read/write characteristics of HDDs.

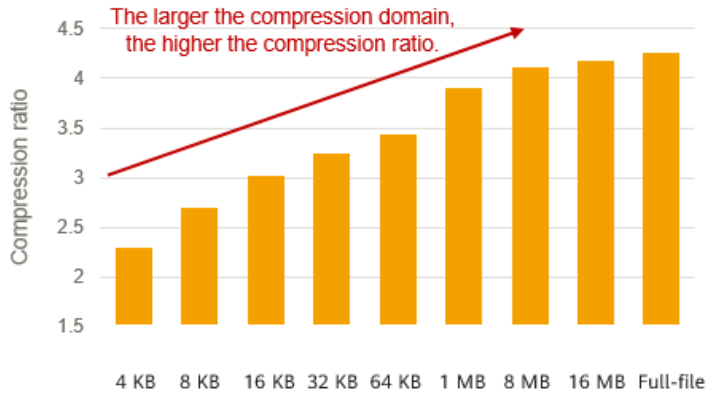
As shown in the preceding figure, after variable-length segmentation is performed for file-level sequential write data, fingerprint calculation and duplication check are performed. For duplicate data, the LBA directly points to an existing ID. New data blocks are sorted based on LBA addresses and then allocated with new IDs. The new IDs are in ascending order and never reused. New data is written to disks after being compressed, and is arranged in the sequence of new IDs on storage media. From the end-to-end aspect, sequential data written by backup software is deduplicated and then stored in sequence on storage media.

6.3 Compression

Data compression involves two processes. Input data blocks are first compressed to smaller ones using a compression algorithm and then compacted before being written to disks.

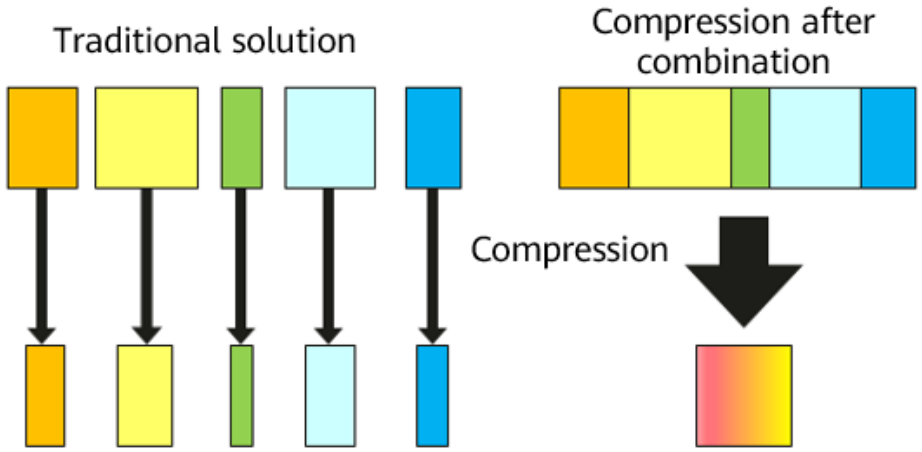
6.3.1 Compression After Combination

According to the test data, the final compression ratio is positively correlated with the size of the data blocks involved in compression at a time.



HyperProtect uses combination after compression. As shown in Figure 6-7, the system combines data blocks with consecutive IDs into a large data block (128 KB) for compression. As backup software involves sequential writes and recovery involves sequential reads, data blocks with consecutive IDs are combined and then compressed. During backup recovery, data blocks with consecutive IDs obtained after one read are hit in the read cache by subsequent read requests initiated by backup software, fastening disk read and improving the overall performance.

Figure 6-7 Compression after combination

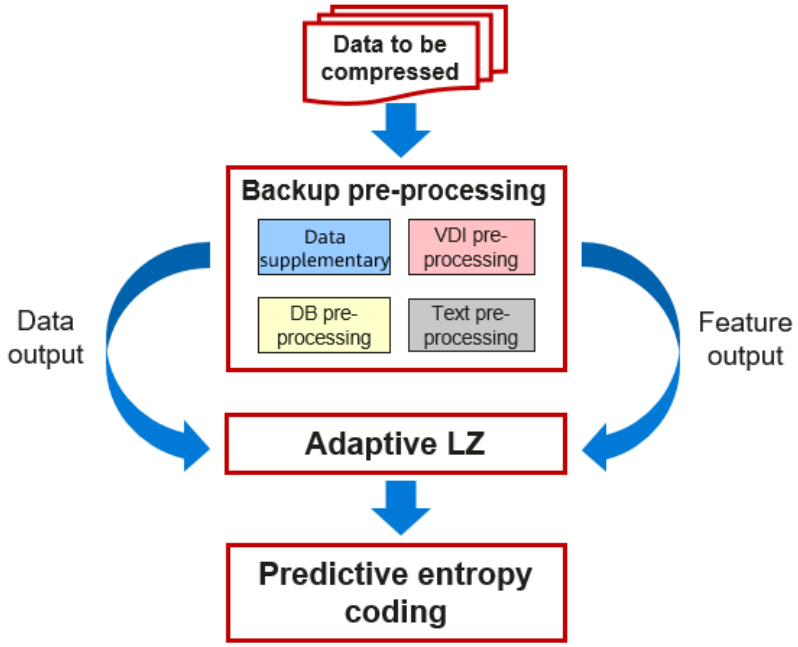


6.3.2 Compression Process

Preprocessing in Backup Scenarios

Before data compression, HyperProtect uses multiple preprocessing algorithms for feature-based processing in main backup scenarios. The preprocessing is to identify data scenarios based on data flow features and then clean (removing identifiers such as tags inserted by software), rearrange (reversible data rearrangement based on data type features for improved

compression ratio), and deduplicate (deleting redundant data based on data features) data to improve the compression ratio. The pre-processed data and identified data features are transmitted to the adaptive LZ module for deduplication and compression.



The pre-processed data and identified data features are output.

Adaptive LZ

Different from the common LZ compression algorithm, the adaptive LZ-based compression algorithm matches multiple preset compression policies based on the data features identified by the preprocessing module to compress the preprocessed data.

Predictive Entropy Encoding

The high-performance predictive encoding algorithm trains predictive models based on backup data features and works with word frequency information and prediction technologies to improve the compression ratio.

With the preprocessing, adaptive LZ, and predictive entropy encoding technologies, HyperProtect backup storage improves the compression ratio by more than 10% in the VM + file sharing backup scenario and by more than 30% in the database backup scenario.

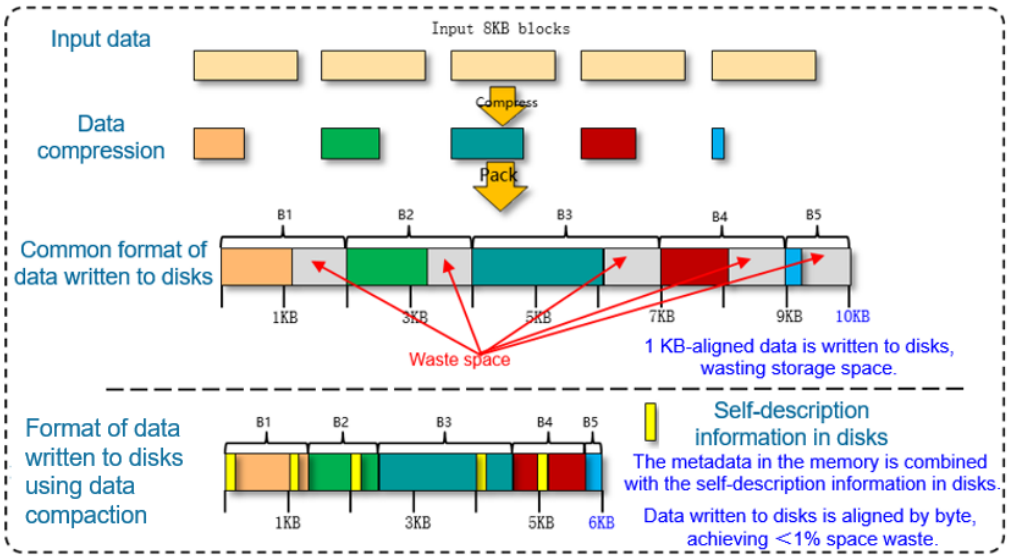
6.3.3 Data Compaction

This section describes how data compaction is implemented.

With HyperProtect, compressed user data is aligned by byte and then compacted to reduce the waste of physical space. This provides a higher reduction ratio than the 1 KB-based alignment generally used in the industry.

As shown in Figure 6-8, byte-level alignment saves 2 KB physical space compared with 1 KB-based alignment, improving the compression ratio.

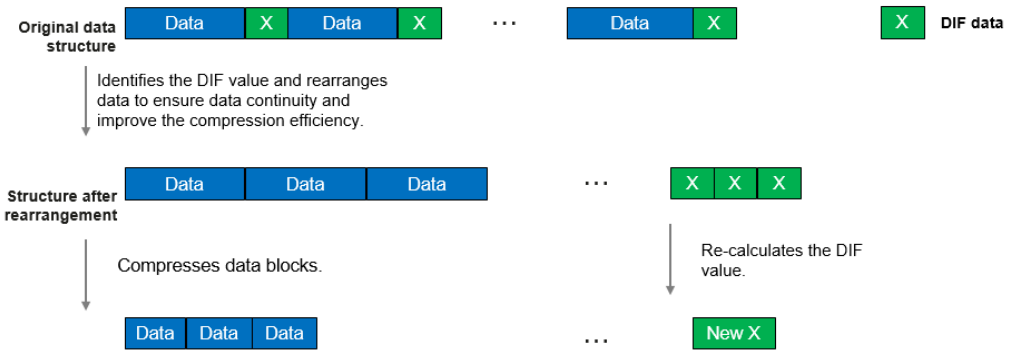
Figure 6-8 Alignment of compressed data by byte



6.3.4 Data Rearrangement

HyperProtect uses the data rearrangement technology. For a grain-granularity data block, HyperProtect inserts data verification information following every 512 bytes of data. Because random data exists in the verification information, the calculation of the compression algorithm may be disturbed by the random data, resulting in a low compression ratio. Therefore, the data rearrangement technology separates the user data from parity data, compresses the user data, and reduces the parity data in a customized way. In this way, a better compression ratio and higher compression and decompression speeds can be obtained.

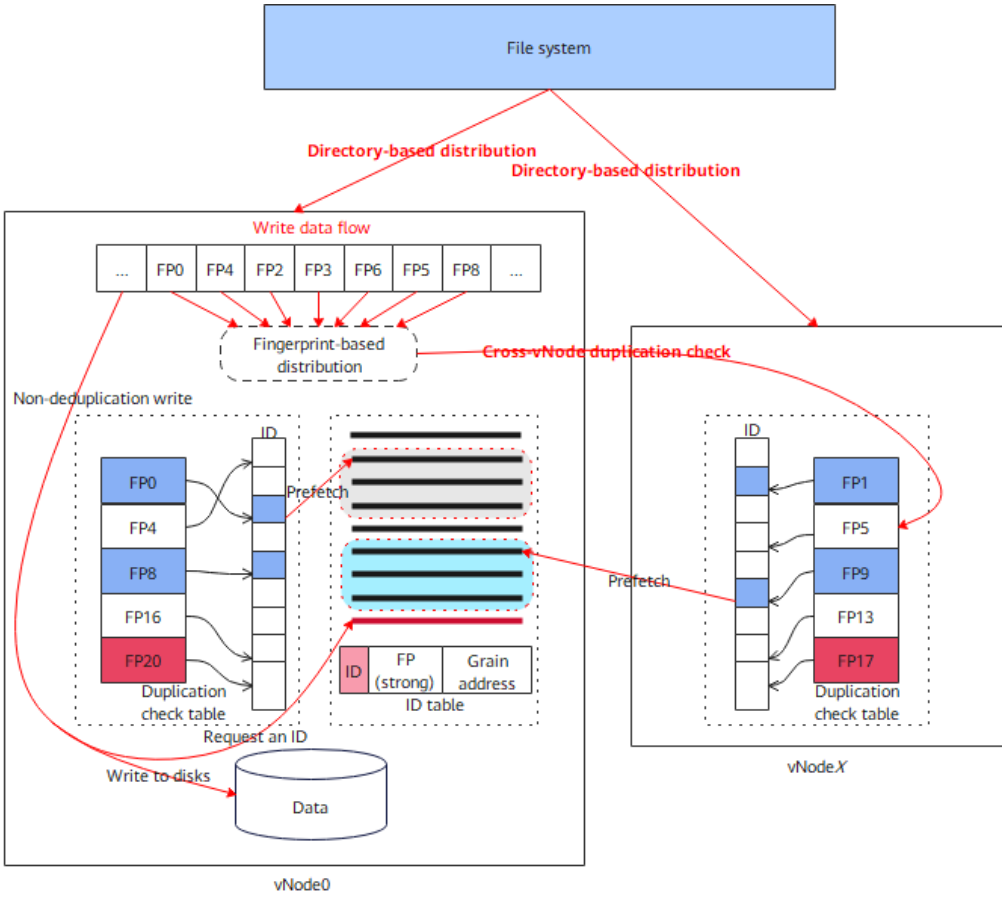
Figure 6-9 Data rearrangement



6.4 Controller-level Deduplication

Deduplication domains can be divided on a per controller basis. A file system is created for each controller to utilize the hardware performance of the entire controller, improving the performance of a single file system and simplifying user configuration.

