百信 HengShan Stor 系列

产品技术白皮书

二〇二三年八月

TEL：0351-7030669　　FAX：0351--7030669　　网址：www.100trust.cn

商标声明


和其他百信商标均为百信信息技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受百信公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，百信公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

百信信息技术有限公司

地址： 山西综改示范区太原唐槐园区横河西二巷 5 号百信信创产业基地

邮编： 030000

网址： http://www.100trust.cn

目 录

1 概述

1.1 当前存储系统面临的挑战

随着云、互联网及智能技术的兴起与普及，企业新兴应用层出不穷，数据快速增长，业务加速创新对于存储能力的快速迭代提出了更高要求。例如，在金融行业，银行要抓住互联网、特别是移动互联网金融崛起带来的机遇，同时也不得不迎接由此带来的挑战：新业务小时级快速上线，以及更精准的用户需求分析等; 在智慧城市、智能制造、电信运营商等领域中，随着新兴业务的激增，业务数据呈现几何倍数的快速增长。企业业务的发展，使得企业数据中心存储系统开始面临新的挑战：

- 新建存储系统周期长与新兴业务快速上线间的矛盾；
- 系统庞大，管理复杂，运维人员压力巨大；
- 存储性能无法满足越来越多的数据并行处理应用需求；
- 客户需求分析、业务数据分析与决策推荐等需求，对大数据、云计算等新技术提出更高的诉求;

新的挑战必然催生新的需求。在行业新兴应用下，也许你理想中的存储系统应该是这样：

- 首先，它是敏捷的，资源可弹性部署、按需获取，支撑新兴业务快速上线；
- 面对数据中心结构化、非结构化等复杂的数据类型，能提供丰富的访问接入支持；
- 可快速海量扩展性能与容量，扩展方式要像堆积木一样简单；
- 提供极致性能，满足并行数据处理需求；
- 在满足那么多功能与性能需求的同时，还能降低 TCO。

百信 HengShan Stor 系列正是这样一款可大规模横向扩展、弹性伸缩的数据中心级智能分布式存储产品。HengShan Stor 系列通过系统软件将存储节点的本地存储资源组织起来构建分布式存储池，可向上层应用提供分布式文件存储服务、分布式对象存储服务、分布式大数据存储服务与分布式块存储服务，以及丰富的业务功能和增值特性; 其

独有的弹性 Erasure Coding（以下简称 EC）技术，能够在保证业务性能前提下为客户提供更多可用空间。产品支持用户根据业务需要灵活购买和部署存储服务，帮助企业轻松应对业务快速变化，实现数据灵活、高效、按需存取能力。

1.2 HengShan Stor 系列亮点

1.2.1 全对称分布式架构，提供极致弹性扩展能力

HengShan Stor 系列采用全对称分布式架构，支持通过横向扩展硬件节点线性增加整系统容量与性能，无需复杂的资源需求规划；系统可轻松扩展至数千节点及 EB 级容量，满足用户的业务规模增长需求。系统提供自动负载均衡策略，数据与元数据均匀分布于各节点，消除元数据访问瓶颈，保障规模扩展场景下的系统性能。系统通过智能分条聚合、IO 优先级智能调度等 FlashLink 技术，配合多级 Cache，大数据直通等系列关键技术，为用户提供高带宽，低时延的极致性能。无论您的数据中心在未来需要扩展 IO 密集型、时延敏感型、大带宽或大容量需求业务，HengShan Stor 系列提供的分布式存储都可以应需承载。

1.2.2 多服务按需使用，非结构化数据存储服务融合互通

HengShan Stor 系列支持文件、对象和大数据三种非结构化数据存储服务的融合互通，一份数据可以被所有非结构化服务共享访问。基于开放架构的 HengShan Stor 系列块服务，兼容容器及多种计算虚拟化平台，为各类云基础架构按需提供横向扩展的数据存储层，您在选择基础架构时无需担心厂商锁定，轻松构建开放的云平台。

- 非结构化数据存储服务（以下简称非结构化服务）：提供基于 NFS 协议、SMB 协议、标准 POSIX 接口和 MPI-IO 库的文件服务，满足 HPC 场景客户对文件系统高带宽，低时延的性能需求。提供兼容 Amazon S3 协议的对象服务与基于 HDFS 的大数据服务。多种非结构化服务之间支持协议互通、数据互访，免数据迁移省空间。
- 结构化数据存储服务（以下简称结构化服务）：提供 SCSI、iSCSI 等标准访问接口协议，支持广泛的虚拟化平台及数据库应用，提供高性能与高扩展能力，满足虚拟化、云资源池及数据库等场景的 SAN 存储需求。

1.2.3 超高密设计，提供极致性能和容量密度

HengShan Stor 系列高密硬件，通过极致密度设计，帮助客户节省机房空间。超大容量设计，帮助客户以更少的设备存储更多的数据；极致的性能密度设计，帮助客户在全闪存趋势下快人一步。

HengShan Stor P9550K 高容量硬件面向极致容量密度设计，采用高内聚结构，实现了盘片密度最大化，同时双层风道+对旋增压风扇设计解决了高密度下的散热问题，是业界首款满足 1.1m 机柜的超高密整机。

HengShan Stor P9950K 高性能硬件面向全闪存极致性能密度设计，全系统采用 PCIe 4.0，是业界最先推出的全 PCIe 4.0 闪存系统；采用小节点设计，配合 Half-Palm NVMe SSD，为客户提供极致性能体验。

2 硬件架构

2.1 硬件介绍

图2-1 百信 HengShan Stor 系列存储系统

HengShan Stor 系列提供提供端到端整体方案兼容性验证，覆盖各类异常场景。推荐典型配置存储设备如下：

表2-1 HengShan Stor 系列典型配置存储设备

分类	节点型号	配置	适用服务			
			块服务	文件服务	对象服务	大数据服务
性能型	HengShan Stor P9950K	5U 80 盘位	-	√	√	√
	HengShan Stor P9920K	2U 12 盘位	√	√	√	√
		2U 24 盘位 /25 盘位	√	-	-	-
均衡型	HengShan Stor P9540K	4U 36 盘位	√	√	√	√

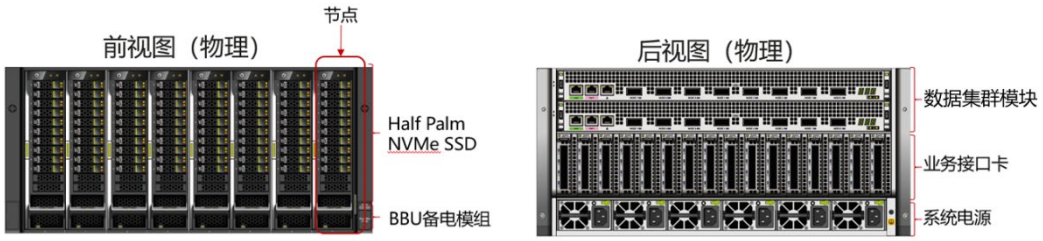
	HengShan Stor P9520K	2U 12 盘位	√	√	√	√
	HengShan Stor P9546K	4U 60 盘位	-	√	√	√
	HengShan Stor P9550K	5U 120 盘位	-	√	√	√
视频型	HengShan Stor P9350K	5U 120 盘位		√	√	
	HengShan Stor P9346K	4U 60 盘位		√	√	
	HengShan Stor P9340K	4U 36 盘位		√	√	
归档型	HengShan Stor P9146K	4U 60 盘位	-	√	√	-
	HengShan Stor P9150K	5U 120 盘位	-	√	√	-

2.1.1 HengShan Stor P9950K 高性能硬件介绍

HengShan Stor P9950K 是 5U 高密全闪型整机，采用高密多节点的设计理念，为客户提供极致存储性能。

- HengShan Stor P9950K 每机箱支持 8 个独立的存储节点，整机支持 80 个 Half-Palm NVMe SSD 盘位，支持 16 张 PCIe 4.0 的业务接口卡，16 个 100GE 数据集群接口。
- HengShan Stor P9950K 每节点 8 个 DDR4 的内存通道，节点内全 PCIe 4.0 设计，每个节点支持 10 个 Half-Palm NVMe SSD。

图2-2 百信 HengShan Stor P9950K

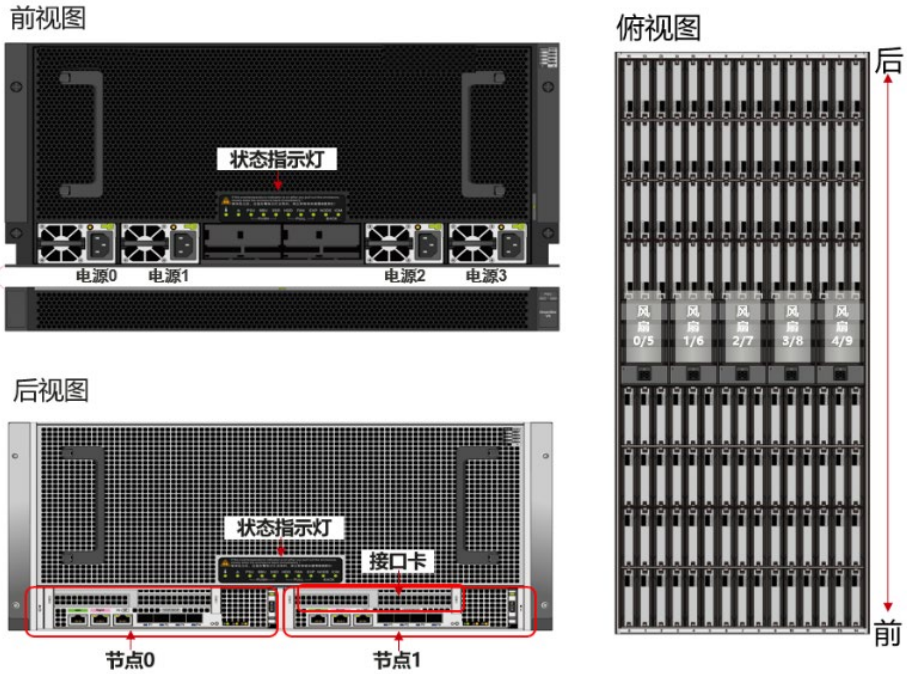


2.1.2 HengShan Stor P9550K 高容量硬件介绍

HengShan Stor P9550K 是 5U 高密大容量整机，采用分布式两节点专有硬件设计，为客户提供极致的可靠性和硬盘密度。

- HengShan Stor P9550K 每机箱 2 节点，整机 120 盘位，提供 24 盘/U 的业界最高密度。
- HengShan Stor P9550K 每节点提供 20 瓦/盘的高能效接入能力，内置滑轨和坦克链设计，全 FRU 设计。

图2-3 百信 HengShan Stor P9550K



2.1.3 HengShan Stor P9546K 高容量硬件介绍

HengShan Stor P9546K 是基于 Kunpeng 920 处理器开发的 4U 高密整机，为客户提供超强可靠性和硬盘密度。

- HengShan Stor P9546K 每机箱 60 盘位，内置滑轨和跟线架，全 FRU 设计，支持使用 1000 mm 及以上深度的机柜。
- HengShan Stor P9546K 有 1 机箱 1 节点和 1 机箱 2 节点两种配置。
- HengShan Stor P9546K 的 1 机箱 1 节点配置，每节点采用两颗 48 核鲲鹏处理器，16 个 DDR4 的内存通道，1 机箱 2 节点配置，每节点采用一颗或两颗 48 核鲲鹏处理器，8 个或 16 个 DDR4 的内存通道，节点内为全 PCIe 4.0 设计；

图2-4 HengShan Stor P9546K 前面板

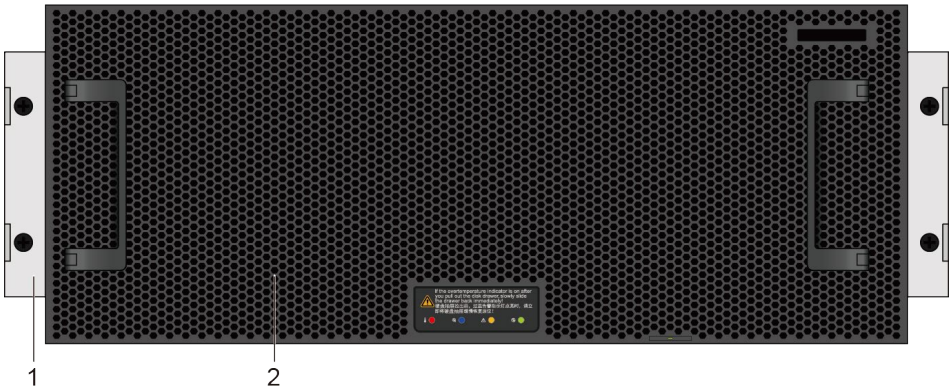


图2-5 HengShan Stor P9546K（1 机箱 1 节点）后面板

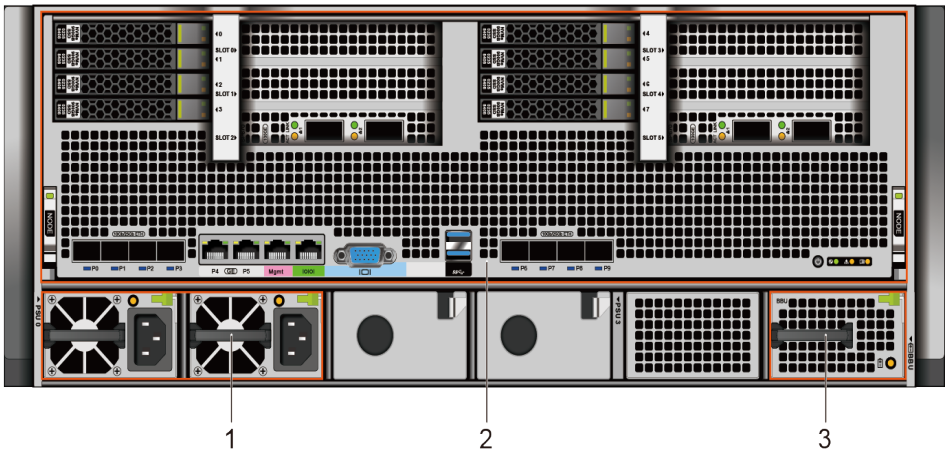
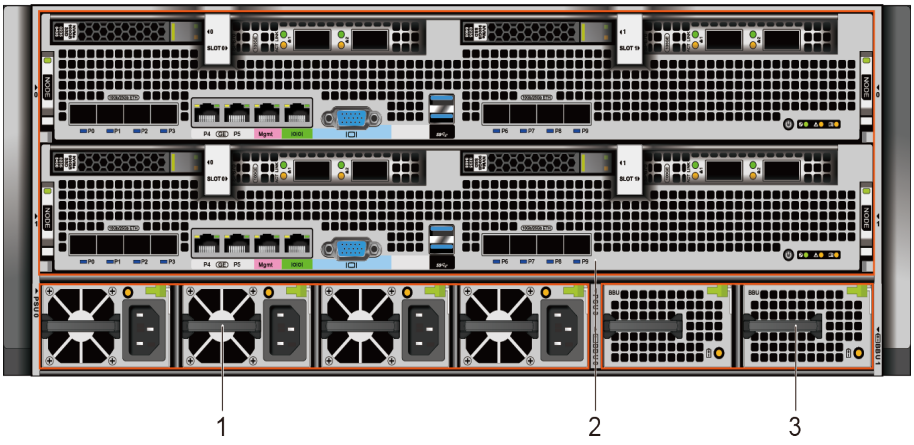


图2-6 HengShan Stor P9546K（1 机箱 2 节点）后面板

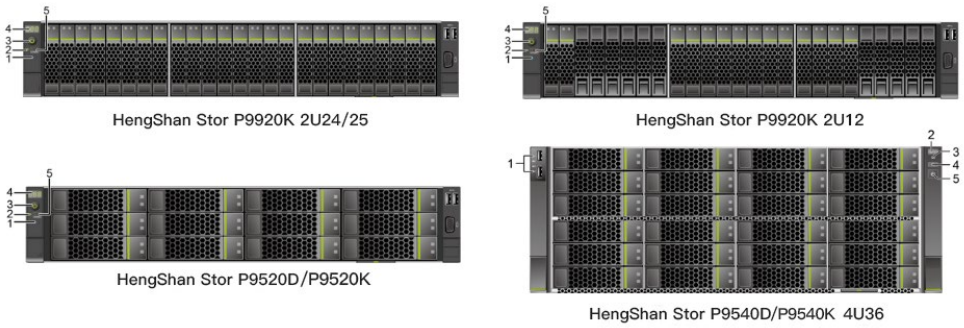


2.1.4 HengShan Stor P9920K/HengShan Stor P9520K/HengShan Stor P9540K 通用硬件介绍

HengShan Stor 系列还支持基于 Arm/x86 处理器的存储产品：

- HengShan Stor P9920K 是 2U 12/24/25 盘位的全闪产品 (2U 24 盘位只支持 x86)，采用 SAS SSD 或 NVMe SSD 主存介质
- HengShan Stor P9520K 是 2U 12/25 盘位的硬件
- HengShan Stor P9540K 是 4U 36 盘位的硬件。

图2-7 百信 HengShan Stor P9920K/P9520K/P9540K



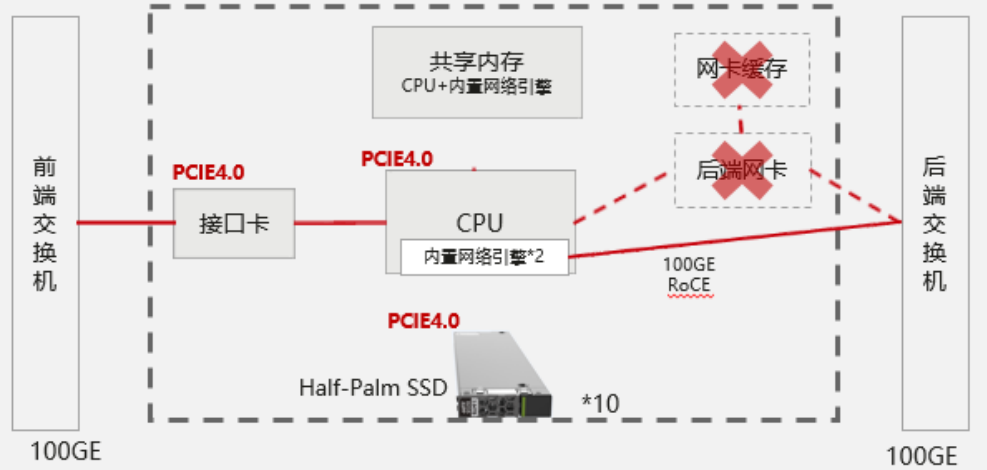
2.2 高密高性能硬件设计

HengShan Stor P9950K 面向全闪存极致性能设计，采用小节点，单机箱 5U 8 节点共 80 盘，单台整机可以实现 160GB/s 的极致单机性能。

2.2.1 前后端 PCIe 4.0

全系统采用 PCIe 4.0 设计，业界最先推出的全 PCIe 4.0 全闪存系统，PCIe 4.0 相比现有的 PCIe 3.0 带宽提升一倍，具备更大带宽，更低时延的优势。

