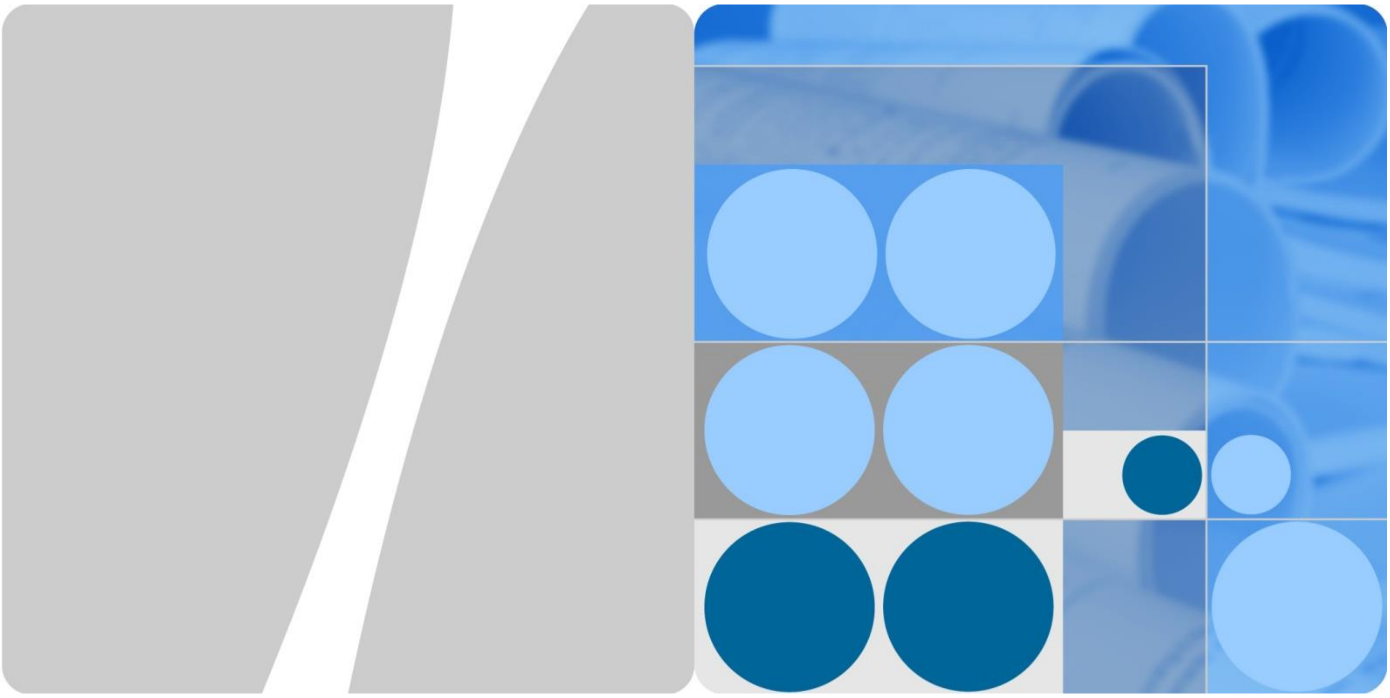


Huawei OceanStor 6800 V5 Mission-Critical Hybrid Flash Storage System

Technical White Paper

Issue	01
Date	2018-07-31

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

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

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

Notice

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

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

Huawei Technologies Co., Ltd.

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

Website: http://e.huawei.com

Contents

1 Executive Summary

Huawei OceanStor 6800 V5 mission-critical hybrid flash storage systems (OceanStor 6800 V5) are designed for enterprise-class applications.

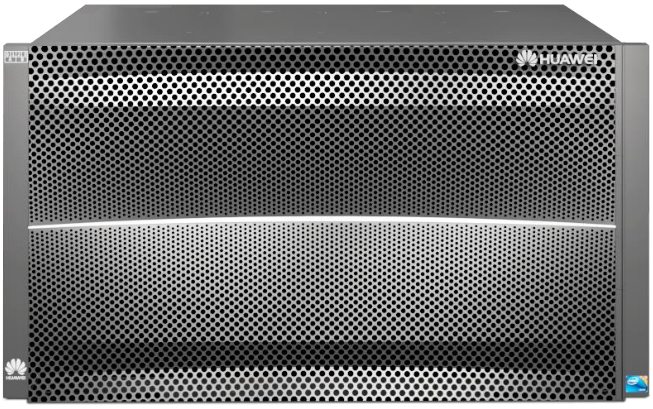
This document describes and highlights the key technologies, unique advantages, and customer benefits of OceanStor 6800 V5. These will be discussed in terms of product positioning, hardware architecture, software architecture, and features.

2 Overview

2.1 OceanStor 6800 V5

Figure 2-1 shows the exterior of OceanStor 6800 V5.

Figure 2-1 Exterior of OceanStor 6800 V5



OceanStor 6800 V5 has the following features:

- Supports up to eight controllers and 3200 2.5-inch disks or 3072 3.5-inch disks.
- Integrates system cache in controllers. Cache capacity can be non-disruptively expanded to 8 TB (1 TB per controller) by adding controllers.

- Supports up to 192 front-end ports, including 8 Gbit/s Fibre Channel, 16 Gbit/s Fibre Channel, GE, 10GE, 10 Gbit/s FCoE, and 56 Gbit/s InfiniBand ports.
- Uses the latest SAS 3.0 technology and supports a maximum of 96 x 4 lane 12 Gbit/s SAS 3.0 ports.

For detailed product specifications, visit http://e.huawei.com/us/products/cloud-computing-dc/storage/massive-storage/6800-v5.

2.2 Customer Benefits

OceanStor 6800 V5 is Huawei's next-generation, enterprise-class hybrid flash storage, designed for mission-critical enterprise applications. With a cloud-ready OceanStor OS, an industry-leading hardware platform, and intelligent management software, OceanStor 6800 V5 delivers top-of-the-line functionality, performance, efficiency, reliability, and ease of use.

OceanStor 6800 V5 fully satisfies the data storage requirements of large-database online transaction processing (OLTP)/online analytical processing (OLAP), file sharing, cloud computing, and many other applications. This makes OceanStor 6800 V5 perfect for application in sectors such as government, finance, telecommunications, energy, and media.

In addition, OceanStor 6800 V5 can be deployed with a wide range of efficient and flexible backup and disaster recovery solutions to ensure business continuity and data security, delivering excellent storage services.

Converged: Accelerated Data Service Efficiency

Powered by the latest OceanStor OS, OceanStor 6800 V5 provides converged and unified resource pools that boast the agility of resource scheduling. This enables free data mobility and helps enterprise IT architectures develop into cloud-based architectures.

- Convergence of all types of flash storage

  Huawei provides comprehensive flash storage products that support interconnection and communication between one another, regardless of their types, levels, and versions. The convergence of data storage, management, and O&M ensures that storage systems can deliver high performance and low latency, as well as the long-term robust reliability of SSDs.
- Convergence of SAN and NAS

  SAN and NAS are converged to provide elastic storage, improve storage resource utilization, and reduce the total cost of ownership (TCO). Block and file data services are converged, enabling storage systems to carry different types of services and boosting the industry-leading performance and functions of SAN and NAS to even greater levels.
- Convergence of storage resource pools

  The built-in heterogeneous virtualization function, SmartVirtualization, enables OceanStor 6800 V5 to take over the storage systems (of different levels, types, and models) of other mainstream vendors, and integrate them into a unified storage resource pool. This can eliminate data silos and enable unified resource management, automation, and service orchestration.

  In addition, data can be automatically migrated from third-party storage to Huawei storage without interrupting services. This reduces the migration time by an average of 60%.
- Convergence of multiple data centers

Networking is simpler. HyperMetro cooperates with HyperVault 3DC to further ensure the continuity of mission-critical services. The active-active data centers (two data centers) can be smoothly upgraded to three data centers, delivering the highest level of service continuity in geo-redundant mode, achieving 64:1 multi-level DCs, providing centralized data disaster recovery and assurance.

Stable and Reliable: 99.9999% High Availability from Products to Solutions

- 4-controller symmetric controller enclosure

  With the SmartMatrix architecture, OceanStor 6800 V5 integrates four controllers into the 6 U space of a controller enclosure. The controllers are then interconnected through a passive backplane.

  In addition, continuous cache mirroring and back-end disk controller interconnection techniques are used, offering industry-leading, 4-controller redundancy, with each of the four controllers acting as a hot backup for each other. Even if three controllers fail at the same time, service stability is protected, maximizing the continuity of mission-critical applications and preventing a single-point running status that is often seen in scenarios where traditional high-end storage systems are upgraded or a controller is faulty.
- Load balancing across controllers

  Load balancing is implemented among controllers, accelerating application access and eliminating performance bottlenecks.
- Full hardware redundancy

  All components and channels are redundant to prevent single points of failure. Fault detection, recovery, and isolation can be independently implemented for each component and channel, ensuring stable system running.
- Unique rapid data restoration technology

  Innovative block-level virtualization reduces the time required to reconstruct 1 TB of data from 10 hours to 30 minutes. When compared with traditional storage systems, OceanStor 6800 V5 reduces the risk of data damage caused by disk failures by 95%.
- DIX+PI end-to-end data protection

  Based on PI and DIX, OceanStor 6800 V5 provides solutions that protect data integrity all the way from application systems and HBAs to storage systems, and disks. This prevents damage to data, further protecting services.
- A wide range of data protection software

  The Hyper series of data protection software includes snapshot, clone, all-in-one backup, remote replication, and other data protection technologies. They protect user data locally, remotely, inside systems, and across different regions, and achieve 99.9999% availability, maximizing business continuity and data availability.
- HyperMetro for both SAN and NAS for core applications

  Huawei launched the innovative integrated active-active solution in the industry. OceanStor 6800 V5 supports HyperMetro for both SAN and NAS, ensuring the high availability of databases and file services. The gateway-free HyperMetro enables the load balancing of active-active mirrors and non-disruptive cross-site takeover. This ensures zero losses of core application data and zero service interruptions.

  In addition, Huawei's HyperMetro solution can be effortlessly upgraded to geo-redundant mode with three data centers.

Fast: Outstanding Performance Achieved to Meet Ever-Increasing Requirements of Enterprise Services

- Flash storage architecture

  OceanStor 6800 V5 uses the flash-oriented system architecture. Based on flash convergence technology, CPU scheduling, cache, redundant array of independent disks (RAID), and interworking between the OceanStor OS and disk drives are all specially designed to suit flash memory.

  OceanStor 6800 V5 can intelligently sense HDDs and SSDs, automatically distinguish media types, and dynamically select the optimal algorithms. This allows the storage to deliver stable I/O response time of less than 1 ms in the event of a large number of service access requests, thereby ensuring optimal performance for critical applications.

- Industry-leading specifications for flash design

  OceanStor 6800 V5 uses next-generation Intel multi-core processors, PCIe 3.0 buses, 12 Gbit/s SAS 3.0 high-speed disk ports, and supports a variety of host ports such as 16 Gbit/s Fibre Channel, 10 Gbit/s FCoE, and 56 Gbit/s InfiniBand host ports. It fully satisfies the needs of video, large file, and other bandwidth-sensitive applications.

- Flexible scalability

  SmartMatrix 2.0 allows for the smooth and linear expansion of system resources to a maximum of 8 controllers, an 8 TB cache, and 3,200 disks. This addresses ever-increasing data storage needs now and in the future and helps customers maximize their return on investment (ROI).

3 System Architecture

3.1 Hardware Architecture

OceanStor 6800 V5 uses the intelligent matrix multi-controller architecture. The storage system can be expanded horizontally, in the unit of controller enclosures, to achieve a linear increase in both performance and capacity.

A controller enclosure uses the four-controller redundancy architecture. Four controllers use the onboard PCIe 3.0 to implement inter-controller cache mirroring. Multiple controller enclosures are scaled out through dedicated switches.

Disks in a disk enclosure are connected to SAS 3.0 back-end full interconnection interface modules in the controller enclosure through two ports. In this way, all of the four controllers can access these disks. SSDs, SAS disks, and NL-SAS disks are supported.

Backup battery units (BBUs) are used to ensure that data in the cache can be quickly written to coffer disks when an OceanStor 6800 V5 encounters a power failure. This protects cache data and achieves zero data loss.

3.1.1 Back-End Full Interconnection

Each controller enclosure of an OceanStor 6800 V5 contains four controllers that are interconnected through PCIe 3.0 high-speed channels provided by the passive backplane in the enclosure. If no more than four controllers are being used, external cables and switches are not required, simplifying deployment and increasing reliability and performance.

Figure 3-1 shows the front view of a 4-controller OceanStor 6800 V5, where four BBUs reside in the upper part, two controllers in the middle, and another two controllers in the lower part. Figure 3-2 shows the internal connections of a controller enclosure.

Figure 3-1 Front view of a 4-controller OceanStor 6800 V5 controller enclosure

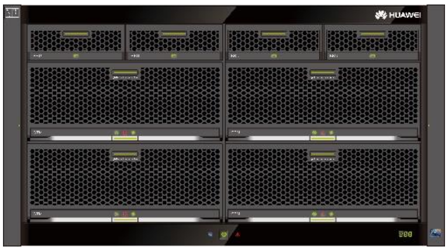
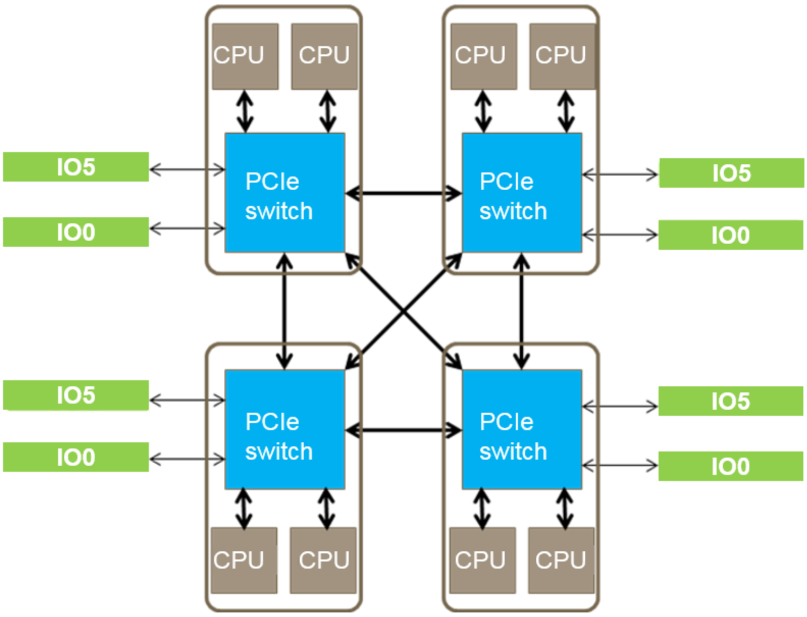


Figure 3-2 Internal connections of a controller enclosure



OceanStor 6800 V5 uses SAS 3.0 back-end, full-interconnection interface modules. Such interface modules are shared with and connected to the four controllers in a controller enclosure. If a controller is faulty, the connections between the controller enclosure and the disk enclosure remain redundant and do not reduce the reliability of the back-end connection.

Figure 3-3 shows the rear view of an OceanStor 6800 V5 controller enclosure. The four slots in the middle house 4 U back-end full-interconnection interface modules. Figure 3-4 shows the back-end connections of an OceanStor 6800 V5.

Figure 3-3 Rear view of an OceanStor 6800 V5 controller enclosure

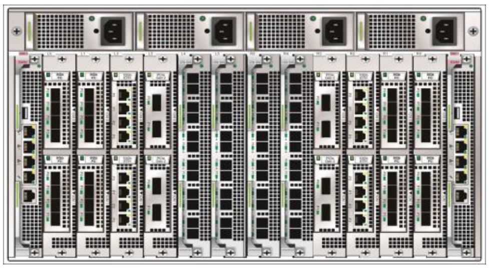
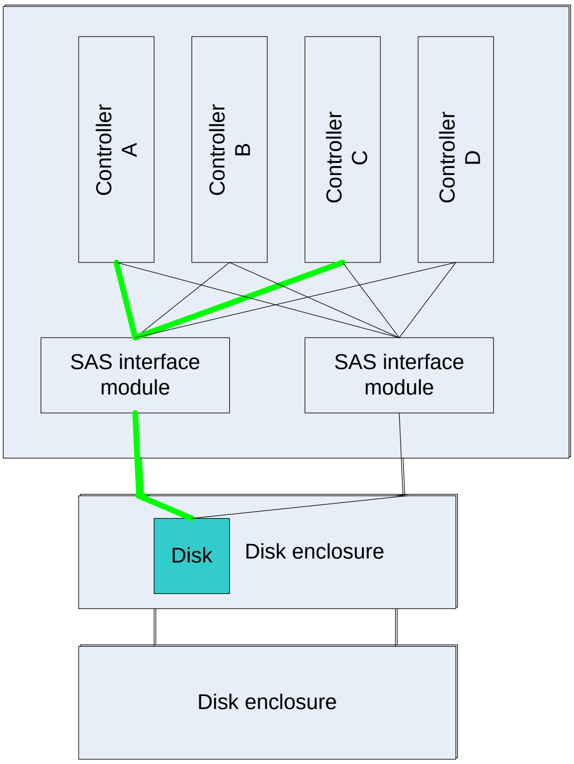


Figure 3-4 Back-end connections of an OceanStor 6800 V5



OceanStor 6800 V5 supports persistent cache, a persistent mirroring technology. Typically, within a controller enclosure, the cache of controller A and the cache of controller B act as mirrors for each other, as do the caches of controllers C and D.

In a traditional architecture, if a controller (say, controller B) becomes faulty, the cache of controller A enters write through mode to ensure data reliability, leading to a significant decrease in performance.

However, in the same scenario, persistent mirroring technology mirrors data from the cache of controller A to the cache of controller C or D (dynamically selected based on internal algorithms), ensuring the reliability of cache data and preventing significant decreases in performance.

OceanStor 6800 V5 can scale out to eight controllers, and each of these controllers has built-in PCIe switching chips. These chips provide 128-lane PCIe 3.0 connections that deliver 256 GB/s interconnection bandwidth among controllers.

NOTE

Controllers in each controller enclosure are connected by PCIe Gen3 x 8 links (that is, 8-lane PCIe 3.0 links).

The number of links connecting controllers is as follows:

- 2 for controllers A to B, and C to D
- 1 for controllers A to C, A to D, B to C, and B to D

A total of eight links exist between the four controllers, and each link provides a bidirectional bandwidth of 2 x 8 x 8 Gbit/s. Therefore, the total bidirectional PCIe interconnection bandwidth is 8 x 2 x 8 x 8 Gbit/s = 1024 Gbit/s, which is 128 GB/s.

3.1.2 Full Hardware Redundancy

All components and channels of the OceanStor 6800 V5 are fully redundant, eliminating single points of failure. Components and channels can detect, repair, and isolate faults independently to ensure stable system running.

Table 3-1 Fully redundant hardware components

Hardware	Component	Redundancy	Fault Impact
Controller enclosure	Controller	1+3	Performance deteriorates accordingly.
	Power module	2+2	None
	Fan module	11+1	None
	BBU module	3+1	None
	Interface module	1+1	None
	Management module	1+1	None
2 U disk enclosure	Expansion module	1+1	None
	Power module	1+1	None
	Fan module	1+1	None
4 U disk enclosure	Expansion module	1+1	None
	Power module	2+2	None
	Fan module	5+1	None

3.1.3 SED Data Encryption

OceanStor 6800 V5 can work with self-encrypting drives (SEDs) and either Internal Key Manager (built-in key management system) or External Key Manager (an independent key management system) to implement static data encryption. The data encryption feature uses the AES 256 algorithm to encrypt user data on storage to ensure the confidentiality, integrity, and availability of user data.

Internal Key Manager

Internal Key Manager is a key management application built in OceanStor 6800 V5. It uses the best practice design of NIST SP 800-57 to manage the AK life cycle of SEDs. Internal Key Manager supports key generation, updating, destruction, backup, and restoration.

Internal Key Manager is easy to deploy, configure, and manage. Therefore, Internal Key Manager is recommended if there are no requirements for FIPS 140-2 and key management is only being used by storage systems in a data center. It is unnecessary to deploy an independent key management system.

External Key Manager

OceanStor 6800 V5 supports the External Key Manager (an independent key management system) that uses the Key Manager Server (KMS) of a third-party system to manage keys.

External Key Manager is a SafeNet Key Secure system that uses standard KMIP + TLS protocols and complies with FIPS 140-2. Therefore, External Key Manager is recommended if FIPS 140-2 is required or multiple systems in a data center require centralized key management.

External Key Manager supports key generation, updating, destruction, backup, and restoration. Two External Key Managers can be deployed to synchronize keys in real time for enhanced reliability.