Figure 6-10 I/O paths for controller-level deduplication



General design:

1. In a single file system, data is distributed to level-1 and level-2 directories to ensure that data (in a single file system) can be written to all vNodes under a controller. Requests at the protocol layer and file system layer are distributed within the controller, fully utilizing array performance.

2. The read and write requests of the FP in the INDEX are distributed in the controller based on the distribution algorithm. The read and write requests of the ID table are still processed in the vNode to which the data belongs. Because the upper-layer data has been distributed, the read and write requests of the ID table at the INDEX layer are also distributed in the controller, fully utilizing the controller hardware performance. Deduplication domains are divided based on the controller granularity. During deduplication, the deduplication table and ID table are queried across vNodes in the controller during duplication check.

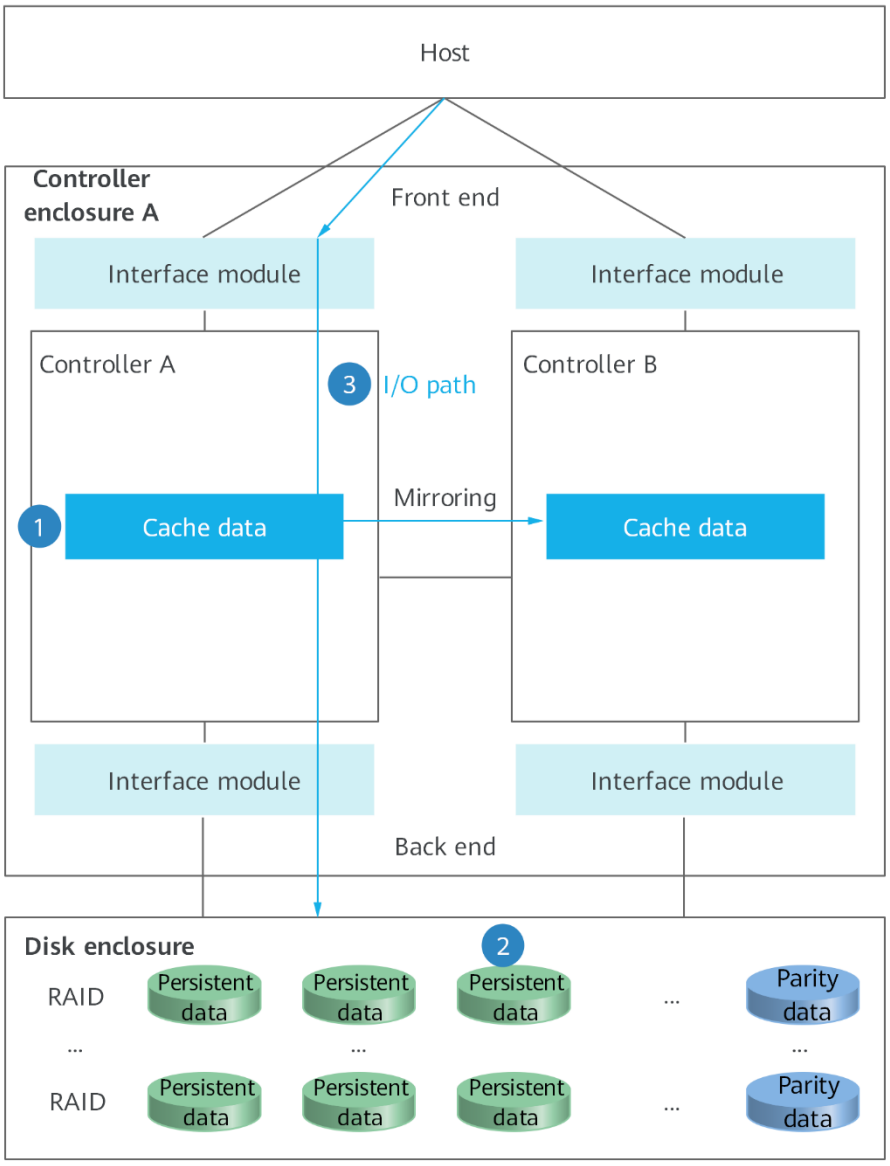
7 System Reliability Design

HyperProtect provides up to 99.9999% availability via data and service availability designs. The active-active distributed architecture ensures that services are not interrupted and faults of interface modules, controllers, and disks are imperceptible to backup applications. This ensures that each backup job can be completed within a stable time window.

7.1 Data Reliability Design

For data written by hosts to storage, HyperProtect undergoes three processes: data caching, data persisting on disks, and data transmitting on paths. The following describes data reliability measures in the three processes.

Figure 7-1 Data reliability overview



7.1.1 Cache Data Reliability

To improve the data writing speed, HyperProtect provides the write cache mechanism. That is, after data is written to the memory cache of a controller and a cross-controller copy, a success message is returned to the host and then cache data is written to disks in the background.

User data stored in the controller memory may be lost if the system is powered off or the controller is faulty. To prevent data loss, the system provides written data mirroring and power failure protection to ensure data reliability.

7.1.1.1 Written Data Mirroring

The full series of HyperProtect backup storage supports written data mirroring.

The data written by the host is first written to the power-failure-protection cache of the two controllers, and then a write success message is returned to the host. If a controller is faulty or reset, the mirrored dirty data in the other controller can be used to directly take over the dirty data of the faulty controller. This ensures that services are not interrupted and data is not lost.

7.1.1.2 Power Failure Protection

HyperProtect has built-in BBUs (backup power). If power outage occurs, BBUs in controllers provide extra power for moving cache data in the memory to the coffer. After the power supply is recovered, the system recovers the cache data in the coffer to the memory during system startup, preventing data loss.

BBUs of the HyperProtect high-end model are located on controllers. When controllers are removed, BBUs are removed along with them. After detecting that a controller is removed, the software module on the controller uses the BBU (backup power) on the controller to back up user cache data to the coffer (space for storing data in the cache in case of power failures), ensuring data reliability. The process of moving cache data to the coffer upon power failures is implemented by the underlying system and does not rely on upper-layer software, thereby not affected by services and further improving user data reliability.

7.1.2 Persistent Data Reliability

HyperProtect uses the intra-disk RAID technology to ensure disk-level data reliability and prevents data loss. As long as the number of faulty disks does not exceed that of the redundant disks, data will not be lost and the redundancy will not decrease.

7.1.2.1 Intra-Disk RAID

In addition to the failure of an entire disk, regional damage may occur on the chips used for storing data. This is called the silent failure (bad block). These bad blocks do not cause the failure of an entire disk, but cause the data access failure on the disk.

Common bad block scanning can detect silently failed data in advance and recover the data. However, disk scanning occupies a large amount of resource. To prevent impact on foreground services, the scanning speed must be controlled. Therefore, if the disk capacity and quantity are large, it takes weeks or even months to scan all disks. In addition, if both the bad block and disk failure occur in the interval between two scans, data may fail to be recovered.

Based on bad block scanning, HyperProtect uses HSSDs to provide the intra-disk RAID feature to prevent silent failures in scanning intervals. Specifically, RAID 4 groups are created for data on SSDs in the unit of dies to implement redundancy, tolerating the failure of a single die without any data loss on SSDs.

7.1.3 Data Reliability on I/O Paths

During data transmission within a storage system, data passes through multiple components over various channels and undergoes complex software processing. Any problem during this process may cause data errors. If such errors cannot be detected immediately, error data can be written to persistent disks, calculated internally, or returned to the host, causing service exceptions.

To resolve the preceding problems, HyperProtect uses the end-to-end protection information (PI) function to detect and correct data errors (internal changes to data blocks) on the transmission path. The matrix verification function ensures that changes to the whole data

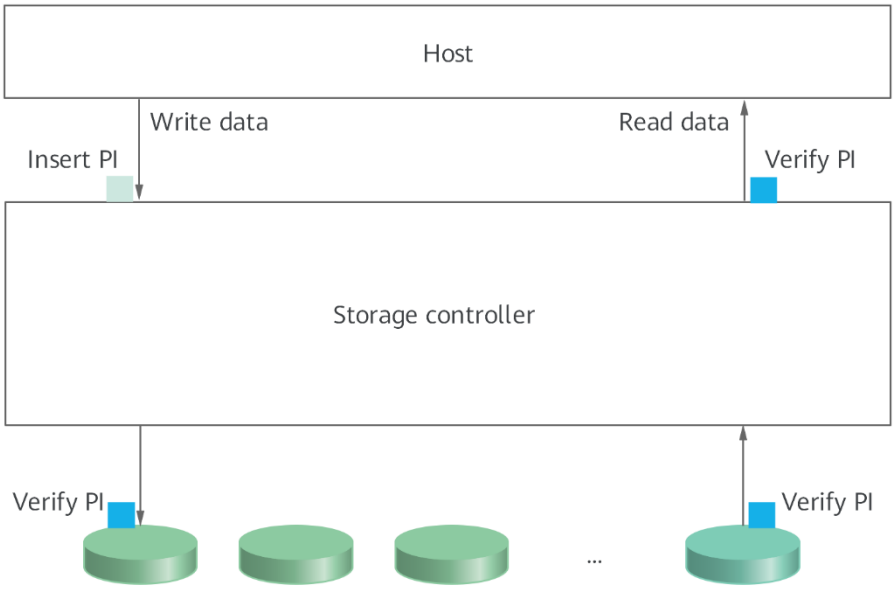
block (the whole data block is overwritten by old data or other data) can be detected. The preceding measures ensure data reliability on I/O paths.

7.1.3.1 End-to-End PI

HyperProtect uses ANSI T10 Protection Information (PI) to ensure data integrity within a storage system. Upon reception of data from a host, the storage array inserts an 8-byte PI field to every 512 bytes of data before performing internal processing.

After data is written to disks, the disks verify the PI fields of the data to detect any change to the data between reception and flushing to the disks. In the following figure, the green block indicates that a PI is inserted to the data. The blue blocks indicate that a PI is calculated for the 512-byte data and compared with the saved 8-byte PI to verify data correctness.