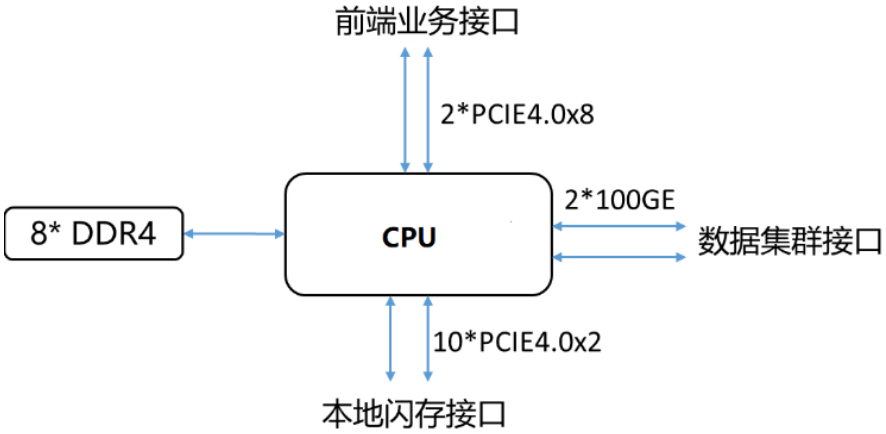
图2-8 百信 HengShan Stor P9950K 前后端 PCIe 4.0



CPU 的前端业务接口带宽分配的 2 个 PCIE4.0X8 的接口，提供的总带宽能力为 20GB/s。HengShan Stor P9950K 采用免后端网卡设计，最大化地发挥了 CPU 的接入能力，节省了成本，同时后端数据集群接口采用 CPU 直出的 2 个 100GE 接口端口，该接口支持 RDMA 技术，极大地降低了网络时延，可以有效的将集群的数据，高效的传输到其他节点上，同时只消耗少量的 CPU。本地闪存接口采用 10 个 PCIE4.0X2 的接口，使用的 SSD 接口协议是 NVMe，使得 CPU 可以直接通过 PCIe 直接访问介质，相比传统的 SAS 协议，每次 I/O 时延可以降低 50%以上。

每个节点里面都采用了最优的带宽配比，可以实现最优的带宽分配关系，前端：集群：介质=1:1:1.25 的带宽能力，充分发挥 CPU 的带宽能力。

图2-9 百信 HengShan Stor P9950K 通道带宽配比

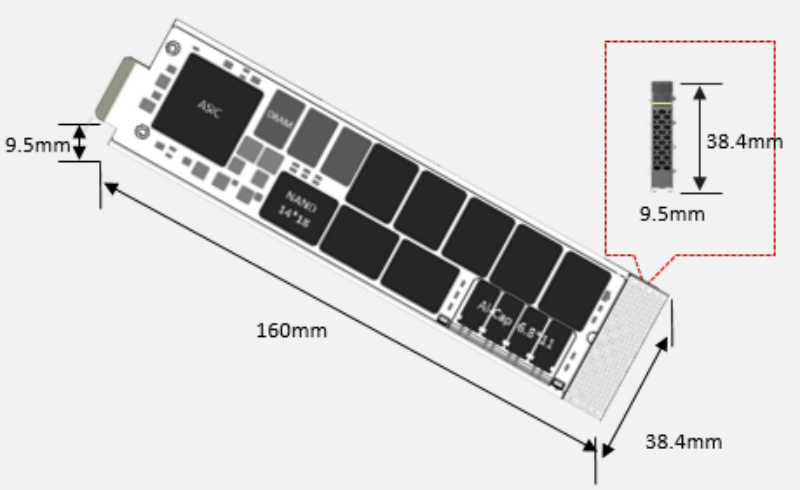


2.2.2 Half-Palm NVMe SSD

HengShan Stor P9950K 采用 NVMe SSD 盘，整机支持 80 个 Half-Palm NVMe SSD，该介质形态是专为 Flash 介质进行设计，容量可达 7.68 TB，较传统 SSD 盘更为轻薄，空

间占用更少性能更优。较传统的 NVMe SSD 盘，该形态可以为每个机架放置更多 SSD，从而在相同容量下释放更多机架空间，节省机房成本。

图2-10 Half Palm NVMe SSD



为了发挥 SSD 的性能，HengShan Stor P9950K 采用的鲲鹏多核处理器提供超强算力，通过优化分支预测算法、提升运算单元数量、改进内存子系统架构等一系列微架构设计，大幅提高了处理器性能，为产品化 CPU 分组分核算法优化系统时延奠定了硬件基础；鲲鹏处理器集成了南桥、网卡、SAS 存储节点三种芯片，做到集成度业界领先，单颗芯片实现 4 颗芯片的功能，同时支持动态调频功能，降低了客户 TCO。

2.2.3 高效散热设计

Flash 介质的功耗随着性能的增加而增加，怎么解决散热问题也是一个难点。在 HengShan Stor P9950K 整机中，总共使用 32 个直径为 40mm 的对旋风扇（对旋风扇位于节点内，随节点拔出后可独立更换），能够提供足够的风量将系统中 Half-Palm NVMe SSD、CPU、内存和接口卡的热量高效的带到空气中，再由机房的空调将这部分热量带出机房外。同时整机采用平行背版设计，平行背板是用于连接 SSD 与 CPU 之间的信号，平行背板的优势在于 SSD 盘与主板之间连接时，在风的流向方向上，没有竖直背板的阻挡，同时背板通过极致的开窗设计，风阻达到最小，能够有效提升给 SSD 盘散热的风量，降低 SSD 盘温度。通过对旋风扇后，使得硬盘在常温时的工作温度低于 35℃，为硬盘营造一个非常舒适的工作温度。

图2-11 Half Palm NVMe SSD 散热外壳

Half Palm NVMe SSD 盘的尺寸为 160mm x 38.4mm x 9.5mm，采用全金属外壳设计，外壳材质采用全压铸铝，导热系数可达 200 W/mK 以上，内部颗粒和芯片能够直接与外壳接触，能够有效的将硬盘内部的热量传导出来，相比其他材质可降低 Flash 颗粒的工作温度 5℃。为了保证处理器的可靠工作，采用了 VC 相变散热器+碳纤维导热垫，有效的保证了 CPU 的工作结温低于 85℃，CPU 安全可靠工作的同时，又可以充分发挥出 CPU 的性能。

2.3 高密大容量硬件设计

普通分布式存储节点在 4U 的空间里，通常最多能放置 36 块 3.5 寸硬盘。而 HengShan Stor 系列在 5U 的空间里，能够放置 120 块 3.5 寸硬盘，1 台设备相当于 3 台 36 盘通用节点，存储密度是其 2.67 倍。存储密度提升带来的最直接价值——节省空间，相对于 36 盘节点，同样的硬盘数，空间占用下降 62.5%。

高密硬件设计需要解决三大难题：

- 保证硬盘的散热效率，这就要求硬盘之间有足够的间距；
- 降低盘框的空间损耗，将更多的空间留给硬盘；
- 所有盘片可以独立维护，这要求将整个盘框能够抽出机柜。为了支撑重 100 kg 以上的硬盘框抽拉，防止机柜倾斜，业界通常采用抱轨设计，侵占了硬盘装配空间。

  HengShan Stor P9550K 创新性地采用双向双层抽屉+坦克链、内缩设计，不仅实现了盘片密度最大化，解决了极致密度要求和操作维护空间矛盾，同时也解决了盘片散热问题和盘片在线独立维护问题。

2.3.1 高内聚结构设计

双向双层抽屉解决了上层盘维护和下层节点、电源 FRU 固定部件分离的问题，通过可活动的坦克链连接这两层。HengShan Stor P9550K 整机节点与电源在下 1U 空间内采用内缩设计，节省了电源和节点出线空间，成为业界首款满足 1.1 米（深度）通用 IT 机柜使用的超高密整机。

表2-2 百信 HengShan Stor P9550K 双向双层抽屉示意

所有硬盘并不需要打开机框上盖进行维护，通过抽屉式抽拉，将硬盘从机框里拉出进行维护，可以从机框前面拉出 60 块硬盘，也可以从机框后面拉出 60 块硬盘维护。“坦克链”设计，使硬盘在操作维护中 120 块盘业务不中断得以实现，硬盘在机框上 4U 箱体里抽拉时，高速信号线缆一端与上 4U 硬盘箱体连接，一端与节点背板连接，所有线缆束缚在“坦克链”里，跟随机框前后移动。

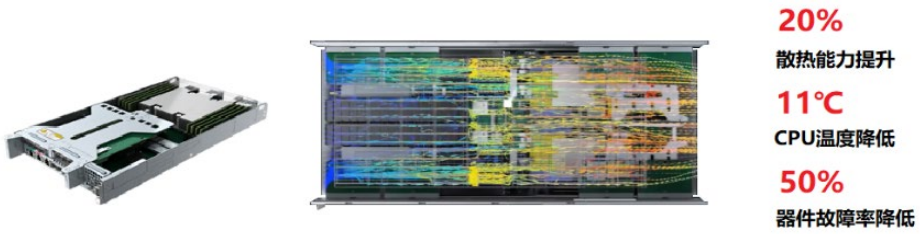
图2-12 百信 HengShan Stor P9550K 框内结构

双向抽拉保证盘框抽拉时，重心始终控制在机框内，这样就不再需要抱轨，大幅压缩了滑动轨道的宽度，将更多的空间留给硬盘和硬盘间的散热风道。使磁盘间距提升了

50%，散热能力提升了 30%，从而支撑 120 块硬盘稳定工作。由于机框的重心在框内，重心偏移降低了 75%，确保了设备在机柜高处安装时的安全性。

2.3.2 高效散热设计

图2-13 百信 HengShan Stor P9550K 散热技术



在解决了硬盘密度问题后，下 1U 空间内两节点中的 CPU、内存等高发热器件也不能忽视，如果不能有效解决散热问题，将面临超高的故障率，甚至宕机。为确保 CPU 的散热，HengShan Stor P9550K 使用了碳纤维导热膜+真空 VC 散热器技术。碳纤维导热膜具有超高的导热系数，质量轻且耐腐蚀，非常适合密度高、尺寸精度要求高的设备。VC 真空均热板，是将多处热源所散发的热流，在短距离内，均匀分布到较大的散热面积，相对于铜导管的线性热传导，具有更高的散热率。在这个配合下，整体升了 20%的散热能力，相对于传统散热技术，CPU 核心温度降低了 11 度。

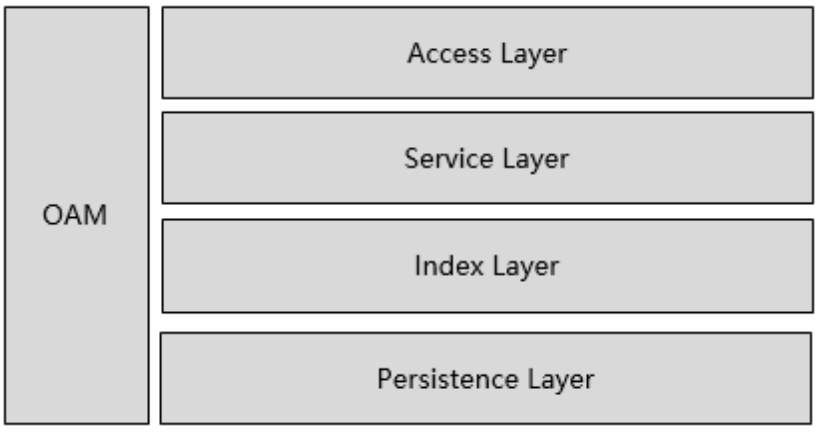
此外，HengShan Stor P9550K 采用了风扇中置结合上下两层独立风道设计。风扇中置，对于前部的 60 块盘是抽风，后部的 60 块盘是吹风，最大化地利用了轴流风扇的能力。节点、电源、电池、SSD 缓存位于机箱下部，形成上下 2 层独立风道，避免硬盘与 CPU、内存等高温环境下相互影响。相对于采用通用服务器的散热技术，器件失效率降低 50%。

3 软件架构

3.1 全对称分布式软件架构

3.1.1 软件逻辑架构

图3-1 HengShan Stor 系列产品软件逻辑架构

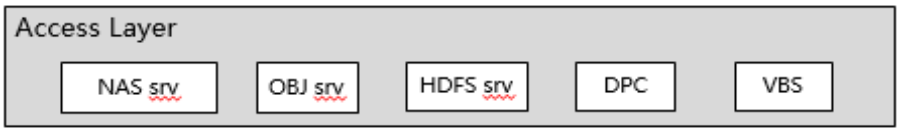


HengShan Stor 系列是一款可大规模横向扩展的智能分布式存储产品，可提供块服务、文件服务、对象服务和大数据服务，用户可以按需部署相应的存储服务。如上图所示，OAM(Operation Administrator and Maintenance)提供 HengShan Stor 系列产品的管控面服务，包括资源管理、业务管理、系统管理、用户管理、安装部署、升级、扩容/缩

容、巡检/信息收集等。IO 面在逻辑架构上从上往下分为：Access Layer，Service Layer，Index Layer 和 Persistence Layer。

3.1.1.1 Access Layer

图3-2 Access Layer 架构功能元素



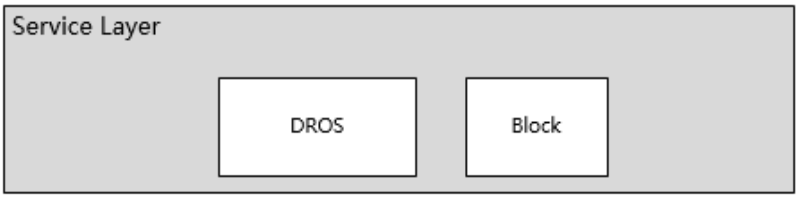
从功能层次上，Access Layer 负责接收并处理各种服务的访问，对于非结构化服务，主要有以下功能模块：

- NAS srv 模块负责提供标准的 NAS 协议(NFS/SMB)接入处理；
- OBJ srv 模块负责提供标准的 S3 对象访问接入处理；
- HDFS srv 模块负责提供标准的 HDFS 协议接入访问处理；
- DPC 模块是针对 HPC 场景设计的兼容标准 POSIX/MPI-IO 语义访问。

对于结构化服务，VBS(Virtual Block Service)提供标准的 iSCSI 协议访问，以及为高性能场景提供的本地 SCSI 语义访问。

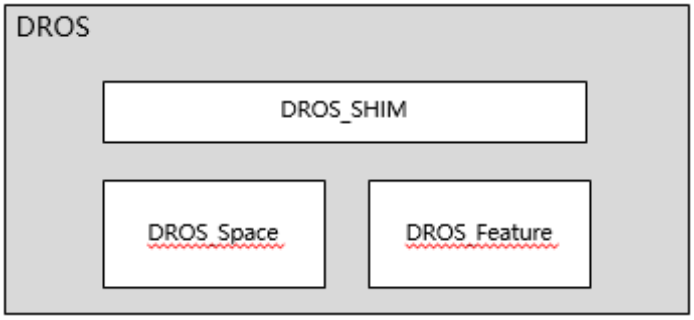
3.1.1.2 Service Layer

图3-3 Service Layer 架构功能元素图



Service Layer 负责提供非结构化服务与结构化服务功能，所有非结化服务处理统一到 DROS(Distributed Relational Object Service)平台，共享一份元数据，一份数据，共享增值特性。结构化服务由单独的 Block 模块进行处理。

图3-4 DROS 架构功能元素图



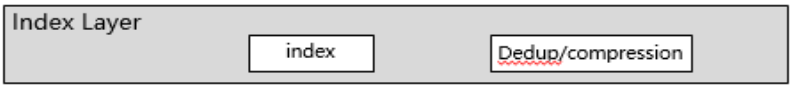
在 DROS 模块内，DROS_SHIM 是语义抽象层，针对文件/大数据/对象三种非结构化服务语义进行统一抽象，上层不同的非结构化语义接入后会把相应的语义操作映射到 DROS_SHIM 提供的功能接口上。

DROS_Space 提供基础的非结构化服务的元数据处理，确保所有非结构化服务共一份元数据。

DROS_Feature 为非结构化服务提供统一的增值服务，如配额，分级，快照等。

3.1.1.3 Index Layer

图3-5 Index Layer 架构功能元素



Index Layer 在 Persistence Layer 之上提供细粒度的空间布局管理能力。

Index 模块负责将 DROS 或 block 中的数据对象映射到下层的 Plog 空间上。对于非结构化服务提供了基于保电内存的缓存能力。

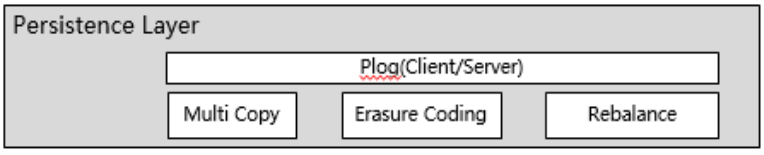
Dedup/compression 模块为结构化服务提供重删压缩能力节省成本。

说明

PLOG（Persistence LOG）是 Persistence Layer 提供的一组按照固定大小管理的物理地址的集合，提供 EC 或多副本的冗余保护能力，只能追加写，不能覆盖写，通过 PlogID+Offset 的方式来访问具体的物理磁盘上的 LBA 地址。

3.1.1.4 Persistence Layer

图3-6 Persistence Layer 架构功能元素



Persistence Layer 负责 HDD、SSD 等存储介质的空间分配与故障管理，通过跨节点的 EC 或副本提供可靠的、分布式化的数据管理能力。Persistence Layer 聚焦于可靠性、性能、扩展性等存储底层相关的能力。向上提供基于 PLOG（persistence log）语义的空间访问能力。

3.1.1.5 微服务形态

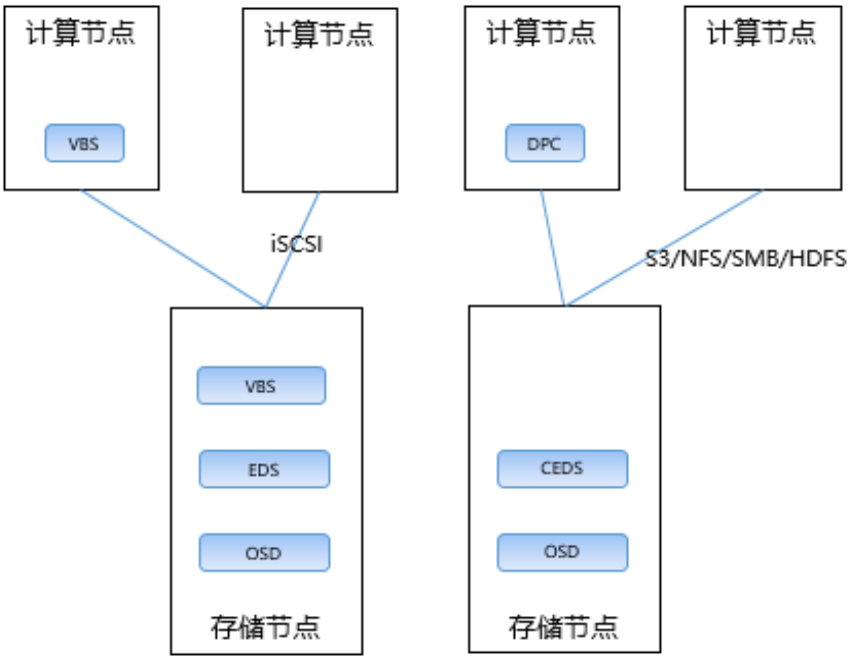
以上介绍了 HengShan Stor 产品的软件逻辑功能架构，整个产品软件是微服务形态发布，以上的软件架构元素，会构建在对应的微服务包内，关键的微服务组件包括：