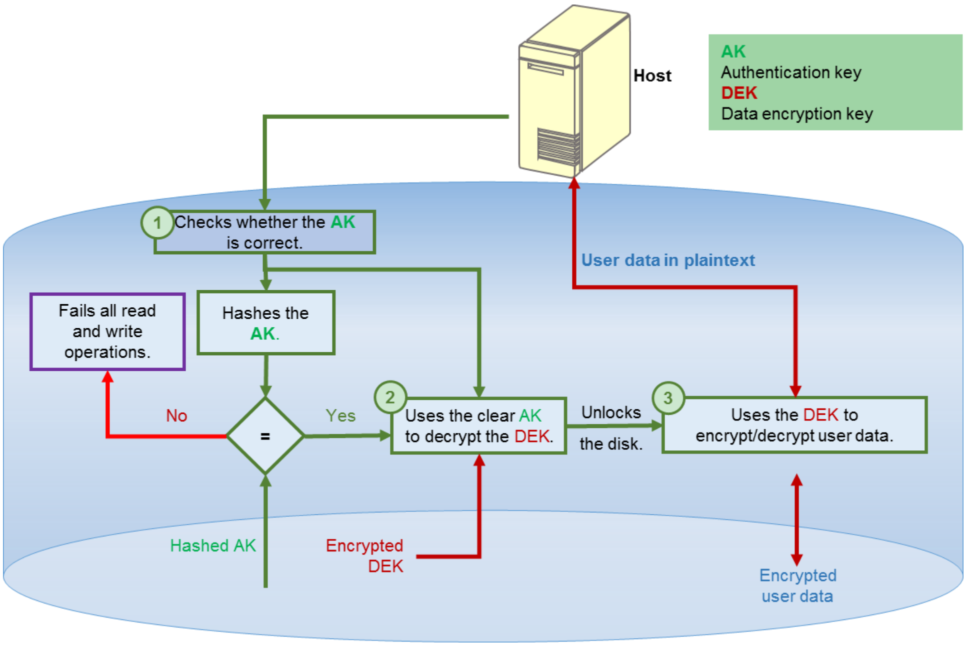
SEDs

SEDs use AKs and data encryption keys (DEKs) to implement two layers of security protection.

- AK mechanism

  After data encryption has been enabled on a storage system, the storage system activates the AutoLock function for an SED, applies an AK from the key manager, and stores the AK on the SED. AutoLock protects the SED and allows only the storage system itself to access the SED. When the storage system accesses an SED, it acquires an AK from the key manager and compares it with the AK stored on the SED. If the acquired AK and stored AK are the same, the SED decrypts the DEK for data encryption or decryption. If they are different, all read and write operations fail.

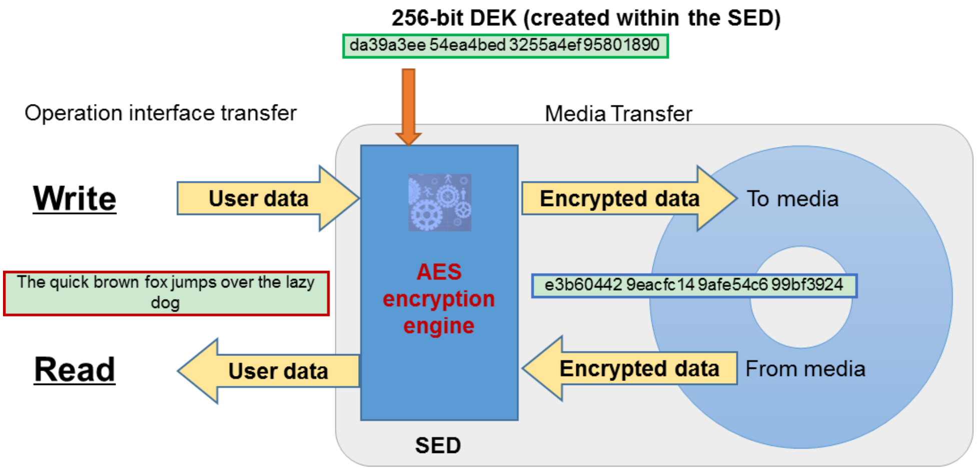
Figure 3-5 AK mechanism


Host
AK
Authentication key
DEK
Data encryption key
1 Checks whether the AK is correct.
Hashes the AK.
Fails all read and write operations.
No
=
Yes
2 Uses the clear AK to decrypt the DEK.
Unlocks the disk.
3 Uses the DEK to encrypt/decrypt user data.
User data in plaintext
Hashed AK
Encrypted DEK
Encrypted user data

- DEK mechanism

  After AutoLock authentication succeeds, the SED uses its hardware circuits and internal DEK to encrypt or decrypt the data. DEK will encrypt data after it has been written to disks. The DEK cannot be acquired separately, meaning that the original information on an SED cannot be recovered mechanically after it is removed from the storage system.

Figure 3-6 Data encryption

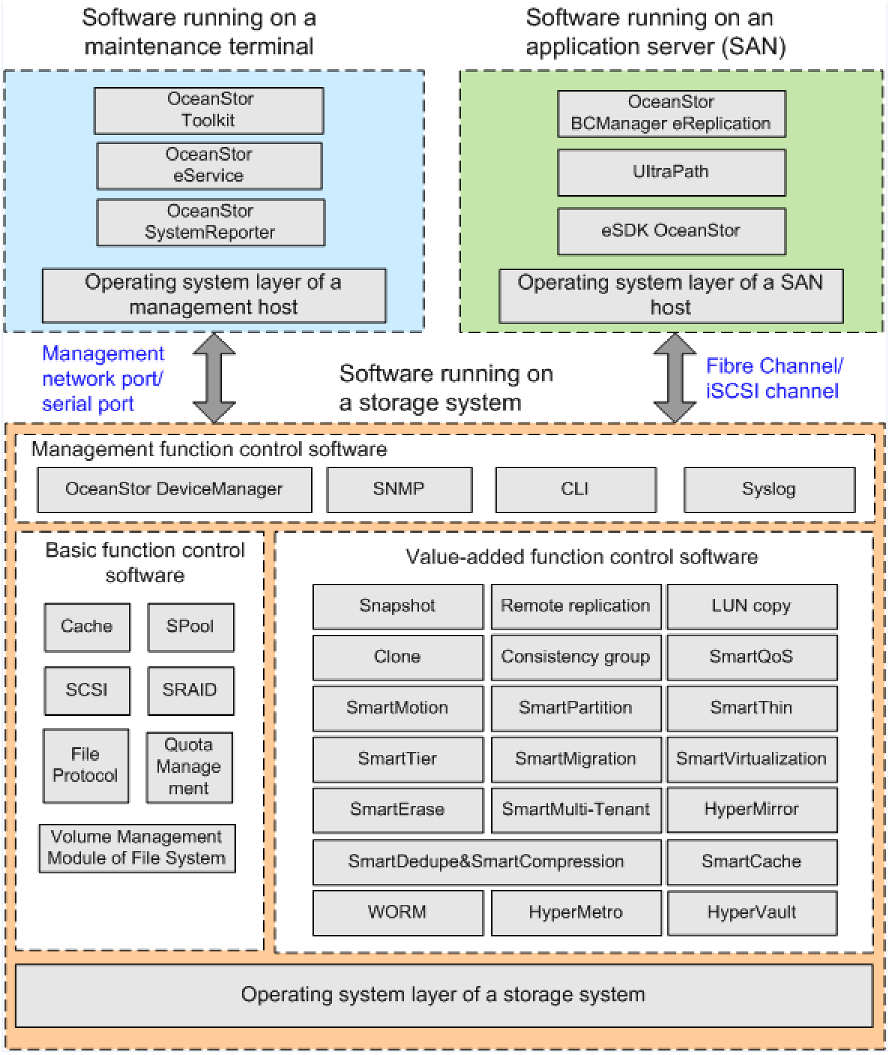

256-bit DEK (created within the SED)
da39a3ee 54ea4bed 3255a4ef95801890
Operation interface transfer
Media Transfer
Write
User data
Encrypted data
To media
The quick brown fox jumps over the lazy dog
AES encryption engine
e3b60442 9eacfc14 9afe54c6 99bf3924
Read
User data
Encrypted data
From media
SED

3.2 Software Architecture

The software suite provided by the OceanStor 6800 consists of software deployed on storage systems, software on maintenance terminals, and software on application servers. These three types of software work jointly to deliver storage, backup, and disaster recovery services in a smart, efficient, and cost-effective manner.

Figure 3-7 shows the software architecture.

Figure 3-7 Software architecture


Software running on a maintenance terminal
OceanStor Toolkit
OceanStor eService
OceanStor SystemReporter
Operating system layer of a management host
Software running on an application server (SAN)
OceanStor BCManager eReplication
UltraPath
eSDK OceanStor
Operating system layer of a SAN host
Management network port/ serial port
Software running on a storage system
Fibre Channel/ iSCSI channel
Management function control software
OceanStor DeviceManager
SNMP
CLI
Syslog
Basic function control software
Cache
SPool
SCSI
SRAID
File Protocol
Quota Manage ment
Volume Management Module of File System
Value-added function control software
Snapshot
Remote replication
LUN copy
Clone
Consistency group
SmartQoS
SmartMotion
SmartPartition
SmartThin
SmartTier
SmartMigration
SmartVirtualization
SmartErase
SmartMulti-Tenant
HyperMirror
SmartDedupe&SmartCompression
SmartCache
WORM
HyperMetro
HyperVault
Operating system layer of a storage system

OceanStor 6800 V5 uses the dedicated OceanStor OS to manage hardware and support the running of storage system software. Basic function control software provides basic data storage and access services, while value-added features are used to provide advanced functions, such as backups, disaster recovery, and performance optimization. Storage systems can be managed by management function control software.

The following describes key technologies in terms of block-level virtualization, SAN and NAS integration, load balancing, data cache, end-to-end data integrity protection, and software features.

3.2.1 Block Virtualization

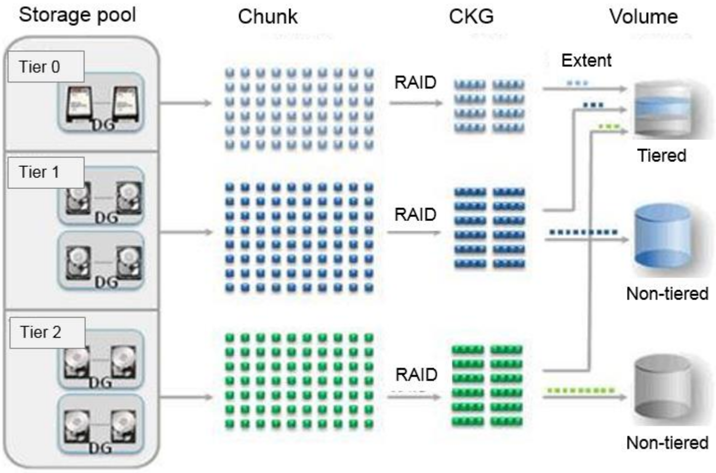
Working Principle

OceanStor 6800 V5 uses the RAID 2.0+ block virtualization technology. Different from traditional RAID that has fixed member disks, RAID 2.0+ enables block virtualization of data on disks. All disks in a storage system are divided into chunks at a fixed size. Multiple chunks from disks are automatically selected at random to form a chunk group (CKG) based on the

RAID algorithm. A CKG is further divided into extents at a fixed size. These extents are allocated to different volumes. Volumes are presented as LUNs or file systems.

Figure 3-8 shows RAID 2.0+.

Figure 3-8 RAID 2.0+ block virtualization



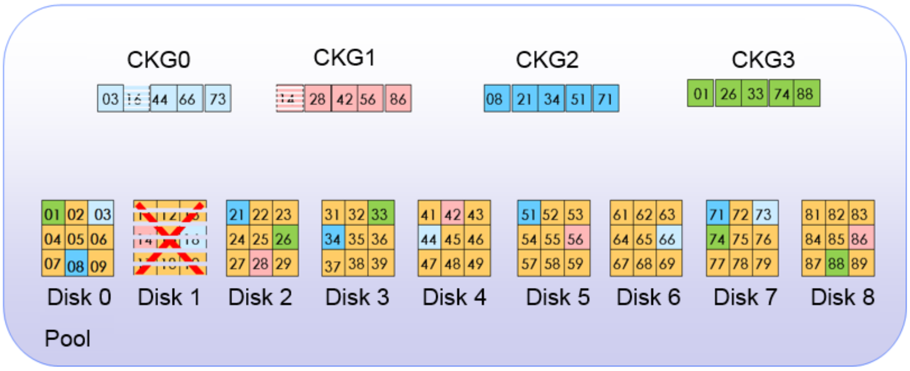
Fast Reconstruction

A RAID group consists of multiple chunks from several physical disks. If a disk fails, other disks participate in reconstructing the data of the faulty disk. More disks are involved in data reconstruction to accelerate the process, allowing 1 TB of data to be reconstructed within 30 minutes.

For example, in a RAID 5 group with nine member disks, if disk 1 becomes faulty, the data in CKG0 and CKG1 is damaged. The storage system then randomly selects chunks to reconstruct the data on disk 1.

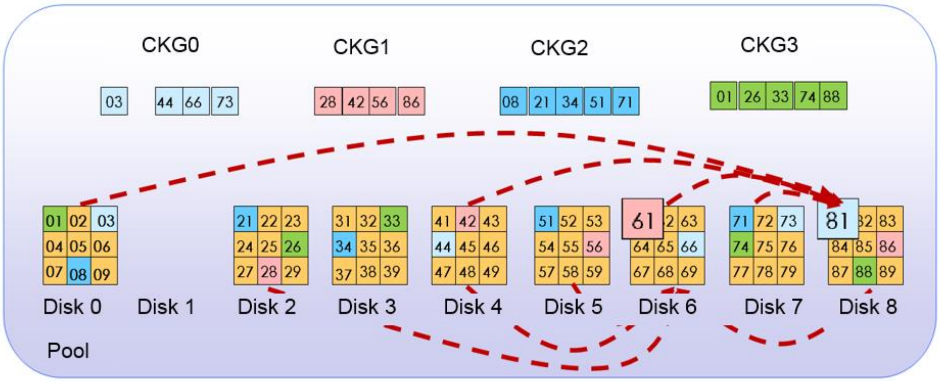
As shown in Figure 3-9, chunks 14 and 16 are damaged. In this case, idle chunks (colored light orange) are randomly selected from the pool to reconstruct data. The system tries to select chunks from different disks.

Figure 3-9 RAID 2.0+ fast reconstruction (1)



As shown in Figure 3-10, chunk 61, on disk 6, and chunk 81, on disk 8, are randomly selected. Data will be reconstructed to these two chunks.

Figure 3-10 RAID 2.0+ fast reconstruction (2)



The bottleneck for traditional data reconstruction typically lies in the target disk (a hot spare disk) because data on all member disks is written to a target disk for reconstruction. As a result, the write bandwidth is the key factor deciding the reconstruction speed. For example, if 2 TB of data on a disk is reconstructed and the write bandwidth is 30 MB/s, it will take 18 hours to complete data reconstruction.

RAID 2.0+ improves data reconstruction in the following two aspects:

1. Multiple target disks

   In the preceding example, if two target disks are used, the reconstruction time will be shortened from 18 hours to 9 hours. If more chunks and member disks are involved, the number of target disks will be equal to that of member disks. As a result, the reconstruction speed linearly increases.

2. Chunk-specific reconstruction

   If fewer chunks are allocated to a faulty disk, less data needs to be reconstructed, further accelerating reconstruction.

RAID 2.0+ shortens the reconstruction time per TB to 30 minutes, greatly reducing the probability of a dual-disk failure.

Load Balancing Among Disks

RAID 2.0+ automatically balances workloads on disks and evenly distributes data from volumes to all disks of a storage system. This prevents individual disks from being overloaded and enhances reliability. As more disks participate in data reads and writes, storage system performance improves.

Maximized Disk Utilization

- Performance

  In a RAID 2.0+ environment, LUNs or file systems are created using storage space from a storage resource pool and are no longer limited by the number of disks in a RAID group, greatly boosting the performance of a single LUN or file system.

- Capacity

The number of disks in a storage resource pool is not limited by the RAID level. This eliminates the chance of usage differences of different RAID groups in traditional volume management environments. Coupled with dynamic LUN or file system capacity expansion, disk space usage is remarkably improved.

Enhanced Storage Management Efficiency

- Easy planning

  It is unnecessary to spend much time in planning storage. Customers simply need to create a storage pool by using multiple disks, set the tiering policies of the storage pool, and allocate space (volumes) from the storage pool.

- Easy expansion of storage pools

  To expand the capacity of a storage pool, customers just need to insert new disks, and the system will automatically distribute data evenly across all disks.

- Easy expansion of volumes

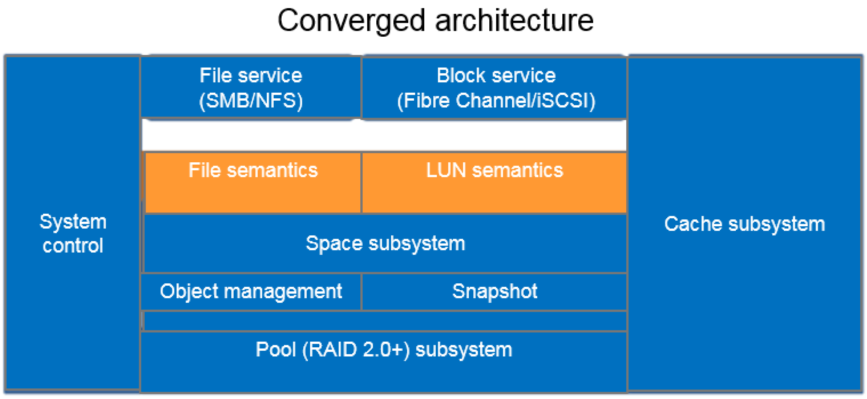
  When customers need to expand the capacity of a volume, they only need to specify the size of the volume to be expanded. The system automatically allocates the required space from the storage pool and adjusts the data distribution of the volume to evenly distribute the volume data across all disks.

3.2.2 SAN and NAS Convergence

OceanStor 6800 V5 adopts a SAN and NAS convergence design. NAS gateways are no longer needed. One set of hardware and software supports both SAN and NAS as well as file access protocols such as Network File System (NFS), Common Internet File System (CIFS), FTP, and HTTP, and file backup protocol Network Data Management Protocol (NDMP). Like SAN, NAS supports Scale-out of eight controllers. Hosts can access any LUN or file system from a front-end host port on any controller.

Figure 3-11 shows the converged architecture of the storage systems. File systems and LUNs directly interact with the space subsystem. The file system architecture is based on objects. Each file or folder acts as an object, and each file system is an object set. LUNs are classified into thin LUNs and thick LUNs. The two types of LUNs come from the storage pool system and space system, instead of file systems. In this way, this converged architecture delivers a simplified software stack and provides a higher storage efficiency than the traditional unified storage architecture. In addition, LUNs and file systems are independent from each other.

Figure 3-11 OceanStor OS software architecture



3.2.3 Load Balancing

SAN Load Balancing

By default, OceanStor 6800 V5 evenly allocates LUNs to controllers and evenly distributes LUN space across all disks in the system.

If there is an I/O path between a host and each controller of the storage system, UltraPath, Huawei proprietary multipathing software, preferably selects the path to the owning controller of the target LUN. If no optimum path is available, the system automatically determines the corresponding controller of the LUN service after I/O requests are delivered to the storage system. Then the Smart Matrix architecture transfers the I/O requests to the corresponding controller.

Even the allocation of LUNs to controllers and the distribution of LUN space among disks will balance the workloads of controllers and disks. SmartMatrix selects the optimum path to deliver I/O requests using UltraPath, allowing the system to reach its optimum performance.

NAS Load Balancing

By default, OceanStor 6800 V5 automatically allocates file systems to controllers. The file system space is evenly distributed to all disks in the system to balance the service loads and disk pressure.

OceanStor 6800 V5 also provides the DNS load balancing feature to intelligently distribute host NFS/CIFS/FTP client connections to service IP addresses configured on different nodes and ports based on service loads, improving system performance and reliability.

When a host uses a domain name to access the NAS service of a storage system, the host sends a DNS request to the built-in DNS server of the storage system to obtain an IP address based on the domain name. If the domain name contains multiple IP addresses, the built-in DNS server selects an IP address with a light load to respond to the host based on the CPU usage, port bandwidth usage, and number of NAS connections of the controllers where IP addresses reside. After receiving the DNS response, the host sends a service request to the destination IP address.

The DNS load balancing feature supports the following load balancing policies: round robin, node CPU usage, node connection quantity, node bandwidth usage, and comprehensive node load.

3.2.4 Data Caching

- Cache distribution

  The physical memory usage of OceanStor 6800 V5 is as follows:

  Physical memory = Cache occupied by the operating system + Read cache + Local write cache + Mirroring write cache + Cache occupied by service features

- Cache types

  There are two types of cache in OceanStor 6800 V5, namely, read cache and write cache.