Figure 7-2 End-to-end PI

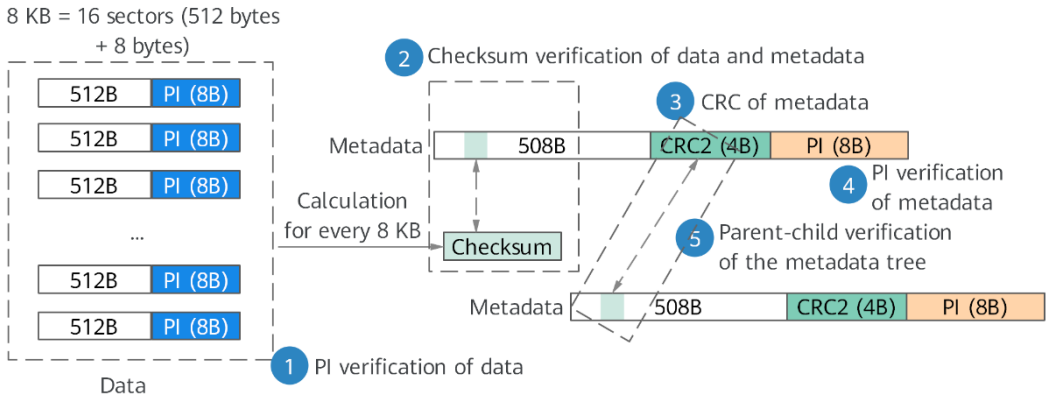


When the host reads data, the disks verify the data to prevent changes to the data. If any error occurs, the disks notify the upper-layer controller software, which then recovers the data by using RAID. To prevent errors on the path between the disks and the front end of the storage array, the storage system verifies the data again before returning it to the host. If any error occurs, the storage system recovers the data using RAID (calculates the data on the faulty disk based on other member disks of the RAID group) to ensure end-to-end data reliability from the front end to the back end.

7.1.3.2 Matrix Verification

Because the internal structure of disks is complex or the read path is long (involving multiple hardware components), various errors may occur due to software defects. For example, a write success is returned but the data fails to be written to disks (write failure); data B is returned when data A is read (read offset); or data that should be written to address A is actually written to address B (write offset). Once such errors occur, the PI verification of the data is passed. If the data is still used, the incorrect data (such as old data) may be returned to the host.

Figure 7-3 Parent-child verification



HyperProtect provides matrix verification to cope with the write failure, read offset, and write offset that may occur on disks. In the preceding figure, each piece of data consists of 512-byte user data and 8-byte PI. Two bytes of the PI are used for cyclic redundancy check (CRC) to ensure reliability of the 512-byte data horizontally (protection point 1). The CRC bytes in 16 PI sectors are extracted to calculate the checksum, which is then saved in a metadata node. If offset occurs in a single or multiple pieces of data (512+8), the checksum of the 16 pieces of data is also changed and becomes inconsistent with that saved in the metadata. This ensures data reliability vertically. After detecting data damage, the storage system uses RAID redundancy to recover the data. This is matrix verification.

7.1.4 Automatic Cross-site Data Repair

The HyperProtect backup storage system provides remote replication to ensure high data reliability. With remote data backup based on remote replication, HyperProtect can automatically recover local corrupted data by using remote backup data. This improves system data reliability and O&M efficiency.

When the end-to-end verification of a read request from a host fails or the background data consistency scanning detects a data damage and the data fails to be restored through local data redundancy, the system automatically reads backup data from the remote end using the remote replication function and returns the data to the host after the verification is successful. Then, the system uses the data read from the remote storage system to restore the damaged data on the local storage system. The entire process does not require manual intervention and does not affect host applications.

7.2 Service Availability

The storage array provides multiple redundancy protection mechanisms for the entire path from the host to the storage array. That is, when a single point of failure occurs on the interface module or link (1), controller (2), and storage media (3) that I/Os pass through, redundant components and fault tolerance measures work together to ensure that services are not interrupted. The storage media supports RAID 5, RAID 6 algorithms. Currently, enclosure redundancy is not supported.

Figure 7-4 Multi-layer redundancy and fault tolerance design (high-end)

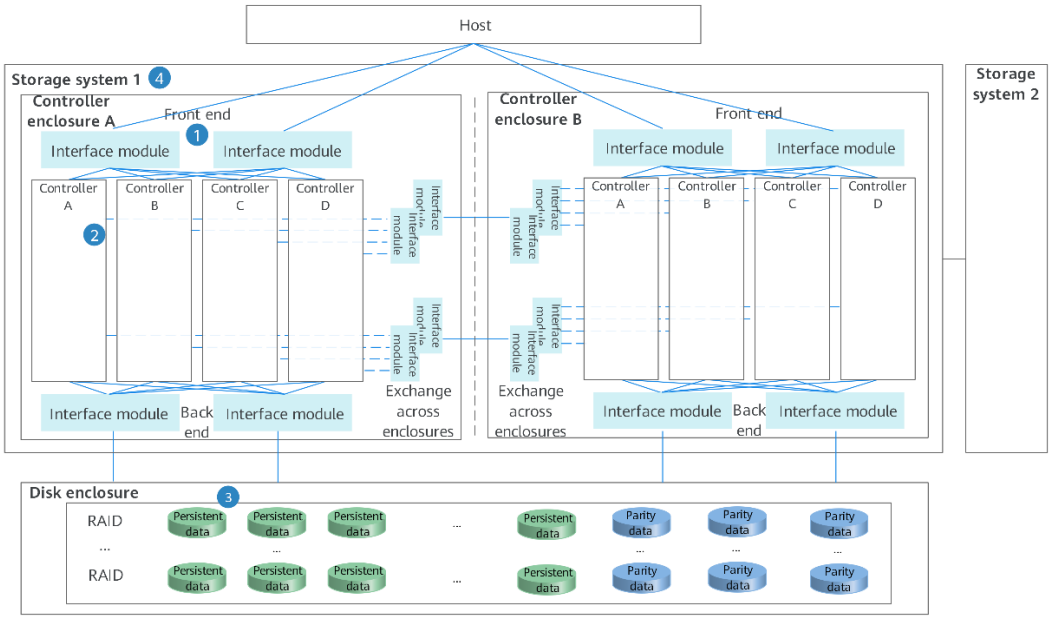
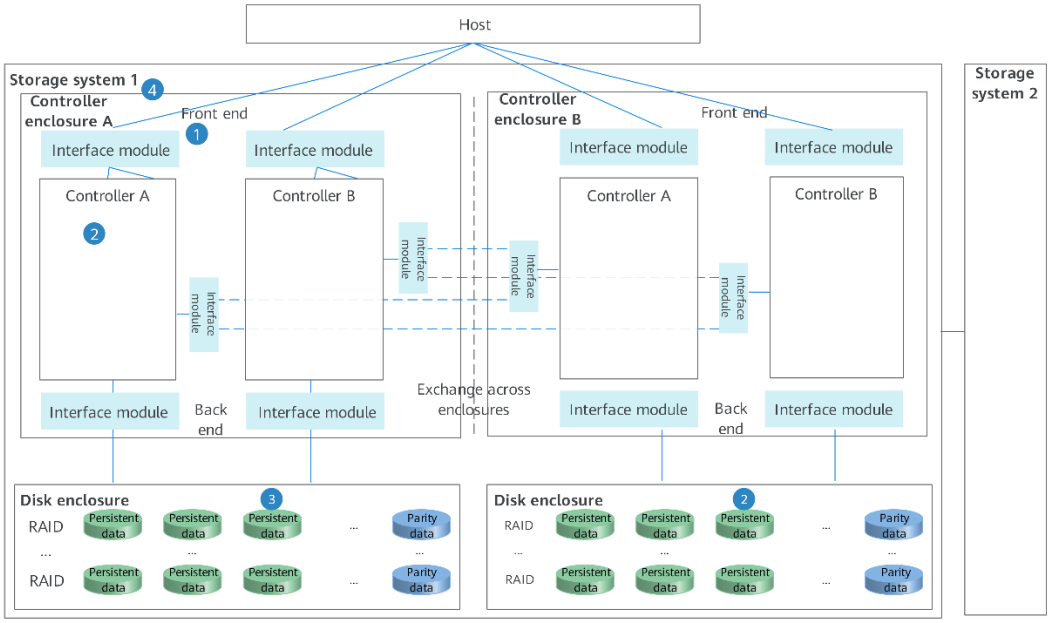


Figure 7-5 Multi-layer redundancy and fault tolerance design (mid-range)



7.2.1 Redundancy Protection on Interface Modules and Links

HyperProtect supports full redundancy. There are redundant front- and back-end interface modules connecting hosts and disks, respectively, as well as redundant links between these modules and controllers. Each front-end interconnect I/O module (FIM) of HyperProtect high-end models is connected to each controller in the controller enclosure. Under the Fibre Channel protocol, the FIM is connected to the host. If a controller is faulty or replaced, the connection between the host and the FIM is not interrupted and therefore, the connection will

not be re-established. After the remaining controllers take over services, the FIM delivers retry or new I/Os to the controllers to ensure service continuity.

7.2.2 Controller Redundancy Protection

HyperProtect provides redundant key controllers to ensure reliability. In typical scenarios, cache data is stored on the current controller and the copy of cache data is stored on another controller. If a controller fails, services can be switched to the controller to which the cache data copy belongs, ensuring service continuity.

7.2.2.1 Continuous Mirroring

Vinchin HyperProtect high-end devices provide the continuous mirroring function. For example, if controller A is faulty, controller B selects controller C or controller D as the cache mirror. If controller D fails, controller D selects controller B and controller C as the cache mirror. This enables the storage system to tolerate failure of three out of four controllers. This design ensures high availability of services in the event that multiple controllers fail successively.

If a controller fails, its services are quickly switched to the mirror controller, and the mirror relationship is re-established for the remaining controllers. The storage system continues to process service requests in write-back mode to ensure system performance, and data mirror redundancy ensures data reliability and service continuity.

7.2.3 Storage Media Redundancy Protection

HyperProtect not only ensures high reliability of a single disk, but also uses multi-disk redundancy to ensure service availability if a single disk is faulty. That is, disk faults or sub-health is detected in a timely manner by using algorithms, and faulty disks are isolated in a timely manner to avoid long-term impact on services. Then, data of faulty disks is recovered by using the redundancy technology. In this case, services can be continuously provided.

7.2.3.1 Fast Isolation of Faulty and Slow Disks

When disks are running properly, HyperProtect monitors the in-position and reset signals. If a disk is removed or faulty, the storage system isolates it. New I/Os are written to other disks and a read success is returned to the host after degraded read of I/Os.

In addition, continuous, long-time operation causes disks to wear and increases the chance of particle failures. As a result, disks respond more slowly to I/Os, which can affect services. In this case, slow disks are detected and isolated in a timely manner so that they cannot further affect services.

A model that compares the average I/O service time of disks is built for HyperProtect based on common features of disks, including the disk type, interface type, and owning disk domain. With this model, slow disks can be detected and isolated within a short period of time, shortening the time when host services are affected by slow disks.

7.2.3.2 Disk Redundancy

HyperProtect supports the following RAID configuration modes, which ensure service continuity in the event of disk failures. The system can tolerate simultaneous failure of at most three disks in a storage pool without any data loss or service interruption.

- RAID 5 uses the EC-1 algorithm and generates one copy of parity data for each stripe. The failure of one disk is allowed.

- RAID 6 uses the EC-2 algorithm and generates two copies of parity data for each stripe. The simultaneous failure of two disks is allowed.

8 Value-added Features

To meet customers' requirements for local protection and remote disaster recovery, HyperProtect provides series software. Snapshot is used to recover local logical errors, Replication is used to implement remote DR protection, and Clone is used to implement quick data cloning.

8.1 Snapshot

Snapshot is a snapshot feature of HyperProtect. It provides different snapshot functions for SAN and NAS services. SAN snapshots, readable and writable, are independent LUN objects in a host view and must be mapped separately. NAS snapshots, read-only, are deployed in the snapshot directory of a file system. They can be directly accessed after being mounted to a file system share.

8.1.1 Snapshot for SAN

This section describes the technical principles and key functions of Snapshot for SAN.

8.1.1.1 Basic Principles

HyperProtect provides readable and writable snapshots for SAN services. A SAN snapshot is accessible to a host only after being added to a mapping of the host. This section details the time point (TP) technology, which is crucial to Snapshot.