表3-1 微服务组件列表

微服务包名称	功能描述	对应的逻辑功能架构元素
DPC(Distributed parallel client)	并行客户端，部署在计算节点，提供标准 posix 与 MPI-IO 访问。	DPC、plog client
VBS(Virtual Block Service)	块客户端，可以提供标准的 iscsi/scsi 访问，可以部署在计算节点，也可以部署在存储节点。	VBS
EDS(Enterprise Data Service)	企业数据服务，提供结构化服务及相应的增值特性。	Block、index layer、plog client
CEDS(Converged Enterprise Data Service)	融合企业数据服务，提供融合的非结构化服务。	NAS srv、OBJ srv、HDFS srv、DROS、index layer、plog client
OSD(Object Storage Device)	提供数据持久化能力。	persistent layer

这些关键组件微服务组件部署示意如下图所示。

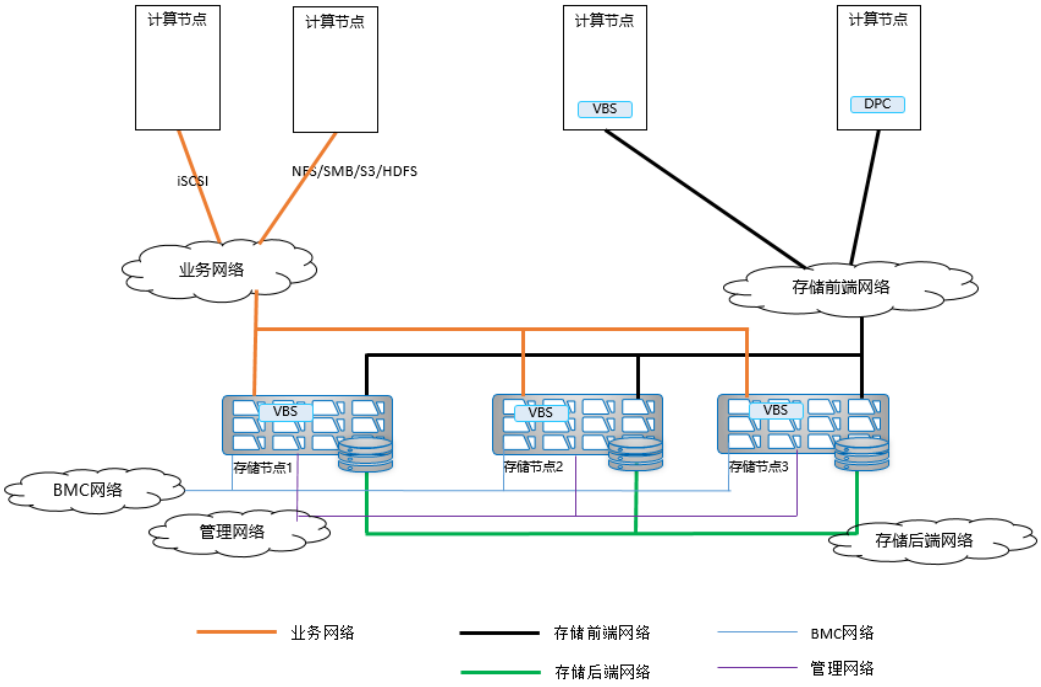
图3-7 HengShan Stor 系列关键微服务组件部署示意图



VBS 微服务组件可以部署在存储节点上对外提供 iSCSI 访问。VBS 微服务组件也可以部署在计算节点上提供更高性能访问。DPC 微服务组件部署在计算节点，为 HPC 场景提供更高性能。

3.1.2 网络逻辑架构

图3-8 HengShan Stor 系列逻辑网络设计示意图



从用户应用访问接入到存储节点 scale-out 扩展，考虑到维护管理，性能，可靠性等因素，HengShan Stor 系列区分定义了网络类型，分为业务网络，存储前端网络，存储后端网络，管理网络，BMC 网络。

表3-2 逻辑网络功能描述

名称	描述	网络类型	业务场景描述
存储业务网络	用于标准协议客户端访问接入	1. TCP/IP 协议 2. IB 协议 3. ROCE 协议	1. 计算节点通过 NFS/SMB/S3/HDFS/FTP/FTPS 协议访问。 2. NFS over RDMA 使用 IB 或 ROCE 网络。 3. 计算节点不部署 VBS 场景下，直接使用 iscsi 访问。
存储前端网络	在计算节点部署 DPC/VBS 时，负责 DPC/VBS 与存储节点间数据通信。	1. TCP/IP 协议 2. IB 协议 3. ROCE 协议	1. 对于性能要求不高场景，可以考虑存储前端网络与存储后端网络合并。标准协议接入场景推荐合并。 2. 文件 DPC 接入场景，建议存

名称	描述	网络类型	业务场景描述
存储后端网络	存储节点间的数据通信网络，用于重构/均衡等后台流量，确保主机 IO 性能不受后台 IO 影响.支持 RDMA	1. TCP/IP 协议 2. IB 协议 3. ROCE 协议	储前端网络与存储后端网络分离部署，使用 RDMA 提升性能。 3. NFS over RDMA 服务在 100GB 存储组网形态下，推荐与存储网络合部。小于 100GB 存储组网则推荐使用单独的存储业务网络。
管理网络	用于存储系统管理和维护	TCP/IP 协议	GE 网口，单网口或者双网口，双网口支持 bond1、bond2 和 bond4 模式，推荐配置成 bond1
BMC 网络	提供远程带外硬件设备管理功能	TCP/IP 协议	GE 网口，远程硬件设备管理

3.2 全局负载均衡

HengShan Stor 系列全对称分布式软件架构分两层路由，一层在 Service layer，确保每个节点都能分担服务处理能力，另一层在 Persistence layer，尽可能的均衡存储空间同时加快故障重构速度。

Service layer 非结构化服务与结构化服务由于语义的差异有不同的路由打散策略，但是都是基于同一个基础路由模型进行映射的。Persistence layer 的路由打散策略是基于 STORE DHT。下面的章节从前端负载均衡与后端负载均衡两个维度进行介绍说明。

3.2.1 前端负载均衡

HengShan Stor 系列非结构化服务与结构化服务前端负载均衡都是基于同一个基础路由模型，下面会先介绍基础路由模型，然后再针对差异化部分分别介绍，对于非结构化服务，由于多协议融合互通，前端访问模型进行了抽象归一（见 3.5 非结构化数据存储服务融合互通章节），这里主要针对文件语义进行介绍。

3.2.1.1 基础路由模型

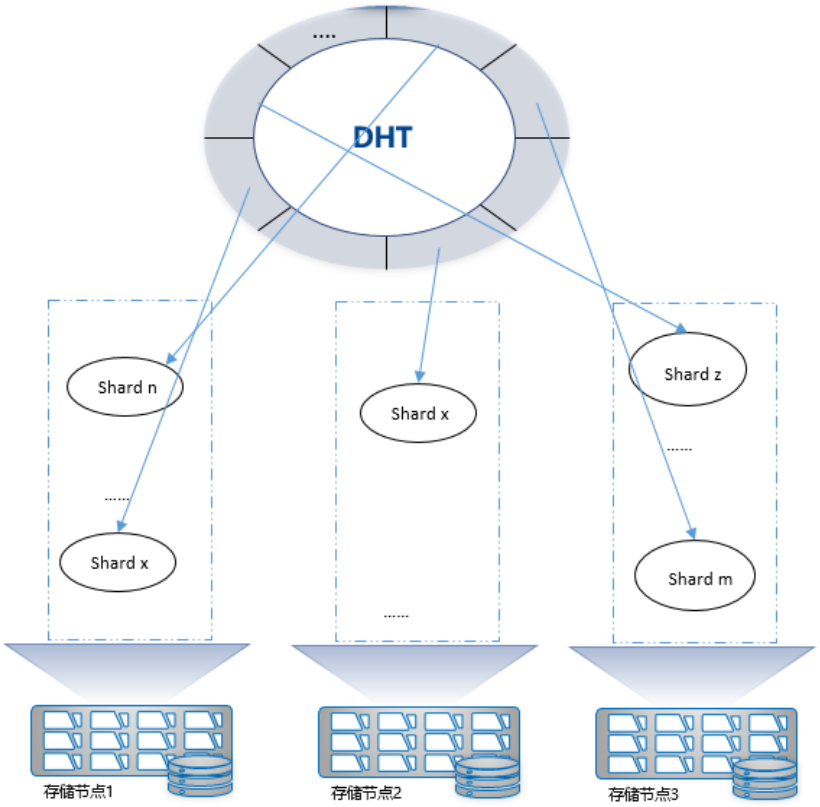
HengShan Stor 系列存储采用 Shard DHT 算法作为 Service layer 的基础路由策略，每一个资源池拥有一个 Shard DHT 哈希环，每一个哈希环有 4096 个 Shard，对于每个存储节点 Node 能够处理的 Shard 数量，为资源池成员存储节点数量均分 4096，比如资源池由 16 个节点组成，那么就由 16 个节点均分 4096，即每个存储节点处理 4096/16=256 个 Shard。

HengShan Stor 系列采用的 Shard DHT 算法具有以下特点：

- 均衡性：数据能够尽可能分布到所有的节点中，这样可以使得所有节点负载均衡。

- 单调性：当有新节点加入系统中，系统会重新做数据分配，以 Shard 为单位迁移粒度，将 SHARD 处理单元对应的数据服务由新节点接管，数据迁移新增节点上处理，现有节点上的数据不需要做很大调整。

图3-9 Shard DHT 路由示意图

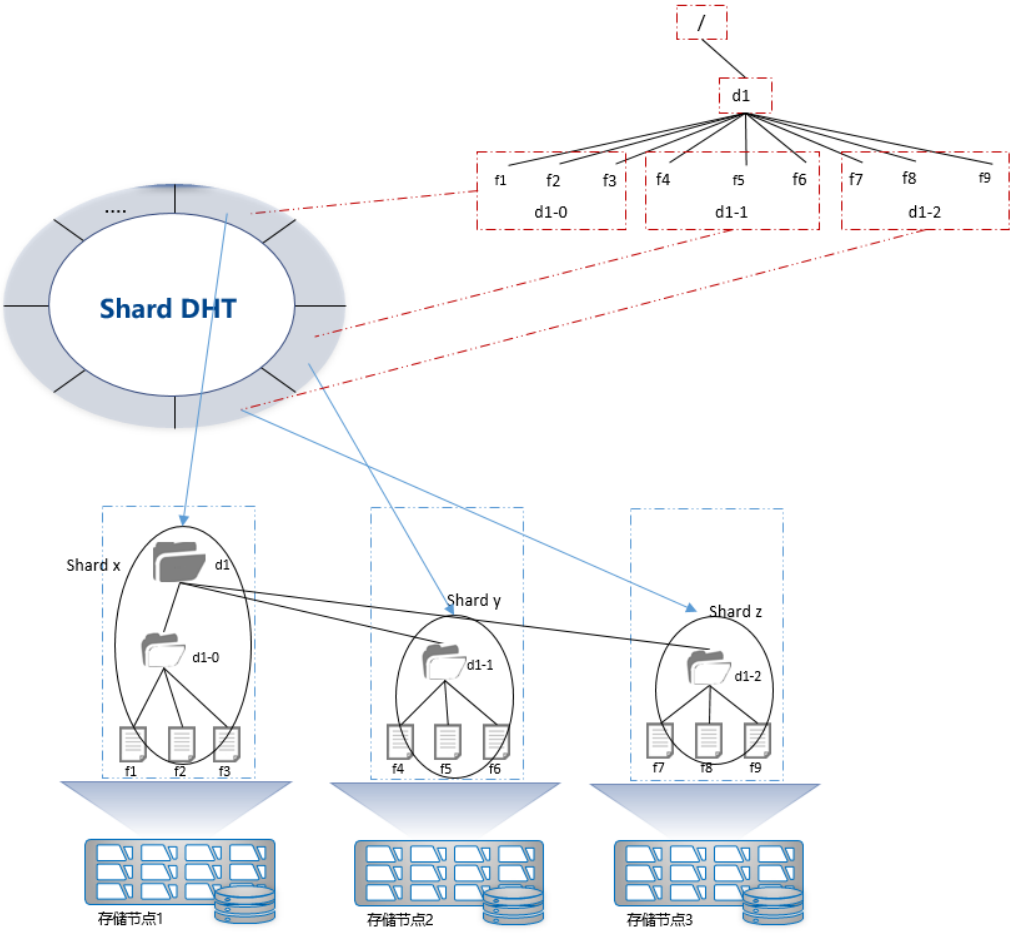


3.2.1.2 文件语义路由映射

HengShan Stor 系列对象会被映射成文件，对象的桶会被映射成根目录，非结构化服务共一份元数据管理，对象、大数据与文件服务使用相同的打散策略。具体来说，为了解决大目录文件过多的性能问题，采用了基于目录分片的分布式打散策略，达到负载均衡的目的。目录分片与目录分片下的子文件归属相同的节点进行处理。以下以文件为例介绍打散策略。

系统在创建目录时会默认创建多个目录分片，为每个目录分片选择一个 Shard。在该目录下创建文件时会 HASH 计算落到哪个目录分片，相关文件的元数据操作都会转发到对应的目录分片归属节点上处理。

图3-10 目录分片路由示意图

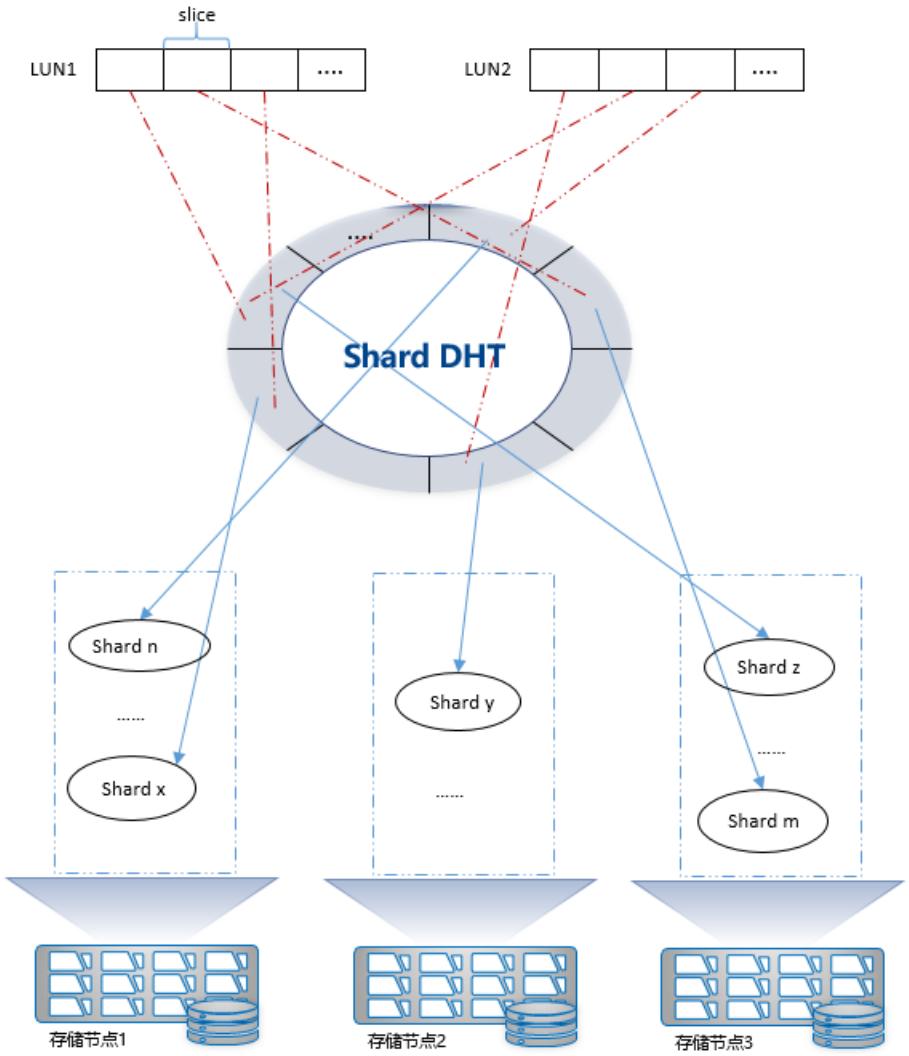


如上图例子所示，d1 目录创建时，系统会为 d1 目录创建 d1-0、d1-1、d1-2 三个目录分片，分别为这三个目录分片选择三个 Shard，根据 Shard 和存储节点的映射关系，决定当前目录分片归属于那个节点。后续 d1 目录下创建的文件会先根据文件名 hash 到一个目录分片，对应的文件元数据操作会转发到相应的目录分片归属节点处理。

3.2.1.3 块语义路由映射

HengShan Stor 系列块存储为了让每个 LUN 的数据更均衡，把每个 LUN 按照固定的粒度（如 4MB）划分成若干个 Slice（分片），每个 Slice 按照“LUN 对应的哈希因子+Shard 起始 LBA”，哈希因子为 LUN ID，计算哈希值，将每个 Slice 落到如下图的 DHT 环上，从而映射到 shard 上。

图3-11 块语义路由示意图



以块服务为例举例说明：

如上层应用发了一个写请求，LUN 1 的 LBA=64MB，Lenth=5MB，那么这块数据首先被分为两个 Slice：S2（64MB~68MB）和 S3（68MB~69MB），根据打散算法，S2 和 S3 分别转到存储节点 1 和存储节点 2 去处理，当都处理完成后，统一返回到上层应用。

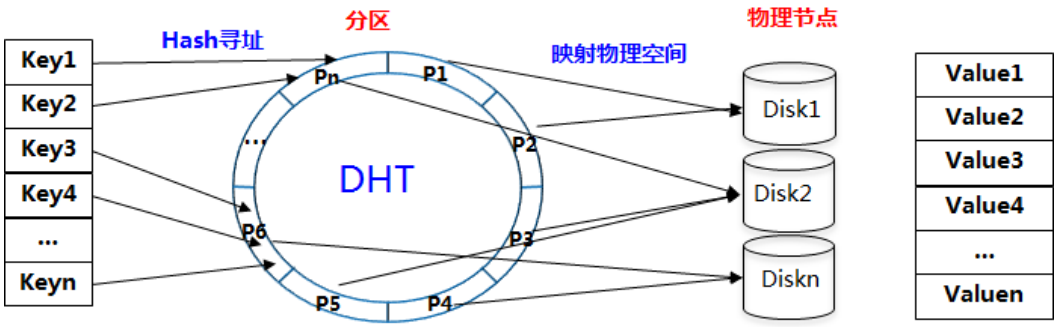
3.2.2 后端负载均衡

请求经过前端负载均衡打散后，在 Service 层归属节点处理完成处理，对于读请求，需要快速定位到数据持久化节点，采用了 STORE DHT 算法进行快速路由。对于写请求，数据最终以 plog 语义写入 Persistence 层，在选择 plog 时，需要考虑到后端节点及硬盘容量的均衡性，采用了动态智能分区和静态选盘算法。

3.2.2.1 STORE DHT

STORE DHT 中的 key 由 PlogID 和 Offset 构成，经过 hash 计算后，确定数据存放在哪一个分区(PT)，进而映射到硬盘的具体位置。分区(PT)支持机柜级、节点级、硬盘级，默认是跨节点组织分区(PT)，系统所有容量按分区(PT)个数均匀分配容量空间。
HengShan Stor 系列采用 DHT 进行数据寻址，具体的算法如下图：