  - Read cache: Data that has been read is saved to the memory (read cache). This eliminates the need to read the same data from disks again, accelerating read efficiency.
  - Write cache: Data that is about to be written onto disks is saved to the memory (write cache). When the amount of data that is saved in the write cache reaches a specified threshold, the data will be saved to disks.

Read cache and write cache reduce disk-related operations, improve read and write performance of storage systems, and protect disks from being damaged due to repeated read and write operations.

If the write cache is not used, all cache can be used as the read cache. Each storage system reserves the minimum read cache to ensure that read cache resources are still available even if the write workload is heavy.

- Cache prefetch

  In the event of a large number of random I/Os, OceanStor 6800 V5 identifies sequential I/Os with the multi-channel sequential I/O identification algorithm. For the sequential I/Os, the storage system uses prefetch and merge algorithms to optimize system performance in various application scenarios.

  The prefetch algorithm supports intelligent prefetch, constant prefetch, and variable prefetch. By automatically identifying I/O characteristics, intelligent prefetch determines whether data is prefetched and determines the prefetch length, ensuring that the system performance meets requirements of different scenarios.

  By default, the storage system adopts the intelligent prefetch algorithm. However, in application scenarios with definite I/O models, users can also configure the storage system to use constant prefetch or variable prefetch. These two algorithms allow users to define prefetch length.

- Cache eviction

  When the cache usage reaches a specified threshold, the cache eviction algorithm calculates the access frequency of each data block based on historical and current data access frequencies. The eviction algorithm then works with the multi-channel sequential I/O identification algorithm to evict unnecessarily cached data. In addition, you can configure the cache priority of a volume and adjust the priority of each I/O for a specific service. Data with low priorities is eliminated first, and high-priority data is cached to ensure the data hit rate.

3.2.5 End-to-End Data Integrity Protection

The ANSI T10 Protection Information (PI) standard provides a way to check data integrity when accessing a storage system. This check is undertaken based on the PI field defined in the T10 standard. This standard adds an 8-byte PI field to the end of each sector to check data integrity. In most cases, the T10 PI is used to ensure the integrity of data in a storage system.

Data Integrity Extensions (DIX), provided by vendors such as Oracle and Emulex, further extend the protection scope of T10 PI. Therefore, DIX+T10 PI can achieve complete end-to-end data protection.

In addition to using T10 PI to ensure the integrity of data in a storage system, OceanStor 6800 V5 also adopts DIX + T10 PI to implement end-to-end data integration protection. A storage system verifies and delivers PI fields of data in real time. If a host does not support PI, the storage system adds the PI fields to the host interface and then delivers the fields. In a storage system, PI fields are forwarded, transmitted, and stored together with user data. Then, before user data is read by a host again, the storage system uses PI fields to check the accuracy and integrity of user data.

3.2.6 Various Software Features

OceanStor 6800 V5 provides the Smart series features to accelerate system efficiency and the Hyper series features to protect data.

- The Smart series features include SmartDedupe, SmartCompression, SmartThin, SmartVirtualization, SmartMigration, SmartTier, SmartQoS, SmartPartition, SmartErase, SmartMulti-Tenant, SmartCache, SmartQuota, and SmartMotion. These software features help users improve storage efficiency and reduce the total cost of ownership (TCO).

- The Hyper series features include HyperSnap, HyperClone, HyperReplication, HyperMetro, HyperVault, HyperCopy, HyperMirror, and HyperLock. These software features help users implement data backup and disaster recovery. In addition, the storage systems can be used in various disaster recovery solutions in which three data centers are deployed.

The storage system can also be deployed in solutions that are integrated with common IT systems. The following is an example of integration with some VM environments.

- VMware vSphere

  The storage system supports VMware vStorage APIs for Array Integration (VAAI), vStorage APIs for Software Awareness (VASA), and Site Recovery Manager (SRM). The vCenter plugin is provided, enabling unified management in vCenter.

- Windows Hyper-V

  The storage system supports Windows Thin space reclamation technology and Windows Offload Data Transfer (ODX). In addition, the System Center plug-in is provided and can be managed by the System Center Operations Manager (SCOM) and System Center Virtual Machine Manager (SCVMM).

3.2.7 Flash-Oriented System Optimization

SSDs deliver high performance for random I/O access, and ensure low latency, however, their erase times are limited. HDDs deliver high performance for sequential I/O access but their erase times are not restricted. Huawei has optimized SSDs, as well as the hybrid storage of SSDs and HDDs used in OceanStor 6800 V5, to achieve better performance and reliability.

- Seamless collaboration between OceanStor OS and Huawei SSD (HSSD) firmware

  SSDs use flash chips that are involved in erasure operations. When erasure operations are being performed, other data in the channels that is involved in the erasure operations is inaccessible. As a result, a latency of 1 ms to 2 ms occurs, leading to performance fluctuations.

  Huawei storage systems use HSSDs. OceanStor OS is designed to work alongside HSSDs to ensure that erasure operations are sequentially performed on multiple HSSDs. OceanStor OS does not read data from HSSDs on which erasures are being performed. Instead, data is read from other HSSDs based on a RAID redundancy mechanism, thereby ensuring stable latency.

- Intelligent SSD perception by cache

  OceanStor 6800 V5 uses different dirty data flushing policies for SSDs and HDDs. When Huawei-certified disks are connected, the storage system automatically identifies the media types. For SSDs, the storage system delays the flushing of active data, reduces the flushing times, and decreases write amplification based on the Least Recently Used (LRU) algorithm. This boosts system performance and prolongs the service life of SSDs.

- Performance optimized using multiple cores

  In terms of a multi-core scheduling mechanism, system performance is optimized for the NUMA architecture. For example, messages related to a single I/O are dispatched to the same CPU to reduce cross-CPU access overheads and increase the CPU cache hit ratio.

  With regard to multi-thread operating efficiency, a data structure design is used to prevent multiple threads from concurrently accessing data on the cache line of the CPU

L1 cache. This eliminates the pseudo-sharing of the CPU L1 cache, improves the CPU L1 cache efficiency, and reduces the CPU overhead in memory-based data access.

4 Smart Series Features

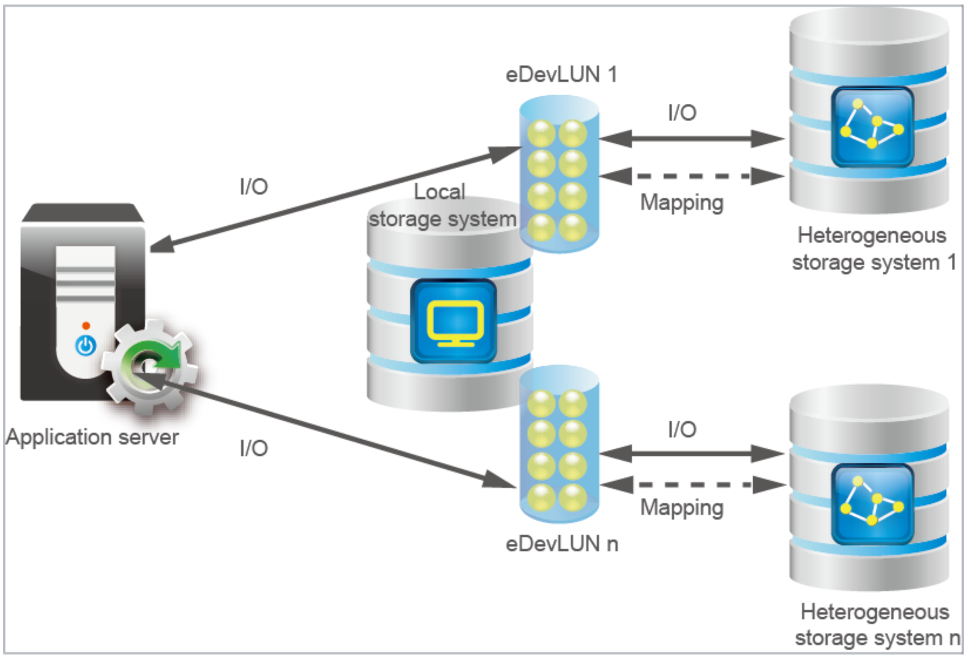
4.1 SmartVirtualiztaion

OceanStor 6800 V5 uses SmartVirtualization to take over heterogeneous storage systems (including other Huawei storage systems and third-party storage systems), protecting customer investments. SmartVirtualization conceals the software and hardware differences between local and heterogeneous storage systems, allowing the local system to use and manage the heterogeneous storage resources as if they were local resources. In addition, SmartVirtualization can work with SmartMigration to migrate data from heterogeneous storage systems online, facilitating device replacement.

Working Principle

SmartVirtualization maps the heterogeneous storage system to the local storage system, which in turn uses external device LUNs (eDevLUNs) to take over and manage the heterogeneous resources. eDevLUNs consist of metadata volumes and data volumes. The metadata volumes manage the data storage locations of eDevLUNs and use the physical space of the local storage system. The data volumes are logical representations of external LUNs, and use the physical space of the heterogeneous storage system. eDevLUNs on the local storage system match external LUNs on the heterogeneous storage system, allowing application servers to access data on the external LUNs through the eDevLUNs.

Figure 4-1 Heterogeneous storage virtualization



SmartVirtualization uses LUN masquerading to set the world wide names (WWNs) and Host LUN IDs of eDevLUNs on OceanStor 6800 V5 to the same values as those on the heterogeneous storage system. After data migration is complete, the host's multipathing software switches over the LUNs online without interrupting services.

Application Scenarios

- Heterogeneous array takeover

  Because customers build data centers over time, the storage arrays they use may come from different vendors. Storage administrators can use SmartVirtualization to manage and configure existing devices, protecting investments.

- Heterogeneous data migration

  Customers may need to replace storage systems with warranty periods that are about to expire or performance that does not meet service requirements. SmartVirtualization and SmartMigration can migrate customer data to OceanStor 6800 V5 online without interrupting host services.

- Heterogeneous disaster recovery

  If service data is stored at two sites having heterogeneous storage systems and requires robust service continuity, SmartVirtualization can work with HyperReplication to allow data on LUNs in heterogeneous storage systems to be mutually backed up. If a disaster occurs, a functional service site takes over services from the failed service site and then recovers data.

- Heterogeneous data protection

Data on LUNs that reside in heterogeneous storage systems may be attacked by viruses or corrupted. SmartVirtualization can work with HyperSnap to instantly create snapshots for LUNs that reside in heterogeneous storage systems, and use these snapshots to rapidly restore data to a specific point in time if the data is corrupted.

4.2 SmartMigration

OceanStor 6800 V5 uses SmartMigration for intelligent data migration based on LUNs. Data on a source LUN can be completely migrated to a target LUN without interrupting ongoing services. SmartMigration supports data migration within a Huawei storage system or between a Huawei storage system and a compatible heterogeneous storage system.

When the system receives new data during migration, it simultaneously writes the new data to both the source and target LUNs and records data change logs (DCLs) to ensure data consistency. After migration is complete, the source and target LUNs exchange information so that the target LUN can take over services.

SmartMigration involves data synchronization and LUN information exchange.

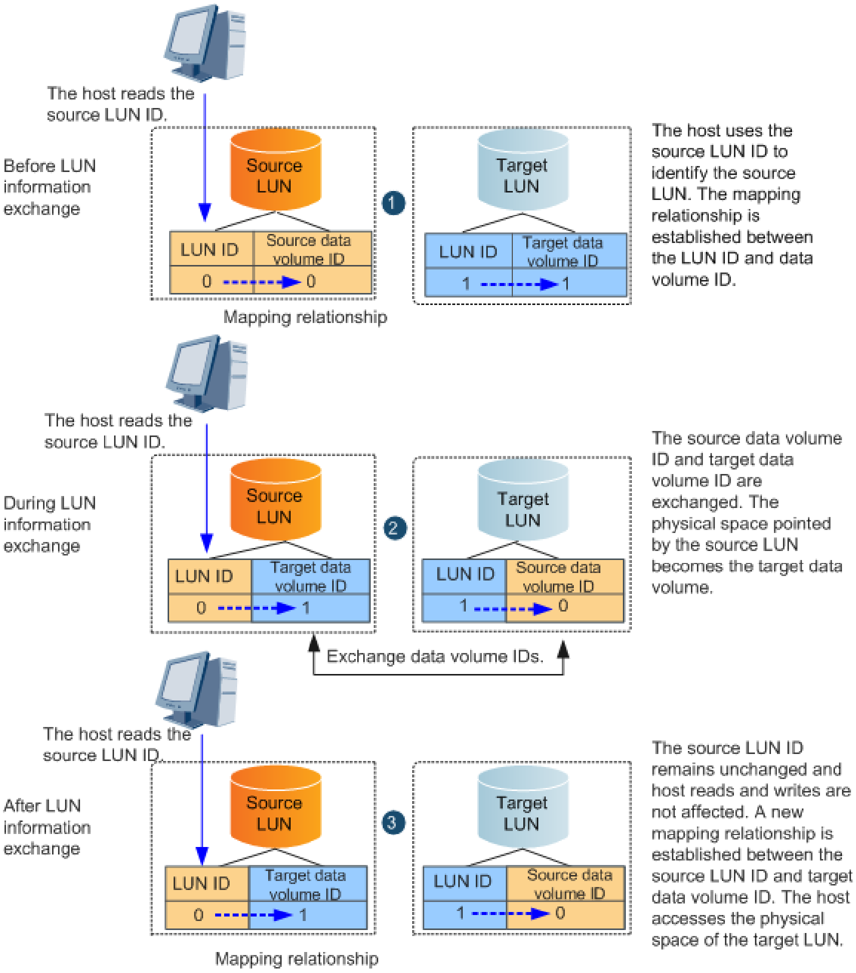
Data Synchronization

1. Before migration, customers must configure the source and target LUNs.
2. When migration begins, the source LUN replicates data to the target LUN.
3. During migration, the host can still access the source LUN, and when the host writes data to the source LUN, the system records the DCL.
4. The system writes the incoming data to both the source and target LUNs.
   - If data is successfully written to both LUNs, the system clears the record in the DCL.
   - If data fails to be written to the target LUN, the storage system identifies the data that failed to be synchronized according to the DCL. Then, the system copies the data to the target LUN. After the data is copied, the storage system returns a write success to the host.
   - If data fails to be written to the source LUN, the system returns a write failure to notify the host to re-send the data. Upon receiving the data again, the system only writes the data to the source LUN.

LUN Information Exchange

After data replication is complete, host I/Os are temporarily suspended, and the source and target LUNs exchange information, as seen in Figure 4-2.

Figure 4-2 LUN information exchange


The host reads the source LUN ID.
Before LUN information exchange
Source LUN
LUN ID
Source data volume ID
0
0
1
Target LUN
LUN ID
Target data volume ID
1
1
The host uses the source LUN ID to identify the source LUN. The mapping relationship is established between the LUN ID and data volume ID.
Mapping relationship
The host reads the source LUN ID.
During LUN information exchange
Source LUN
LUN ID
Target data volume ID
0
1
2
Target LUN
LUN ID
Source data volume ID
1
0
The source data volume ID and target data volume ID are exchanged. The physical space pointed by the source LUN becomes the target data volume.
Exchange data volume IDs.
The host reads the source LUN ID.
After LUN information exchange
Source LUN
LUN ID
Target data volume ID
0
1
3
Target LUN
LUN ID
Source data volume ID
1
0
The source LUN ID remains unchanged and host reads and writes are not affected. A new mapping relationship is established between the source LUN ID and target data volume ID. The host accesses the physical space of the target LUN.
Mapping relationship

The LUN information exchange is instantaneous, and does not interrupt services.

Application Scenarios

- Storage system upgrades with SmartVirtualization

  SmartMigration works with SmartVirtualization to migrate data from legacy storage systems (from Huawei or other vendors) to new Huawei storage systems. This improves service performance and data reliability.

- Data migration for capacity, performance, and reliability adjustments

4.3 SmartDedupe and SmartCompression

SmartDedupe and SmartCompression provide data deduplication and compression functions to shrink data for file systems and thin LUNs. This saves space while reducing the TCO of the enterprise IT architecture.

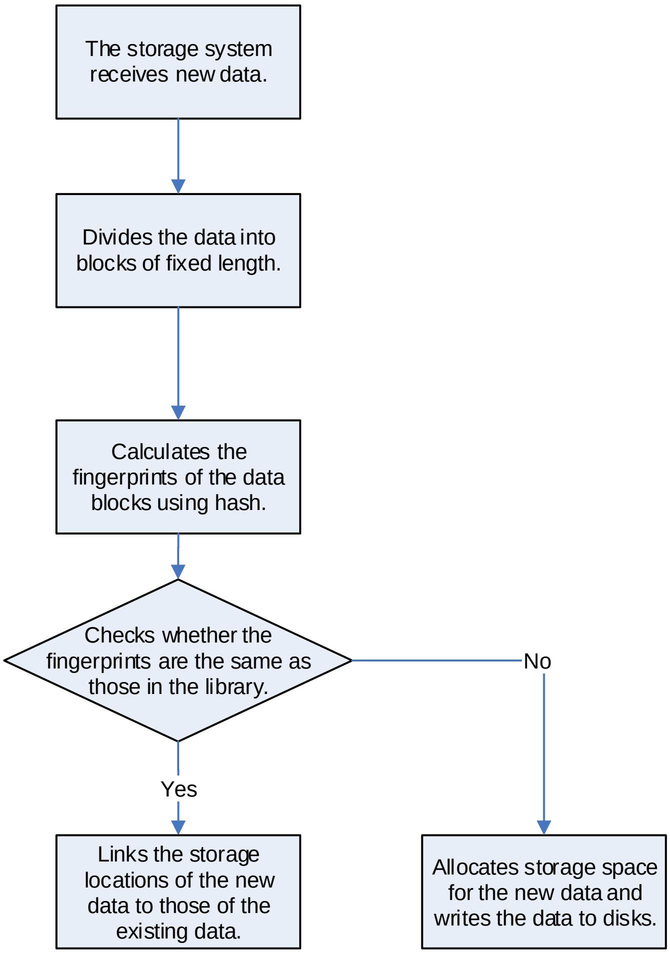
SmartDedupe

OceanStor 6800 V5 uses SmartDedupe to implement inline deduplication for file systems and thin LUNs. In inline deduplication mode, the storage system deduplicates new data before writing it to disks.

The data deduplication granularity is consistent with the minimum data read and write unit (grain) of file systems or thin LUNs. Users can specify the grain size (4 KB to 64 KB) when creating file systems or thin LUNs, so that OceanStor 6800 V5 implements data deduplication based on different granularities.

Figure 4-3 shows how OceanStor 6800 V5 deduplicates data.

Figure 4-3 Deduplication process


The storage system receives new data.
Divides the data into blocks of fixed length.
Calculates the fingerprints of the data blocks using hash.
Checks whether the fingerprints are the same as those in the library.
No
Yes
Links the storage locations of the new data to those of the existing data.
Allocates storage space for the new data and writes the data to disks.

The process is as follows:

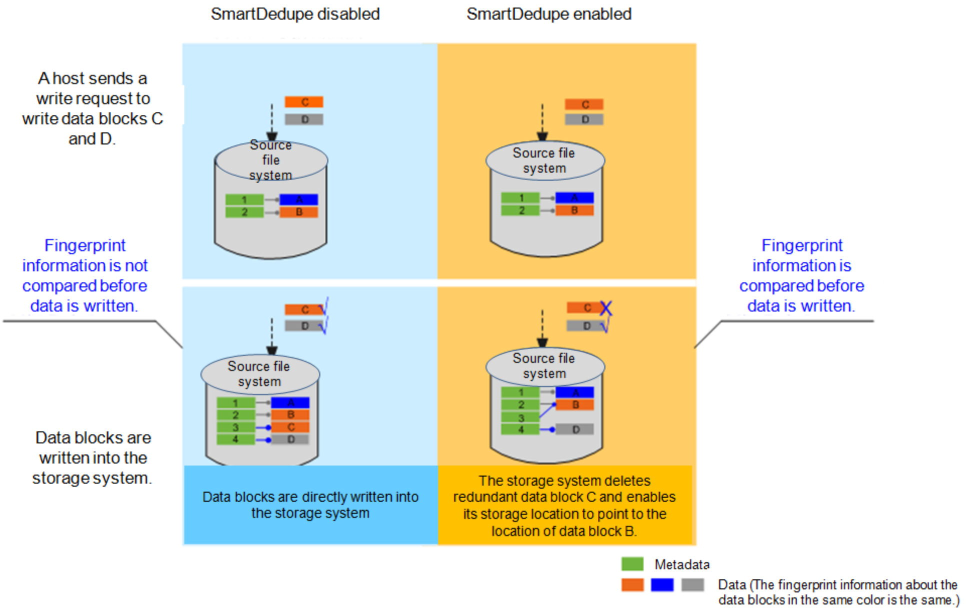
1. The storage system divides new data into blocks based on the deduplication granularity.
2. The storage system compares the fingerprints of new data blocks with those of existing data blocks, kept in the fingerprint library. If identical fingerprints are not found, the storage system writes new data blocks. If identical fingerprints are found, the system does the following:

- With byte-by-byte comparison disabled (default), the system identifies the data blocks as duplicates. It will not allocate storage space for these duplicate blocks, and instead links their storage locations with those of the existing data blocks.
- With byte-by-byte comparison enabled, the storage system will compare the new data blocks with the existing data blocks byte by byte. If they are the same, the system identifies duplicate data blocks. If they are different, the system writes the new data blocks.