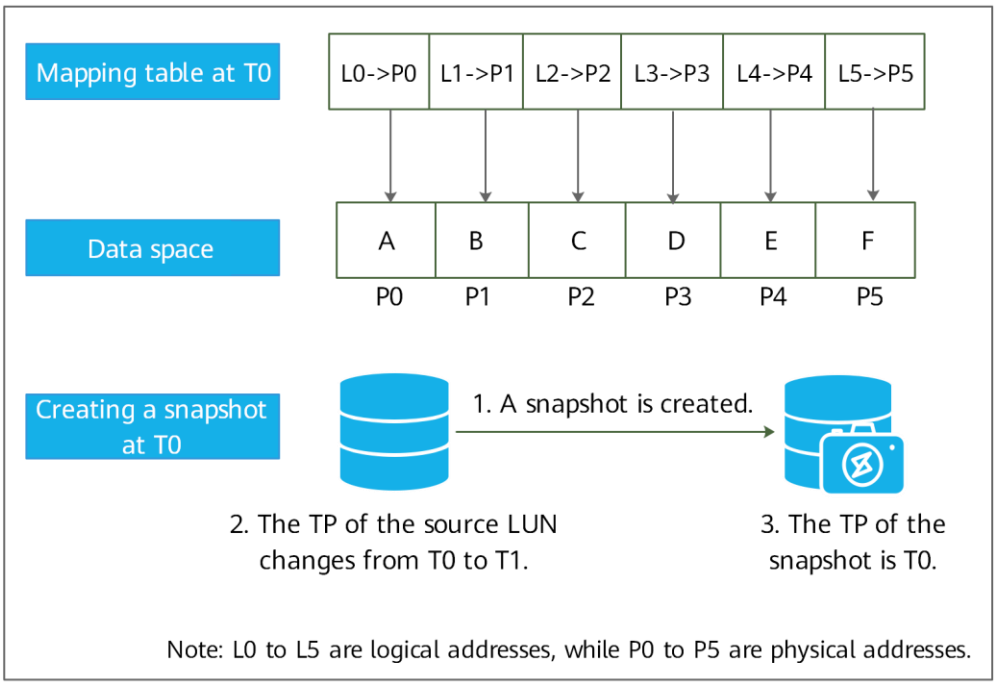
TP

HyperProtect uses the multi-TP technology to implement basic data protection features. All local and remote data protection features use this technology to obtain data copies and ensure consistency.

This technology adds a TP attribute to LUNs. When a snapshot is created for a LUN, the value of TP attribute of the source LUN is incremented, while the TP attribute of the snapshot is the original TP at the snapshot creation.

In the example of Figure 8-1, the current TP of the source LUN is T0 and the user creates a snapshot for the source LUN.

Figure 8-1 Snapshot principles

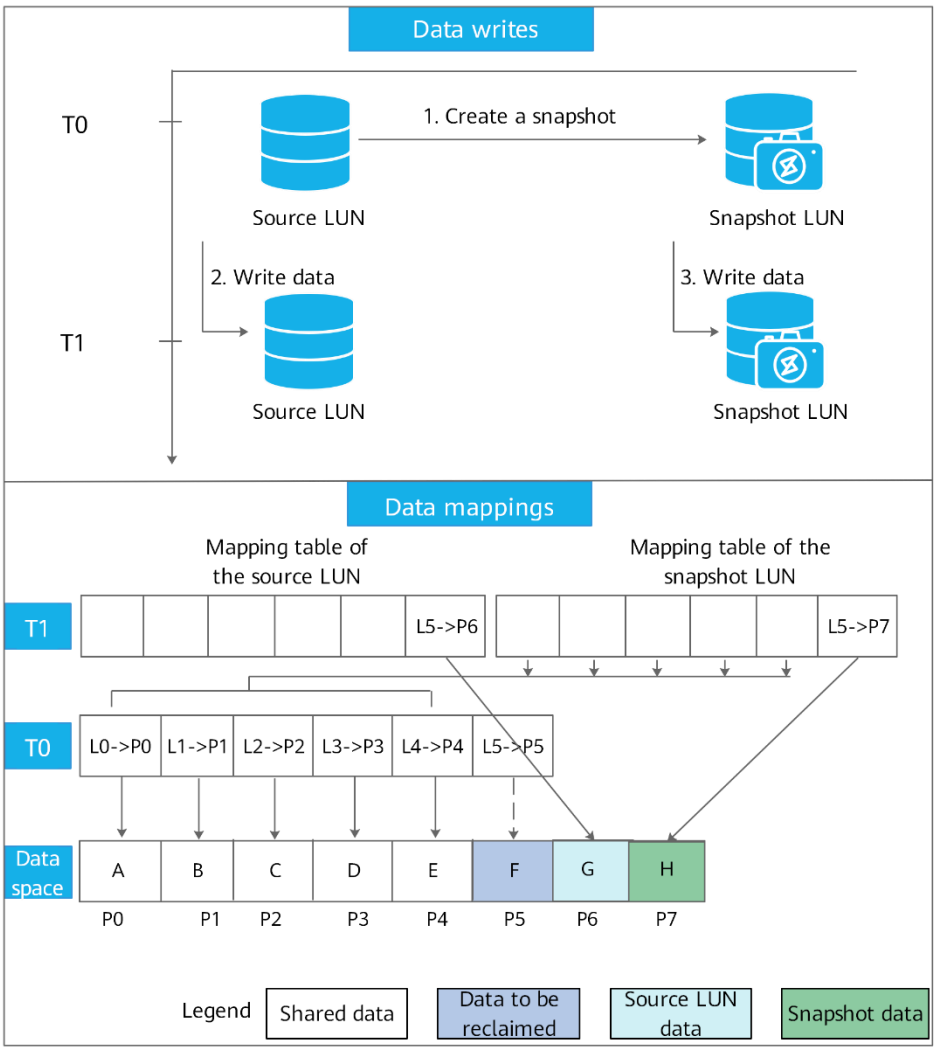


Because a snapshot is created at T0, the TP attribute of the source LUN changes from T0 to T1. The TP attribute of the snapshot is T0. When the source LUN is read, data at T1 is read; when the snapshot is read, data at T0 is read (ABCDEF in this example).

Reading and Writing a Snapshot

The host reads and writes data on the source LUN at the latest TP, or reads and writes data on the snapshot at the original TP, as shown in Figure 8-2.

Figure 8-2 Reading and writing a snapshot



- Reading from the source LUN

  After a snapshot is created for the source LUN, the TP of the source LUN is updated from T0 to T1. The read requests on the source LUN will read the data within the [T0, T1] time range on the source LUN (from the latest data to the old data according to mapping entries in the mapping table). This will not cause any new performance overhead.

- Reading from the snapshot

  The latest TP of the snapshot is T0. When the snapshot is read, the data at T0 is returned if the mapping table of the snapshot is not empty. If the mapping table of the snapshot is empty, a TP redirection is triggered and the data at T0 of the source LUN is read.

- Writing to the source LUN

  When new data is written to the source LUN, the write requests carry the latest T1. The system uses the logical address of the new data and T1 as the key, and uses the address where the new data is stored in the SSD storage pool as the key value.

- Writing to the snapshot

When new data is written to the snapshot, the write requests carry the latest T0 of the snapshot. The system uses the logical address of the new data and T0 as the key, and uses the address where new data is stored in the SSD storage pool as the key value.

Because the read and write requests on the source LUN or snapshot carry corresponding TPs, the metadata can be quickly located, minimizing the impact on performance.

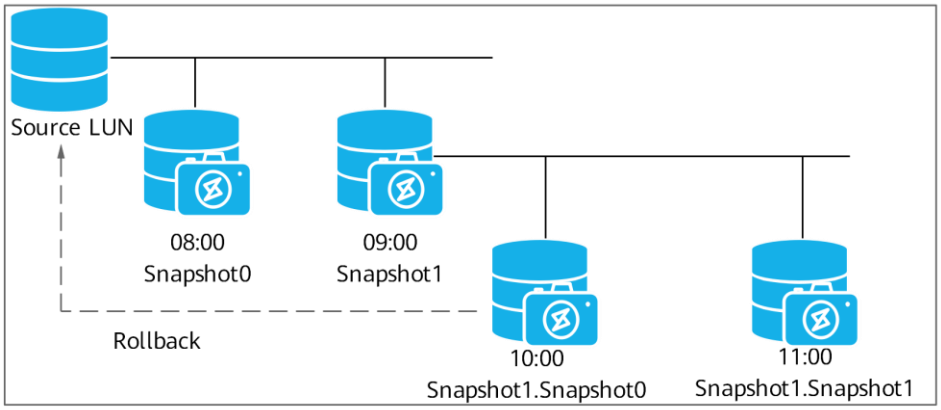
8.1.1.2 Cascading Snapshots

To protect writable snapshots, you can use cascading snapshots.

With this feature, you can create child snapshots for a parent snapshot. On HyperProtect, it supports up to eight levels of cascading snapshots.

Cascading snapshots that share a source LUN support cross-level rollback without any level constraints. In Figure 8-3, Snapshot1 is created for the source LUN at 9: 00, and Snapshot1.Snapshot0 is a cascading snapshot of Snapshot1 at 10:00. You can roll back the source LUN using the data of Snapshot1.Snapshot0 or Snapshot1. In addition, you can roll back the Snapshot1 using the data of Snapshot1.Snapshot0.

Figure 8-3 Cascading snapshots and cross-level rollback

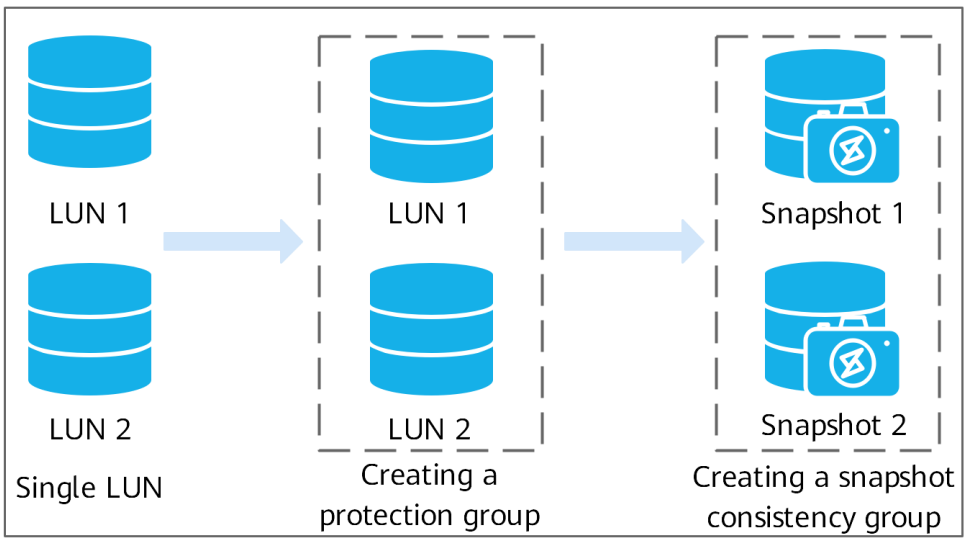


8.1.1.3 Snapshot Consistency Group

Snapshot supports snapshot consistency groups. This chapter describes cascading snapshot consistency groups.

For LUNs that are dependent on each other, you can create a snapshot CG for these LUNs to ensure data consistency. For example, the data files, configuration files, and logs of an Oracle database are usually saved on different LUNs. Snapshots for these LUNs must be created at the same time to guarantee that the snapshot data is consistent in time.

Figure 8-4 Working principles of the snapshot consistency group



1. Create a LUN protection group and add LUNs to it.
2. Create a snapshot consistency group for the protection group. The snapshots in the snapshot consistency group share the same TP.

8.1.2 Snapshot for NAS

HyperProtect uses the redirect-on-write (ROW) mechanism to create snapshots for file systems. ROW allows the file system to write newly created or modified data to a new space instead of overwriting the original data, ensuring high reliability and scalability of the file system. With ROW, file system snapshots can be created in seconds and do not occupy additional space unless the source files are deleted or modified.