图3-12 STORE DHT 路由示意图



HengShan Stor 系列将哈希空间设置为 2^32，并将该哈希空间划分为 N 等份，每 1 等份是 1 个分区（PT），这 N 等份按照硬盘数量进行均分。例如：系统 N 默认为 3600，假设当前系统有 36 块硬盘，则每块硬盘承载 100 个分区。上述“分区-硬盘”的映射关系在系统初始化时会分配好，后续会随着系统中硬盘数量的变化会进行调整。该映射表所需要的空间很小，HengShan Stor 系列产品节点会在内存中保存该映射关系，用于进行快速路由。

举例说明：

应用需要访问 PlogID+Offset 地址的 4KB 长度的数据，首先通过 PlogID+Offset 构造 key，对该 key 进行 HASH 计算得到哈希值，并对 N 取模，得到分区(PT)号，根据内存中记录的“分区-硬盘”映射表可得知数据归属的硬盘。

说明

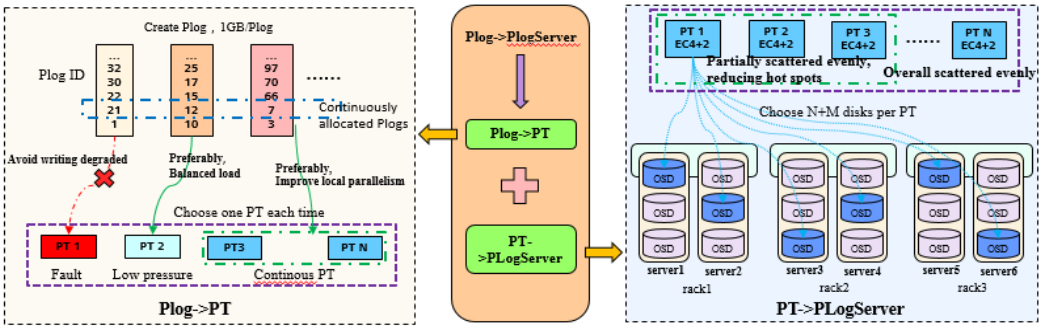
结构化服务支持配置机柜级冗余。

3.2.2.2 动态智能分区和静态选盘算法

在数据持久化的过程中，HengShan Stor 系列采取了优化改进两层算法，从根源上优化分布式存储的性能和可靠性：

- 创建 Plog，为 Plog 选择分区(PT)的动态智能分区算法；
- 分区(PT)选择 OSD 的局部静态选盘算法。

图3-13 DHT 动态智能分区和静态选盘算法



动态智能分区算法主要引入自适应的负反馈机制实现高可靠高性能，主要改进和实现目标如下：

- 写入可靠性不降级：如果 Plog 对应的分区(PT)落到故障盘，直接将此 Plog 弃置，选择新的 plog 写入数据。
- 负载均衡，消除热点：在随机访问场景，当前轮询或哈希分布式算法都不能完全保证数据布局均衡、磁盘访问性能均衡；另外盘故障恢复、慢盘热点、流控 QoS 都会影响系统性能。HengShan Stor 系列会周期性的采集盘、节点、分区(PT)的可用容量以及 IO 压力情况，智能识别热点、快慢盘。

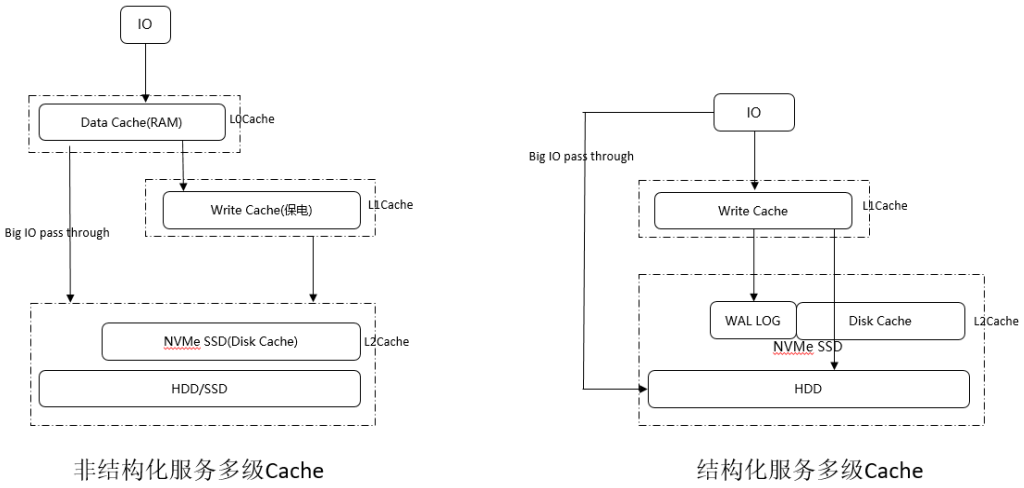
局部静态选盘算法主要解决分区(PT)到 OSD 实例(硬盘)映射的局部均衡最优的问题，在以下场景实现了优化：

- 扩容节点时候不影响可靠性前提下的数据均衡性能优化，减少无效数据迁移。在算法上改进传统按盘均衡为按照分区(PT)组循环挑选分区(PT)，组内均衡既不会降低可靠性也大大降低了数据的无效迁移。
- 多副本场景，优化主分区(PT)的均衡性。通常算法仅关注首次主副本的均衡性，一旦出现节点或磁盘故障，就必然带来许多数据块重新选择主副本的问题，新的主副本是否均衡布局影响当前业务性能也会影响后续数据增量同步的性能。现有技术是个静态固定选主的方案，即主副本坏后选择备 1 再备 2 的顺序。HengShan Stor 系列采用动态选主方案，会根据备副本所在磁盘的 Busy 情况进行选择，避免故障后某些盘成为热点。
- 纠删码场景，提高数据重构并发度，提升性能。同样采用分区(PT)组和全局 Disk 的相关度实现小范围计算，将从单个正常 Disk 读取的数据块越少从而参与恢复磁盘数量越多进行均衡性衡量。

3.3 多级缓存加速

HengShan Stor 系列产品采用多级缓存技术逐层对 IO 进行聚合加速，提升系统整体 IO 性能。在 SSD 与 HDD 混配场景下，构建内存、SSD、HDD 的多层缓存机制，减少下 HDD 盘的 IO 提升整体性能。在全闪存配置场景下，构建内存、SSD 的多层缓存机制，以获取极致性能体现。

图3-14 多级缓存示意图



全闪存配置下，非结构化服务没有 L2Cahce， 结构化服务没有 L2Cache 中的 Disk Cache。

- **非结构化服务多级缓存：**

  HengShan Stor 系列非结构化服务的节点角色通常分为：接入节点，文件归属节点， EC 数据持久化节点。我们会在这三个角色分别构建缓存进行加速。如上图所示，L0 位于系统的接入节点，L1 位于文件的归属节点，L2 位于数据持久化节点。

  如果是标准协议访问，请求接收的存储节点就是接入节点，如果是 DPC（Distributed Parallel Client，分布式并行客户端）访问，DPC 就是接入节点。

  L0Cache 是不持久化的，计算节点应用的异步写 IO 直接写入 L0Cache 就返回成功，当应用下发 flush 命令或 L0Cache 缓存智能聚合后会写到后端存储，流程说明见 3.4.1 非结构化数据存储服务关键 IO 流程章节。

  L1Cache 基于保电内存构建，同时会在节点间镜像，确保节点掉电或故障 cache 数据不丢失。

- **结构化服务多级缓存：**

  HengShan Stor 系列结构化服务不支持保电内存，为实现缓存加速，把 SSD 从功能上分为两个部分：

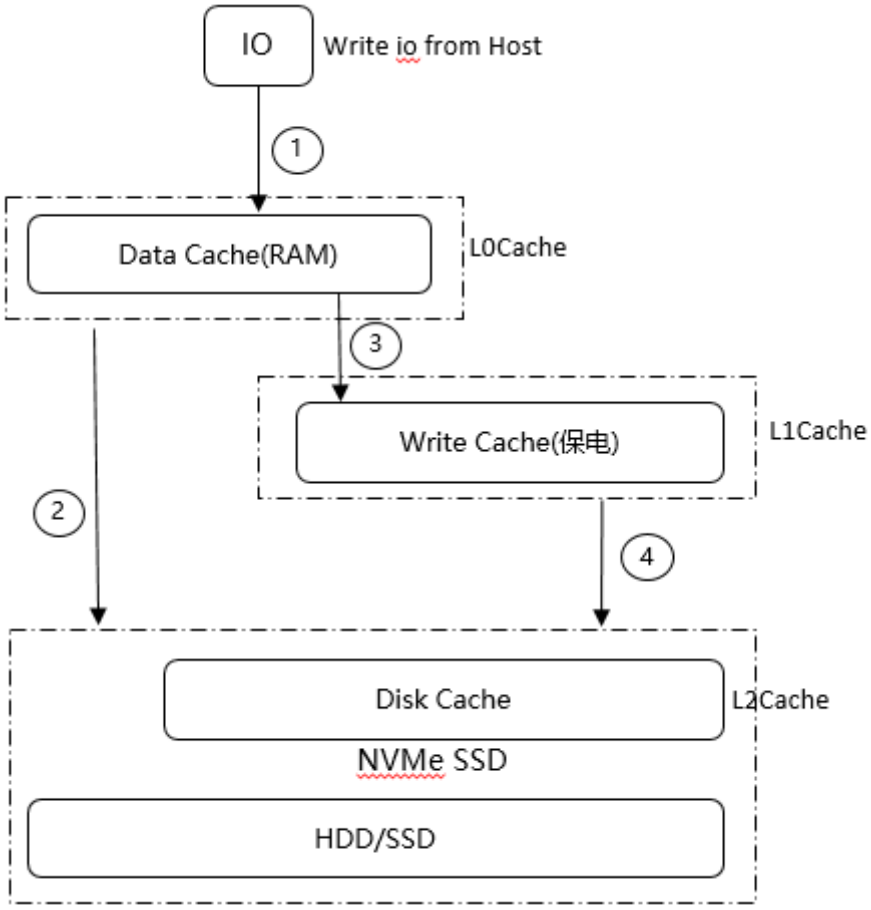
  第一个部分给 L1Cache 做持久化使用，通常为主机业务服务，即主机的 Write IO 通过此部分用来确保数据临时持久化，不会因掉电导致数据丢失，对于这部分的 Cache，我们通常称为 WAL Cache，通常使用小比例的跨节点 EC，保证数据的可靠性；小 IO 在 L1Cache 进行智能聚合，聚合成大块 IO，会以大比例 EC 写入后端 HDD 介质。

  第二部分通常是作为主存机械盘的缓存区域 L2Cache，确保数据在进入 HDD 机械盘之前，先通过这个部分的 Cache 将数据缓存住，这样 IO 不需要到达机械盘，从而降低了时延，对于这部分的 Cache，通常称之为 Disk Cache。如果是全闪存配置，不会构建 Disk Cache。

3.3.1 写缓存加速

- **非结构化服务写 IO：**

图3-15 非结构化服务 Write Cache



a. 主机发送的异步写 IO，会写入接入节点的 Data Cache(RAM)，然后返回成功。同步写 IO 会透写 DataCache，会根据 IO 大小走第 b 步或第 c 步。写入 Data Cache(RAM)的数据达到一定阈值后会下刷，根据聚合的 IO 大小决定是走第 b 步还是第 c 步；

说明

异步写入 DataCache 的数据，除了上述水位刷盘，根据语义标准，在主机下发 flush 时会把相关数据下刷，以确保给用户返回 flush 成功后，数据不丢失。

b. 如果聚合成大 IO，会直接完成 EC 冗余计算，然后把 EC 条带数据跨节点写入数据的持久化节点，元数据转发到归属节点写入 L1Cache，然后返回；

c. 如果没有聚合成大 IO，数据会转发到归属节点写入 L1Cache，更新元数据，同时镜像到其它节点，确保节点故障不丢失，然后返回；

d. 归属节点在后台会把写入 L1Cache 的小块 IO 聚合成大块（满分条），计算 EC 后跨节点写入数据的持久化节点。在 HDD 混配场景下，为了提升 HDD

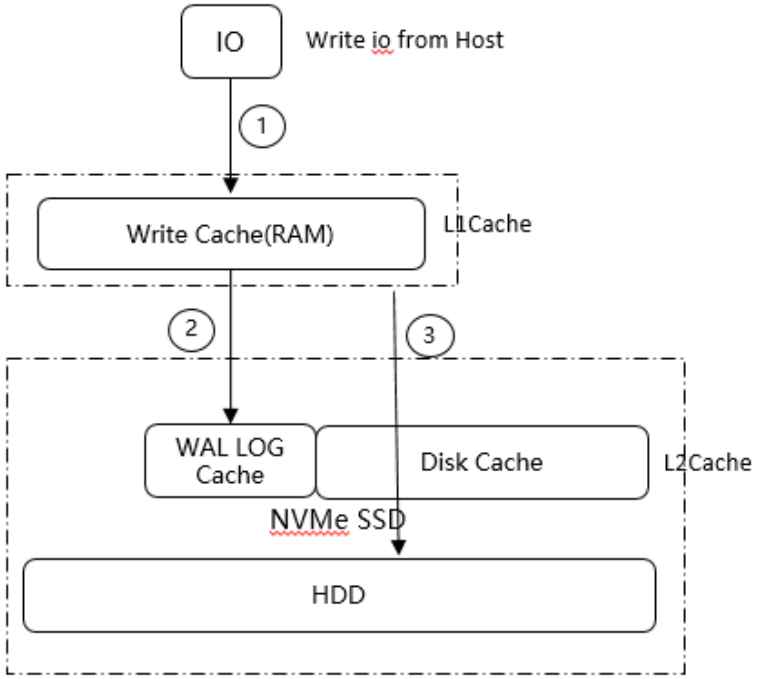
盘的吞吐率，EC 拆分后写入持久化节点的数据，小 IO 会先刷入 L2Cache 进行聚合，聚合成大块后再写入 HDD。

说明

L1Cache 尽量聚合成满分条，以 20+2 的 EC，条带单元为 32KB 为例，L1Cache 聚合成 32KB*20 大小的块就可以满分条写，以减少写惩罚，但是 32KB 的块直接写 HDD 盘不能最大化发挥出磁盘的吞吐率，对于小于 256KB 的块会先进 L2Cache 进行聚合，聚合到超过 256K 后再一次写入 HDD 盘，如果 L1Cahce 能聚合 32KB*20*8，也就是写 HDD 是 256KB 的数据，就不需要进入 L2Cache 聚合了。

- **结构化服务写 IO：**

图3-16 结构化服务 Write Cache



结构化服务主要使用 SSD 作为缓存盘进行加速。如上图所示。

a. VBS 发送的写 IO（图中 Write IO From Host），会转发到归属节点写入 L1Cache,如果 IO 较大直接执行步骤 c 透写返回；

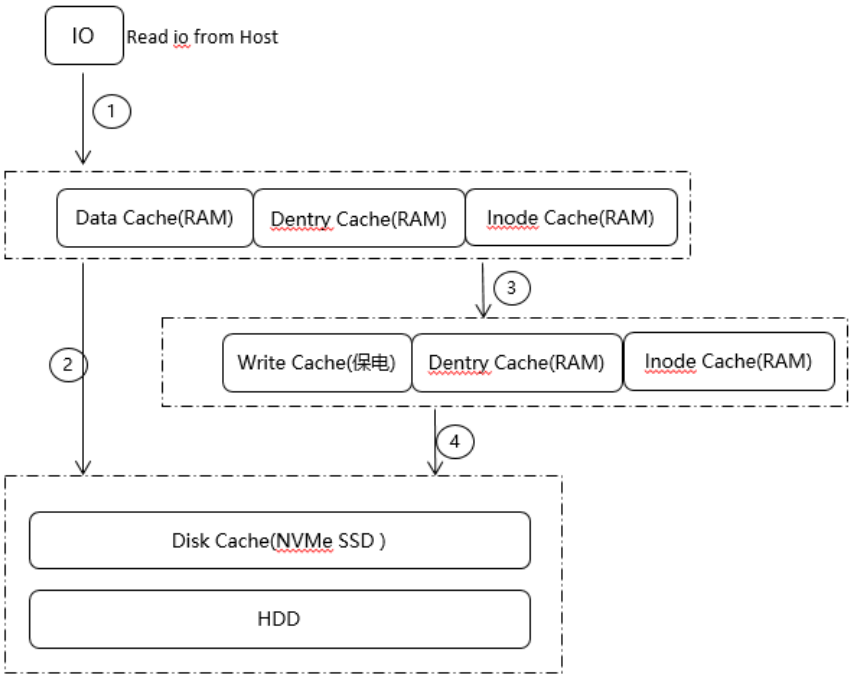
b. 写入 L1Cache 的数据同时以日志的方式（采用固定的 2+2 小分片 EC）跨节点记录到 WAL LOG Cache 中，成功后返回；

c. L1Cache 中的数据会进行 IO 排序重整并等待满分条以副本或 EC 的方式跨节点写入数据最终的持久化节点。在 HDD 混配场景下，为提升 HDD 盘的吞吐率，对于大块 IO 会直接写到 HDD 中，小块 IO 会先在 Disk Cache 中缓存，凑成大 IO 后再写入 HDD。

3.3.2 读缓存加速

- **非结构化服务读 IO：**

图3-17 非结构化服务 Read Cache



a. 主机发送的读 IO 操作（图中 Read io from Host），会优先在接入节点的数据缓存中查找，如果命中就返回。为了提升元数据的性能，在接入节点也构建了 dentry 与 inode 的缓存，如果不命中，根据读取的 io 大小决定是走第 b 步还是第 c 步（元数据不命中，会走第 c 步）；

b. 如果要读取的 io size 较大，会直接从数据的持久化节点读取；

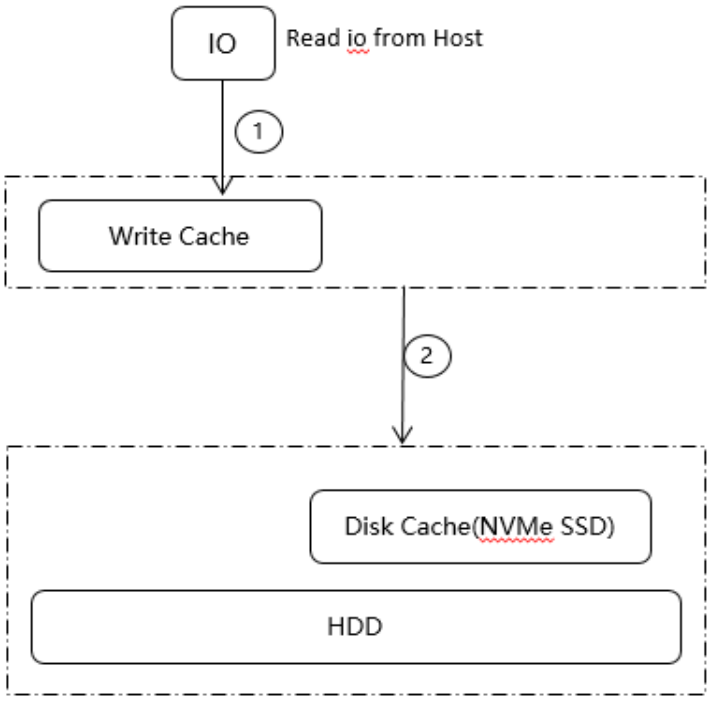
c. 如果要读取的 io size 较小，数据会转发到归属节点处理，归属节点会优先在 Write Cache(保电)内存中查找（因为有可能数据还没有下盘），如果命中就返回；

d. 归属节点写缓存中不命中就发往数据持久化节点读取。

在数据持久化节点读取时，按如下顺序进行查找 Disk Cache(NVMe SSD) -> HDD。

- **结构化服务读 IO：**

图3-18 结构化服务 Read Cache



HengShan Stor 系列结构化服务的读缓存，第一层为内存 Cache; 第二层为基于 SSD 的 Disk Cache，SSD Cache 采用热点读机制，系统会统计每个读取的数据，并统计热点访问因子，当达到阈值时，系统会自动缓存数据到 SSD 中，同时会将长时间未被访问的数据移出 SSD。