The following is an example of the process:

A file system has data blocks A and B, and an application server writes data blocks C and D to the file system. C has the same fingerprint as B, while D has a different fingerprint from A and B. Figure 4-4 shows how the data blocks are processed when different data deduplication policies are used.

Figure 4-4 Data processing with SmartDedupe enabled and disabled


SmartDedupe disabled
SmartDedupe enabled
A host sends a write request to write data blocks C and D.
C
D
Source file system
1
2
A
B
Fingerprint information is not compared before data is written.
Fingerprint information is compared before data is written.
C √
D √
C X
D √
Source file system
1
2
3
4
A
B
C
D
Data blocks are written into the storage system.
Data blocks are directly written into the storage system
The storage system deletes redundant data block C and enables its storage location to point to the location of data block B.
Metadata
Data (The fingerprint information about the data blocks in the same color is the same.)

SmartCompression

Inline and post-process compression is available in the industry. OceanStor 6800 V5 uses inline compression, which compresses new data before it is written to disks. Inline compression has the following advantages when compared to post-process compression:

- Requires less initial storage space, lowering the initial investment of customers.
- Generates fewer I/Os, applicable to SSDs, which restrict the number of reads and writes.
- Compresses data blocks after snapshots are created, saving space.

SmartCompression compresses data blocks based on the user-configured compression policy. The storage system supports the following compression policies:

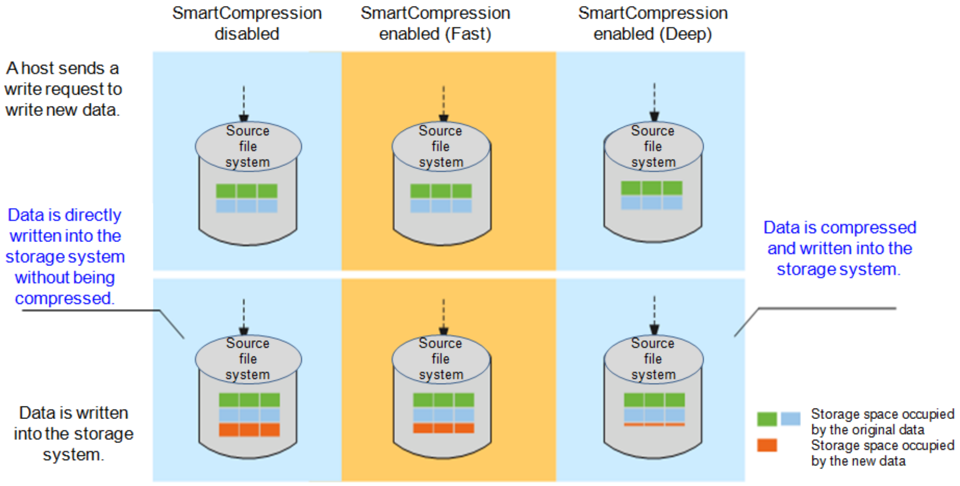
- Fast policy (default)

  This policy has a higher compression speed but lower efficiency in capacity saving.
- Deep policy

This policy significantly improves capacity saving efficiency but takes longer to perform compression and decompression.

Figure 4-5 shows how data blocks are processed when different data compression policies are used.

Figure 4-5 Data processing with SmartCompression enabled and disabled


SmartCompression disabled
SmartCompression enabled (Fast)
SmartCompression enabled (Deep)
A host sends a write request to write new data.
Source file system
Data is directly written into the storage system without being compressed.
Data is compressed and written into the storage system.
Data is written into the storage system.
🟩🟦 Storage space occupied by the original data
🟧 Storage space occupied by the new data

Interworking of SmartDedupe and SmartCompression

SmartDedupe and SmartCompression can work together. When they are both enabled, data is deduplicated and then compressed, saving more storage space.

SmartDedupe and SmartCompression, provided by the OceanStor 6800 V5, work in in-line mode. When the functions are enabled, new data is deduplicated and compressed. When the functions are disabled, deduplicated data cannot be restored.

4.4 SmartTier

SmartTier for Block

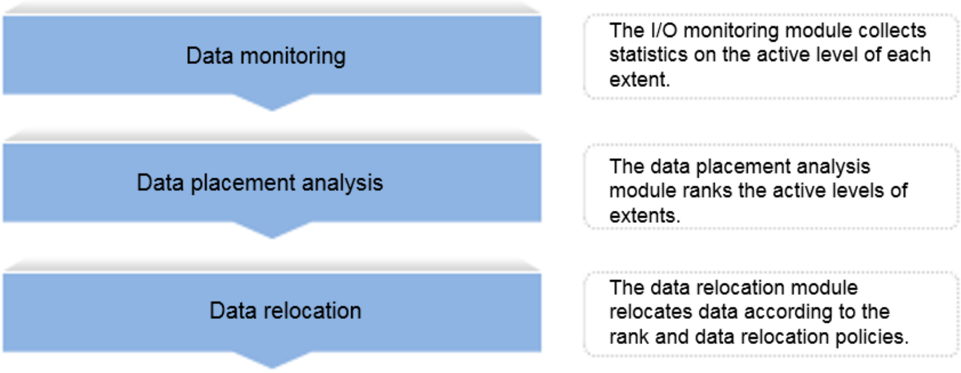
OceanStor 6800 V5 uses SmartTier for dynamic storage tiering.

SmartTier categorizes storage media into three storage tiers based on performance: high-performance tier (SSDs), performance tier (SAS disks), and capacity tier (NL-SAS disks). Storage tiers can be used independently or together to provide data storage space.

SmartTier performs intelligent data storage based on LUNs, segmenting data into extents (with a default size of 4 MB, configurable from 512 KB to 64 MB). SmartTier collects statistics on and analyzes the activity of data based on extents and matches the data of various activity levels with proper storage media. More-active data will be promoted to higher-performance storage media (such as SSDs), whereas less-active data will be demoted to more cost-effective storage media with larger capacities (such as NL-SAS disks).

SmartTier implements data monitoring, placement analysis, and data relocation, as shown in the following figure:

Figure 4-6 SmartTier implementation


Data monitoring
The I/O monitoring module collects statistics on the active level of each extent.
Data placement analysis
The data placement analysis module ranks the active levels of extents.
Data relocation
The data relocation module relocates data according to the rank and data relocation policies.

Data monitoring and data placement analysis are automated by the storage system, and data relocation is initiated manually or by a user-defined policy.

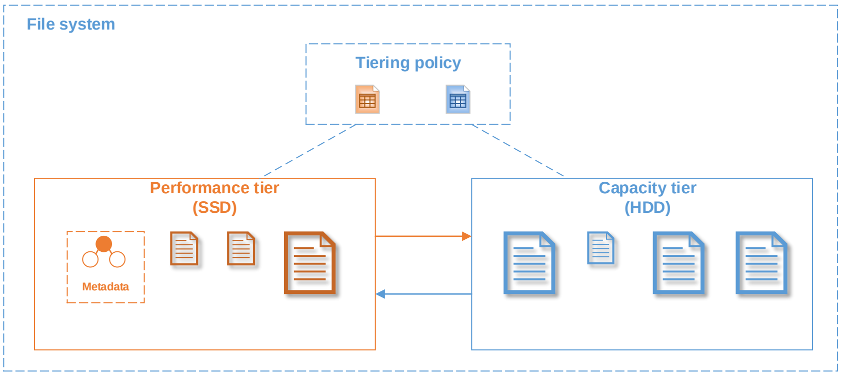
SmartTier improves storage system performance and reduces storage costs to meet enterprise requirements of both performance and capacity. By preventing historical data from occupying expensive storage media, SmartTier ensures effective investment and eliminates energy consumption caused by idle capacities. This reduces the TCO and optimizes the cost-effectiveness.

SmartTier for File

SmartTier also applies to file systems. It helps customers simplify data life cycle management, improve media usage, and reduce costs. SmartTier dynamically relocates data by file, among different media based on user-defined tiering policies.

A storage pool can be composed of SSDs and HDDs. SmartTier automatically promotes files to high-performance media (SSDs) or demotes files to large-capacity media (HDDs, including SAS and NL-SAS disks) based on user-configured tiering policies. Users can specify tiering policies by file name, file size, file type, file creation time, and SSD usage. Figure 4-7 shows the working principles of SmartTier.

Figure 4-7 SmartTier working principles



SmartTier features:

- Custom tiering policies

  Users can flexibly define tiering policies by file name, file size, file type, file creation time, SSD usage, or a combination of these to meet requirements in various scenarios.

- File access acceleration

  By default, file system metadata is stored in SSDs, which facilitates the locating of files and directories, thereby accelerating file access.
- Intelligent flow control

  File relocation increases CPU and disk loads. The storage system performs intelligent flow control for relocation tasks based on service pressure, minimizing the impact of data relocation on service performance.
- Saved cost

  SmartTier enables tiered storage. The storage system saves data on SSDs and HDDs, ensuring service performance at lower costs when compared with All Flash Arrays (AFAs).
- Simplified management

  SmartTier supports tiered storage within a file system. It automatically relocates cold data to HDDs, archiving data without requiring other features or applications, which simplifies data life cycle management. Users are not aware of this seamless data relocation.

SmartTier applies to scenarios in which file life cycle management is required, such as financial check images, medical images, semiconductor simulation design, and reservoir analysis. The services in these scenarios have demanding performance requirements in the early stages and low performance requirements later. The following describes an example.

In the reservoir analysis scenario, small files are imported to the storage system for the first time. These small files are frequently accessed and have high performance requirements. After small files are processed by professional analysis software, large files are generated, which have low performance requirements. Users can configure tiering policies based on file sizes. To be specific, small files are stored on SSDs and large files are stored on HDDs (such as low-cost NL-SAS disks). In this way, SmartTier helps reduce customer's costs while meeting performance requirements.

4.5 SmartThin

SmartThin enables the storage system to allocate storage resources on demand. SmartThin does not allocate all available capacity in advance, and instead presents a virtual storage capacity larger than the physical storage capacity. In this way, you see the storage space as being larger than the actual allocated space. When you begin to use the storage, SmartThin provides only the required space. If the allocated storage space is about to use up, SmartThin triggers storage resource pool expansion to add more space. The expansion process is not noticeable by users and causes no system downtime.

SmartThin applies to:

- Core businesses that have demanding requirements for continuity, such as bank transaction systems

  SmartThin allows customers to conduct online capacity expansion without interrupting businesses.
- Businesses with application system data usage that fluctuates unpredictably, such as email services and online storage services

  SmartThin enables physical storage space to be allocated on demand, preventing wasted resources.

- Businesses that involve various systems with diverse storage requirements, such as telecom carrier services

  SmartThin allows different applications to contend for physical storage space, improving space utilization.

4.6 SmartQoS

SmartQoS dynamically allocates storage system resources to meet the performance objectives of applications.

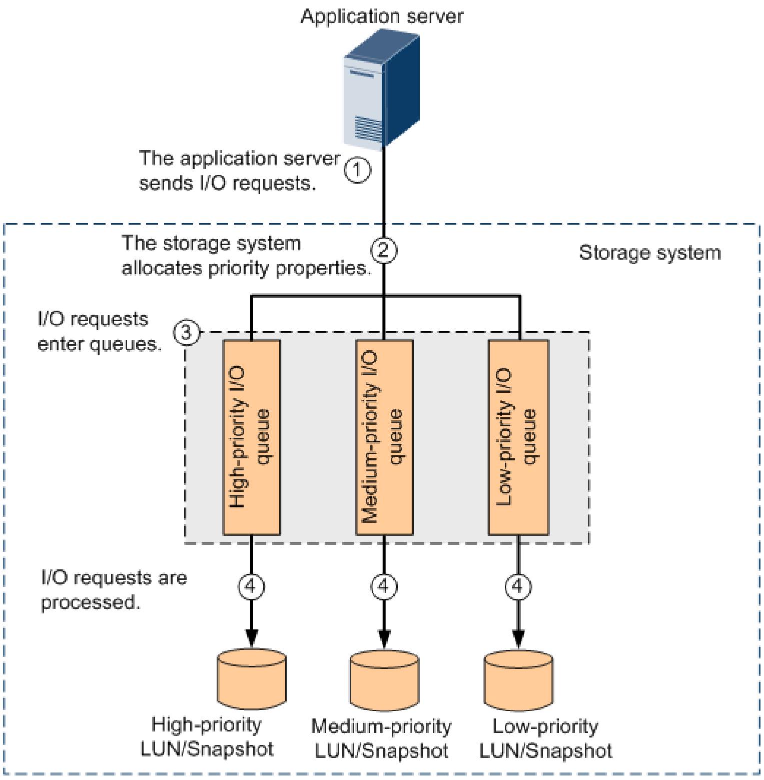
SmartQoS enables you to set upper limits for IOPS or bandwidth for specific applications. Based on the upper limits, SmartQoS can accurately limit the performance of these applications, preventing them from contending for storage resources with critical applications.

SmartQoS uses LUN-, FS-, or snapshot-specific I/O priority scheduling and I/O traffic control to guarantee service quality.

- I/O priority scheduling

  This schedules resources based on application priorities. When allocating system resources, a storage system prioritizes the resource allocation requests initiated by high-priority services. If there is a shortage of resources, a storage system allocates more resources to the high-priority services to meet their QoS requirements.

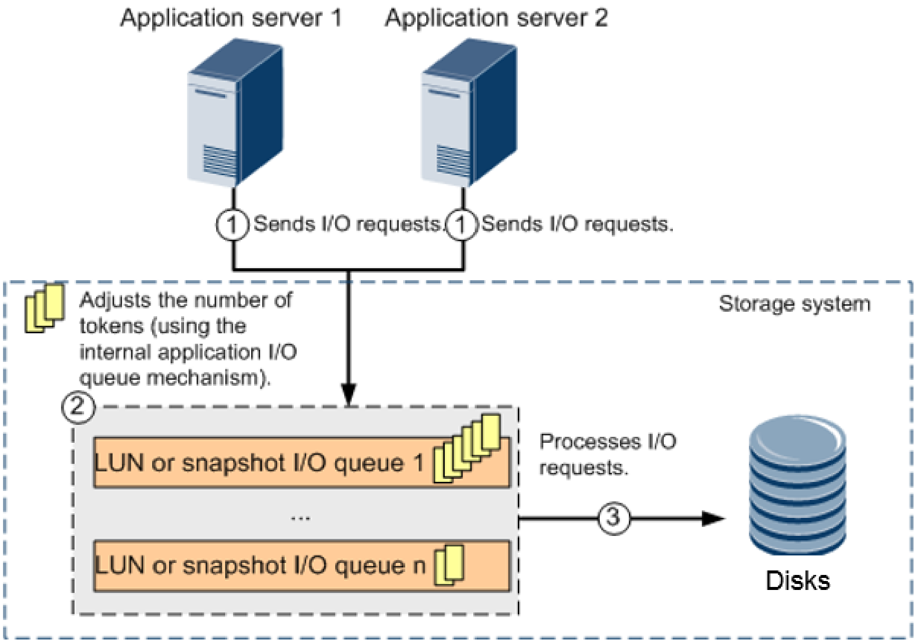
Figure 4-8 I/O priority scheduling process


Application server
The application server sends I/O requests.
1
The storage system allocates priority properties.
2
Storage system
I/O requests enter queues.
3
High-priority I/O queue
Medium-priority I/O queue
Low-priority I/O queue
I/O requests are processed.
4
4
4
High-priority LUN/Snapshot
Medium-priority LUN/Snapshot
Low-priority LUN/Snapshot

- I/O traffic control

This limits the traffic of some applications by limiting their IOPS or bandwidth, thereby preventing these applications from affecting other applications. I/O traffic control involves I/O request processing, token distribution, and dequeuing control.

Figure 4-9 Managing LUN or snapshot I/O queues


Application server 1
Application server 2
1 Sends I/O requests.
1 Sends I/O requests.
Adjusts the number of tokens (using the internal application I/O queue mechanism).
Storage system
2
LUN or snapshot I/O queue 1
...
LUN or snapshot I/O queue n
Processes I/O requests.
3
Disks

4.7 SmartPartition

SmartPartition is a smart cache partitioning technique provided by OceanStor 6800 V5. SmartPartition ensures high performance of mission-critical applications by partitioning cache resources. An administrator can allocate a cache partition of a specific size to an application. The storage system then ensures that the application has exclusive permission to use its allocated cache resources.

NOTE

Cache is the most critical factor that affects the performance of a storage system.

- For a write service, a larger cache size means a higher write combination rate and higher write hit ratio (write hit ratio of a block in a cache).
- For a read service, a larger cache size means a higher read hit ratio.

Different types of services have different cache requirements.

- For a sequential service, the cache size must meet the I/O combination requirements.
- For a random service, a larger cache size indicates that I/Os are more likely to fall onto stripes within the cache, thereby improving performance.

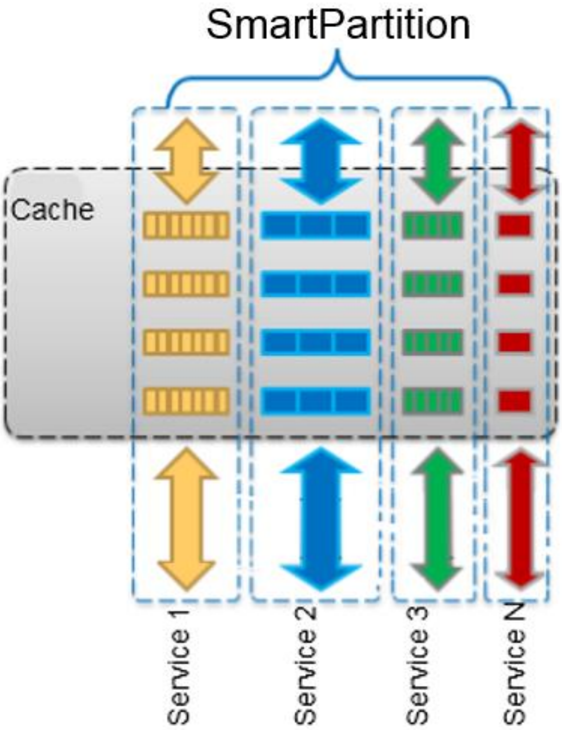
SmartPartition can be used with other QoS techniques (such as SmartQoS) for better QoS effects.

Working Principle

SmartPartition allocates cache resources to services (the actual control objects are LUNs and file systems) based on partition sizes, thereby ensuring the QoS of mission-critical services.

Figure 4-10 illustrates the SmartPartition working principle.

Figure 4-10 SmartPartition working principle



Technical Highlights

- Intelligent partition control

  Based on user-defined cache sizes and QoS policies, SmartPartition automatically schedules system cache resources to ensure optimal system QoS and the required partition quality.

- Ease of use

  SmartPartition is easy to configure. All configurations take effect immediately without a need to restart the system. Users do not need to adjust partitions, thereby improving the usability of partitioning.

Application Scenarios

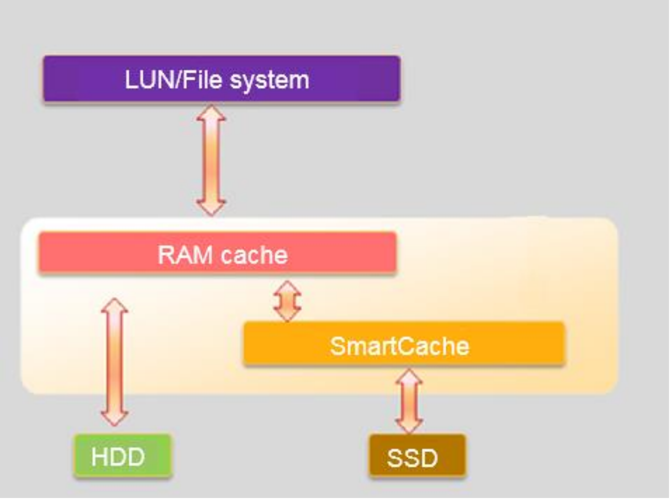
SmartPartition is applicable in scenarios where multiple applications exist, for example:

- A multi-service system. SmartPartition can be used to ensure high performance of core services
- VDI scenario. SmartPartition can be used to ensure high performance for important users
- Multi-tenant scenarios in cloud computing systems

4.8 SmartCache

SmartCache, serving as a read cache module of a storage system, uses SSDs to store clean hot data that RAM cache cannot hold. Figure 4-11 shows the logical architecture of SmartCache.