A file system snapshot only copies and stores the root node of the file system. No user data is copied, so creation can be completed in 1 to 2 seconds. Before data is modified, the snapshot file set and the source file system share the same file system space, so no additional space is required by the snapshot.

Because the newly created snapshot contains no data except a group of pointers pointing to the source file system, users access data in the source file system when they access the snapshot data. The snapshot retains its unique data only when the source file system data is modified. As the unique data is protected by the snapshot, the data can be deleted to release the space only after the snapshot is deleted.

As updates to the source file system accumulate, original data blocks in the source file system are successively copied to the snapshot fileset. However, new data does not occupy the space in the snapshot because the snapshot contains only the image of the source file system when

the snapshot is created. When you want to recover data at the snapshot point in time, you can use the rollback function to immediately recover the data in the file system to the state at the snapshot point in time, offsetting data damage or data loss from the damaged source file system caused by misoperations or viruses after the snapshot point in time.

Exercise caution when performing snapshot rollback because its changes are irreversible. A rollback can restore data to a specified point in time, but data written after that time will be lost with the snapshot after a rollback. If several specific files are damaged or incorrectly modified or deleted, you do not need to roll back the entire file system. Instead, you can manually copy these files from the snapshot at a specific point in time to the source file system. Because snapshot rollback will cause the loss of file system data or the snapshot, services may be interrupted if you are accessing the data during the rollback. Exercise caution when performing snapshot rollback.

A snapshot created with a specified security attribute is a secure snapshot. Users cannot delete a secure snapshot during its validity period. To delete a secure snapshot during its validity period, the user must enter the serial port mode and log in as a security administrator.

8.2 Remote Replication

Replication is a remote replication feature provided by HyperProtect to implement asynchronous data replication (Replication/A), supporting intra-city and remote disaster recovery solutions.

HyperProtect remote replication refers to asynchronous remote replication.

As for asynchronous remote replication, data on the primary LUN is periodically synchronized to the secondary LUN. Production service performance is not affected by the data transfer latency. However, some data may lose if a disaster occurs.

Replication provides the storage array-based consistency group function for asynchronous remote replication to ensure the crash consistency of cross-LUN applications in disaster recovery replication. The consistency group function protects the dependency of host write I/Os across multiple LUNs, ensuring data consistency between secondary LUNs at the DR site.

Replication enables data to be replicated using Fibre Channel and IP networks including IPv4 and IPv6. Data can be transferred between the primary and secondary storage systems through Fibre Channel or IP links.

8.2.1 Remote Replication for SAN

The working principles of the technology are as follows:

1. After an asynchronous remote replication relationship is established between a primary LUN at the primary site and a secondary LUN at the remote replication site, an initial synchronization is implemented by default to copy all data from the primary LUN to the secondary LUN.
2. When the initial synchronization is complete, the data status of the secondary LUN becomes Consistent (data on the secondary LUN is a copy of data on the primary LUN at a certain point in time in the past). Then I/Os are processed as shown in the following figure.

Technical highlights:

- Load balancing

  When the copy task is started in remote replication for a LUN, the task is executed by all controllers concurrently. The workload of the task is distributed to all controllers based on the data layout, improving the replication bandwidth and reducing the impact on front-end services.

- Data compression

  In asynchronous remote replication, both Fibre Channel and IP links support data compression by using the fast LZ4 algorithm. Link compression can be enabled or disabled as required. Data compression reduces the bandwidth required by asynchronous replication. In the testing of a database application with 100 Mbit/s bandwidth, data compression saves half of the bandwidth.

- Quick response to host requests

  In asynchronous remote replication, after a host writes data to the primary LUN at the primary site, the primary site immediately returns a write success to the host before the data is written to the secondary LUN. In addition, data is synchronized from the primary LUN to the secondary LUN in the background, which does not affect the access to the primary LUN. Replication does not synchronize incremental data from the primary LUN to the secondary LUN in real time. Therefore, the amount of data loss depends on the synchronization interval (ranging from 3 seconds to 1440 minutes; 30 seconds by default), which can be specified based on site requirements.

- Splitting, primary/secondary switchover, and rapid fault recovery

  Replication supports splitting, synchronization, primary/secondary switchover, and recovery after disconnection. Disaster recovery tests and service switchover are supported in various fault scenarios.

- Consistency group

  Consistency groups are supported in database scenarios. Multiple LUNs, such as log LUNs and data LUNs, can be added to a consistency group so that data on these LUNs is from a consistent time in the case of periodic synchronization or fault. This facilitates data recovery at the application layer.

- Various supported replication links

  Replication supports both Fibre Channel and IP replication links.

- Cross-array snapshot data synchronization

  User snapshots of the primary LUN can be synchronized to the secondary LUN in sequence. This function is recommended when host data consistency is required, ensuring transaction consistency between snapshot data and host service logic.

8.2.2 Remote Replication for NAS

8.2.2.1 Working Principles of NAS Replication

Similar to asynchronous replication for SAN, asynchronous remote replication for NAS periodically replicates data based on ROW snapshots. The difference lies in the service objects.

The data written by the host is periodically replicated to the secondary file system in the background. Replication periods are defined by users. The system records the addresses of incremental data in each period but does not record the data content. At the beginning of each period, a snapshot is created for the primary file system. The system reads the incremental data since last replication and replicates it to the secondary file system. After the incremental replication is complete, the data in the secondary file system is consistent with that in the primary file system. In each periodic replication, the secondary file system is inconsistent

before all incremental data is replicated. Then, a snapshot is created for the secondary file system when each periodic replication is complete and the secondary file system forms a data consistency point. If the next periodic replication is interrupted by a production site fault or link failure, the system, through file system asynchronous remote replication (Replication for File System), uses the snapshot to roll back the secondary file system to the state at the last successful replication for guaranteed data consistency. Replication for File System provides replication link deduplication, splitting, incremental resynchronization, interruption upon a fault, and automatic restoration.

8.2.2.2 Deduplicated Replication

As a typical backup scenario, NAS asynchronous replication features high data duplication ratio in multiple replication jobs. In a network environment with limited resources, highly redundant data must be transmitted at a high speed to ensure data security. It poses an ultimate requirement on the link replication capability, that is, minimizing duplicate data transmission to reduce the required network bandwidth, reducing WAN usage costs, and accelerating replication. The deduplication technology is indispensable to eliminate duplicate data. If the data deduplication technology can be used in replication links, the replication efficiency will be greatly improved, bringing great benefits to customers. Therefore, HyperProtect introduces and optimizes the data deduplication technology to implement deduplicated replication, significantly reducing the amount of data copied in replication jobs and saving bandwidth resources required for replication jobs.

Technical highlights:

1. Deduplication of redundant data ensures higher copy bandwidth and improves service data security.

2. Data compression is supported. After deduplication, data can be compressed to further reduce the amount of data to be sent.

3. It enables customers to back up main storage data even in limited network environments.

8.3 Clone Function

Clone on Vinchin HyperProtect backup storage products allows the system to create a complete physical copy of the source LUN's data on the target LUN. The target LUN can be an existing one or automatically created when the Clone pair is created. The source and target LUNs that form a Clone pair must have the same capacity. The target LUN can either be empty or have existing data. If the target LUN has data, the data will be overwritten by the source LUN of Clone. After the Clone pair is created, you can synchronize data. During the synchronization, the target LUN can be read and written immediately, even if the synchronization is suspended. Clone supports incremental synchronization and recovery. You can create a Clone consistency group using a LUN protection group to protect data consistency on a group of source LUNs.

8.3.1 Data Synchronization from the Source LUN to the Target LUN

When a clone pair starts synchronization, the system generates an instant snapshot for the source LUN, and then synchronizes the snapshot data to the target LUN. Any subsequent write operations are recorded in the DCL. When synchronization is performed again, the system compares the data of the source and target LUNs, and only synchronizes the differential data to the target LUN. The data written to the target LUN between the two

synchronizations will be overwritten. To retain the existing data on the target LUN, you can create a snapshot for it before synchronization.

8.3.2 Reverse Synchronization

If the source LUN is damaged, data on the target LUN can be recovered to the source LUN. Both full and incremental reverse synchronizations are supported. When recovery starts, the system generates a snapshot for the target LUN and synchronizes the snapshot data to the source LUN. For incremental recovery, the system compares the data of the source and target LUNs, and only synchronizes the differential data.

8.3.3 Immediately Available Clone LUNs

The Clone data synchronization status includes **Synchronizing**, **Sync paused**, **Unsynchronized**, or **Normal**. In different states, the read and write I/Os of the source LUN and target LUN are processed in different ways.

1. When Clone is in the normal or unsynchronized state:

   The host reads and writes the source or target LUN directly.

2. When Clone is in the synchronizing or paused state:

   The host reads and writes the source LUN directly.

   For read operations on the target LUN, if the requested data is found on the target LUN (the data has been synchronized), the host reads the data from the target LUN. If the requested data is not found on the target LUN (the data has not been synchronized), the host reads the data from the snapshot of the source LUN.

   For write operations on the target LUN, if a data block has been synchronized before the new data is written, the system overwrites this block. If a data block has not been synchronized, the system writes the new data to this block and stops synchronizing the source LUN's data to it. This ensures that the target LUN can be read and written before the synchronization is complete.

8.3.4 Clone Consistency Group

Clone allows you to create a consistency group for a LUN protection group. A Clone consistency group contains multiple Clone pairs. When you synchronize or restore a consistency group, data on all of its member LUNs is always at a consistent point in time, ensuring data integrity and availability.

9 System Serviceability Design

This chapter describes how to manage the HyperProtect storage through various interfaces (including DeviceManager, CLI, RESTful API, SNMP, and SMIS) and describes the upgrade mode transparent to hosts.

9.1 System Management

HyperProtect provides device management interfaces and integrated northbound management interfaces. Device management interfaces include a graphic management interface DeviceManager and a command-line interface (CLI). Northbound interfaces are RESTful APIs, supporting SNMP, evaluation tools, and third-party network management plug-ins.

9.1.1 DeviceManager

DeviceManager is a built-in HTML5-based management system for HyperProtect. It provides wizard-based GUI for efficient management. Users can enter https://*Storage management IP address*:8088/ on the browser to use DeviceManager. On DeviceManager, you can perform almost all required configuration and management operations.

You can use the following functions on DeviceManager:

- Storage space management: includes management of the storage pool, LUNs, and mappings between LUNs and hosts.
- Data protection management: uses snapshots and replication to protect data.
- Configuration task: provides background tasks for complex configuration operations to trace the procedure of the configuration process.
- Fault management: monitors the status of storage devices and management units on storage devices. If faults occur, alarms will be generated and troubleshooting suggestions and guidance will be provided.

- Performance and capacity management: supports real-time performance and capacity monitoring, historical performance and capacity data collection and query, and performance data association analysis.
- Security management: supports the management of users, roles, permissions, certificates, and keys.

DeviceManager provides a simple interactive interface. Users can complete configuration tasks only in a few operations, improving user experience.

9.1.1.1 Storage Space Management

9.1.1.1.1 Flexible Storage Pool Management

HyperProtect manages storage space by using storage pools. A storage pool consists of multiple SSDs and can be divided into multiple LUNs or file systems for hosts to use. You can use one storage pool to manage all of the space or create multiple storage pools.

- The entire storage system uses only one storage pool.

  This is the simplest method. You only need to create one storage pool with all disks during system initialization.
- Multiple storage pools are divided to isolate different applications.

  If you want to use multiple storage pools to manage space and isolate fault domains of different applications, you can manually create storage pools at any time in either of the following ways: You can specify number of disks used to create a storage pool, and the system automatically selects the disks that meet the requirements. Alternatively, you can manually select specific disks to create a storage pool.

9.1.1.2 Configuration Task

DeviceManager automatically creates a background configuration task after a user submits a complex configuration operation. The task is running on the storage background. In this way, the user can perform other operations while the task is still being executed.