存储节点在收到 VBS 发送的读 IO 操作时，会进行如下步骤处理：

a. 从内存“Write Cache”中查找是否存在所需 IO 数据，如果存在，则直接返回，否则执行第 b 步；

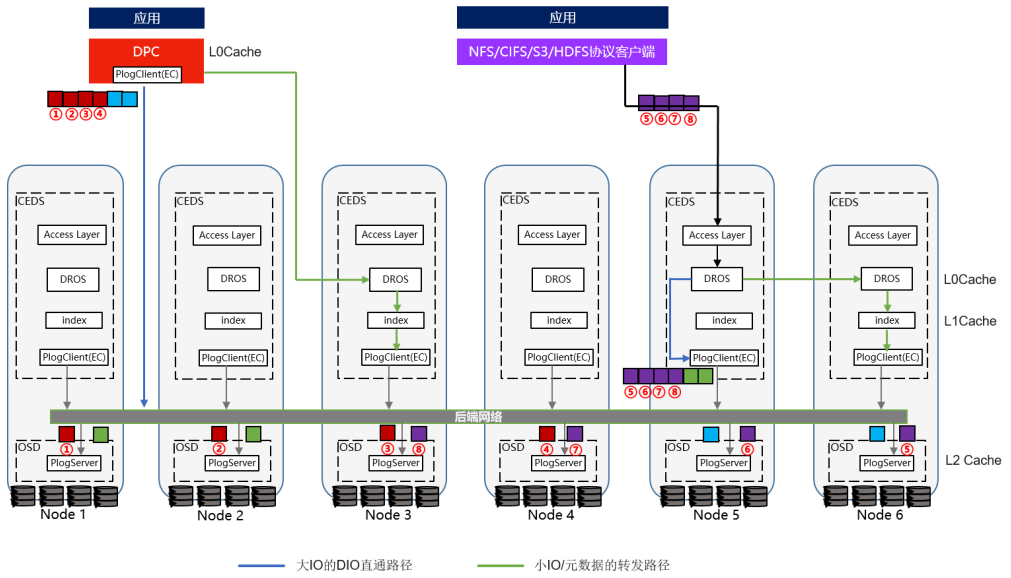
b. 读请求发到数据持久化节点(OSD)，先从 SSD 的 Disk Cache 中查找是否存在所需 IO 数据，如果存在则直接返回。如果不存在则从硬盘中查找到所需的 IO 数据并返回，同时增加该 IO 数据的热点访问因子，如果热点访问因子达到阈值，则会被缓存在 SSD 的 Disk Cache 中。

3.4 关键 IO 流程

3.4.1 非结构化数据存储服务关键 IO 流程

DROS 是 HengShan Stor 系列的非结构化数据管理平台，非结构化服务是基于此平台提供，基础 IO 都进行了归一。为了降低网络开销，对于大 IO 进行读写直通的处理(DIO)，不需要把大块数据转到归属节点再写 Persistence 层，直接从接入节点访问 persistence 层。

图3-19 非结构化服务 IO 流程示意



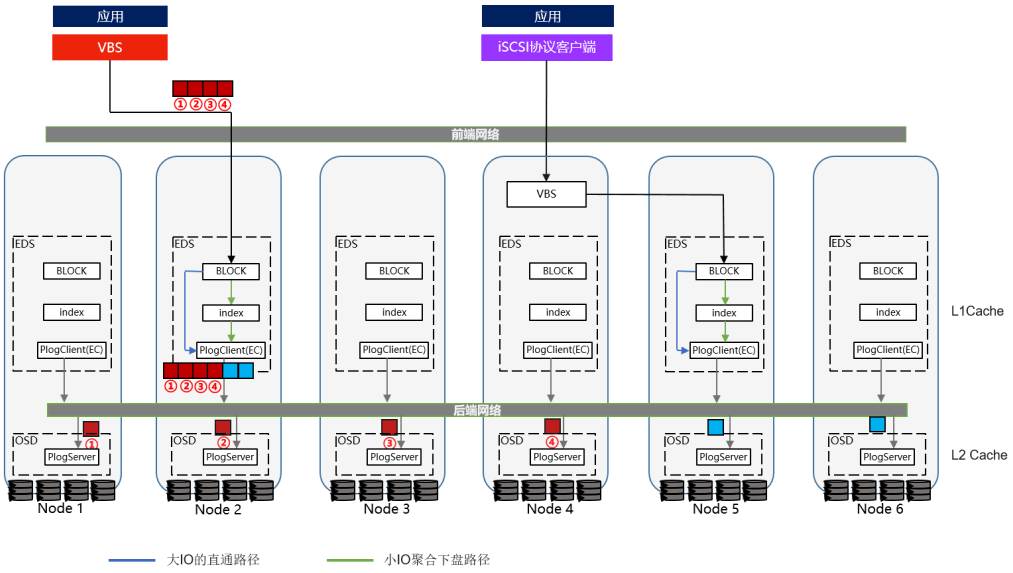
如上图所示，DPC 做为接入节点时，异步 IO 数据块 1，2，3，4 写入 DPC 侧的 L0Cache，从 L0Cache 往下刷数据时，如果数据在 L0Cache 聚合成了大 IO，就直接满分条计算 EC 冗余后直通写到 OSD，然后把元数据转发到归属节点。如果是小 IO 就直接转发到归属节点处理，归属节点针对小 IO 会在 L1Cache 中，L1Cache 把小 IO 聚合成大 IO 再计算 EC 写入 OSD。标准协议接入访问，存储节点作为接入节点，异步写 IO 数据块 5，6，7，8 写入 L0Cache，L0Cache 往下刷时，聚合成大块 IO 计算宗元 EC 冗余后写到 OSD，然后把元数据转发到归属节点写入 L1Cache，如果是小 IO 转发到归属节点的 CEDS 进入 L1Cache 等待聚合下盘。

说明

创建、删除、设置属性等元数据操作属于小 IO，会直接走转发路径到归属节点上处理。

3.4.2 结构化数据存储服务关键 IO 流程

图3-20 结构化服务 IO 流程示意



上层应用下发写 IO 请求到存储服务的 VBS（Virtual Block Service），VBS 收到该 IO 请求，根据第一层的 DHT hash 算法将数据转到指定存储节点；由这个存储节点上的 EDS（BLOCK+Index）模块处理该数据；

EDS 接收到写 IO 请求后，如果是小 IO，以小比例 EC 形式写入 L1Cache，同时该 EDS 所在存储节点的内存中仍然保持一份该数据，EDS 返回写 IO 成功给 VBS，再由 VBS 返回给上层应用。待 L1Cache 内存中的数据聚合到更大的块，聚合成满分条计算 EC 冗余后，通过 PLOG 接口写到 OSD。如果是大 IO，就不需要进入 L1Cache 进行聚合，直接按用户配置的 EC 比例计算冗余，写入 OSD。

说明

结构化服务的 VBS 可以部署在存储节点，也可以部署到计算节点上。

3.5 非结构化数据存储服务融合互通

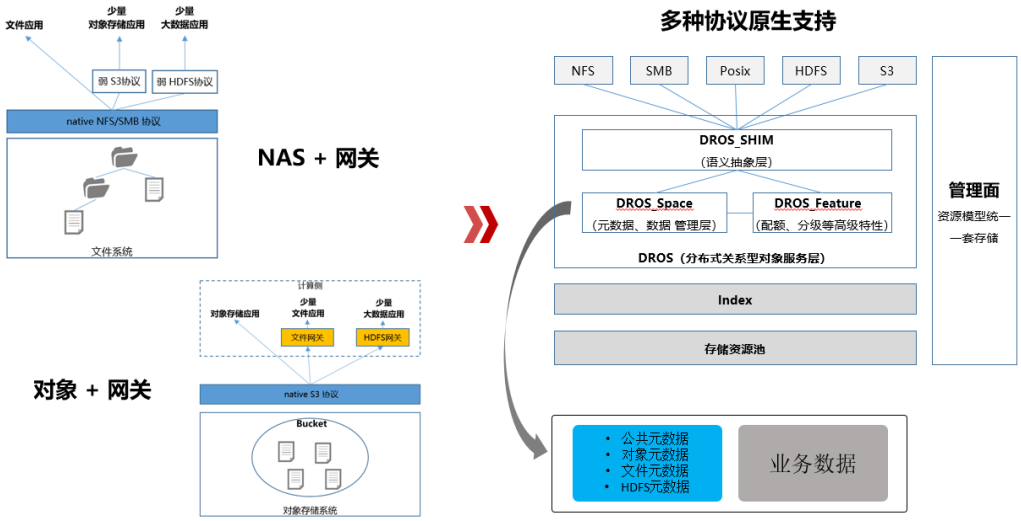
万物互联、云和智能化驱动数据高速增长，越来越多的数据被产生，其中非结构化数据成为主力，但由于文件、对象、大数据这些服务的差异性，导致多服务烟囱式构建，用户最终要同时使用多套存储系统，才可以解决数据的全生命周期管理。以自动驾驶场景为例，不同的处理阶段需要访问不同类型的数据，端到端数据处理流程需要多种协议或者多套存储系统参与。在这种解决方案中，用户的主要痛点如下：

- **空间浪费**：同一份数据需要在多套存储系统中存储多份，浪费存储空间，大大增加了用户的存储成本。
- **效率低下**：同一份数据需要跨多种存储系统复制，一般的对象存储系统性能较差，随着数据量越来越多，这种复制可能需要几小时，几天甚至更长的时间，大

大影响了用户的数据处理效率。在上面讲的自动驾驶领域，这种复制消耗的时间，将直接影响最终汽车的研发效率。

- **运维困难：**由于是多套存储系统，设备管理、系统操作等种种差异，会大大增加管理人员的运维难度。

图3-21 HengShan Stor 系列非结构化服务融合互通示意

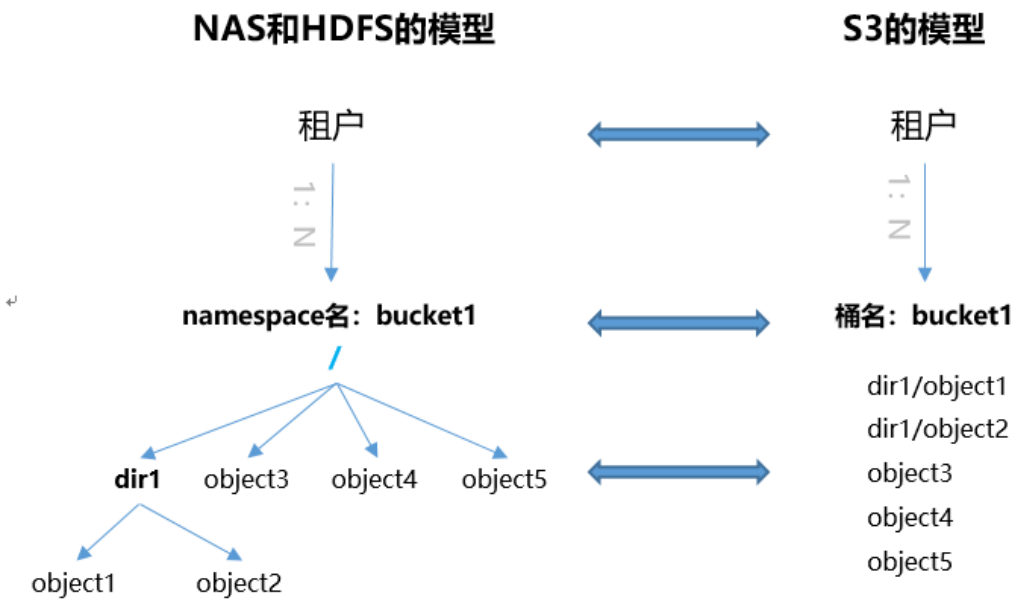


业界支持非结构化数据融合互通的架构主要有两种：以“文件存储系统”为底座的互通架构和以“对象存储系统”为底座的互通架构。这两种架构互通基本都是基础 IO 语义的数据互通，元数据和数据管理系统仅可以偏向一类生态应用，只有一类 native 语义，另一类要么不支持，要么协议很弱，无法支撑上层生态应用。为解决这些问题，HengShan Stor 系列推出了文件/对象/大数据三个非结构化服务融合互通的技术方案，具备如下关键能力：

- **管理面统一：**资源模型统一，如租户/用户、网络、存储池等；
- **共享存储资源池：**非结构化服务在同一个存储资源池(storage pool)内提供，多服务共享一份数据；
- **IO 语义互通：**全 Native 语义、协议无损，无需外部插件；
- **高级特性互通：**非结构化服务共享高级特性且互通，如 QoS、分级和配额等特性共享，无语义损失，扩展性、性能、可服务性等能力统一。

HengShan Stor 系列非结构化服务共享存储资源池(storage pool)，这是多服务语义互通的前提（一份数据写入，多种协议可访问）。为了实现语义的互通，统一了多种非结构化服务的概念模型。HengShan Stor 系列非结构化服务提供统一的多 Namespace，文件服务称之为文件系统，对象服务称之为 Bucket。各个 Namespace 的用户数据相互隔离，系统资源争抢通过 SmartQos 机制控制，一个 Namespace 可以同时支持大数据/文件/对象服务。

图3-22 Namespace 与桶的映射关系示意



基于 DROS 底座，为了实现非结构化服务数据互通，Namespace 映射为对象的 bucket，把文件的目录结构与对象的 URL 进行了映射，映射关系示意如上图所示，Namespace 被映射成桶 bucket1，目录/dir1 下的文件 object1, object2，被映射成桶 bucket1 下的对象 dir1/object1, dir1/object2，根目录下的文件 object3, object4, object5 被映射成桶 bucekt1 下的对象 object3, object4, object5。

反过来用户从 S3 对象侧创建的对象也会相应的映射成对应的目录结构，如果一个空的 Namespace(映射成桶 bucket1)，用户从 S3 对象侧在桶 bucket1 下创建 dir1/object3 对象，对应的 DROS 会为这个对象创建相应的目录 dir1，及目录 dir1 下的文件 object3.如果用户从 S3 对象侧在桶 bucket1 下创建 object3 对象，对应的 DROS 会在 Namespace 的根目录下创建文件 object3。

如此可以保证数据一份数据，从文件写入，对象可参访问，对象写入文件也可以访问。

说明

上述例子是两级文件目录，多级目录是相同的映射关系，如/dir1/dir2/object 映射到对象就是桶 bucket1 下的对象/dir1/dir2/object。

基础的模型的统一，只能解决最基础的多服务互通访问，NAS/HDFS/对象各自的服务语义是有差异的，如锁、权限等。解决了上述的基础模型映射后，IO 语义互通的还有三个主要关键点：

- **语义无损互通**

  如 S3 协议的 ListObjects 接口、多段、多版本接口；

  如 HDFS 协议的 Concat 接口。

- **权限互通**

  同一个文件被多个协议同时访问时，如何做到权限互通，保证数据安全。

- **锁互通**

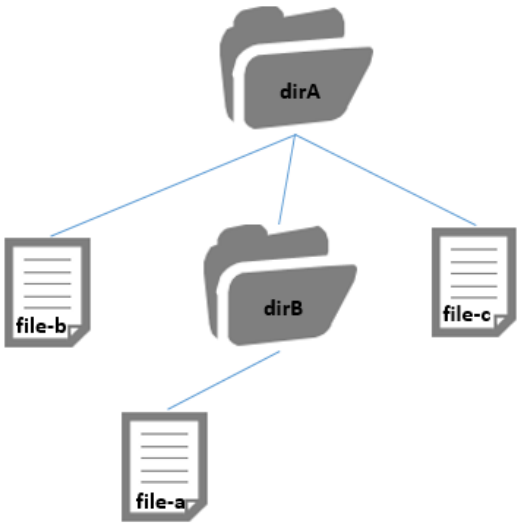
  同一个文件被多个协议访问时，不同协议的锁如何做到互通，如何保证数据并发访问的一致性。

3.5.1 语义无损互通

3.5.1.1 S3 协议 ListObjects 接口

S3 协议的 ListObjects 类似于 NAS 和 HDFS 的 readdir 接口，但是 S3 的 ListObjects 具备更强大的功能，支持：列举结果按字典序排序、按前缀/标记检索和分页等，而且该接口无目录语义，而是将每个文件的全路径的 path 作为列举结果。

图3-23 传统的目录元数据结构图



如上图所示，使用 NFS 协议创建的目录 dirA，子目录 dirB，以及文件 file-a,file-b,file-c。如果使用 NFS 协议调用 dirA 的 readdir 接口，那么返回值为：

file-b

dirB

file-c

NFS 等协议的 readdir 结果是不要求有序的，而且仅返回当前目录的子目录或文件。

如果使用 S3 协议调用 ListObjects 接口，prefix=dirA，那么 list 结果为：

dirA/dirB/file-a

dirA/file-b

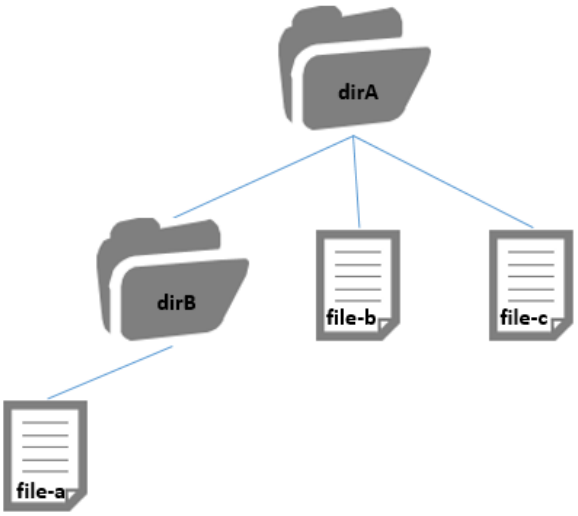
dirA/file-c

从结果中可以看出 S3 协议是要求全路径返回的，且结果是按字典序排序的。

传统互通厂商，目录元数据基本都采用上面这种全局无序的结构，无法支持标准 S3 协议的 ListObjects 接口。

基于这些需求，HengShan Stor 系列的目录元数据组织支持有序结构存储，实际系统中存储的元数据结构如下图：

图3-24 HengShan Stor 系列的目录元数据结构图



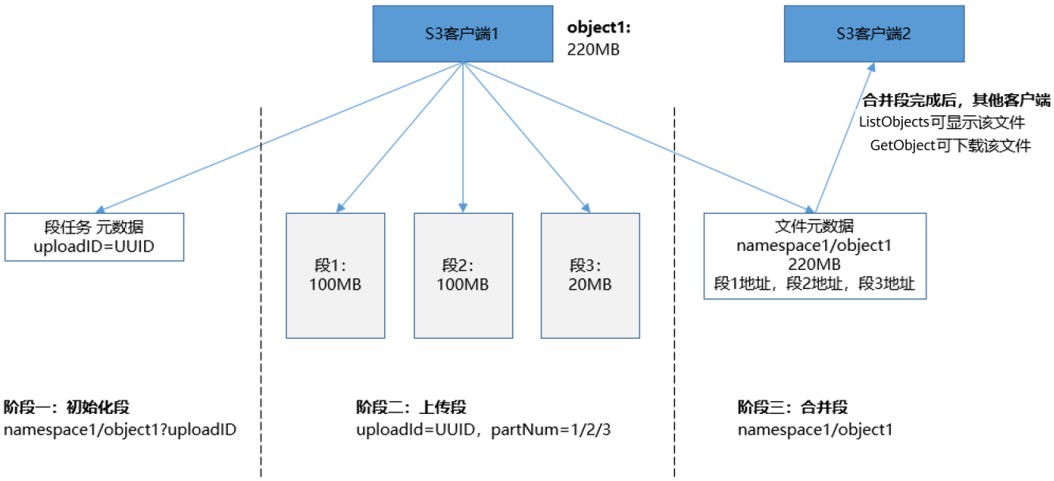
使用 NFS、SMB 等其他任何协议写入的文件，在系统内部都支持采用这种元数据布局，从而保证在多协议互通场景下，HengShan Stor 系列可以提供 native 的 ListObjects 能力，兼容该接口的各种功能参数。