Figure 4-11 Logical architecture of SmartCache



SmartCache improves performance when accessing hot data through a LUN or file system. The working principle is as follows:

1. After a LUN or file system is enabled with SmartCache, RAM cache delivers hot data to SmartCache.
2. SmartCache establishes a mapping relationship between the data and the SSD in the memory and stores the data on the SSD.
3. When the host delivers a new read I/O to the storage system, the system preferentially looks for the required data in RAM cache.
   - If the required data cannot be found, the system then looks for the required data in SmartCache.
   - If the required data is found in SmartCache, the corresponding data is read from the SSD and returned to the host.

When the amount of data buffered in SmartCache reaches the upper limit, SmartCache selects cache blocks according to the LRU algorithm, clears mapping items in the lookup table, and eliminates data on the buffer blocks. Data writes and elimination are continually performed, ensuring that data stored on SmartCache is frequently accessed data.

Application Scenarios

SmartCache applies to services that have hotspot areas and intensive random read I/Os, such as databases, OLTP applications, web services, and file services.

4.9 SmartErase

As a head cannot read data from or write data to the same point every time, newly written data cannot precisely overwrite the original data. For this reason, some data will remain. Dedicated devices can be used to obtain copies of the original data (data shadow), and the more times data is overwritten, the less residual data that exists.

SmartErase employs overwriting to destroy data on LUNs. SmartErase provides two methods for destroying data: DoD 5220.22-M and customized

- DoD 5220.22-M

  DoD 5220.22-M is a data destruction standard that was introduced by the US Department of Defense (DoD). This standard provides a software method for destroying data on writable storage media, namely, three times of overwriting:

  - Using an 8-bit character to overwrite all addresses
  - Using the complementary codes of the character (complements of 0 and 1) to overwrite all addresses
  - Using a random character to overwrite all addresses

- Customized

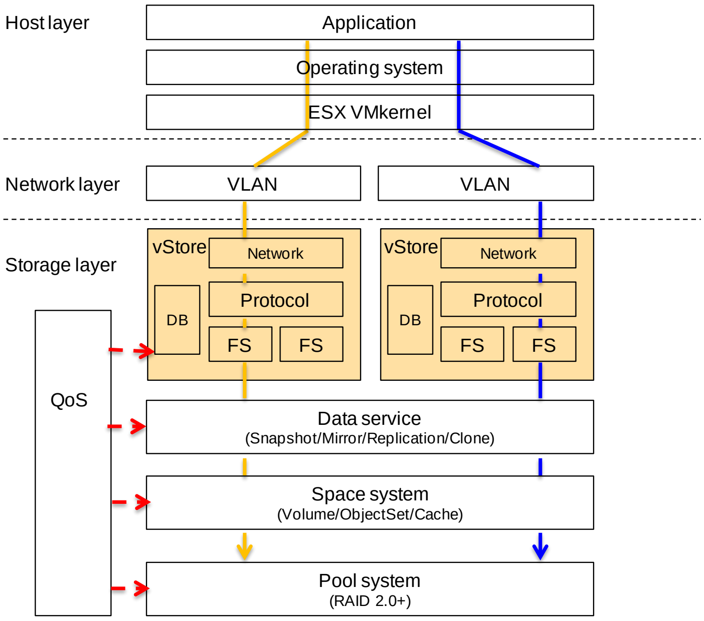
  For customized overwriting, a system generates data based on internal algorithms and uses that data to overwrite all addresses of LUNs a specific number of times. The number of times for overwriting range from 3 to 99 (7 by default).

4.10 SmartMulti-Tenant

SmartMulti-Tenant allows the creation of multiple virtual storage systems (vStores) in a physical storage system. vStores can share the same storage hardware resources in a multi-protocol unified storage architecture, without affecting the data security or privacy of each other.

SmartMulti-Tenant implements management, network, and resource isolation, which prevents data access between vStores and ensures security.

Figure 4-12 Logical architecture of SmartMulti-Tenant



- Management isolation

  Each vStore has its own administrator. vStore administrators can only configure and manage their own storage resources through the GUI or RESTful API. vStore administrators support role-based permission control. When being created, a vStore administrator is assigned a role specific to its permissions.

- Service isolation

  Each vStore has its own file systems, users, user groups, shares, and exports. Users can only access file systems belonging to the vStore through logical interfaces (LIFs).

  Service isolation includes: service data isolation (covering file systems, quotas, and snapshots), service access isolation, and service configuration isolation (typically for NAS protocol configuration).

  - Service data isolation

    System administrators assign different file systems to different vStores, thereby achieving file system isolation. File system quotas and snapshots are isolated in the same way.

  - Service access isolation

    Each vStore has its own NAS protocol instances, including the SMB service, NFS service, and NDMP service.

  - Service configuration isolation

    Each vStore can have its own users, user groups, user mapping rules, security policies, SMB shares, NFS shares, AD domain, DNS service, LDAP service, and NIS service.

- Network isolation

  VLANs and LIFs are used to isolate the vStore network, preventing illegal host access to vStore's storage resources.

  vStores use LIFs to configure services. A LIF belongs only to one vStore to achieve logical port isolation. You can create LIFs from GE ports, 10GE ports, bond ports, or VLANs.

4.11 SmartQuota

In a NAS file service environment, resources are provided to departments, organizations, and individuals as shared directories. Because each department or person has unique resource requirements or limitations, storage systems must allocate and restrict resources, based on the shared directories, in a customized manner. SmartQuota can restrict and control resource consumption for directories, users, and user groups, perfectly tackling all of these challenges.

SmartQuota allows you to configure the following quotas:

- Space soft quota

  Specifies a soft space limit. If any new data writes are performed and would result in this limit being exceeded, the storage system reports an alarm. This alarm indicates that space is insufficient and asks the user to delete unnecessary files or expand the quota. The user can still continue to write data to the directory.

- Space hard quota

  Specifies a hard space limit. If any new data writes are performed and would result in this limit being exceeded, the storage system prevents the writes and reports an error.

- File soft quota

  Specifies a soft limit on the file quantity. If the number of used files exceeds this limit, the storage system reports an alarm. This alarm indicates that the file resources are insufficient and asks the user to delete unnecessary files or expand the quota. The user can still continue to create files or directories.

- File hard quota

  Specifies a hard limit on the file quantity. If the number of used files for a quota exceeds this limit, the storage system prevents the creation of new files or directories and reports an error.

SmartQuota employs space and file hard quotas to restrict the maximum number of resources available to each user. The process is as follows:

1. In each write I/O operation, SmartQuota checks whether the accumulated quota (Quotas of the used space and file quantity + Quotas of the increased space and file quantity in this operation) exceeds the preset hard quota.
   - If yes, the write I/O operation fails.
   - If no, follow-up operations can be performed.
2. After the write I/O operation is allowed, SmartQuota adds an incremental amount of space and number of files to the previously used amount of space and number of files. This is done separately.
3. SmartQuota updates the quota (used amount of space and number of files + incremental amount of space and number of files) and allows the quota and I/O data to be written into the file system.

The I/O operation and quota update succeed or fail at the same time, ensuring that the used capacity is correct during each I/O check.

NOTE

If the directory quota, user quota, and group quota are concurrently configured in a shared directory in which you are performing operations, each write I/O operation will be restricted by these three quotas. All types of quota are checked. If the hard quota of one type of quota does not pass the check, the I/O will be rejected.

SmartQuota does the following to clear alarms: When the used resource of a user is lower than 90% of the soft quota, SmartQuota clears the resource over-usage alarm. In this way, even though the used resource is slightly higher or lower than the soft quota, alarms are not frequently generated or cleared.

4.12 SmartMotion

In the IT industry, enterprises and administration departments are faced with numerous challenges concerning capacity, performance, and costs related to data storage. Enterprises cannot accurately assess the growth of service performance when purchasing storage systems. In addition, as service volume grows, it is hard to adjust existing services after disks are added to legacy storage systems.

To address the preceding problems, enterprises must develop a long-term performance requirement plan in the initial stages of IT system construction.

SmartMotion dynamically migrates data and evenly distributes data across all disks, resolving the problems facing customers. Customers need to assess only recent performance requirements when purchasing storage systems, significantly reducing initial purchase costs and total TCO. If the system performance requirements increase with the service volume, customers only need to add disks to storage systems. After the disks are added, SmartMotion migrates data and evenly distributes the original service data across all disks, notably improving service performance.

SmartMotion is implemented based on RAID 2.0+. For RAID 2.0+, the space of all the disks in a disk domain is divided into fixed CKs. When CKGs are required, disks are selected in a pseudo-random manner and the CKs from these disks compose CKGs based on a RAID algorithm. All CKs are then evenly distributed across all disks.

When disks are added into a disk domain, the storage system starts SmartMotion. The implementation of a SmartMotion task is performed as follows:

1 Selects the first CKG that is not load-balanced.

2 Selects disks for the CKG in a pseudo-random manner.

- If the selected disks are consistent with the original disks of the CKG, this CKG is skipped and the process goes back to step 1.
- If the selected disks are inconsistent, the process goes to step 3.

3 Compares the original disks of the CKG with the newly selected disks and computes the mapping between the source disks and the target disks based on disk differences. Then, selects the source disks and target disks.

4 Traverses all the source disks for the CKG, allocates new CKs from the target disks, and migrates data from the source disks to the target disks to release the source disks.

5 After all CKGs in the system are traversed, the SmartMotion task is complete. Otherwise, the process goes back to step 1 and processes the next CKG.

After the SmartMotion task is complete, disks are selected for all CKGs in a pseudo-random manner and all required data is migrated. All CKs are evenly distributed across all available disks.

5 Hyper Series Features

5.1 HyperSnap

5.1.1 HyperSnap for Block

OceanStor 6800 V5 uses HyperSnap to quickly generate a consistent image, that is, a duplicate, for a source LUN at a point in time without interrupting the services that are running on the source LUN. The duplicate is available immediately after being generated, and reading or writing the duplicate has no impact on source data. HyperSnap helps with online backups, data analysis, and application testing. It works based on the mapping table and copy-on-write (COW) technology.

Technical Highlights

- Zero-duration backup window

  Traditional backup deteriorates application servers' performance, or even interrupts ongoing services. Therefore, a traditional backup task can be executed only after application servers are stopped or during off-peak hours. A backup window refers to the data backup duration, which is the maximum downtime tolerated by applications. HyperSnap can back up data online, and requires a backup window that takes almost zero time and does not interrupt services.
- Less occupied disk capacity

After creating a consistent copy of a source LUN, HyperSnap uses a COW volume to save data on the source LUN at the snapshot point in time upon the first update. The size of the COW volume is independent of the source LUN size but dependent on the amount of data changed on the source LUN. If the amount of changed data is small, the snapshot captures a consistent copy of the source LUN and uses a small disk space. The consistent copy can be used for service tests, saving disk space.

- Quick data restoration

  Data backed up using traditional offline backup methods cannot be read online. Long-time data restoration is required before a usable duplicate of the source data that was backed up at the specific point in time is available. HyperSnap can directly read the snapshot volume to obtain data on the source volume at the snapshot point in time. This allows it to quickly restore data in the case of data corruption on the source volume.

- Data consistency by consistency group

  For OLTP applications, snapshots for multiple pieces of source LUN data must be created at the same time. In this way, associated application data distributed on different LUNs can be kept at the same point in time. For example, the management data, service data, and log information of an Oracle database application are distributed on different source LUNs. Consistent copies of the three source LUNs must be created at the same time. Otherwise, the three source LUNs cannot be restored to the same point in time, losing data dependency. HyperSnap provides snapshot consistency groups to resolve this problem. I/Os on multiple source LUNs are frozen at the snapshot point in time, and a snapshot is generated for the frozen I/Os.

- Continuous data protection through timing snapshots

  OceanStor 6800 V5 allows snapshots to be created for a source LUN at multiple points in time. Working together with ReplicationDirector on the host, HyperSnap can create or delete snapshots at minute-level intervals. In addition, a snapshot policy can be set to automate the activation and stopping of snapshot tasks. As time elapses, snapshots are generated at multiple points, implementing continuous data protection at a low cost.

- Snapshot copy

  A snapshot copy backs up the data of a snapshot at the snapshot activation point in time. It does not back up data written to the snapshot after the snapshot activation point in time. The snapshot copy and source snapshot share the COW volume space of the source LUN, but the private space is independent. The snapshot copy is a writable snapshot and is independent of the source snapshot. The read and write processes of a snapshot copy are the same as those of a common snapshot.

  Snapshot copy allows users to obtain multiple data copies of a snapshot for various purposes.

5.1.2 HyperSnap for File

OceanStor 6800 V5 uses HyperSnap to quickly generate a consistent image, that is, a duplicate, for a source file system at a certain point in time without interrupting services running on the source file system. This duplicate is available immediately after being generated, and reading or writing the duplicate does not impact the data on the source file system. HyperSnap helps with online backups, data analysis, and application testing. HyperSnap can:

- Create file system snapshots and back up these snapshots to tapes.
- Provide data backups of the source file system so that end users can restore accidentally deleted files.
- Work together with HyperReplication and HyperVault for remote replication and backup.

HyperSnap works based on ROW file systems. In a ROW file system, new or modified data does not overwrite the original data but instead is written to newly allocated storage space. This ensures enhanced data reliability and high file system scalability. ROW-based HyperSnap, used for file systems, can create snapshots in seconds. The snapshot data does not occupy any additional disk space unless the source files are deleted or modified.

Technical Highlights

- Zero-duration backup window

  A backup window refers to the maximum backup duration tolerated by applications before data is lost. Traditional backup deteriorates file system performance, or can even interrupt ongoing applications. Therefore, a traditional backup task can only be executed after applications are stopped or if the workload is comparatively light. HyperSnap can back up data online, and requires a backup window that takes almost zero time and does not interrupt services.

- Snapshot creation within seconds

  To create a snapshot for a file system, only the root node of the file system needs to be copied and stored in caches and protected against power failure. This reduces the snapshot creation time to seconds.

- Reduced performance loss

  HyperSnap makes it easy to create snapshots for file systems. Only a small amount of data needs to be stored on disks. After a snapshot is created, the system checks whether data is protected by a snapshot before releasing the data space. If the data is protected by a snapshot, the system records the space of the data block that is protected by the snapshot but is deleted by the file system. This results in a negligible impact on system performance. Background data space reclamation contends some CPU and memory resources against file system services only when the snapshot is deleted. However, performance loss remains low.

- Less occupied disk capacity

  The file system space occupied by a snapshot (a consistent duplicate) of the source file system depends on the amount of data that changed after the snapshot was generated. This space never exceeds the size of the file system at the snapshot point in time. For a file system with little changed data, only a small storage space is required to generate a consistent duplicate of the file system.

- Rapid snapshot data access

  A file system snapshot is presented in the root directory of the file system as an independent directory. Users can access this directory to quickly access the snapshot data. If snapshot rollback is not required, users can easily access the data at the snapshot point in time. Users can also recover data by copying the file or directory if the file data in the file system is corrupted.

  If using a Windows client to access a CIFS-based file system, a user can restore a file or folder to the state at a specific snapshot point in time. To be specific, a user can right-click the desired file or folder, choose **Restore previous versions** from the short-cut menu, and select one option for restoration from the displayed list of available snapshots containing the previous versions of the file or folder.

- Quick file system rollback

  Backup data generated by traditional offline backup tasks cannot be read online. A time-consuming data recovery process is inevitable before a usable duplicate of the source data at the backup point in time is available. HyperSnap can directly replace the file system root with specific snapshot root and clear cached data to quickly roll the file system back to a specific snapshot point in time.

You must exercise caution when using the rollback function because snapshots created after the rollback point in time are automatically deleted after a file system rollback succeeds.

- Continuous data protection by timing snapshots

  HyperSnap enables users to configure policies to automatically create snapshots at specific time points or at specific intervals.

  The maximum number of snapshots for a file system varies depending on the product model. If the upper limit is exceeded, the earliest snapshots are automatically deleted. The file system also allows users to periodically delete snapshots.

  As time elapses, snapshots are generated at multiple points, implementing continuous data protection at a low cost. It must be noted that snapshot technology cannot achieve real continuous data protection. The interval between two snapshots determines the granularity of continuous data protection.

5.2 HyperClone

5.2.1 HyperClone for Block

HyperClone generates a complete physical copy of a source LUN at a point in time without interrupting ongoing services. If the clone is split, writing data to and reading data from the physical copy do not affect source LUN data.

Working Principle

HyperClone is implemented through a combination of bitmap and COW, and a combination of bitmap and dual-write (where data is written to the primary and secondary LUNs simultaneously). The working principle is as follows:

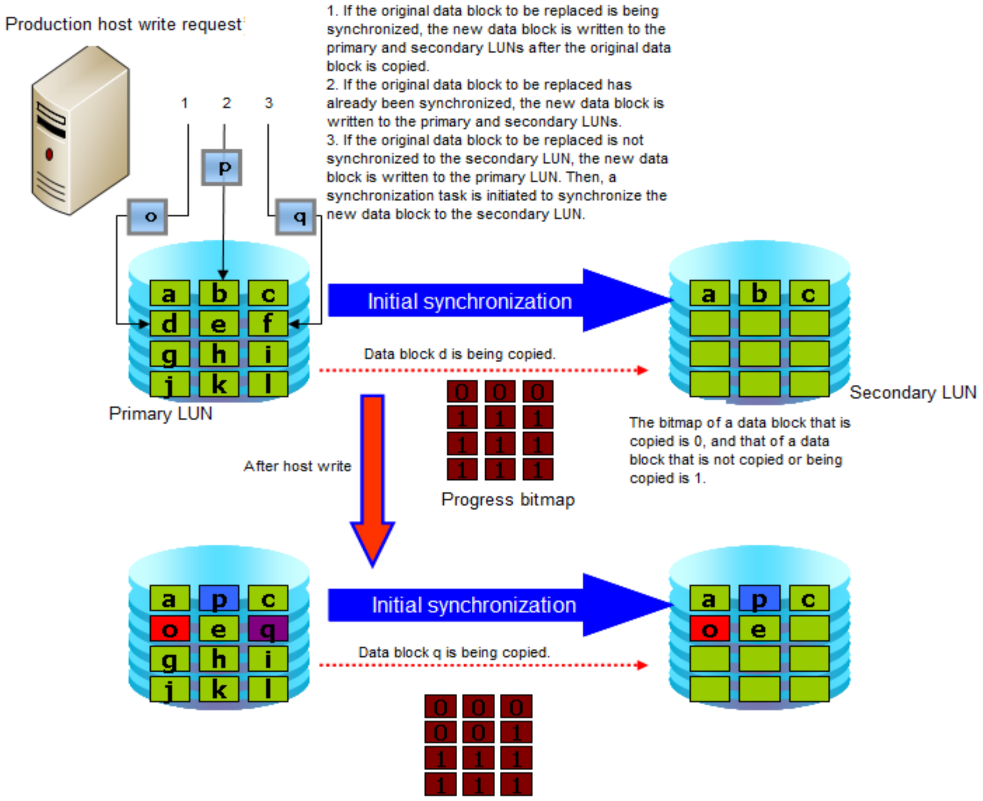
After a secondary LUN is added to a clone group, all data in the primary LUN is replicated to the secondary LUN by default. This is called initial synchronization, and a progress bitmap reflects the synchronization process. If the primary LUN receives a write request from the production host during initial synchronization, the storage system checks the synchronization progress, and performs subsequent operations as follows:

- If the write-targeted data block has not been synchronized to the secondary LUN, data is written to the primary LUN and the storage system notifies the host of a write success. Then, the data is synchronized to the secondary LUN during the subsequent synchronization task.
- If the write-targeted data block has already been synchronized, data is written to both the primary and secondary LUNs.
- If the write-targeted data block is being synchronized, the storage system waits until the data block is copied. Then, the storage system writes data to both the primary and secondary LUNs.

After the initial synchronization is complete, the clone group can be split. After splitting the clone group, the primary and secondary LUNs can be used separately for testing and data analysis. Changing the data in a primary or secondary LUN does not affect the other LUN, and the progress bitmap records data changes to both LUNs.

Figure 5-1 illustrates the HyperClone working principle.

Figure 5-1 HyperClone working principle



1. If the original data block to be replaced is being synchronized, the new data block is written to the primary and secondary LUNs after the original data block is copied.
2. If the original data block to be replaced has already been synchronized, the new data block is written to the primary and secondary LUNs.
3. If the original data block to be replaced is not synchronized to the secondary LUN, the new data block is written to the primary LUN. Then, a synchronization task is initiated to synchronize the new data block to the secondary LUN.

The bitmap of a data block that is copied is 0, and that of a data block that is not copied or being copied is 1.