For example, the background configuration task mechanism applies to protection group-based protection, and cross-device protection. Suppose that a user needs to configure protection for a protection group that has hundreds of LUNs. It takes a long time and hundreds of operations to complete the configuration. However, the background configuration task mechanism helps the user complete the configuration on the background, freeing the user from long-term waiting.

- Task steps and progress

  A complex task usually contains multiple executable steps. You can check which task step is being executed and view the overall task execution progress (%) on DeviceManager.
- Tasks can be executed only in the background

  After a configuration task is submitted, the task is automatically executed in the background. You can close the DeviceManager page without waiting for the task to complete. You can also submit multiple configuration tasks. If the resources on which the tasks depend do not conflict, the tasks are automatically executed in sequence in the background.
- Resuming tasks from the breakpoint

  If the system is powered off unexpectedly during the task execution, the task will be executed at the breakpoint and all the steps will be performed after the system is restarted.

- Retrying failed tasks manually

  If an exception occurs during the execution of a step, the task is automatically interrupted and the cause is displayed. For example, a task cannot be executed because the storage space is insufficient during files system creation. If you manually rectify the fault that causes task interruption, you can manually restart the task after the fault is rectified and the task continues from the break point.

9.1.1.3 Fault Management

9.1.1.3.1 Monitoring Status of Hardware Devices

This function provides hardware views in a what-you-see-is-what-you-get manner and uses colored digits to make status of different hardware more distinguishable. 0 shows the status statistics of hardware components.

You can further navigate through a specific device frame and its hardware components, and view the device frame in the device hardware view. In the hardware view, you can monitor the real-time status of each hardware component and learn the physical locations of specific hardware components (such as ports and disks), facilitating hardware maintenance.

You can query the real-time health status of the disks, ports, interface modules, fans, power modules, BBUs, controllers, and disk enclosures. In addition, disks and ports support performance monitoring. You can refer to 9.1.1.4 Performance and Capacity Management for more details.

9.1.1.3.2 Alarm and Event Monitoring

This function provides you with real-time fault monitoring information. If a system fault occurs, the fault is pushed to the home page of DeviceManager in real time.

A dedicated page is available for you to view information about all alarms and events and also provides you with troubleshooting suggestions.

You can also receive alarm and event notifications through syslog, email, and SMS (a dedicated SMS modem is required).

- Allows users to configure a batch of recipient email addresses for receiving alarm notifications at the same time.
- Allows users to configure a batch of phone numbers for receiving alarm notifications at the same time.

9.1.1.4 Performance and Capacity Management

Performance data collection and analysis are essential to daily device maintenance. Because the performance data volume is large and data analysis consumes many system resources, an extra server is often required for installing dedicated performance data collection and analysis software, making performance management complex.

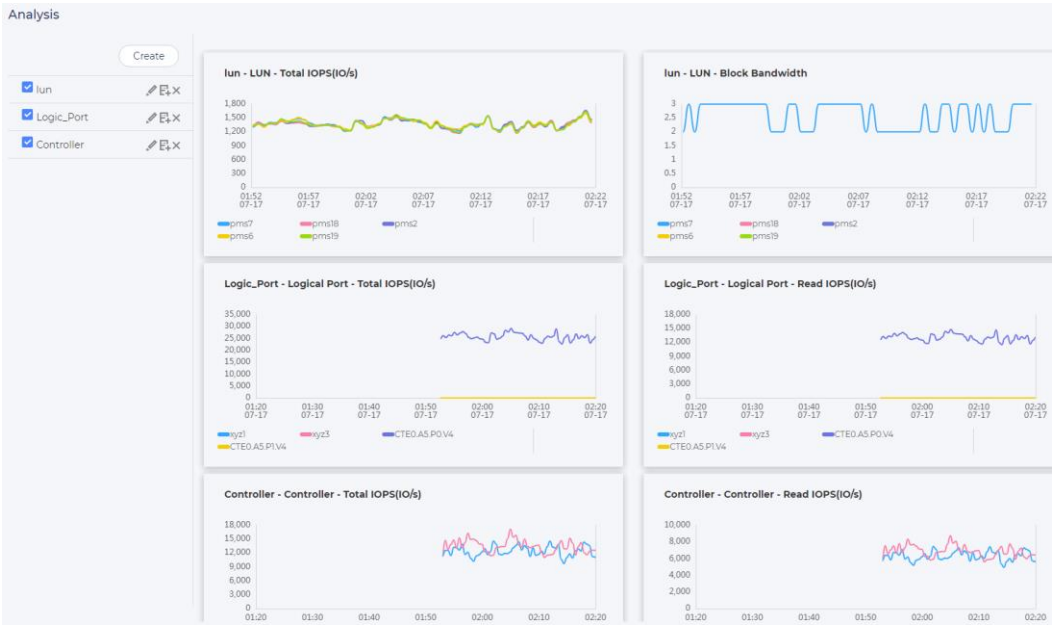
HyperProtect has a built-in performance and capacity data collection and analysis component that is ready for use. The component is specially designed to consume minimal system resources.

9.1.1.4.1 Built-In Performance Data Collection and Analysis Capabilities

HyperProtect has built-in performance collection and analysis software. You do not need to install the software separately. You can collect, store, and query historical performance and capacity data of a maximum of the past three years. Alternatively, you can specify the

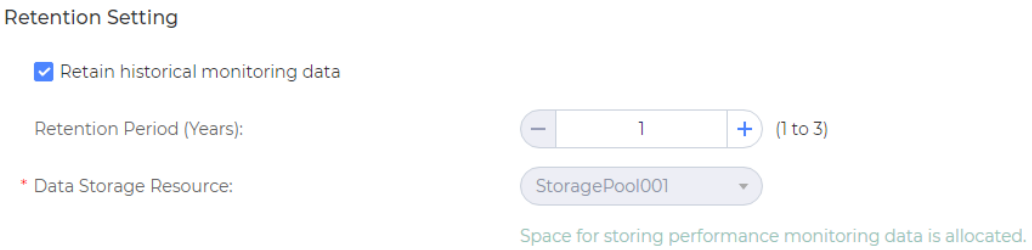
performance data retention period as required. Performance data of multiple specific objects can be collected and analyzed, such as controllers, ports, disks, file systems, remote replication, and replication links. Multiple performance indicators can be monitored for each object, such as bandwidth, IOPS, average I/O response time, and usage.

Different objects and performance indicators can be displayed in the same view to help you analyze performance issues. You can specify the desired performance indicators to analyze the top or bottom objects, so that you can locate overloaded objects more efficiently and tune performance more precisely. The following figure shows performance monitoring. For details about monitoring objects, monitoring indicators, and performance analysis functions, see the online help.



9.1.1.4.2 Independent Data Storage Space

To store collected performance and capacity data, a dedicated storage space is required. DeviceManager provides a dedicated configuration page for users to select a storage pool for storing data.



You can also customize the data retention period. The maximum retention period is three years. DeviceManager automatically calculates the required storage space based on your selection and allocates the required storage space in the storage pool.

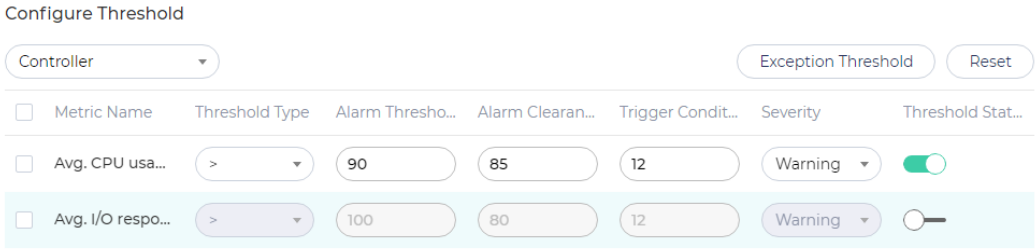
9.1.1.4.3 Capacity Prediction

DeviceManager supports the prediction of system and storage pool capacity usage in the next year. Users can formulate a better resource usage plan in advance and schedule resources based on the prediction results.

9.1.1.4.4 Performance Threshold Alarm

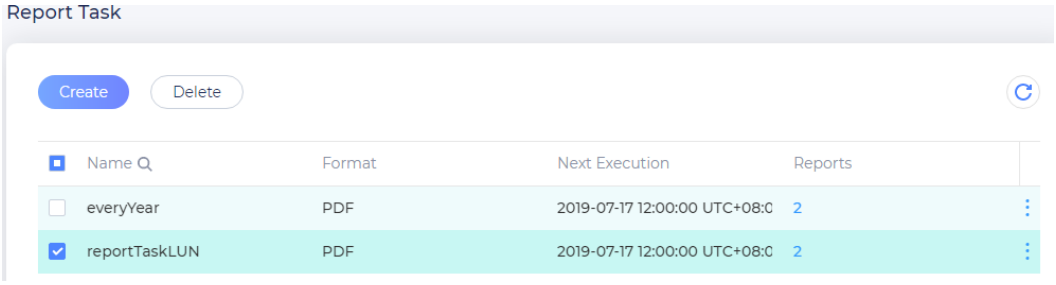
This function allows you to configure threshold alarms for objects such as controllers, ports, file systems, and replication. The alarm threshold, flapping period, and alarm severity can be customized.

Different storage resources carry different types of services. Therefore, common threshold alarms may not meet requirements. Performance management allows you to set threshold rules for specified objects to ensure high accuracy for threshold alarms.



9.1.1.4.5 Scheduled Report

Performance and capacity reports for specific objects can be generated periodically. Users can learn about the performance and capacity usage of storage devices periodically.



Reports can be generated by day, week, or month. You can set the time when the reports are generated, the time when the reports take effect, and the retention period of the reports. You can select a report file format. Currently, the *.pdf and *.csv formats are supported.

You can select the objects for which you want to generate a performance report. All the objects for which you want to collect performance statistics can be included in the report. You can also select the performance indicators to be displayed in the report. The capacity report collects statistics on the capacity of the entire system and storage pools.

You can create multiple report tasks. Each report task can be configured with its own parameters. The system automatically generates reports according to the task requirements.

9.1.2 CLI

The Command Line Interface (CLI) allows administrators and other system users to manage and maintain the storage system. It is based on the secure shell protocol (SSH) and supports key-based SSH access.

9.1.3 RESTful API

RESTful APIs of HyperProtect allow system automation, development, query, and allocation based on HTTPS interfaces. With RESTful APIs, you can use third-party applications to control and manage arrays and develop flexible management solutions for HyperProtect.

9.1.4 SNMP

The storage system reports alarms and events through SNMP traps and allows you to use SNMP interfaces to query storage alarms and events, collect data performance, and query information of objects managed on the storage system.

9.1.5 SMI-S

The Storage Management Initiative Specification (SMI-S) is a storage standard management interface developed and maintained by the Storage Networking Industry Association (SNIA). Many storage vendors participate in defining and implementing SMI-S. The SMI-S interface is used to configure storage hardware and services. Storage management software can use this interface to manage storage devices and perform standard management tasks, such as viewing storage hardware, storage resources, and alarm information. HyperProtect supports SMI-S 1.6.1, 1.6.0, and 1.5.0.

9.1.6 Tools

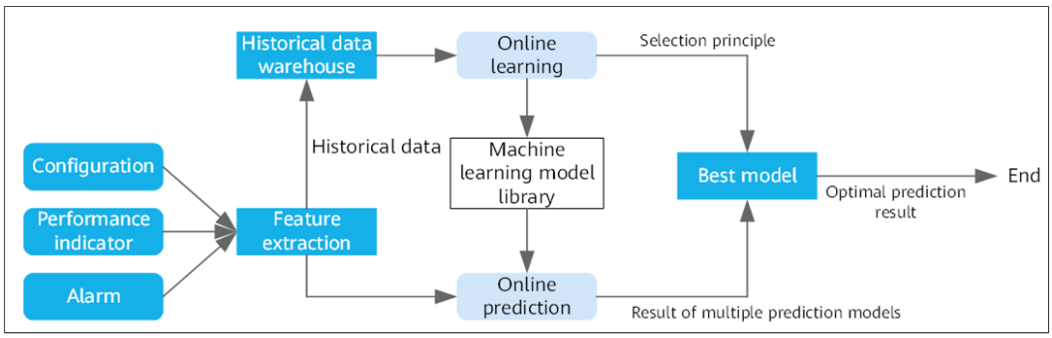
HyperProtect provides diversified tools for pre-sales assessment and post-sales delivery. These tools effectively and quickly help deploy, monitor, analyze, and maintain HyperProtect.