3.5.1.2 S3 协议多段接口

多段上传接口对 S3 协议而言是一个极其重要的接口，几乎所有的 S3 应用都使用多段接口来上传几 MB 以上的中大文件。所以，如果说不支持多段接口的互通，基本上可以等价于不支持 S3 协议的互通。

然而，这个接口在其他协议上是不支持的，而且也找不到类似的接口。S3 的多段接口原理如下：

图3-25 S3 协议多段接口原理图



初始化段：生成一个多段任务 UUID。

上传段：按段号上传实际的数据，此过程支持客户端多线程或多客户端并发上传段数据，提升上传性能。比如客户端有个 220MB 的文件，分 3 个线程上传，线程 1 上传 0~100MB，线程 2 上传 100MB~200MB，线程 3 上传 200MB~220MB，三个线程并发互不影响，上传失败了只需重试自己的段数据即可。

合并段：当所有段上传完成后，客户端执行合并段操作，系统会生成 object1 的文件元数据，元数据中记录这个文件是由哪些段组成的。此步骤，仅仅是元数据操作，不会真正的合并数据，所以效率极高。

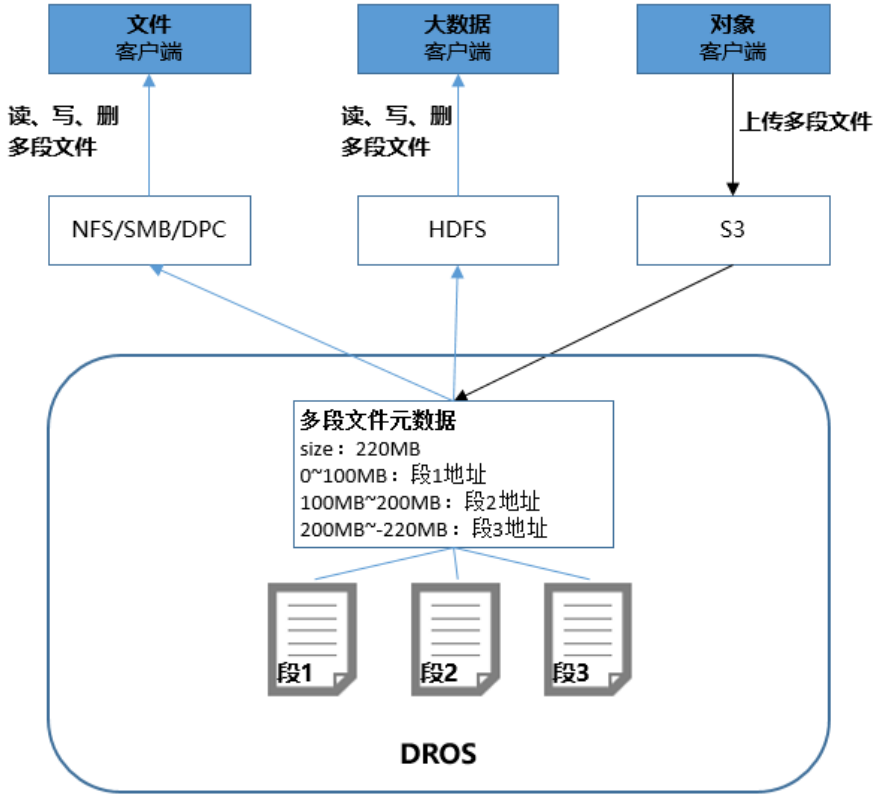
只有阶段三合并段完成后，这个文件才对外可见，其他客户端才可以下载或删除该文件。

从以上原理介绍可以看出，S3 协议的多段接口不仅具备客户端高并发、高性能上传的特点，而且还具备类似“断点续传”的功能，不同的段之间互不影响，上传失败只用重试自己的段。

业界的互通架构，一般是不支持多段文件的跨协议访问的，或者仅支持其他协议可读不可写，所以不能算支持 S3 协议的互通。

而 HengShan Stor 系列的多段接口支持其他所有协议的可读、可写。具体原理如下图：

图3-26 多段处理示意图



如上图所示，对于多段文件，DROS 会记录特殊的元数据结构，会记录文件每个 offset 对应的段文件地址。对于其他协议的主要操作，实现原理如下：

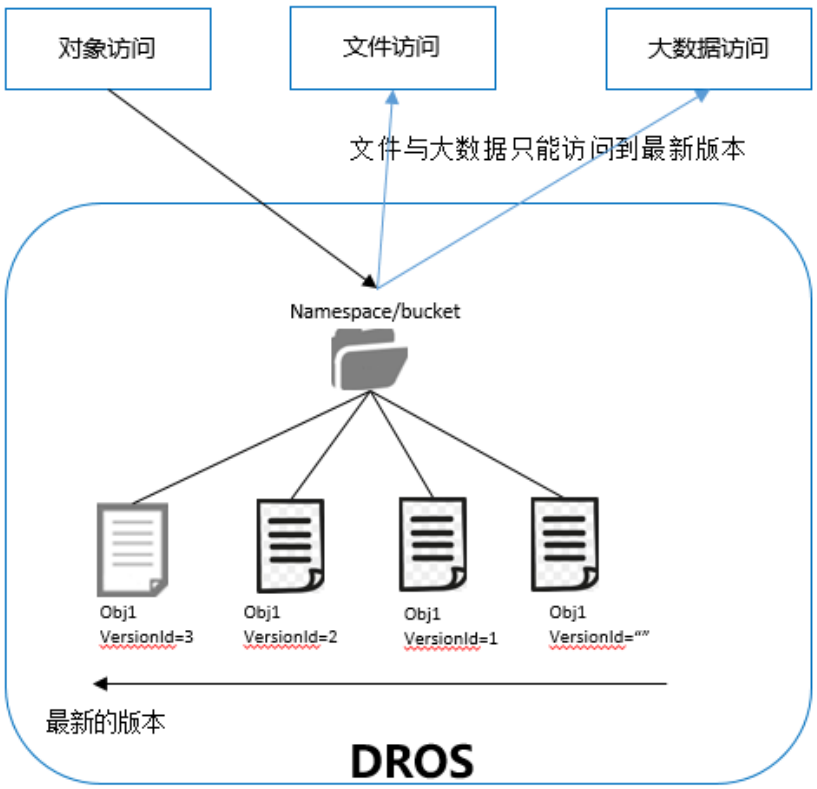
- 偏移读操作：将 offset 转换为对应段内的偏移读。
- 修改写操作：将 offset 转换为对应段内的偏移写。
- Truncate 操作：会将 truncate offset 对应的段数据回收掉

3.5.1.3 S3 协议多版本接口

S3 协议多版本控制是对象一个重要的接口功能，利用多版本控制，可以在一个桶中保留多个版本的对象，让用户更方便的检索与还原各个版本，在意外操作或应用程序故障时快速恢复数据。

HengShan Stor 系列的多版本在桶级别设置开关，默认禁用，可以通过 API 或管理控制台启用、暂停。禁用状态下，上传的每个对象具有“空”的 versionId，上传时会覆盖同名的对象。启用状态下，上传的每个对象具有“非空”的 versionId，上传同名对象会生成新的版本，不会覆盖老的同名对象，大数据与文件访问时并不会看到历史版本的信息，只会看到最新版本的文件名与数据。

图3-27 多版本处理示意图



多版本一旦启用永久生效，但可在启用和暂停模式之间切换。

3.5.2 权限互通

支持 NFS/SMB/DPC/FTP/FTPS、HDFS 和 S3 跨协议访问权限控制，实际上就是需要达到各个协议之间权限信息能互通，并能对每个协议的用户进行精确的权限控制，如 SMB 用户设置的权限也能对 NFS 用户生效。但是由于当前每个协议的用户并不是一致

的（SMB 的用户在 AD 域上，NFS 的用户可能在 LDAP/NIS 上），对应的用户标识方式也不一致（SMB 的用户使用 SID，NFS 的用户使用 UID/GID），导致的各个协议设置的权限很难互通（权限是以用户/用户组为管理对象）。

说明：以下重点阐述 SMB、NFS 和 S3 协议的互通，HDFS/DPC/FTP/FTPS 的 user 管理、权限管理基本和 NFS 一致。

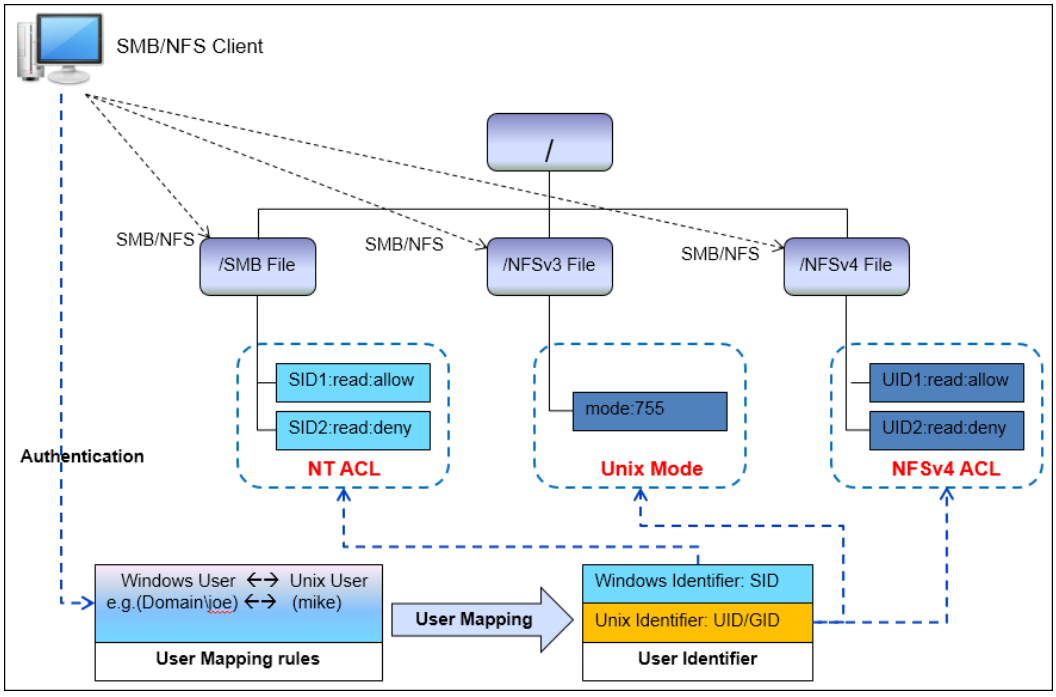
3.5.2.1 NFS 与 SMB

为了达到权限互通的目的，首要目标就是需要对各个协议的用户进行关联，我们使用用户映射的方案（指定某个 Windows 用户和某个 Unix 用户对应）来达到这一目的。在用户映射方案下，SMB 或者 NFS 用户统一标识应该包含：

- 用户的 SID 和所属用户组的 SID，用于 SMB 共享权限的鉴定以及 NT ACL 的鉴定。
- 用户的 UID 和所属用户组的 GID，用于 UGO 的权限鉴定。

在 SMB 或者 NFS 用户进行用户映射后得到统一的用户标识，后续应用在 ACL 安全鉴权环节，根据设置的 ACL 类型获取对应的用户标识进行权限判断即可，参考下图：

图3-28 SMB-NFS 跨协议访问用户映射示意



SMB-NFS 跨协议访问业务中，对于进行用户映射的时机，需要尽量减少对文件 IO 性能的影响，尽量在非 IO 数据流上去进行。针对不同协议的特点，区别如下：

- 在 SMB 协议中，对于一个 SMB 会话，只会在建立会话时进行一次用户认证动作，确定用户权限信息，用于后续该会话的文件访问鉴权。因此 SMB 业务中，用户映射动作放在用户认证环节进行。
- 在 NFS 协议中，区别于 SMB 协议，每一次文件操作请求中，客户端均会将用户信息发送给存储系统，无法做到一次认证，长期使用。因此将用户映射动作放到

真正 ACL 鉴权的时刻，只有当文件的 ACL 为 NT ACL 的时候，才需要进行用户映射，这样可以减少大量的不必要的用户映射操作。

权限控制行为：

一个文件系统或 DTree（Directory Tree）可通过设置不同的安全类型，来满足不同应用场景的权限控制管理需求。例如，若一个文件系统主要应用于 NFS 业务访问，SMB 业务只进行简单的读取业务，则可设置该文件系统的安全类型为 UNIX，文件权限控制以 Mode/posix ACL 为准；若一个文件系统主要应用于 SMB 业务访问，NFS 业务只进行简单的读取业务，则可设置该文件系统的安全类型为 NTFS，文件权限控制以 NT ACL 为准。

当前 HengShan Stor 系列存储产品暂只支持 Unix 安全类型，该安全类型下，Windows 或 Linux 客户端访问行为模式如下表：

表3-3 Window/linux 客户端访问行为说明

-		Window Client Access	Linux Client Access
鉴权		SMB 用户映射到 NFS 用户，使用 Mode 鉴权	NFS 用户 Mode 鉴权
创建文件	Owner	SMB owner:映射后 NFS 用户 NFS owner:映射后 NFS 用户	SMB owner:映射后 NFS 用户 NFS owner: 登录 NFS 用户
	继承	Mode bit	客户端设置
设置 ACL		视条件允许或忽略	不支持
查询 ACL		映射后 ACL	不支持
Chmod		NA	成功
修改 Owner		忽略	成功

3.5.2.2 S3 与其它协议

S3 协议不同于其他协议，它不采用 UGO 鉴权、也不采用 user 级 ACL 鉴权。S3 协议的权限控制主要支持以下几种方式：

表3-4 S3 协议权限控制说明

控制对象	权限机制	说明
User	IAM Policy	给指定的 user、usergroup 授予指定的权限。

控制对象	权限机制	说明
Bucket （对应 HengShan Stor 系列的 Namespace）	1. Bucket ACL 2. Bucket Policy	Bucket ACL 一般是跨租户间简单的 read/write/readwrite 等权限控制；而 BucketPolicy 控制的粒度更细，比如可以控制特定前缀的对象，或控制 IP 段或 https 等。
Object	Object ACL	区别于其他协议，S3 协议的这个 ACL 是跨租户授权的，不支持租户内的 user 间授权。

基于该协议的差异，HengShan Stor 系列支持将 S3 的用户映射为 unix 用户，从而达到权限互通的目的，具体机制如下：

1. 管理员首先建立“S3 用户”到“unix 用户”的映射；
2. 映射成功后，S3 协议写入的所有文件都会记录正确的 UID 和 GID 等信息，此时其他协议访问时，即可按 UGO 或 posix ACL 进行鉴权。

图3-29 用户映射模型图

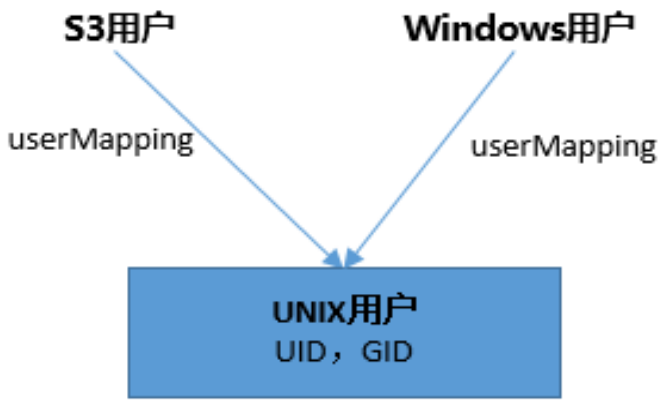
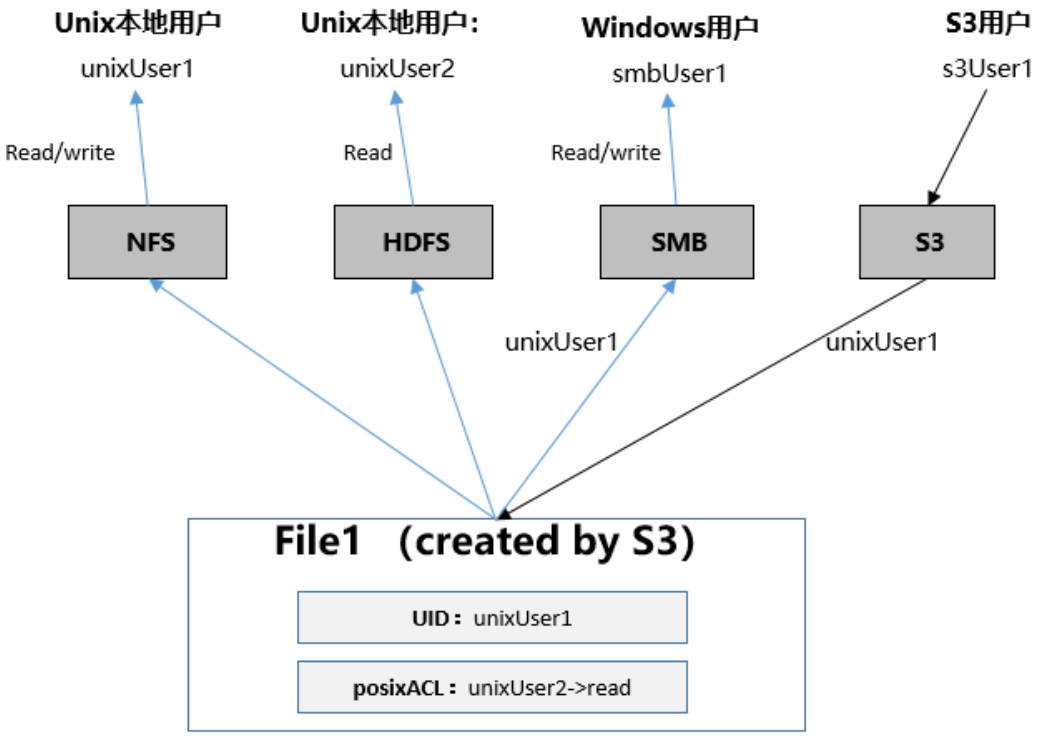