Technical Highlights

- 1-to-16 mode

  HyperClone allows you to assign a maximum of 16 secondary LUNs for a primary LUN. A clone in 1-to-N mode can back up multiple copies of source data for various data analyses.

- Zero-duration backup window

  HyperClone backs up data without interrupting services, ensuring a backup window that takes almost no time.

- Dynamic adjustment of copy speeds

  You can manually change the copy speed to prevent conflicts between synchronization tasks and a production services. If a storage system has detected that the service load is heavy, you can manually lower the copy speed to free system resources for services. When the service load is light, you can increase the copy speed to mitigate service conflicts during peak hours.

- Reverse synchronization

  If data on the primary LUN is incomplete or corrupted, you can recover the original service data by performing incremental reverse synchronization from the secondary LUN to the primary LUN.

- Automatic recovery

  If a problem occurs, for example, the primary or secondary LUN fails, the corresponding clone created on the system will enter a disconnected state. After this problem is resolved, the clone is recovered based on a specified recovery policy.

  - If the specified policy is automatic recovery, the clone automatically enters the synchronizing state, and differential data is incrementally synchronized to the secondary LUN.
  - If the specified policy is manual recovery, the clone waits to be recovered and you must manually initiate synchronization.

  Incremental synchronization greatly reduces the fault/disaster recovery time.

- Clone consistency group

  In OLTP applications, you must, typically, simultaneously split multiple clone pairs to obtain data copies at the same point in time. In this way, associated data that is distributed to different LUNs can be maintained at the same point in time. HyperClone can split multiple clone pairs simultaneously, freezing data on multiple primary LUNs at the point in time at which the split was performed, and obtaining consistent copies of the primary LUNs.

Application Scenarios

- Data backup

  HyperClone can generate multiple physical copies of a primary volume and allow multiple services to access data concurrently.

- Data recovery and protection

  If primary LUN data is corrupted by a virus or human error, or is physically damaged, a data copy from the secondary LUN taken at a suitable point in time can be reversely copied to the primary LUN. Then, the primary LUN can be restored to its state when the data copy was created.

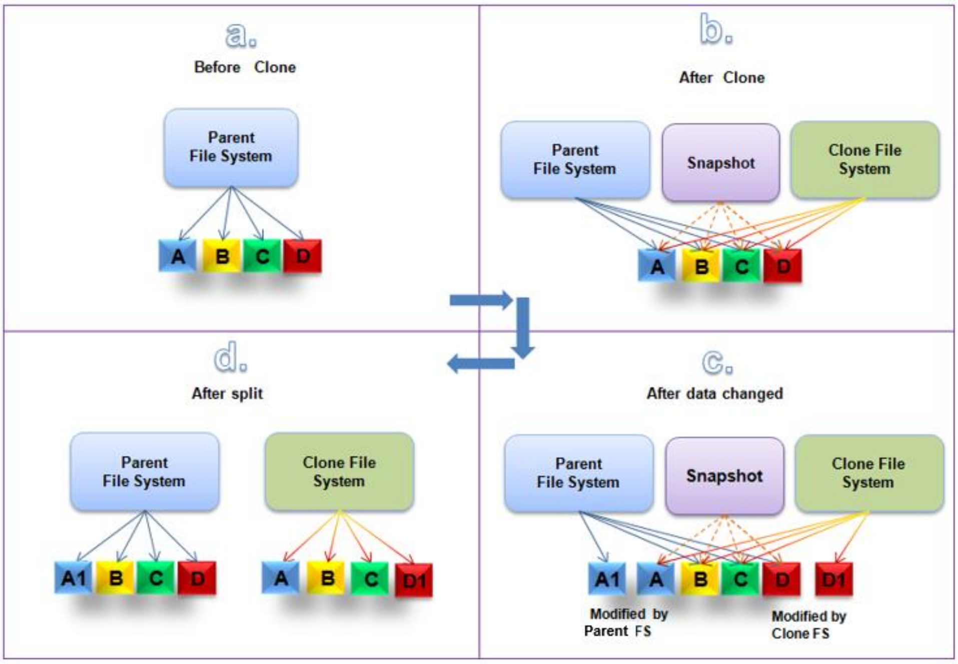
5.2.2 HyperClone for File

HyperClone creates a clone file system, which is a copy, for a parent file system at a specified point in time. Clone file systems can be shared to clients exclusively to meet the requirements of rapid deployment, application tests, and DR drills.

Working Principle

A clone file system is a readable and writable copy taken from a point in time that is based on redirect-on-write (ROW) and snapshot technologies.

Figure 5-2 Working principle of HyperClone for File



- As shown in Figure a, the storage system writes new or modified data onto the newly allocated space of the ROW-based file system, instead of overwriting the original data. The storage system records the point in time of each data write, indicating the write sequence. The points in time are represented by serial numbers, in ascending order.
- As shown in Figure b, the storage system creates a clone file system as follows:
  - Creates a read-only snapshot in the parent file system.
  - Copies the root node of the snapshot to generate the root node of the clone file system.
  - Creates an initial snapshot in the clone file system.

  This process is similar to the process of creating a read-only snapshot during which no user data is copied. Snapshot creation can be completed in one or two seconds. Before data is modified, the clone file system shares data with its parent file system.
- As shown in Figure c, modifying either the parent file system or the clone file system does not affect the other system.
  - When the application server modifies data block A of the parent file system, the storage pool allocates new data block A1 to store new data. Data block A is not released because it is protected by snapshots.
  - When the application server modifies data block D of the clone file system, the storage pool allocates new data block D1 to store new data. Data block D is not released because its write time is earlier than the creation time of the clone file system.
- Figure d shows the procedure for splitting a clone file system:
  - Deletes all read-only snapshots from the clone file system.
  - Traverses the data blocks of all objects in the clone file system, and allocates new data blocks in the clone file system for the shared data by overwriting data. This splits shared data.
  - Deletes the associated snapshots from the parent file system.

After splitting is complete, the clone file system is independent of the parent file system. The time required to split the clone file system depends on the size of the share data.

Technical Highlights

- Rapid deployment

  In most scenarios, a clone file system can be created in seconds and can be accessed immediately after being created.
- Saved storage space

  A clone file system shares data with its parent file system and occupies extra storage space only when it modifies shared data.
- Effective performance assurance

  HyperClone has a negligible impact on system performance because a clone file system is created based on the snapshot of the parent file system.
- Splitting a clone file system

  After a clone file system and its parent file system are split, they become completely independent of each other.

5.3 HyperReplication

OceanStor 6800 V5 uses HyperReplication in synchronous mode (HyperReplication/S) and asynchronous mode (HyperReplication/A) to implement remote replication. Developed on the OceanStor OS unified storage software platform, OceanStor 6800 V5 is compatible with the replication protocols of all Huawei OceanStor converged storage products. Therefore, OceanStor 6800 V5 can interconnect with all new or legacy Huawei converged storage systems to construct highly flexible disaster recovery solutions.

OceanStor 6800 V5 supports HyperReplication/S for Block, HyperReplication/A for Block, and HyperReplication/A for File.

5.3.1 HyperReplication/S for Block

Working Principle

HyperReplication/S maintains data consistency between primary and secondary LUNs based on a log mechanism. The working principle of HyperReplication/S is as follows:

After a synchronous remote replication relationship is established between primary and secondary LUNs, initial synchronization is implemented to copy all data from the primary LUN to the secondary LUN.

After initial synchronization is complete, I/Os are processed as follows:

1 The primary site receives a write request from a production host. HyperReplication sets the differential log value to differential for the data block that corresponds to the request.

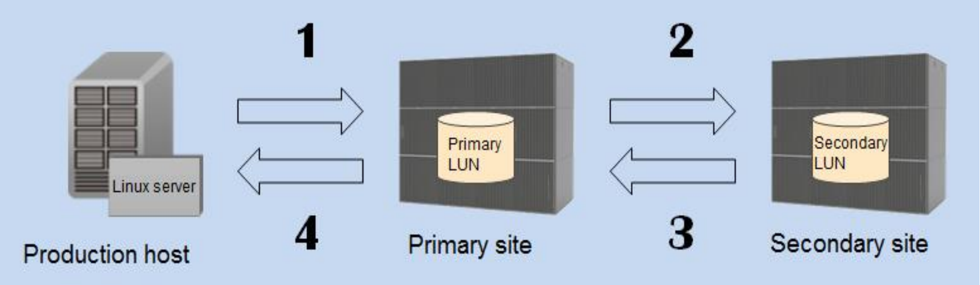
2 The requested data is written to both the primary and secondary LUNs. When writing data to the secondary LUN, the primary site sends the data to the secondary site over a preset link.

3 If data is successfully written to both the primary and secondary LUNs, the corresponding differential log value is changed to non-differential. If data is not

successfully written, the value remains differential, and the data block is copied again during the next synchronization.

4 The primary site returns a write acknowledgement to the production host.

Figure 5-3 Working principle of HyperReplication/S



Technical Highlights

- Zero data loss

  HyperReplication/S synchronizes data from the primary LUN to the secondary LUN in real time, ensuring zero recovery point objective (RPO).

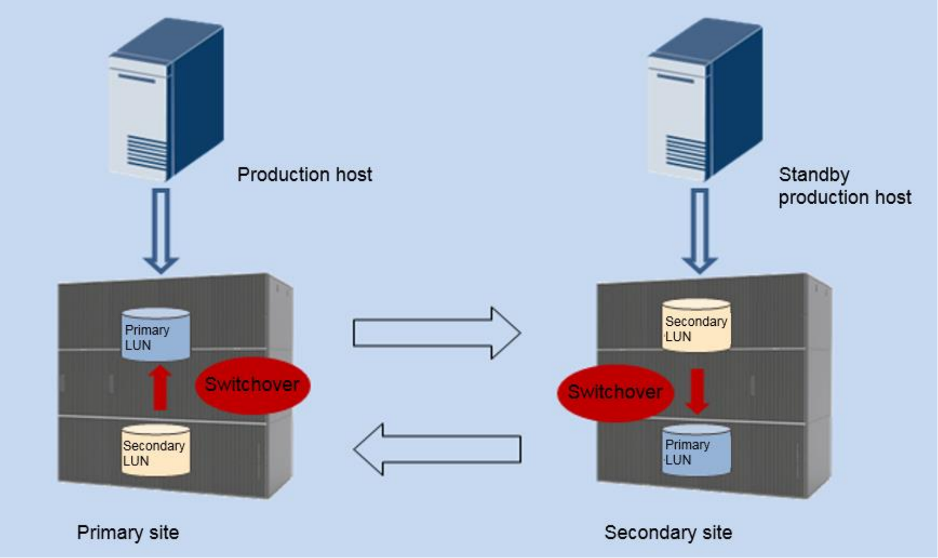
- Support for the split mode

  In split mode, write requests initiated by the production host are delivered only to the primary LUN. This mode meets certain user needs, such as temporary link maintenance, network bandwidth expansion, and data being saved at a certain point in time on the secondary LUN.

- Primary/Secondary switchover

  HyperReplication/S supports primary/secondary switchover. In the following figure, the primary LUN at the primary site becomes the new secondary LUN after the switchover, and the secondary LUN at the secondary site becomes the new primary LUN. This process requires only some simple operations on the host side. The major operation, which can be done in advance, is to map the new primary LUN to the standby production host. Then, the standby production host at the secondary site takes over services and delivers subsequent I/O requests to the new primary LUN.

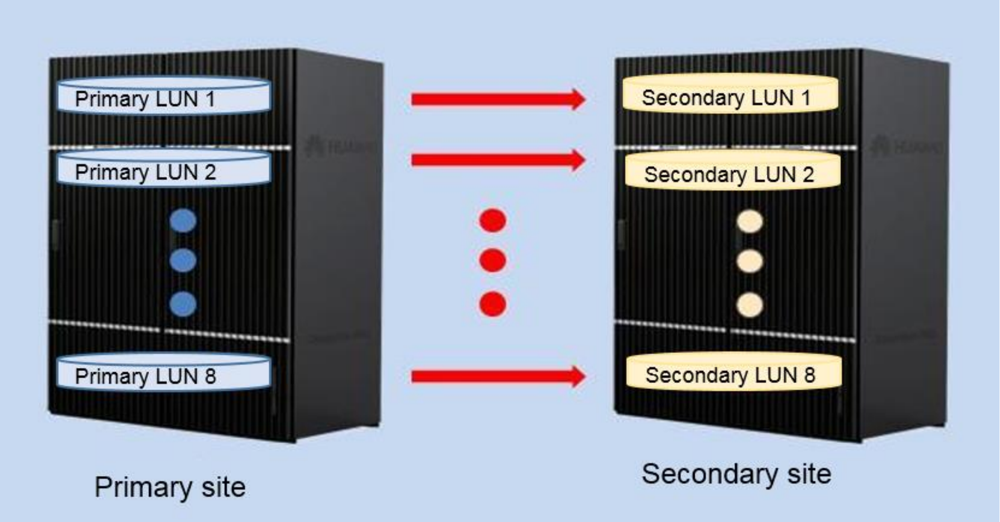
Figure 5-4 Primary/Secondary switchover



- Support for consistency groups

  HyperReplication/S provides the consistency group function to ensure that data is simultaneously replicated among LUNs. HyperReplication/S allows you to add remote replication pairs to a consistency group. When performing splitting, synchronization, or a primary/secondary switchover for a consistency group, the operations apply to all members of the consistency group. In addition, if a fault occurs, all members of the consistency group enter simultaneously the disconnected state.

Figure 5-5 Consistency group of HyperReplication/S



Application Scenarios

HyperReplication/S applies to local data disaster recovery and backup, which are scenarios where the primary site is near the secondary site. An example of this is intra-city disaster recovery. For HyperReplication/S, a write success acknowledgement is returned to the production host only after the data in the write request is written to both the primary site and secondary site. If the primary site is far from the secondary site, the write latency of foreground applications is relatively high, affecting foreground services.

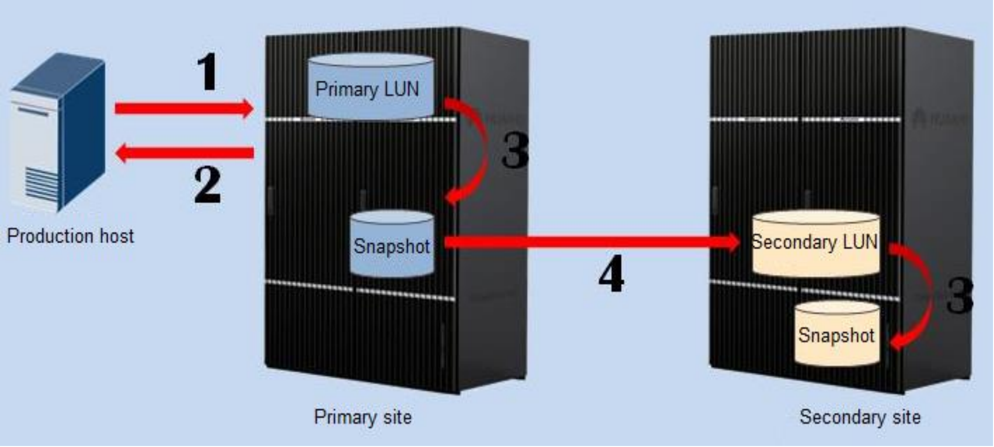
5.3.2 HyperReplication/A for Block

Working Principle

The working principle of HyperReplication/A is similar to that of HyperReplication/S: After an asynchronous remote replication relationship is set up between primary and secondary LUNs, initial synchronization is implemented to copy all data from the primary LUN to the secondary LUN. After the initial synchronization is complete, the data status of the secondary LUN is changed to **Synchronized** or **Consistent**. Then, I/Os are processed as follows:

1. The primary site receives a write request from a production host.
2. The primary site writes the new data to the primary LUN and immediately sends a write acknowledgement to the host.
3. Incremental data is automatically synchronized from the primary LUN to the secondary LUN based on a user-defined synchronization period. This can range from 1 to 1440 minutes. (If the synchronization type is **Manual**, synchronization must be triggered manually.)
4. Before synchronization begins, a snapshot is generated for the primary and secondary LUNs respectively. The snapshot of the primary LUN ensures that the data read from the primary LUN during synchronization remains unchanged. The snapshot of the secondary LUN backs up the data of the secondary LUN in case an exception during synchronization causes the data to become unavailable.
5. During synchronization, data is read from the snapshot of the primary LUN and copied to the secondary LUN. After synchronization is complete, the snapshots of the primary and secondary LUNs are discarded, and the system waits for the next synchronization.

Figure 5-6 Working principle of HyperReplication/A



Technical Highlights

- Data compression and data encryption

  HyperReplication/A uses the AES-256 algorithm to support data encryption specific to iSCSI links. It supports data compression specific to iSCSI links. The data compression ratio varies significantly depending on service data type. The maximum compression ratio of database services is 4:1.

- Quick response to host requests

  After a host writes data to the primary LUN at the primary site, the primary site immediately returns a write acknowledgement 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, without impacting the access to the primary LUN. HyperReplication/A does not synchronize incremental data from the primary LUN to the secondary LUN in real time. Therefore, the amount of data lost is determined by the synchronization period. This can range from 3 seconds (default value) to 1440 minutes, and be specified based on site requirements.

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

  HyperReplication/A supports splitting, synchronization, primary/secondary switchover, and recovery functions.

- Consistency groups

  You can create and delete consistency groups, create and delete HyperReplication pairs in a consistency group, and split pairs. When performing splitting, synchronization, or a primary/secondary switchover for a consistency group, the operations apply to all members of the consistency group.

Application Scenarios

HyperReplication/A applies to remote data disaster recovery and backup, which are scenarios where the primary and secondary sites are far from each other, or the network bandwidth is limited. For HyperReplication/A, the write latency of foreground applications not affected by the distance between the primary and secondary sites.

5.3.3 HyperReplication/A for File

HyperReplication/A supports the long-distance data disaster recovery of file systems. It copies all content of a primary file system to the secondary file system. This implements remote disaster recovery across data centers and minimizes the performance deterioration caused by remote data transmission. HyperReplication/A also applies to file systems within a storage system for local data disaster recovery, data backup, and data migration.

HyperReplication/A implements data replication based on the file system object layer, and periodically synchronizes data between primary and secondary file systems. All data changes made to the primary file system since the last synchronization will be synchronized to the secondary file system.

Working Principle

- Object layer-based replication

  HyperReplication/A implements data replication based on the object layer. The files, directories, and file properties of file systems consist of objects. Object layer-based replication copies objects from the primary file system to the secondary file system without considering complex file-level information, such as dependency between files and directories, and file operations, simplifying the replication process.

- Periodical replication based on ROW

  HyperReplication/A implements data replication based on ROW snapshots.

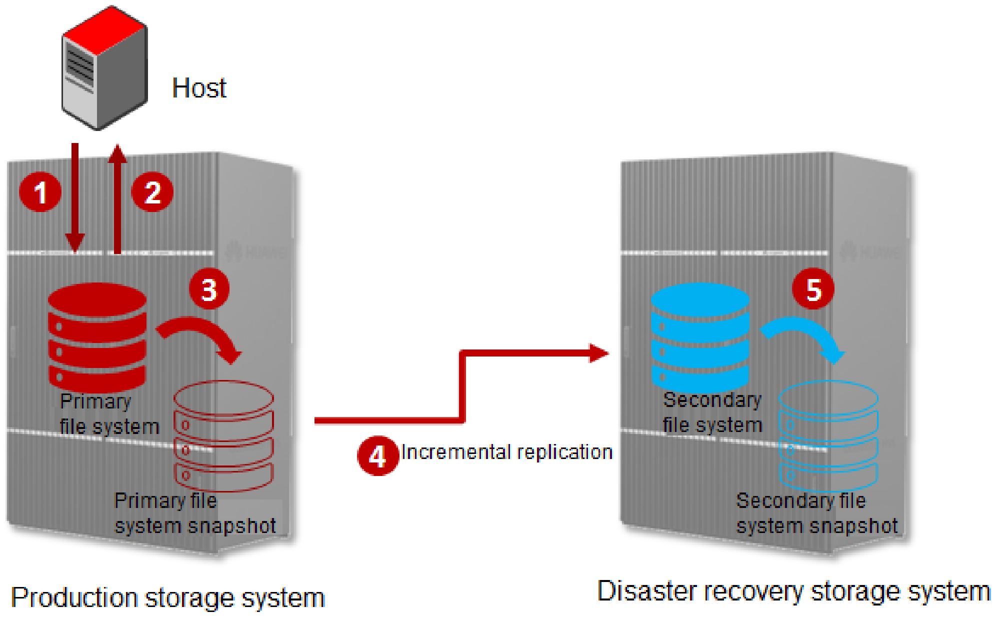
  - Periodic replication improves replication efficiency and bandwidth utilization. During a replication period, the data that was written most recently is always copied. For example, if data in the same file location is modified multiple times, the data written last is copied.