9.1.7 Capacity Prediction

The system capacity changes are affected by multiple factors. The traditional single prediction algorithm cannot ensure the accuracy of prediction results. HyperProtect ensures rationality and accuracy of prediction results from the following aspects:

- HyperProtect uses multiple prediction model clusters for online prediction, outputs the prediction results of multiple models, and then selects the best prediction results based on the selection rules recommended by online prediction. At the same time, based on the historical capacity data, HyperProtect periodically trains and verifies itself in the light of linear prediction, recognizable trends, periods, and partial changes to optimize the model parameters, ensuring that the optimal prediction algorithm is selected.
- The prediction algorithm model can accurately identify various factors that affect capacity changes, for example, sudden capacity increase and decrease caused by major events, irregular trend caused by capacity reclamation of existing services, and capacity hops caused by new service rollout. In this way, the system capacity consumption can be predicted more accurately.

Figure 9-1 Working principles of capacity prediction



Responsibilities of each component:

- Data source collection: collects configurations, performance indicators, and alarm information to reduce the interference of multiple factors on machine learning and training results.
- Feature extraction: uses algorithms to transform and extract features automatically.
- Historical database warehouse: stores historical capacity data of the latest year.
- Online training
  - Uses a large number of samples for training to obtain the measurement indicator statistics predicted by each model and output the model selection rules.
  - For the current historical data, performs iteration for a limited number of times to optimize the model algorithm.
- Machine learning model library: includes ARIMA, fbprophet, and linear prediction models.
- Online prediction: performs prediction online using the optimized models and outputs the prediction results of multiple models and the mean absolute percentage error (MAPE) values of measurement indicators.

$$\mathrm{M} = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

  In the preceding formula, At indicates the actual capacity and Ft indicates the predicted capacity.
- Best model selection: weights the model statistics of online training and results of online prediction, and selects the optimal prediction results.

9.1.8 Disk Health Prediction

Disks are the basis of a storage system. Although various redundancy technologies are used in storage systems, they only tolerate the failure of a limited number of disks. For example, RAID 5 tolerates the failure of only one disk. If two disks fail, the storage system stops providing services to ensure data reliability. Disks are the largest consumables in a storage system. Therefore, disk service life is the most concerned topic for many users. SSDs are electronic components and their service life prediction indicators are limited. In addition, the number of read and write requests varies every day, which further complicates disk service life prediction.

HyperProtect collects the Self-Monitoring, Analysis and Reporting Technology (S.M.A.R.T.) information, I/O link information, as well as reliability indicators of HSSDs, and enters such information to hundreds of HSSD failure prediction models, implementing accurate SSD service life prediction. HyperProtect uses intelligent AI algorithms to predict disk risks, detect failed disks, and replace risky disks in advance, preventing faults and improving system reliability.

- Data source collection

  Disk vendors provide the S.M.A.R.T. data of disks. The S.M.A.R.T. data can indicate the running status of the disks and help predict risky disks in a certain range. However, it is difficult to ensure the accuracy of prediction results. HyperProtect uses AI to dynamically analyze S.M.A.R.T. changes of disks, performance indicator fluctuations, and disk logs, ensuring more accurate prediction results.
- S.M.A.R.T

For SSDs, the interfaces provide SCSI log page information that records the current disk status and performance indicators, such as the grown defect list, non-medium error, and read/write/verify uncorrected errors.

- Performance indicators

  Workload information such as the average I/O size distribution per minute, IOPS, bandwidth, and number of bytes processed per day, as well as performance indicators such as the latency and average service time are included.

- Disk logs

  I/O error codes collected by storage systems, DIF errors, degradation errors, slow disk information, slow disk cycles, and disk service life information are included.

- Feature extraction

  Based on massive historical big data, feature transformation and feature extraction are automatically performed using algorithms.

- Analysis platform
  - Online training: Trainings is performed based on model algorithms, and these model algorithms are optimized through iteration of a limited number of times.
  - Online prediction: Optimized training models are used to predict disk failures.

- Prediction results

  HyperProtect tests and verifies massive SSD data and can accurately predict the SSD service life based on disk failure prediction models.

9.1.9 Device Health Evaluation

Generally, after a device goes online, users perform inspection to prevent device risks, which has two disadvantages:

- Inspection frequency

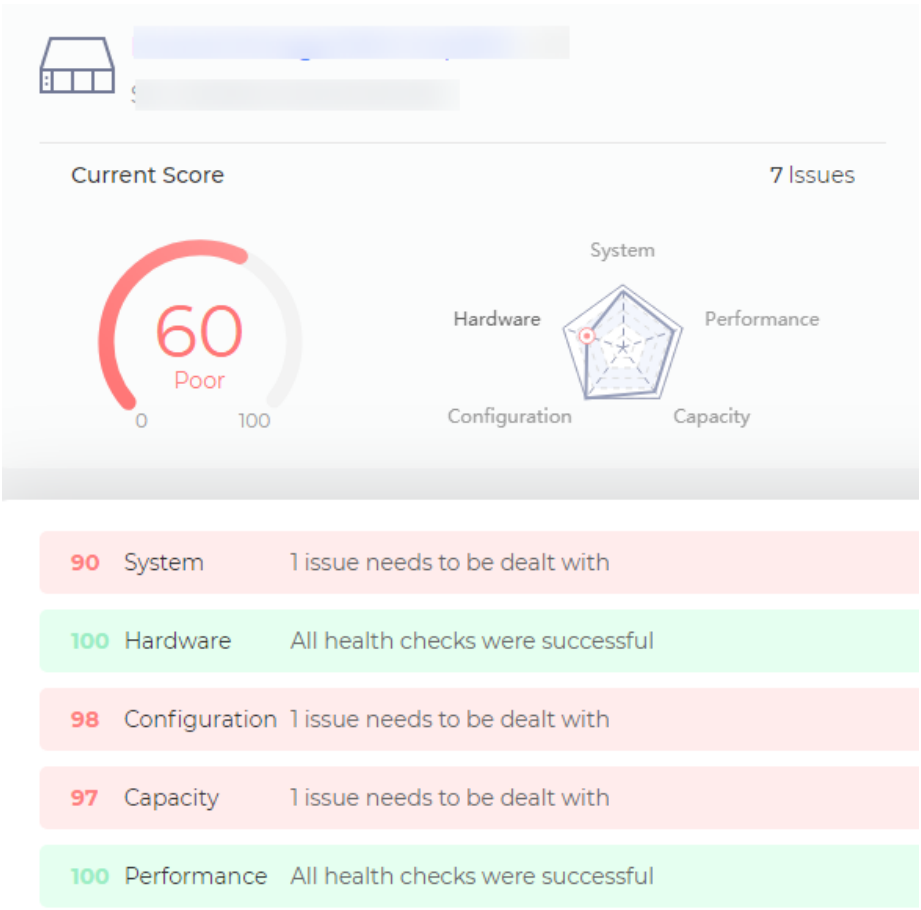
  Inspection is generally performed monthly or quarterly. As a result, users cannot detect problems in a timely manner.

- Inspection depth

  Users can only check whether the current device is faulty. System, hardware, configuration, capacity, and performance risks are not analyzed.

HyperProtect evaluates device health in real time from system, hardware, configuration, capacity, and performance dimensions, detects potential risks, and displays device running status based on health scores. In addition, HyperProtect provides solutions to prevent risks.

Figure 9-2 Device health evaluation details

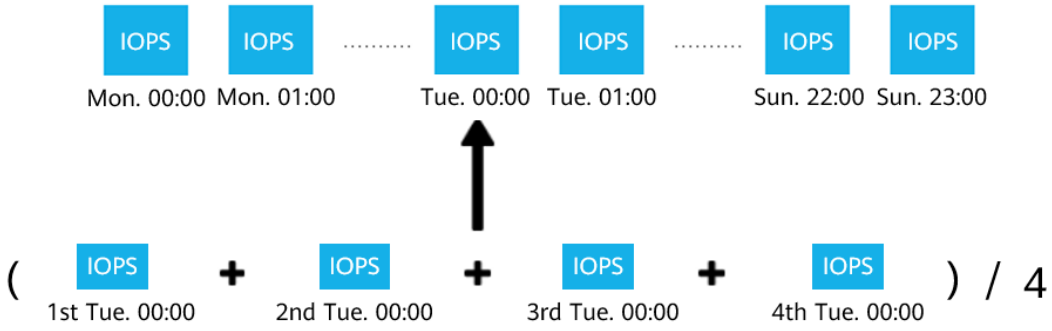


9.1.10 Performance Fluctuation Analysis

Periodic service operations or temporary changes (such as online upgrade, capacity expansion, and parts replacement) should be performed during off-peak hours to avoid affecting online services. Traditionally, O&M personnel estimate the proper time window according to experiences or through performance indicators of a past period of time.

Based on historical device performance data, HyperProtect analyzes performance fluctuations from the aspects of load, IOPS, bandwidth, and latency. Users can view the service period rules from the four dimensions and select a proper time window to perform periodic operations (such as scheduled snapshot) or temporary service changes (such as online upgrade, capacity expansion, and parts replacement) to prevent impact on services during peak hours.

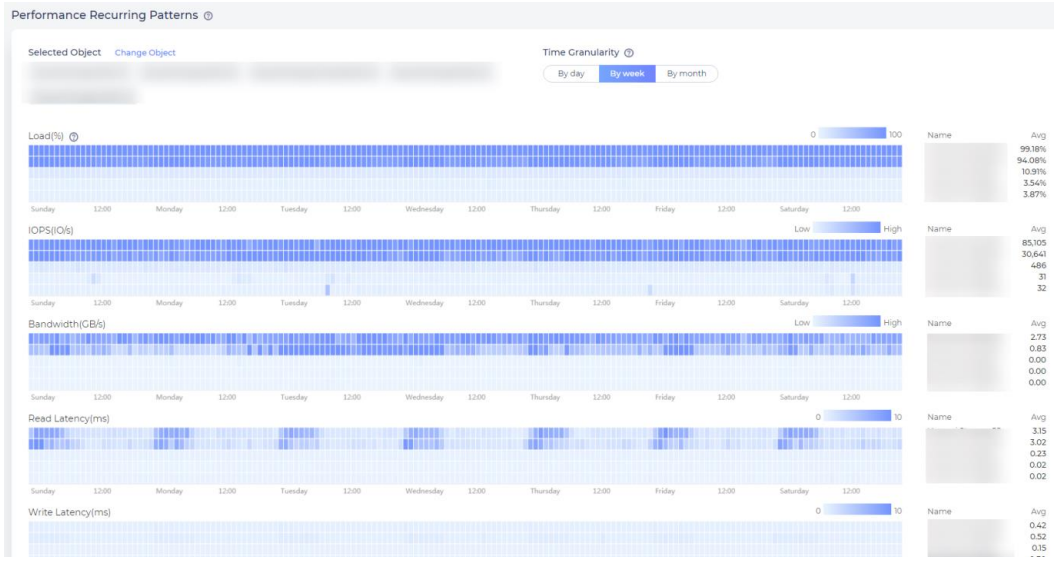
Figure 9-3 Working principles of weekly performance fluctuation analysis



HyperProtect calculates performance indicator values of each hour from Monday to Sunday based on performance data in the past four weeks. For example, the IOPS is calculated as follows: Calculate the sum of the IOPS on each hour in the past four weeks and then divide the sum by 4 to obtain the weekly performance fluctuation. For daily and monthly calculation, the methods are similar.