图3-30 S3 和其他协议权限互通的示例图

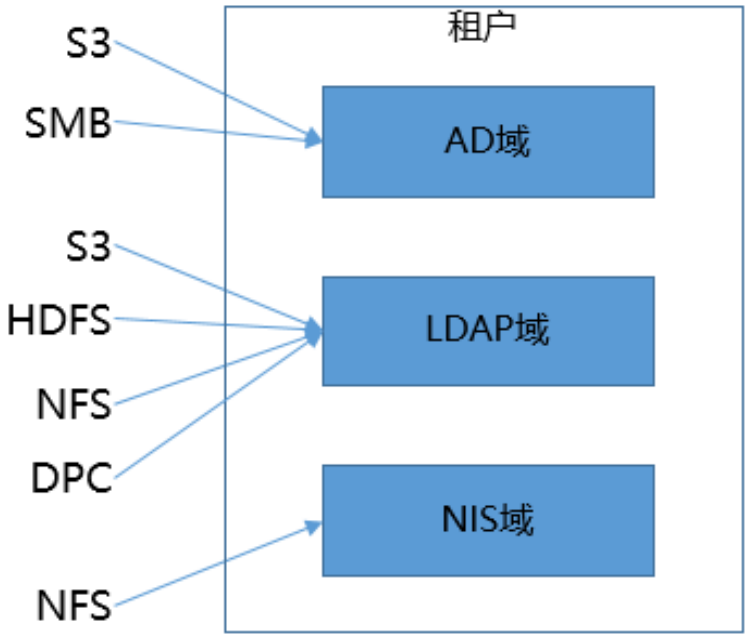


如上图所示，S3 用户和 Windows 用户都映射为 unixUser1 用户，使用 S3 协议写入的文件就会记录正确的 uid 为 unixUser1，然后，NFS 协议和 SMB 协议就可以正确的读、写这个文件。同时 HDFS 协议还可以为该文件设置 ACL，运行 unixUser2 可以 read 该文件，设置完成后 HDFS 协议的 unixUser2，还可以 read 该文件。从而实现，多个不同协议的权限互通。

3.5.2.3 域用户

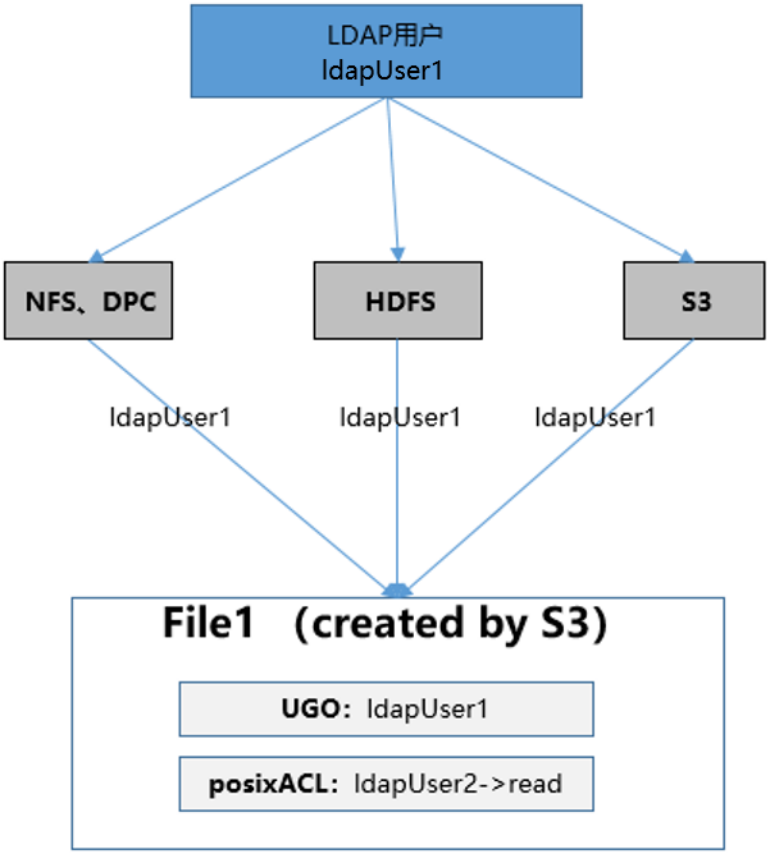
业界标准 NFS、SMB、HDFS、DPC 和 S3 协议支持的域认证服务不同，HengShan Stor 系列支持的域认证能力如下：

图3-31 不同协议的域用户



从图中可以看出，同一个租户的 LDAP 域用户支持 S3、HDFS、NFS 和 DPC，所以对于 LDAP 域用户，HengShan Stor 系列天然支持这些协议的权限互通，无需建立用户 mapping。

图3-32 LDAP 域用户的权限互通原理图



如上图所示，S3 协议使用 LDAP 域用户 ldapUser1，创建的文件 File1，此文件的 UID 即为 ldapUser1，这样以来文件客户端和大数据客户端，统一使用 ldapUser1 访问该文件时，自然就具备相应的权限，无需 usermapping。

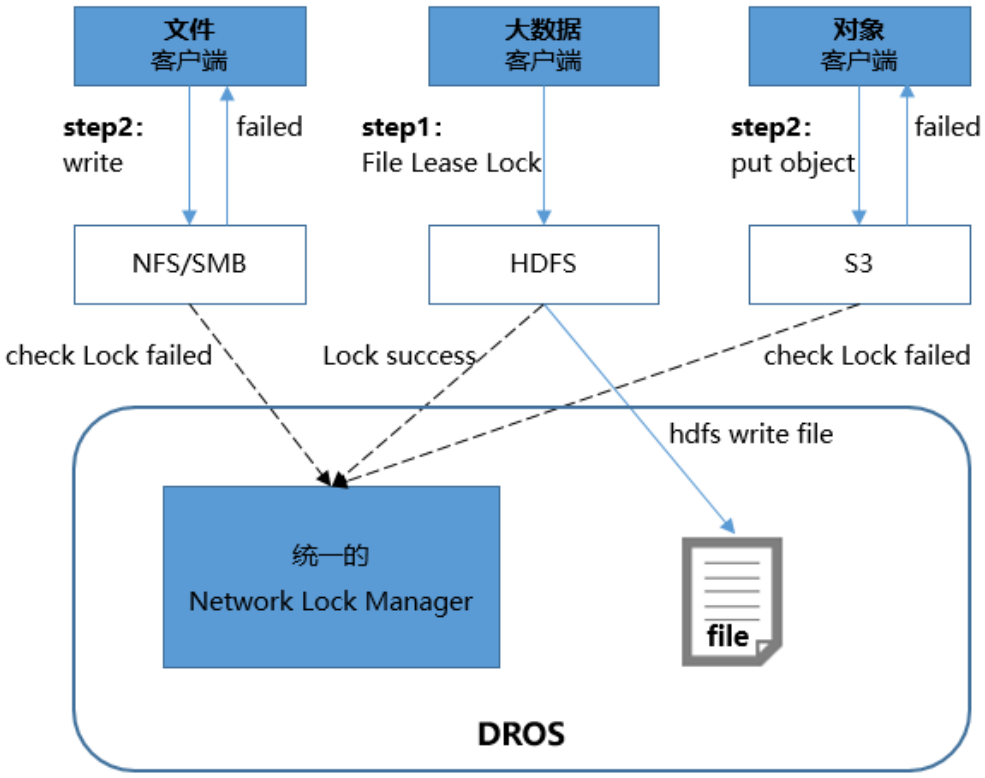
如果 windows 用户通过 SMB 协议访问该文件，那么需要将 windows 用户建立映射到这个 ldap 域用户上。

3.5.3 锁互通

HengShan Stor 系列拓展了 S3、HDFS 协议的锁能力，任何一种协议一旦加强制锁（支持文件级和 BRL(Byte Range Lock)级）成功，其他协议来修改写/追加写都会失败，从而保证了多协议并发访问时的数据一致性。

HengShan Stor 系列架构上具备统一的分布式锁管理服务 NLM，多协议并发访问都需要检查加锁是否成功。

图3-33 锁互通原理示意

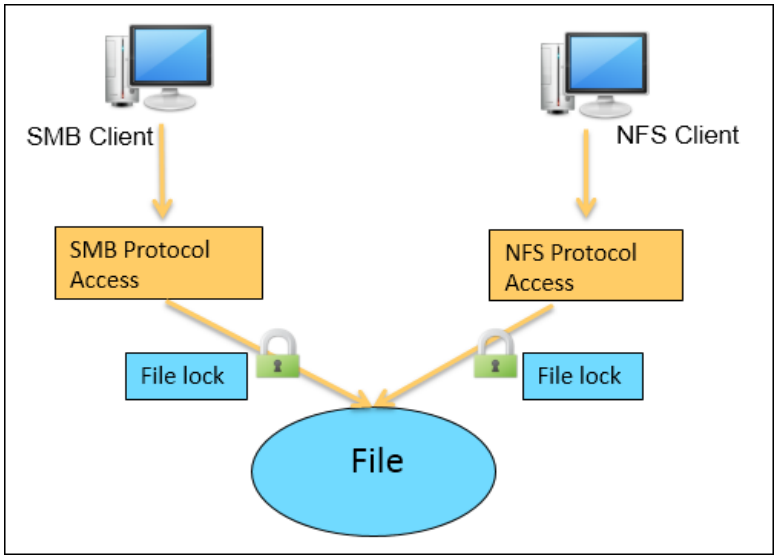


如上图所示，NFS/SMB、HDFS 和 S3 协议并发操作同一个 file，如果 HDFS 协议优先申请 lease LOCK 成功，那么其他协议再去修改同一个 file 的时候，也要检查该文件是否已经被 Lock 住，如果已经被 Lock 住，那么其他协议就会检查锁失败，最终 IO 会返回失败。HDFS 协议加锁成功后，就可以修改写这个 file。

在文件加锁方面，SMB 协议和 NFS 协议最大的差别在于，SMB 有文件锁，而 NFSv3 没有文件锁，SMB 和 NFS 虽然都有范围锁，但是范围锁并不能和文件锁进行互斥。针对不同的应用场景，在协议文件锁互斥上采取不同的策略：

- 如果文件系统安全模式为 UNIX，则主要应用 NFS 协议生产数据，此模式下 NFSv3 协议不主动增加文件锁检测，只有在其应用主动加锁的情况下才会进行锁互斥的检测；
- 如果文件系统安全模式为 NTFS，则主要应用 SMB 协议生产数据，此模式下 NFSv3 协议在打开文件时会主动增加文件锁检测，如下图所示。

图3-34 NFS 与 SMB 锁互斥示意图



此产品版本中，文件系统安全模式仅支持 UNIX 模式，因此只有在业务主动加锁过程中才会做锁互斥检查，跨协议场景下锁互斥检查情况如下：

表3-5 锁互斥检查情况描述

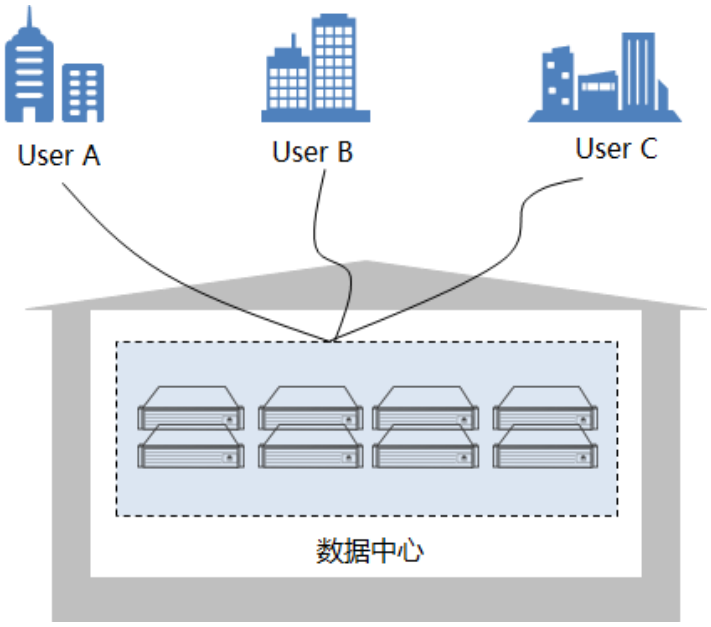
Lock 1	Lock 2	检查点
NLM	NLM	owner + type+ range
NLM	SMB OPEN	type + share access
NLM	SMB BRL	type + range
SMB OPEN	NLM	type + share access
SMB OPEN	SMB OPEN	desire access + share access
SMB BRL	NLM	type + range
SMB BRL	SMB BRL	type + share access

其中 type 为 Shared 或者 Exclusive。

4 增值特性：效率提升

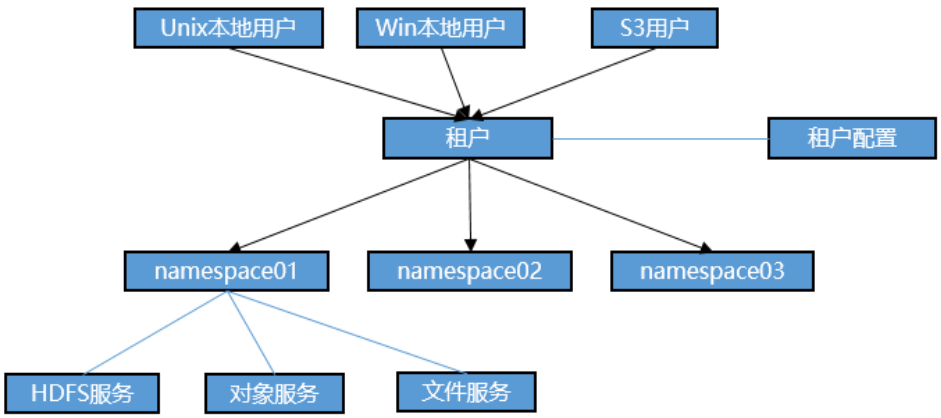
4.1 多租户（SmartMulti-Tenant）

图4-1 多租户示意



如上图所示，HengShan Stor 系列非结构化服提供多租户管理能力，不同租户之间的数据逻辑隔离，便于资源划分，价值点主要体现在：一套系统集中管理，资源被多个租户共享，减少初始投资，数据逻辑隔离提升租户的安全性。

图4-2 租户管理模型示意



一个租户可以有多个 Namespace，每个 Namespace 可以同时提供大数据、对象和文件服务，一个 Namespace 只能属于一个租户。多租户特性提供基于租户粒度的统一资源管理，以租户为单位分配和管理资源。多个租户共享同一套物理存储系统，租户间资

源隔离，确保安全性和隐私。多租户管理提供了一个通用的可扩展的多租户管理模型，通过多租户管理可实现租户级的配置，如下表所示。

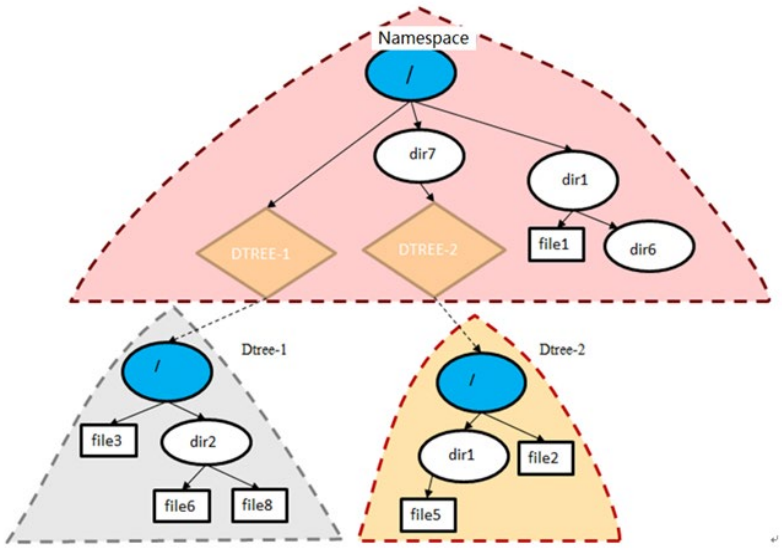
表4-1 租户级的配置

租户级的配置	说明
用户管理	支持 AD 用户，LDAP 用户，NIS 用户的管理。
QOS 配置	支持租户级的 SmartQos 配置。
网络管理	支持租户级子网配置。
数据加密	支持租户级的 SmartEncryption 配置。
审计日志	支持租户级的 SmartAuditlog 配置。
元数据检索	支持租户级的 SmartIndexing 配置。

4.2 配额（SmartQuota）

配额是一种限制或者统计所使用的存储空间和文件数量的技术。HengShan Stor 系列非结构化服务支持 SmartQuota 配额特性，可以应用于帐户、Namespace 或者 DTree（Directory Tree），管理员通过配置合适的配额策略，限制帐户、Namespace 或 DTree 中资源的使用量，并在资源使用量达到配置的阈值级别时发送告警通知。

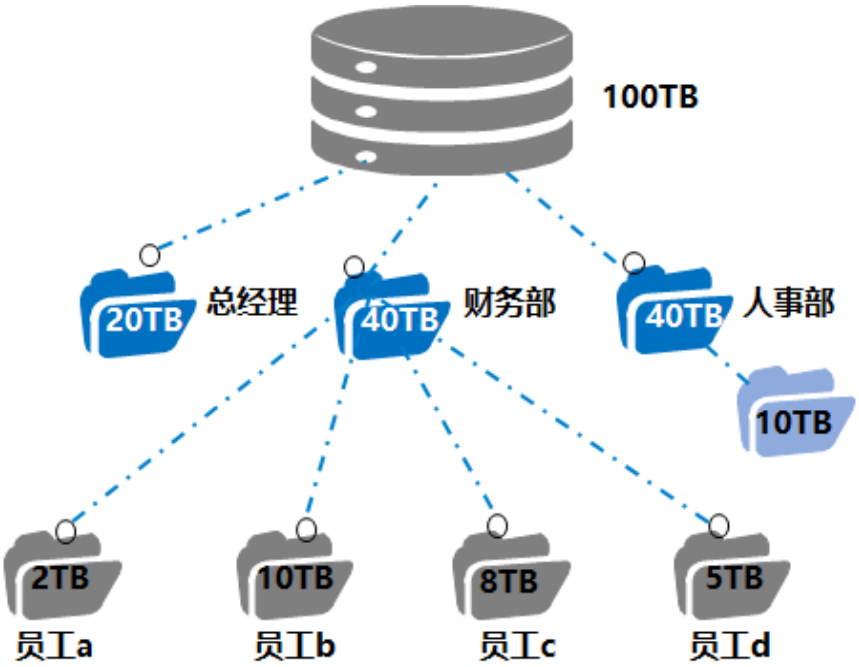
图4-3 DTree 概念示意



DTree 可以在包括文件系统根目录在内的任意目录下创建，但 DTree 间不能嵌套。

SmartQuota 提供简单易用的配置方式，使管理员能合理地为帐户、用户、用户组、DTree 分配存储资源，避免因某个帐户、用户、用户组、DTree 的过度使用造成其他用户无法正常工作，甚至影响系统运行，避免由于存储资源使用的失控，可能造成的系统崩溃，提高了系统的安全性。

图4-4 配额示意图



应用层下来的文件读写创建请求，会先经过 quota 模块，分析出本次写操作受影响的配额配置项，逐项地进行配额项的门限值比较，根据门限值允许或禁止此次文件操作。如果受影响的所有配额控制项允许，写请求才能继续进行，同时更新配额的统计值，一旦写操作尝试超过配额限制则阻止本次写入。