- File systems and their snapshots employ ROW to process data writes. Regardless of whether a file system has a snapshot, data is always written to the new address space, and service performance will not decrease even if snapshots are created. Therefore, HyperReplication/A has a slight impact on production service performance.

Written data is periodically replicated to the secondary file system in the background. Replication periods are defined by users. The addresses, rather than the content of incremental data blocks in each period, are recorded. During each replication period, the secondary file system is incomplete before all incremental data is completely transferred to the secondary file system.

After the replication period ends and the secondary file system becomes a point of data consistency, a snapshot is created for the secondary file system. If the next replication period is interrupted because the production center malfunctions or the link goes down, HyperReplication/A can restore the secondary file system data to the last snapshot point, ensuring consistent data.

Figure 5-7 Working principle of HyperReplication/A for File


Host
1
2
3
Primary file system
Primary file system snapshot
4
Incremental replication
5
Secondary file system
Secondary file system snapshot
Production storage system
Disaster recovery storage system

1. The production storage system receives a write request from a production host.
2. The production storage system writes the new data to the primary file system and immediately sends a write acknowledgement to the host.
3. When a replication period starts, HyperReplication/A creates a snapshot for the primary file system.
4. The production storage system reads and replicates snapshot data to the secondary file system based on the incremental information received since the last synchronization.
5. After incremental replication is complete, the content of the secondary file system is the same as the snapshot of the primary file system. The secondary file system becomes the point of data consistency.

Technical Highlights

- Splitting and incremental resynchronization

  If you want to suspend data replication from the primary file system to the secondary file system, you can split the remote replication pair. For HyperReplication/A, splitting will stop the ongoing replication process and later periodical replication.

  After splitting, if the host writes new data, the incremental information will be recorded. You can start a synchronization session after splitting. During resynchronization, only incremental data is replicated.

  Splitting applies to device maintenance scenarios, such as storage array upgrades and replication link changes. In such scenarios, splitting can reduce the number of concurrent tasks so that the system becomes more reliable. The replication tasks will be resumed or restarted after maintenance.

- Automatic recovery

  If data replication from the primary file system to the secondary file system is interrupted due to a fault, remote replication enters the interrupted state. If the host writes new data when remote replication is in this state, the incremental information will be recorded. After the fault is rectified, remote replication is automatically recovered, and incremental resynchronization is automatically implemented.

- Readable and writable secondary file system and incremental failback

  Normally, a secondary file system is readable but not writable. When accessing a secondary file system, the host reads the data on snapshots generated during the last backup. After the next backup is completed, the host reads the data on the new snapshots.

  A readable and writable secondary file system applies to scenarios in which backup data must be accessed during replication.

  You can set a secondary file system to readable and writable if the following conditions are met:

  - Initial synchronization has been implemented. For HyperReplication/A, data on the secondary file system is in the complete state after initial synchronization.
  - The remote replication pair is in the split or interrupted state.

  If data is being replicated from the primary file system to the secondary file system (the data is inconsistent on the primary and secondary file systems) and you set the secondary file system to readable and writable, HyperReplication/A restores the data in the secondary file system to the point in time at which the last snapshot was taken.

  After the secondary file system is set to readable and writable, HyperReplication/A records the incremental information about data that the host writes to the secondary file system for subsequent incremental resynchronization. After replication recovery, you can replicate incremental data from the primary file system to the secondary file system or from the secondary file system to the primary file system (a primary/secondary switchover is required before synchronization). Before a replication session starts, HyperReplication/A restores target end data to a point in time at which a snapshot was taken and the data was consistent with source end data. Then, HyperReplication/A performs incremental resynchronization from the source end to the target end.

  Readable and writable secondary file systems are commonly used in disaster recovery scenarios.

- Primary/Secondary switchover

  Primary/secondary switchover exchanges the roles of the primary and secondary file systems. These roles determine the direction in which the data is copied. Data is always copied from the primary file system to the secondary file system.

  Primary/secondary switchover is commonly used for failback during disaster recovery.

- Quick response to host I/Os

  All I/Os generated during file system asynchronous remote replication are processed in the background. A write success acknowledgement is returned immediately after host data is written to the cache. Incremental information is recorded and snapshots are created only when data is flushed from cache to disks. Therefore, host I/Os can be responded to quickly.

5.4 HyperMetro

HyperMetro, an array-level active-active technology provided by OceanStor 6800 V5, enables two storage systems to work in active-active mode in two different locations up to 100 km away from each other. For example, the systems could be in the same equipment room or just in the same city.

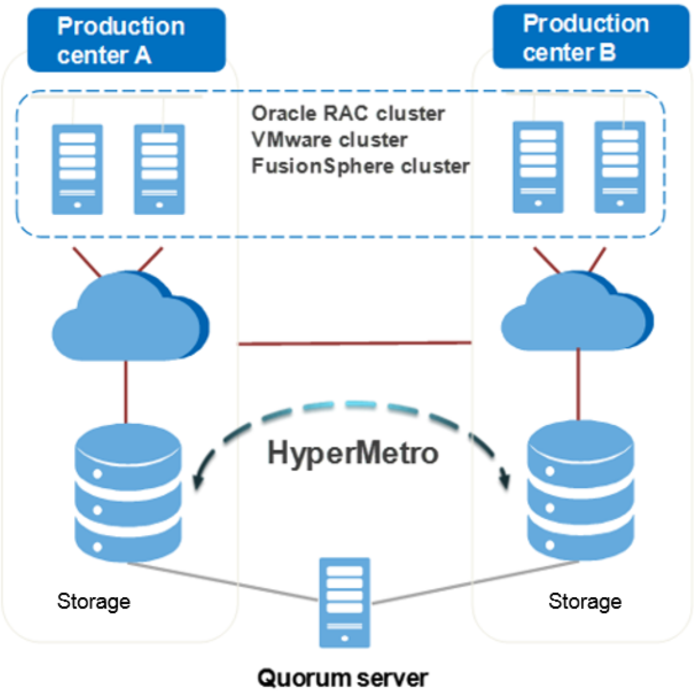
OceanStor 6800 V5 supports both HyperMetro for Block and HyperMetro for File.

5.4.1 HyperMetro for Block

HyperMetro allows two LUNs from two storage arrays to maintain real-time data consistency and to be accessible to hosts. If one storage array fails, hosts automatically choose the path to the other storage array for service access. If the links between the storage arrays are interrupted, a quorum server deployed at a third location determines which storage array continues providing services.

HyperMetro supports both Fibre Channel and IP networking (GE/10GE).

Figure 5-8 Architecture of HyperMetro for Block



Technical Highlights

- Gateway-free active-active solution

  Simple networking makes deployment easy. The gateway-free design improves reliability and performance because it ensures one less possible failure point, and eliminates the 0.5 ms to 1 ms latency caused by a gateway.

- Active-active mode

  Storage arrays in two data centers are accessible to hosts, implementing load balancing across data centers.

- Site access optimization

  UltraPath is optimized specifically for active-active scenarios. It can identify region information to reduce cross-site access, reducing latency. UltraPath can read data from the local or remote storage array. However, when the local storage array is working properly, UltraPath preferentially reads data from and writes data to the local storage array. This prevents data reads and writes across data centers.

- FastWrite

  In a common SCSI write process, a write request goes back and forth twice between two data centers to complete two interactions, Write Alloc and Write Data. FastWrite optimizes the storage transmission protocol and reserves cache space on the destination array for receiving write requests. Write Alloc is omitted and only one interaction is required. FastWrite halves the time required for data synchronization between two arrays, improving the overall performance of the HyperMetro solution.

- Service granularity-based arbitration

  If links between two sites fail, HyperMetro can enable some services to run preferentially in data center A and others in data center B based on service configurations. Compared with traditional arbitration, where only one data center provides services, HyperMetro improves resource usage of hosts and storage systems and balances service loads. Service granularity-based arbitration is implemented based on LUNs or consistency groups.

- Automatic link quality adaptation

  If multiple links exist between two data centers, HyperMetro automatically balances loads among links based on the quality of each link. The system dynamically monitors link quality and adjusts the load ratio between links to minimize the retransmission rate and improve network performance.

- Compatibility with other features

  HyperMetro can work with SmartThin, SmartTier, SmartQoS, and SmartCache. HyperMetro can enable heterogeneous LUNs managed by the SmartVirtualization feature to work in A/A mode. HyperMetro can also work with HyperSnap, HyperClone, HyperMirror, and HyperReplication to form a more complex, advanced data protection solution, such as the Disaster Recovery Data Center Solution (Geo-Redundant Mode), which uses local A/A and remote replication.

- Dual quorum servers

  HyperMetro supports dual quorum servers. If one quorum server fails, its services are seamlessly switched to the other, preventing a single point of failure (SPOF) and improving the reliability of the HyperMetro solution.

5.4.2 HyperMetro for File

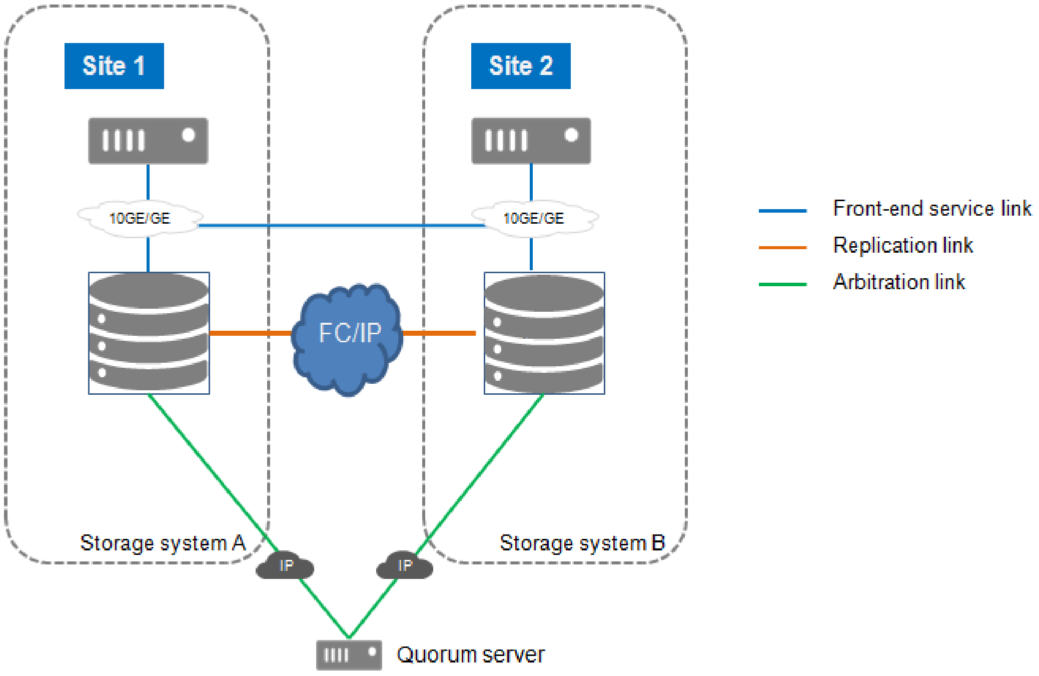
HyperMetro enables hosts to virtualize the file systems of two storage systems as a single file system on a single storage system. In addition, HyperMetro keeps data in both of these file systems consistent. Data is read from or written to the primary storage system, and is synchronized to the secondary storage system in real time. If the primary storage system fails, HyperMetro uses vStore to switch services to the secondary storage system, without losing any data or interrupting any applications.

HyperMetro provides the following benefits:

- High availability with geographic protection
- Easy management
- Minimal risk of data loss, reduced system downtime, and quick disaster recovery
- Negligible disruption to users and client applications

HyperMetro supports both Fibre Channel and IP networking (GE/10GE).

Figure 5-9 Architecture of HyperMetro for File



Technical Highlights

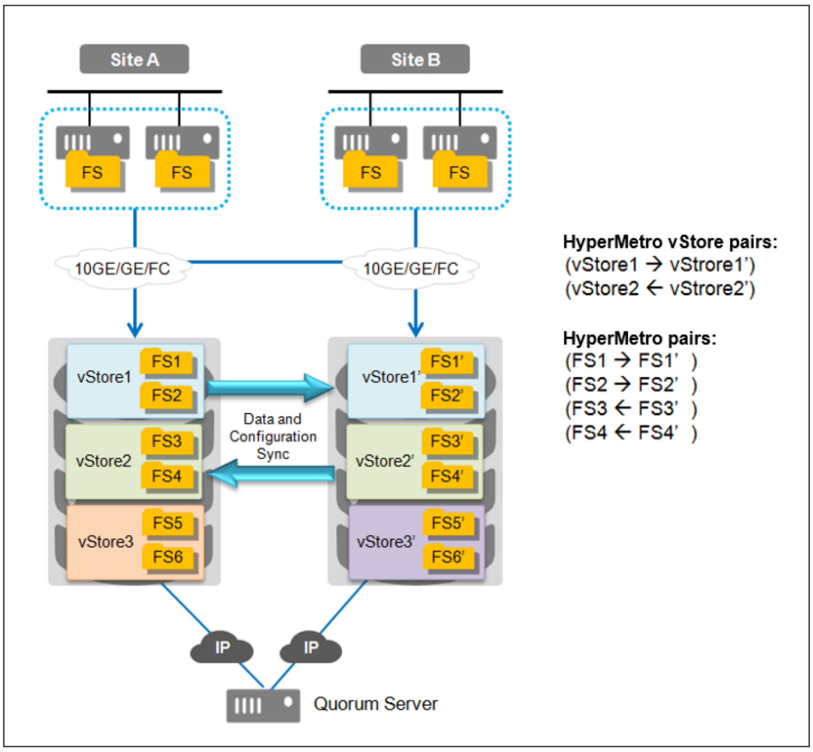
- Gateway-free solution

  With the gateway-free design, host I/O requests do not need to be forwarded by storage gateway, avoiding corresponding I/O forwarding latency and gateway failures and improving reliability. In addition, the design simplifies the cross-site high availability (HA) network, making maintenance easier.
- Simple networking

  The data replication, configuration synchronization, and heartbeat detection links share the same network, simplifying the networking. Either IP or Fibre Channel links can be used between storage systems, making it possible for HyperMetro to work on all-IP networks, improving cost-effectiveness.
- vStore-based HyperMetro

Traditional cross-site HA solutions typically deploy cluster nodes at two sites to implement cross-site HA. These solutions, however, have limited flexibility in resource configuration and distribution. HyperMetro can establish pair relationships between two vStores at different sites, implementing real-time mirroring of data and configurations. Each vStore pair has an independent arbitration result, providing true cross-site HA capabilities at the vStore level. HyperMetro also enables applications to run more efficiently at two sites, ensuring better load balancing. A vStore pair includes a primary vStore and a secondary vStore. If either of the storage systems in the HyperMetro solution fail or if the links connecting them go down, HyperMetro implements arbitration on a per vStore pair basis. Paired vStores are mutually redundant, maintaining service continuity in the event of a storage system failure.

Figure 5-10 vStore-based HyperMetro architecture



- Automatic recovery

  If site A breaks down, site B becomes the primary site. Once site A recovers, HyperMetro automatically initiates resynchronization. When resynchronization is complete, the HyperMetro pair returns to its normal state. If site B then breaks down, site A becomes the primary site again to maintain host services.

- Easy upgrade

  To use the HyperMetro feature, upgrade your storage system software to the latest version and purchase the required feature license. You can establish a HyperMetro solution between the upgraded storage system and another storage system, without the need for extra data migration. Users are free to include HyperMetro in initial configurations or add it later as required.

- FastWrite

  In a common SCSI write process, a write request goes back and forth twice between two data centers to complete two interactions, Write Alloc and Write Data. FastWrite optimizes the storage transmission protocol and reserves cache space on the destination array for receiving write requests, while Write Alloc is omitted and only one interaction is required. FastWrite halves the time required for data synchronization between two arrays, improving the overall performance of the HyperMetro solution.

- Self-adaptation to link quality

  If there are multiple links between two data centers, HyperMetro automatically implements load balancing among these links based on quality. The system dynamically monitors link quality and adjusts the load ratio between links to minimize the retransmission rate and improve network performance.

- Compatibility with other features

  HyperMetro can be used with SmartThin, SmartQoS, and SmartCache. HyperMetro can also work with HyperVault, HyperSnap, and HyperReplication to form a more complex and advanced data protection solution, such as the Disaster Recovery Data Center Solution (Geo-Redundant Mode), which uses HyperMetro and HyperReplication.

- Dual quorum servers

  HyperMetro supports dual quorum servers. If one quorum server fails, its services are seamlessly switched to the other, preventing a single point of failure (SPOF) and improving the reliability of the HyperMetro solution.

5.5 HyperVault

OceanStor 6800 V5 provides an all-in-one backup feature called HyperVault to implement file system data backup and recovery within or between storage systems. HyperVault can work in either of the following modes:

- Local backup

  Data backup within a storage system. HyperVault works with HyperSnap to periodically back up a file system, generate backup copies, and retain these copies based on user-configured policies. By default, five backup copies are retained for a file system.

- Remote backup

  Data backup between storage systems. HyperVault works with HyperReplication to periodically back up a file system. The process is as follows:

  1. A backup snapshot is created for the primary storage system.
  2. The incremental data between the backup snapshot and its previous snapshot is synchronized to the secondary storage system.
  3. After data is synchronized, a snapshot is created on the secondary storage system.

     By default, 35 snapshots can be retained on the backup storage system.

Technical Highlights

- High cost efficiency

  HyperVault can be seamlessly integrated into the primary storage system and provide data backup without additional backup software. Huawei-developed storage management software, OceanStor DeviceManager, allows you to configure flexible backup policies and efficiently perform data backup.

- Fast data backup

  HyperVault works with HyperSnap to achieve second-level local data backup. For remote backup, the system performs full backup the first time, and then only backs up incremental data blocks. This allows HyperVault to provide faster data backup than software that backs up data every time.

- Fast data recovery

  HyperVault uses snapshot rollback technology to implement local data recovery, without requiring additional data resolution. This allows it to achieve second-level data recovery. Remote recovery, which is incremental data recovery, can be used when local recovery cannot meet requirements. Each copy of backup data is a logically full backup of service data. The backup data is saved in its original format and can be accessed immediately.

- Simple management

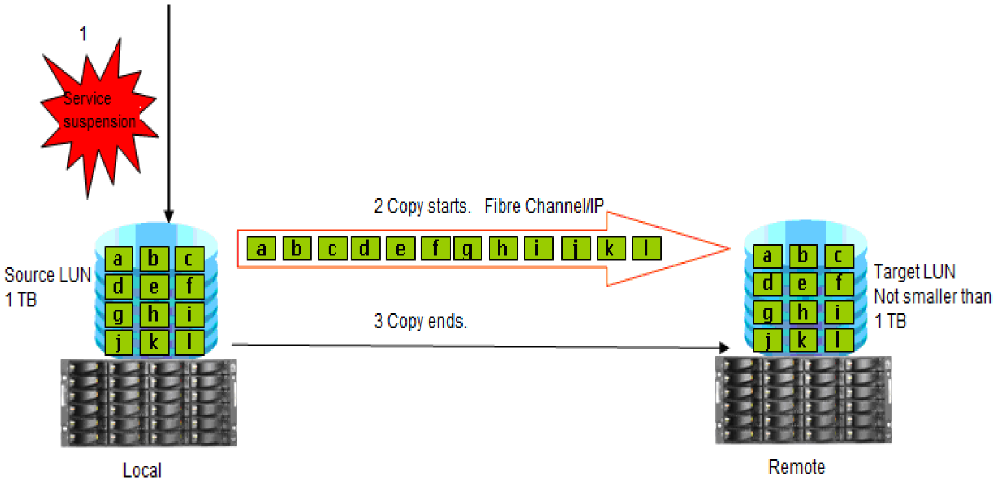
  Only one primary storage system, one backup storage system, and native management software, OceanStor DeviceManager, are required. This mode is simpler and easier to manage than old network designs, which contain primary storage, backup software, and backup media.

5.6 HyperCopy

OceanStor 6800 V5 uses HyperCopy to copy data from a source LUN to a target LUN within a storage system or between storage systems.

HyperCopy implements full copy, which copies all data from a source LUN to a target LUN. Figure 5-11 illustrates the working principle of HyperCopy.

Figure 5-11 HyperCopy working principle



1. A user suspends services to which HyerCopy is applied.

   This prevents services from being interrupted during full LUN copy.

2. A user triggers full LUN copy.

   Data can be copied to a target LUN over a Fibre Channel or IP link. The target LUN's capacity must be greater than or equal to that of the source LUN. Otherwise, data cannot be copied successfully.

   During copying, the progress is displayed.

OceanStor 6800 V5 can implement full LUN copy by reading snapshot volumes without interrupting services, ensuring a zero backup window.