Users can view the service performance statistics by day, week, or month as required, as shown in the following figure.

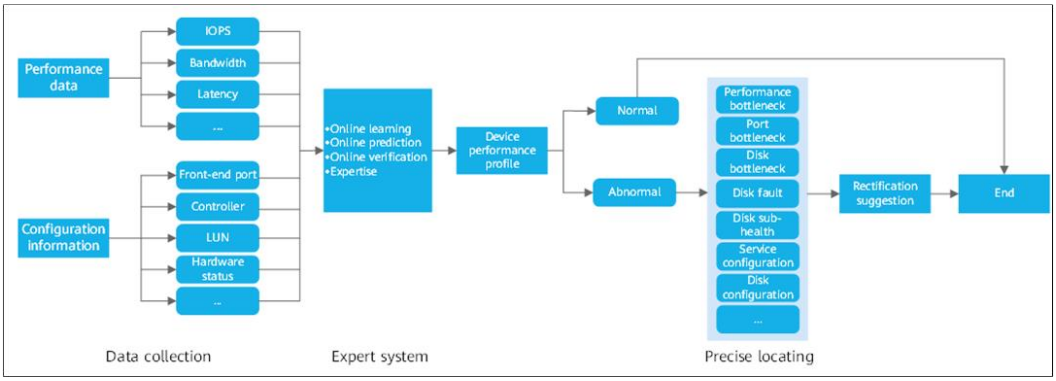
Figure 9-4 Weekly performance fluctuation



9.1.11 Performance Exception Detection

Enterprises are concerned most about service running. However, because performance problems are complicated and difficult to identify and solve in advance, such problems get worse before being detected, affecting services and causing losses to enterprises. HyperProtect provides the performance exception detection function. For service latency, the deep learning algorithm is used to learn service characteristics based on historical performance data. HyperProtect obtains device performance profiles which show real-time exceptions, precisely locates faults, and provides rectification suggestions.

Figure 9-5 Working principles of performance exception detection

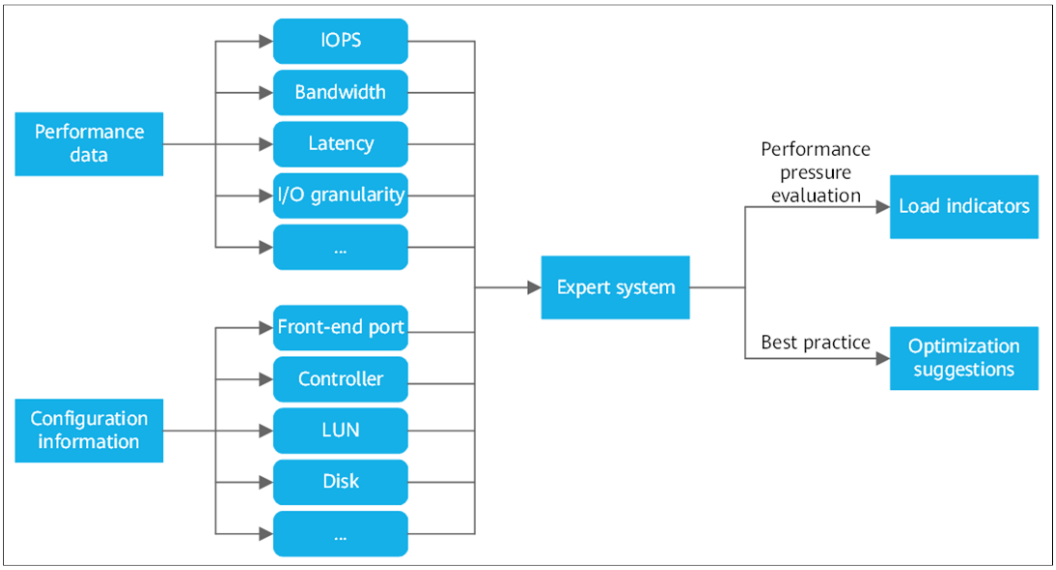


9.1.12 Performance Bottleneck Analysis

After services go online, O&M personnel are concerned about device performance pressure and stable service running. Due to complicated factors that affect device performance, such as hardware configuration, software configuration, service type, and performance data, multiple indicators need to be compared and analyzed concurrently and manually. Therefore, performance pressure evaluation and performance tuning are big challenges.

HyperProtect performance bottleneck analysis covers device configurations and performance data. Users can make adjustment based on these suggestions to ensure stable service running.

Figure 9-6 Working principles of performance bottleneck analysis



9.2 Non-Disruptive Upgrade (NDU)

Storage system upgrade usually requires controller restart and service switchover between controllers to ensure service continuity. This greatly affects service performance, and a large amount of host information must be collected to evaluate potential upgrade risks. Host

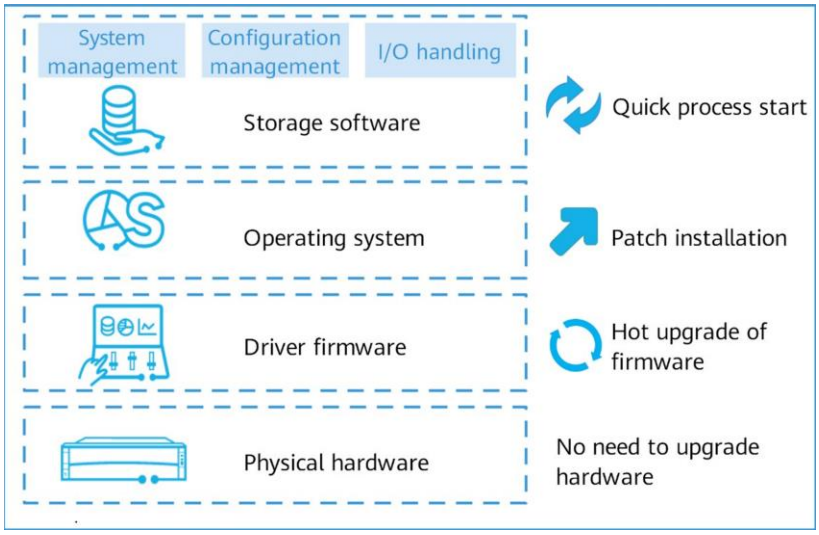
information collection requires host accounts and involves a large amount of information. Certain information cannot be automatically evaluated and demands manual intervention, making an upgrade complex.

HyperProtect provides an upgrade mode that does not require controller restart. In this upgrade mode, services are not affected and performance can quickly recover after the upgrade is complete. During an upgrade, the links between the storage system and host are not interrupted and no link switchover is performed, eliminating the need for collecting host information and avoiding compatibility risks caused by path switchover.

Component-based Upgrade

A storage system can be divided into physical hardware, driver firmware, OS, and storage software. They are upgraded in different ways to complete the system upgrade of HyperProtect, as shown in Figure 9-7.

Figure 9-7 Component-based upgrade



- Physical hardware does not need to be upgraded.
- Driver firmware, including the BIOS, CPLD, and interface module firmware, supports hot upgrade without controller restart.
- The OS is upgraded using hot patches.
- Storage software is user-mode processes. Earlier processes are killed and new processes are started using upgraded codes, which are completed within seconds.

The component-based upgrade and front-end connection keepalive techniques eliminate the need of controller restart and link switchover, and ensure no impact on host services.

Connection Keepalive

In scenarios where front-end interconnect I/O modules are used, links between hosts and the storage system are established over the I/O modules. After receiving service requests from hosts, the I/O modules distribute the requests to the controller. Host I/Os received during the service process restart in the controller are stored in the I/O modules and then distributed to the controller after the service processes become normal. Service process restart takes a very

short time (1 to 2 seconds). In this way, I/Os delivered by hosts do not time out and the hosts are unaware of the restart.

In scenarios where front-end interconnect I/O modules are not used, the daemon process keeps the links (Fibre Channel and iSCSI links) between the controller and hosts connected during service process startup. Service process startup takes a very short time (1 to 2 seconds). In this way, I/Os delivered by hosts do not time out and the hosts are unaware of the startup.

Zero Performance Loss and Short Upgrade Duration

HyperProtect does not involve service switchover. Therefore, an upgrade has nearly no impact on host performance, and performance can recover to 100% within 2 seconds. As controller restart and link switchover are not involved, you do not need to collect host information or perform compatibility evaluation. The upgrade process (from package import to upgrade completion) takes less than 30 minutes, 10 minutes in general. The upgrade of the I/O handling process takes only 10 seconds. This ensures that the upgrade has the minimum impact on the storage system.

10 System Security Design

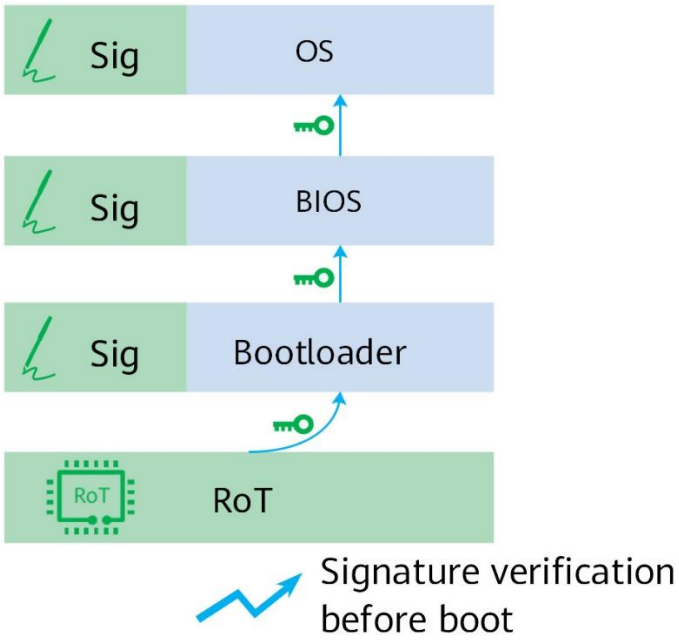
Storage security must be safeguarded by technical measures. Data integrity, confidentiality, and availability must be monitored. Secure boot and access permission control as well as security policies based on specific security threats to storage devices and networks further enhance system security. All these measures prevent unauthorized access to storage resources and data. Storage security includes device, network, service, and management security. This chapter describes software integrity protection, secure boot, and data encryption capabilities related to system security. The secure boot technology guarantees that startup components are verified during startup of storage devices to prevent startup files from being tampered with. The disk encryption feature is used to protect data stored on disks and prevent data loss caused by disk loss. For more information about security technologies, see the security technical white paper.

10.1 Software Integrity Protection

The product software package, upgrade package, and firmware package may be tampered with from the R&D phase to the onsite phase. The digital signature of the product software package is used to protect integrity of upgrade packages used from the R&D phase to the onsite phase. A software package uses an internal digital signature and a product package digital signature. After the software package is sent to the user over the network, the upgrade module of the storage system verifies the digital signature and performs the upgrade only after the verification is successful. This ensures integrity and uniqueness of the upgrade package and internal software modules.

10.2 Secure Boot

Figure 10-1 Secure boot



After a device is powered on, the initial startup module starts, and then verification is performed level by level. After the verification is successful, the device starts. Digital signatures are used to verify firmware integrity to prevent firmware and OSs from being tampered with. The related technologies are as follows:

- The root of trust (RoT) is integrated into the chip to prevent software and physical attacks, providing the highest level of security in the industry.
- Software integrity is ensured by two levels of digital signatures (root key + level-2 key), and software uniqueness is ensured by digital certificates.
- The RSA 2048/4096 algorithm is used, which is of the top security level in the industry.
- Level-2 keys (code signing keys) can be revoked.
- The built-in RoT of the CPU can prevent malicious tampering, such as tampering of flash firmware and replacement of the system disk.

11 Service Support

Vinchin welcomes your suggestions for improving our documentation.

If you have comments, send your feedback to the following mailbox.

Mail to: technical.support@vinchin.com

Your suggestions will be seriously considered and we will make necessary changes to the document in the next release.