Technical Highlights

- Multiple copy methods

  OceanStor 6800 V5 supports LUN copy within one storage system and between storage systems. Data can be copied from the local/target storage system to the target/local storage system. One-to-many LUN copy is provided to generate multiple copies for a source LUN.

- Dynamic adjustment of the copy speed

  HyperCopy allows a storage system to dynamically adjust the copy speed so that LUN copy does not affect production services. When a storage system detects that the service load is heavy, it dynamically lowers the LUN copy speed to make system resources available to services. When the service load is light, the storage system dynamically increases the copy speed, mitigating service conflicts in peak hours.

- Support for third-party storage systems

  LUN copy can be implemented within OceanStor storage systems or between OceanStor storage systems and Huawei-certified third-party storage systems. Table 5-1 outlines the storage systems supported by LUN copy.

Table 5-1 Storage systems supported by LUN copy

Storage System Where a Source LUN Resides	OceanStor Storage System Where a Target LUN Resides	Huawei-Certified Third-Party Storage System Where a Target LUN Resides
OceanStor storage system	Supported	Supported
Huawei-certified third-party storage system	Supported	N/A

- IP network-based LUN copy

  Regarding LUN copies between storage systems, most vendors in the industry support only Fibre Channel-based LUN copy. OceanStor 6800 V5 supports both Fibre Channel-based and IP network-based LUN copies. Customers can flexibly choose between these based on their site requirements. Furthermore, with the popularization of IP networks, IP network–based LUN copy features low costs, easy deployment, and simple maintenance.

5.7 HyperMirror

OceanStor 6800 V5 uses HyperMirror for volume mirroring.

HyperMirror creates two physical copies for a LUN. The space for each copy can come from either a local storage pool or an external LUN, with each copy having the same virtual storage capacity as its mirror LUN. When a server writes data to a mirror LUN, the storage system simultaneously writes the data to the LUN's copies. When a server reads data from a mirror LUN, the storage system reads data from one copy of the mirror LUN. Even if one mirror copy of a mirror LUN is temporarily unavailable (for example, when the storage system where the storage pool resides is unavailable), servers can still access the LUN. Then, the storage system records the LUN areas to which data has been written and synchronizes these areas after the mirror copy recovers.

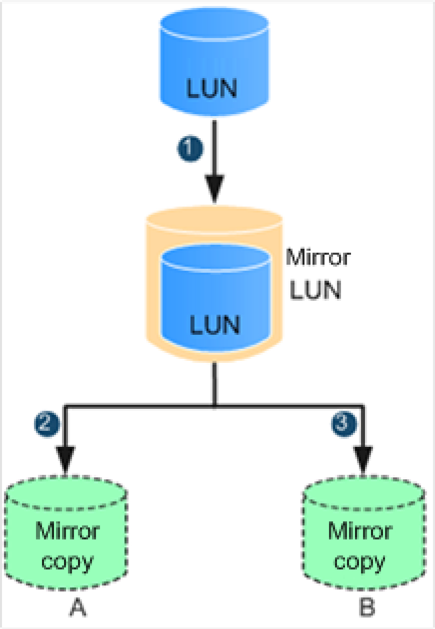
Working Principle

HyperMirror implementation involves mirror LUN creation, synchronization, and splitting.

Mirror LUN creation

Figure 5-12 shows the process for creating a mirror LUN.

Figure 5-12 Process for creating a mirror LUN



1. A user creates a mirror LUN for a local or external LUN. The mirror LUN has the same storage space, properties, and services as the source LUN. Host services are not interrupted during creation.
2. Local mirror copy A is automatically generated during mirror LUN creation. The storage space is swapped from the mirror LUN to mirror copy A. Mirror copy A synchronizes data from the mirror LUN.
3. A user creates mirror copy B for the mirror LUN, and mirror copy B copies data from mirror copy A. In doing so, the LUN with mirror copies A and B has the space mirroring function.

The following describes the process when a host sends an I/O request to the mirror LUN.

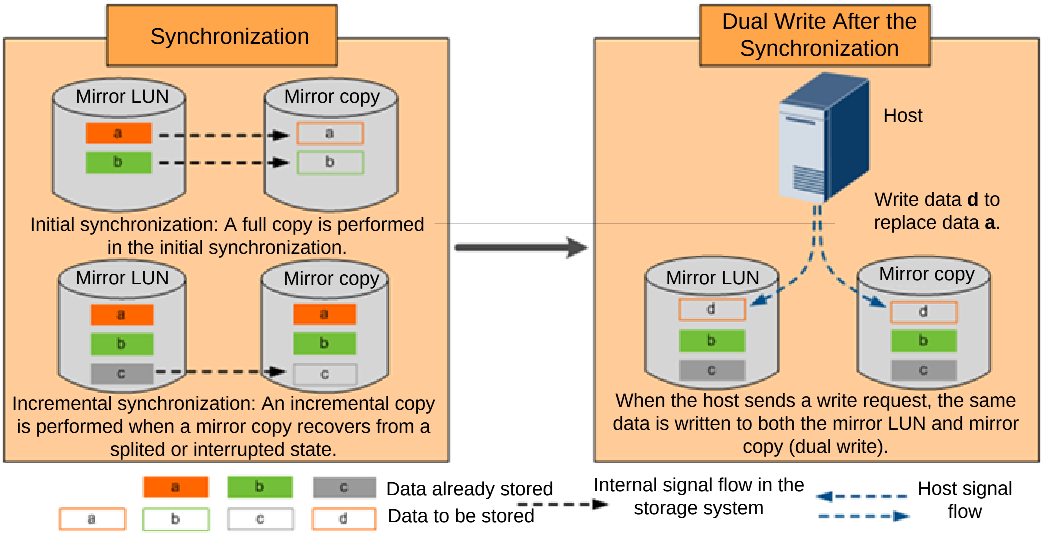
1 When a host sends a read request to the mirror LUN, the storage system reads data from the mirror LUN and its mirror copies in round-robin mode. If the mirror LUN or one mirror copy malfunctions, host services are not interrupted.

2 When a host sends a write request to the mirror LUN, the storage system writes data to the mirror LUN and its mirror copies in dual-write mode.

Synchronization

Figure 5-13 illustrates the synchronization process. When a mirror copy recovers from a fault or the data on a mirror copy becomes complete, that mirror copy copies incremental data from the other mirror copy. This ensures data consistency between the mirror copies.

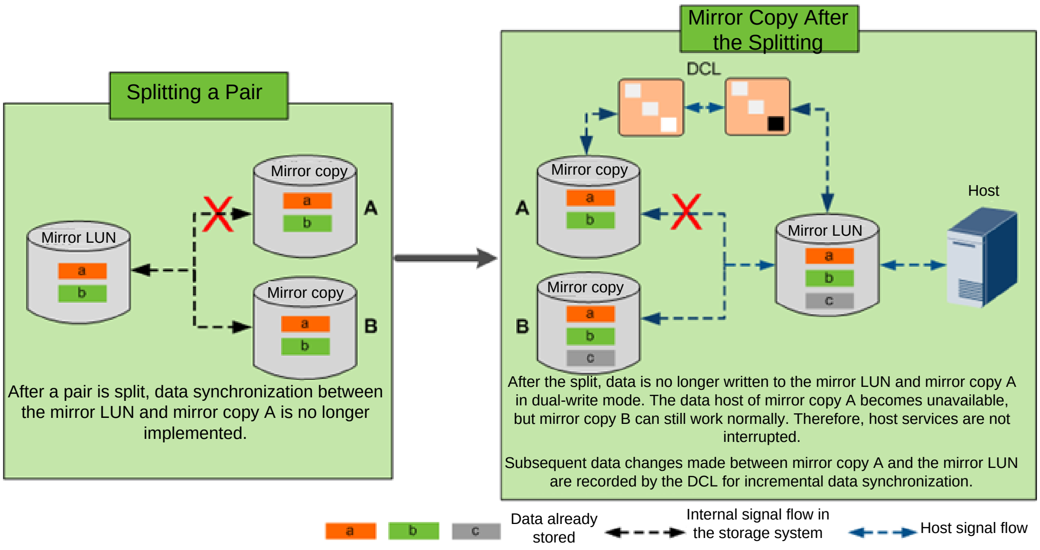
Figure 5-13 Synchronization process


Synchronization
Mirror LUN
Mirror copy
a
b
Initial synchronization: A full copy is performed in the initial synchronization.
Mirror LUN
Mirror copy
c
Incremental synchronization: An incremental copy is performed when a mirror copy recovers from a splited or interrupted state.
Dual Write After the Synchronization
Host
Write data **d** to replace data **a**.
Mirror LUN
Mirror copy
d
When the host sends a write request, the same data is written to both the mirror LUN and mirror copy (dual write).
Data already stored
Data to be stored
Internal signal flow in the storage system
Host signal flow

Splitting

A mirror copy can be split from its mirror LUN. After the mirror copy is split, the mirror LUN cannot perform mirroring on the mirror copy. Subsequent data changes made between the mirror copy and the mirror LUN are recorded by the DCL for incremental data synchronization when the mirroring relationship is restored.

Figure 5-14 Splitting


Splitting a Pair
Mirror LUN
Mirror copy
A
B
a
b
After a pair is split, data synchronization between the mirror LUN and mirror copy A is no longer implemented.
Mirror Copy After the Splitting
DCL
Host
Mirror LUN
Mirror copy
c
After the split, data is no longer written to the mirror LUN and mirror copy A in dual-write mode. The data host of mirror copy A becomes unavailable, but mirror copy B can still work normally. Therefore, host services are not interrupted.
Subsequent data changes made between mirror copy A and the mirror LUN are recorded by the DCL for incremental data synchronization.
Data already stored
Internal signal flow in the storage system
Host signal flow

Technical Highlights

- High data reliability within a storage system

  HyperMirror creates two independent mirror copies for a LUN. If one mirror copy malfunctions, host services are not interrupted, significantly improving data reliability.

- Robust data reliability of heterogeneous storage systems

  When SmartVirtualization is used to take over LUNs of heterogeneous storage systems, HyperMirror is employed to create a mirror LUN and local mirror copies for each heterogeneous LUN. Services will not be interrupted if heterogeneous storage systems are unstable or their links are down.

- Minor impact on host performance

  Mirror copies generated by HyperMirror reside in the cache of their LUN, and concurrent write and round-robin read technologies are implemented between mirror spaces. This ensures host service performance is not affected.

- Ensured host service continuity

  HyperMirror allows mirror copies to be created online for ongoing LUNs. In this way, host services are unaware of any changes made to LUN data space.

5.8 HyperLock

With the explosive growth of information, increasing importance has been pinned on secure access and application. To comply with laws and regulations, important data such as case documents from courts, medical records, and financial documents can be read but not written within a specific period. Therefore, measures must be taken to prevent such data from being tampered with. In the storage industry, Write Once Read Many (WORM) is the most common method used to archive and back up data, ensure secure data access, and prevent data tampering.

Huawei's WORM feature is called HyperLock. A file protected by WORM can enter the read-only state immediately after data is written to it. In the read-only state, the file can be read, but cannot be deleted, modified, or renamed. WORM can prevent data from being tampered with, meeting the data security requirements of enterprises and organizations.

File systems that WORM has been configured for are called WORM file systems and can only be configured by administrators. There are two WORM modes:

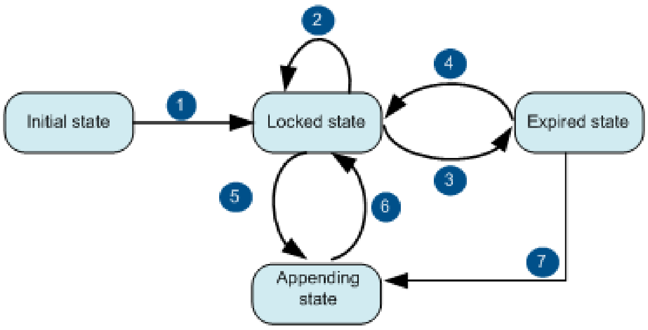
- Regulatory Compliance WORM (WORM-C for short): applies to archive scenarios where data protection mechanisms are implemented to comply with laws and regulations.
- Enterprise WORM (WORM-E): mainly used by enterprises for internal control.

Working Principle

With WORM, data can be written to files once only, and cannot be rewritten, modified, deleted, or renamed. If a common file system is protected by WORM, files in the WORM file system can be read only within the protection period. After WORM file systems are created, they must be mapped to application servers using the NFS or CIFS protocol.

WORM enables files in a WORM file system to be shifted between initial state, locked state, appending state, and expired state, preventing important data from being tampered with within a specified period. Figure 5-15 shows how a file shifts from one state to another.

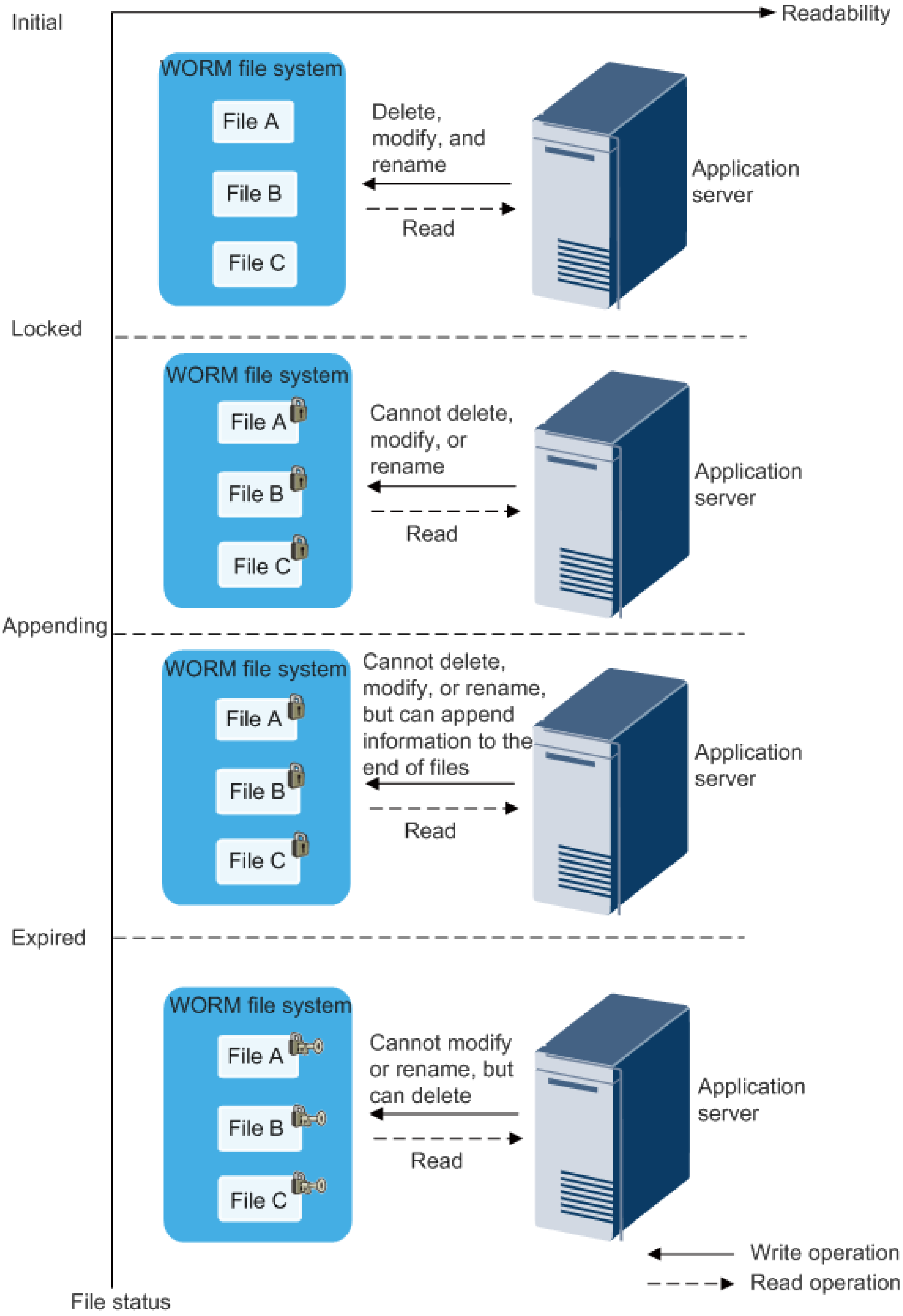
Figure 5-15 File state shifting



1. Initial to locked: A file can be shifted from the initial state to the locked state using the following methods:
   - If the automatic lock mode is enabled, the file automatically enters the locked state after a change is made and a specific period of time expires.
   - You can manually set the file to the locked state. Before locking the file, you can specify a protection period for the file or use the default protection period.
2. Locked to locked: In the locked state, you can manually extend the protection periods of files. Protection periods cannot be shortened.
3. Locked to expired: After the WORM file system compliance clock reaches the file overdue time, the file shifts from the locked state to the expired state.
4. Expired to locked: You can extend the protection period of a file to shift it from the expired state to the locked state.
5. Locked to appending: You can delete the read-only permission of a file to shift it from the locked state to the appending state.
6. Appending to locked: You can manually set a file in the appending state to the locked state to ensure that it cannot be modified.
7. Expired to appending: You can manually set a file in the expired state to the appending state.

You can save files to WORM file systems and set the WORM properties of the files to the locked state based on service requirements. Figure 5-16 shows the reads and writes of files in all states in a WORM file system.

Figure 5-16 Read and write of files in a WORM file system


Initial
Readability
WORM file system
File A
File B
File C
Delete, modify, and rename
Read
Application server
Locked
Cannot delete, modify, or rename
Appending
Cannot delete, modify, or rename, but can append information to the end of files
Expired
Cannot modify or rename, but can delete
Write operation
Read operation
File status

5.9 3DC

OceanStor 6800 V5 can be used in various SAN and NAS disaster recovery solutions.

SAN 3DC solutions are deployed on:

- Cascading/Parallel networks equipped with HyperMetro + HyperReplication/S
- Cascading/Parallel networks equipped with HyperMetro + HyperReplication/A
- Ring networks equipped with HyperMetro + HyperReplication/A
- Cascading/Parallel networks equipped with HyperReplication/S + HyperReplication/A
- Cascading/Parallel networks equipped with HyperReplication/A + HyperReplication/A
- Ring networks equipped with HyperReplication/S + HyperReplication/A

NAS 3DC solutions are deployed on:

- Cascading/Parallel networks equipped with HyperMetro + HyperReplication/A
- Cascading/Parallel networks equipped with HyperMetro + HyperVault
- Cascading/Parallel networks equipped with HyperReplication/A + HyperReplication/A

Two data centers equipped with HyperMetro or HyperReplication/S + HyperReplication/A can be flexibly expanded to three data centers without requiring external gateways.

For details about 3DC solutions supported by OceanStor 6800 V5, visit http://storage.huawei.com/en/index.html.

6 Best Practices

For best practices of OceanStor 6800 V5, visit http://storage.huawei.com/en/html/OceanStor_V5_en.html.

A Appendix

A.1 More Information

For more information about OceanStor 6800 V5, visit:

http://e.huawei.com/us/products/cloud-computing-dc/storage/massive-storage/6800-v5

For more information about Huawei storage, visit:

http://e.huawei.com/en/products/cloud-computing-dc/storage

For after-sales support, visit our technical support website:

http://support.huawei.com/enterprise/en

For pre-sales support, visit:

http://e.huawei.com/en/how-to-buy/contact-us

You can also contact your local Huawei office:

http://e.huawei.com/en/branch-office

A.2 Feedback

Huawei welcomes your suggestions for improving our documentation. If you have comments, please send your feedback to storagedoc@huawei.com.

We seriously consider all suggestions we receive and will strive to make necessary changes to the document in the next release.

A.3 Acronyms and Abbreviations

Table A-1 Acronyms and abbreviations

Acronym and Abbreviation	Full Spelling
AK	Authentication Key
BBU	Backup Battery Unit
CK	Chunk

Acronym and Abbreviation	Full Spelling
CKG	Chunk Group
CIFS	Common Internet File System
COW	Copy-On-Write
DCL	Data Change Log
DEK	Data Encryption Key
DIX	Data Integrity Extensions
DoD	Department of Defense
eDevLUN	External Device LUN
HA	High Availability
KMS	Key Manager Server
LRU	Least Recently Used
NDMP	Network Data Management Protocol
NFS	Network File System
ODX	Offload Data Transfer
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
RAID	Redundant Array of Independent Disks
RPO	Recovery Point Objective
ROW	Redirect-On-Write
SED	Self-Encrypting Drive
SPOF	Single Point of Failure
SRM	Site Recovery Manager
SCOM	System Center Operations Manager
SCVMM	System Center Virtual Machine Manager
TCO	Total Cost of Ownership
VAAI	VMware vStorage APIs for Array Integration
VASA	vStorage APIs for Software Awareness
WWN	World Wide Name
WORM	Write Once Read Many