配额类型包括帐户配额、用户配额、用户组配额和目录配额，指定了应用配额限制的帐户、用户、用户组或者 DTree。

- 用户配额

  可以对特定命名空间或者 DTree 应用用户配额；
- 用户组配额

  可以对特定命名空间或者 DTree 应用用户组配额；
- 目录配额

  应用于 DTree 或命名空间的根，对所有用户限制配额使用，命名空间和 DTree 的目录配额可以嵌套配置，对于形成嵌套关系的配额配置，系统按照相关配额项都不超越限制的原则限制用户资源。
- 帐户配额

  应用于帐户，用于控制该帐户下所有命名空间使用量的总和。

配额阈值分为硬阈值、软阈值、建议阈值三种类型

表4-2 配额阈值说明

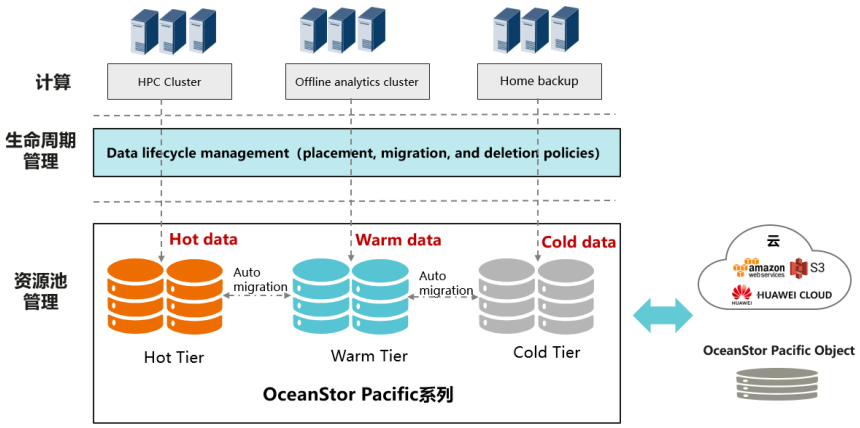
阈值类型	说明
建议阈值	当存储配额到达建议阈值时，不会限制数据写入，仅上报告警信息。
软性阈值	当存储配额到达软性阈值时，上报告警信息，在宽限时间内允许数据继续写入，超期后立即限制数据写入，并再次上报告警信息。 配置软性阈值时，可同时配置宽限时间。 宽限时间未配置的情况下，只上报告警信息，不限制新增数据写入。
硬性阈值	当存储配额达到硬性阈值时，立即限制数据写入，同时上报告警信息。

4.3 分级存储（SmartTier）

对企业的业务应用来讲，并不是所有的数据都具有相同的使用价值。随着时间的推移，有些数据被频繁访问，而有些数据很少被访问，有些数据甚至在最近几年内都没有被访问到。经过科学的统计和分析发现，数据信息的使用价值是有其生命周期规律可循的。通常新生成的信息会经常被访问，使用价值高。随着时间的推移，这些新生信息的使用频率不断下降，直到多年不被访问。这种信息的使用价值也将逐年较低。这些大量的低使用价值数据既占用了高性能、高可靠的宝贵的系统资源、严重影响性能，又占用了大量存储空间，但往往这些数据又由于政策法规、数据仓库建设等原因不能删除。如何解决这些不常用的数据的保存问题，是目前企业面临的数据管理问题之一。自动分级存储技术通过对数据进行搬迁，将数据存放到合适的存储空间中，很好的解决了上述客户遇到的问题。

HengShan Stor 系列非结构化服务提供的 SmartTier 分级特性，允许将同一个存储池内的不同类型物理节点划分成不同的硬盘池。硬盘池是指具有相同特征（物理类型/访问性能）的节点的集合。同时，也允许对接资源池外的支持对象协议的异构设备（蓝光/云/HengShan Stor）。SmartTier 允许用户基于文件池策略定义工作流中数据的价值，将高价值的文件放置在高可用性、高性能的存储设备上，低价值的文件放置在成本较低的、性能和可用性规格较低的设备上。

图4-5 分级示意图



如上图所示的 Tier，是指在 HengShan Stor 存储池内，具有相同特点/能力的硬盘池的集合；在 HengShan Stor 存储池外，对接异构设备。HengShan Stor 的存储空间被分为热、温、冷三个等级，每个硬盘池都有对应一种分级等级，如 SSD 节点硬盘池为“热”，SAS 节点硬盘池为“温”，SATA 或 NL-SAS 节点硬盘池为“冷”。每个等级可包括若干个硬盘池。同一存储等级内部多个硬盘池间系统自动实现负载均衡，其中包括压力均衡和容量均衡。HengShan Stor 对接异构设备，作为成本更低的存储池，存储长时间不用的归档数据。

SmartTier 允许管理员设置基于文件属性的文件池策略，使得数据可以根据存储策略在不同的硬盘池(硬盘池是 EC 数据分布的逻辑资源池，还是 EC 数据故障隔离域)或异构设备之间自动迁移，从而在保证高性能的同时，更合理的使用存储空间。也允许用户手动规划自己的业务，根据自己的业务特点进行分级，把某些重要的目录，文件存放在性能较高的分级上。这些策略包括文件名、文件大小、创建时间、修改时间、访问时间、状态修改时间、UID/GID。针对对象还有一些独有属性，如前缀匹配，对象 tag 匹配，当前版本过期时间匹配，历史版本过期时间匹配。

SmartTier 支持智能分级，该特性内置 ML 学习模块，支持 workload 热度、迁移时间、磁盘水位、Tier UP/Down、文件/目录、文件 size、各层容量共 7 种自动分级维度，无需人工干预，管理员可以无需设置具体分级条件，实现数据的自动智能分级流动。

SmartTier 的分级策略分为：放置策略、迁移策略和删除策略。

- **放置策略：**

  用于确定文件数据初始放置位置，该类策略条件包括：FS/DTree、文件名、UID/GID。如果未匹配到任何策略，根据系统默认的放置策略决定放置位置。只支持将数据放置在 HengShan Stor 存储池。

- **迁移策略：**
  - **周期性迁移策略：**用于确定文件数据搬迁的目标位置。该类策略条件包括：FS/DTree、文件名、文件大小、创建时间、修改时间、访问时间、状态修改时间、UID/GID。系统自动按照策略将数据搬迁到指定等级的硬盘池中。数据搬迁的时间可配置，避开业务的高峰期，默认在每日凌晨零时执行。可以迁移到内部硬盘池或异构设备。
  - **一次性迁移策略：**用于确定文件数据搬迁的目标位置。该类策略条件包括：文件名、文件大小、创建时间、修改时间、访问时间、状态修改时间、UID/GID。系统自动按照策略将数据搬迁到指定等级的硬盘池中。顾名思

义，该搬迁策略只会在配置后执行一次，只有执行当时符合策略的文件数据才会搬迁。可以迁移到内部硬盘池或异构设备。可以从异构设备取回数据到 HengShan Stor 存储池。

- **删除策略：**
  - **周期性删除策略：**周期性删除指定的文件。该类策略条件包括：文件名、文件大小、创建时间、修改时间、访问时间、状态修改时间、UID/GID。对于开启 S3 服务的命名空间，还可以指定前缀、标签、过期策略、非当前版本过期天数等条件。系统自动按照删除策略，直接删除文件，无需用户干预。
  - **一次性删除策略：**用于用户删除指定的文件。与周期删除策略不同，该任务只会执行一次。

说明

硬盘池中的节点应为同一物理类型的节点，否则该硬盘池的性能将会受到影响。系统会统计硬盘池硬盘空间的使用情况，如果占用的硬盘空间超过系统设置的高水位，则不允许新的文件存放/搬迁到该硬盘池。

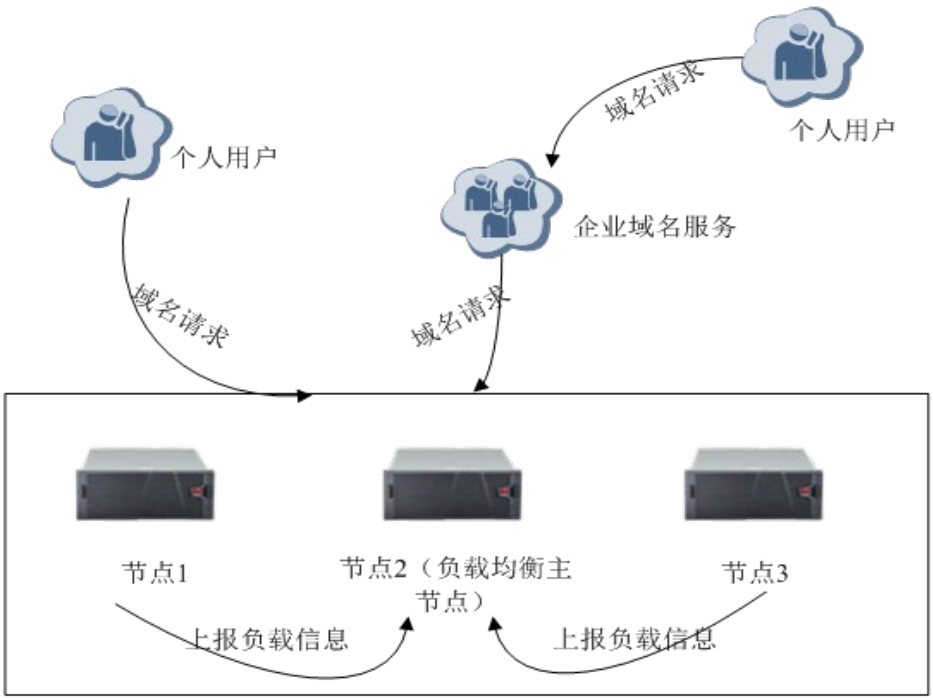
系统最多可以配置 16 个硬盘池，每个硬盘池至少包括 3 个物理节点，否则该硬盘池的状态为“不可用”。当系统中只有一个硬盘池时不建议使用分级存储功能。

异构设备只支持提供标准 AWS S3 协议的存储设备。

4.4 负载均衡（SmartEqualizer）

SmartEqualizer 为百信分布式存储文件服务提供了负载均衡特性，SmartEqualizer 周期性的采集存储系统各个节点的 CPU 利用率、网络吞吐量等信息，提供一种可配置负载策略的负载均衡服务。

图4-6 负载均衡服务示意图



HengShan Stor 系列负载均衡服务是一种一主多备的服务。在存储集群形成初期，各个节点通过 Paxos 算法协商选出一个节点为 SmartEqualizer 的主节点。在存储集群任意时刻，有且仅有一个 SmartEqualizer 的主节点(主节点故障时，系统会自动再选一个新主)。各个节点周期性采集当前的节点负载信息，节点负载信息包括节点 CPU 核数、CPU 主频、内存大小、网络网卡信息、当前 CPU 利用率、当前内存利用率、当前网络吞吐量、当前 NAS 客户端连接数等信息。各个节点将采集好的负载信息统一发往 SmartEqualizer 的主节点。主节点将收集好的负载信息作为负载均衡的依据，当存储集群收到新的 NAS 挂载请求时，则根据用户配置的负载均衡策略选择合适的节点处理 NAS 挂载请求。

存储系统对外提供统一的域名访问，而域名查询请求服务被集成在了 SmartEqualizer 服务之内。在客户端进行域名查询请求的时刻，SmartEqualizer 基于当前所配置的负载策略进行负载计算，返回合适的节点 IP 给客户端，供客户端接入访问存储系统。

HengShan Stor 系列负载均衡服务目前支持如下的负载均衡策略，可以供用户按照实际的环境进行配置。

- 轮询方式（默认策略）：按顺序依次选择节点处理客户端连接请求。
- CPU 使用率：选择 CPU 使用率最低的节点处理客户端连接请求。
- 节点连接数：选择文件服务连接数量最低的节点处理客户端连接请求。
- 节点吞吐量：选择网络吞吐量最低的节点处理客户端连接请求。

如果客户端没有 DNS 服务器，那么客户端可以配置本机的 DNS 服务器地址为存储系统的 DNS IP，直接访问存储系统的 DNS 服务；如果客户端有自己的 DNS 服务器，那么可以在该服务器上配置域名转发功能，将访问的域名请求转发到存储系统的 DNS 服务。

说明

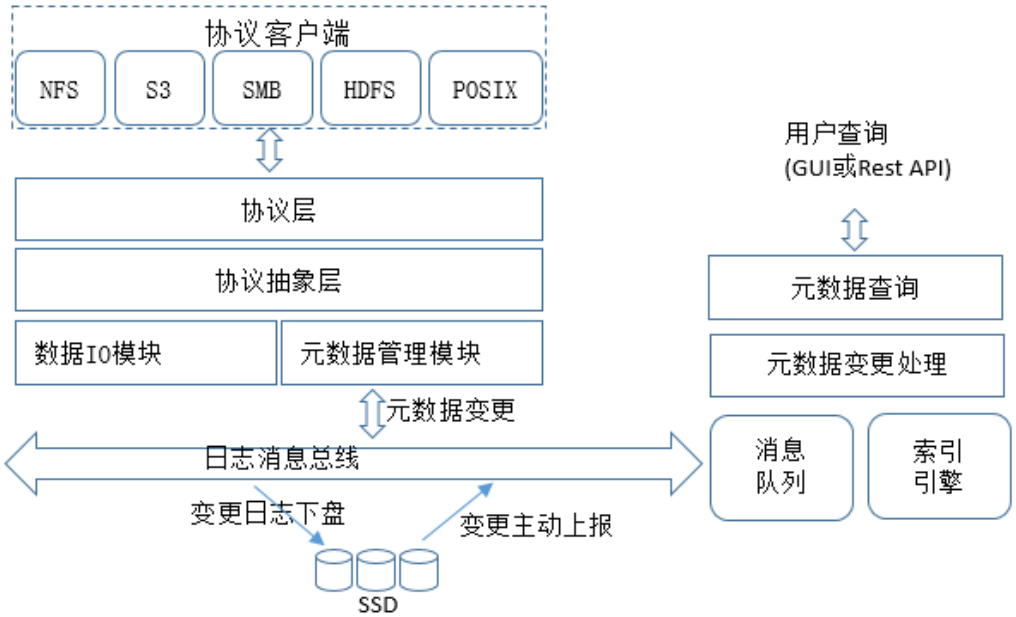
SmartEqualizer 当前版本只给 HDFS 服务、NAS 服务提供域名服务。对象服务有单独的 DNS 负责域名管理。

4.5 元数据检索（SmartIndexing）

随着信息技术的不断发展，全球数据不断增长，人们对于海量数据的管理也越来越重视。海量数据场景，用户需要基于元数据才能对海量数据进行管理，例如获取后缀名为 jpeg 的所有图片文件列表、获取文件大小大于 10GBytes 的文件列表、获取给定日期之前创建的文件列表，在用户能够快速获取符合条件的文件或是对象列表之后才能高效管理对应的数据。传统 Linux 上通过 find 命令查找目标文件的过程即为简单的元数据检索，但这种方式在文件多、目录层次深或复杂查询条件的场景下，效率差不能满足客户需求。

HengShan Stor 系列非结构化服务提供了基于非结构化存储池的元数据检索功能，支持以命名空间(或文件系统)为粒度开启或是关闭元数据检索功能。对于开启了元数据检索的文件系统，在前台业务 IO 变更文件、对象或是目录的元数据之后，异步、主动推送变更的元数据到检索系统。有别于传统存储产品采用的周期性元数据扫描，在避免对生产系统带来幅性能影响的同时，满足用户对海量数据的快速检索。

图4-7 元数据检索原理示意



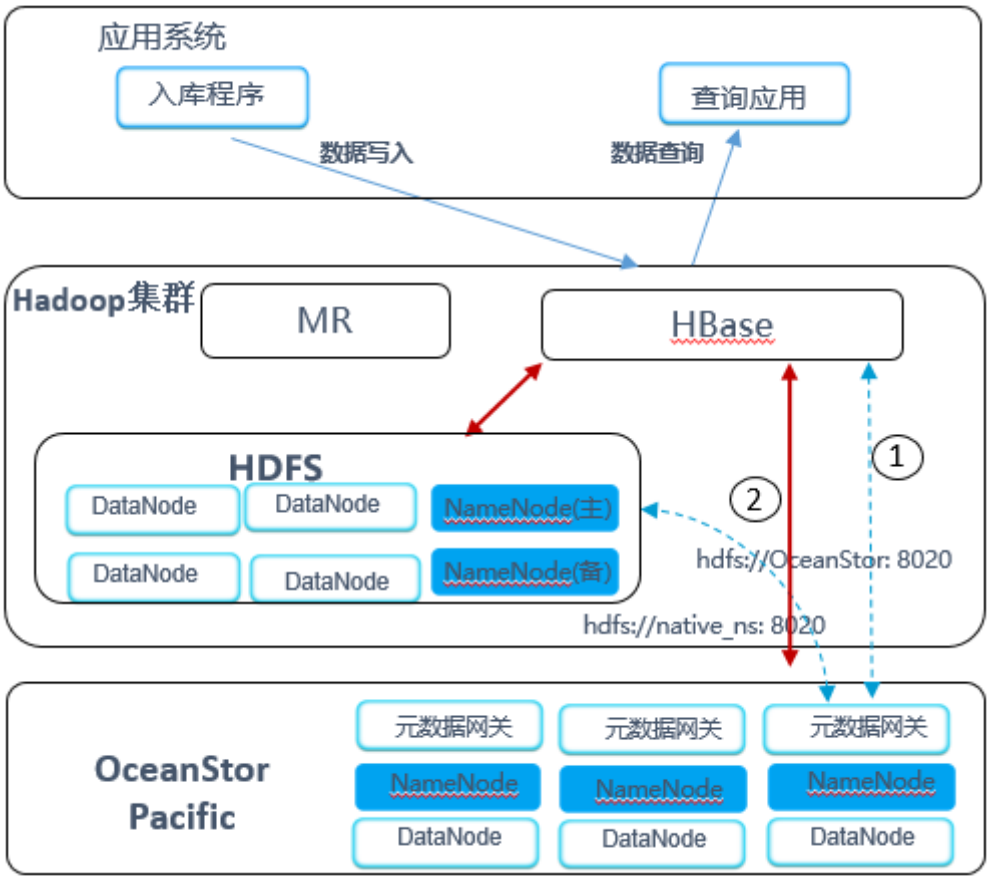
来自不同协议的元数据变更操作（例如创建文件，修改文件的权限、用户和组，修改文件数据）都将转发到对应的文件或是目录的归属节点，并由该归属节点上的元数据管理模块进行处理。在进行元数据变更持久化的同时，HengShan Stor 系列非结构化服务存储引入快速事务处理同时将元数据变更日志和变更之后的元数据写入加速磁盘介质。事务的引入可以确保要么变更日志和元数据变更都成功，要么都失败，从而保证

了元数据的一致性。落盘之后的变更日志将通过日志消息总线主动上报给元数据检索的“消息队列”，最终由“元数据变更处理”模块读取并更新 “索引引擎”。

4.6 智能纳管（SmartTakeover）

HengShan Stor 系列大数据服务提供了 SmartTakeover 元数据网关方案，对外提供全局 Namespace，实现在不迁移数据的前提下对其它 HDFS Namespace 进行纳管。在用户现有的大数据平台下可以实现无缝接入，对上层业务无影响。

图4-8 大数据元数据网关原理示意



修改配置：

1. Hadoop 集群中，将 HBase/Hive/Spark/Presto/openLooKeng 等组件的数据访问路径更改到 HengShan Stor 系列大数据服务的 NameNode。
2. 给元数据网关增加原 HDFS 的 NameNode 路径。

数据访问路径说明：

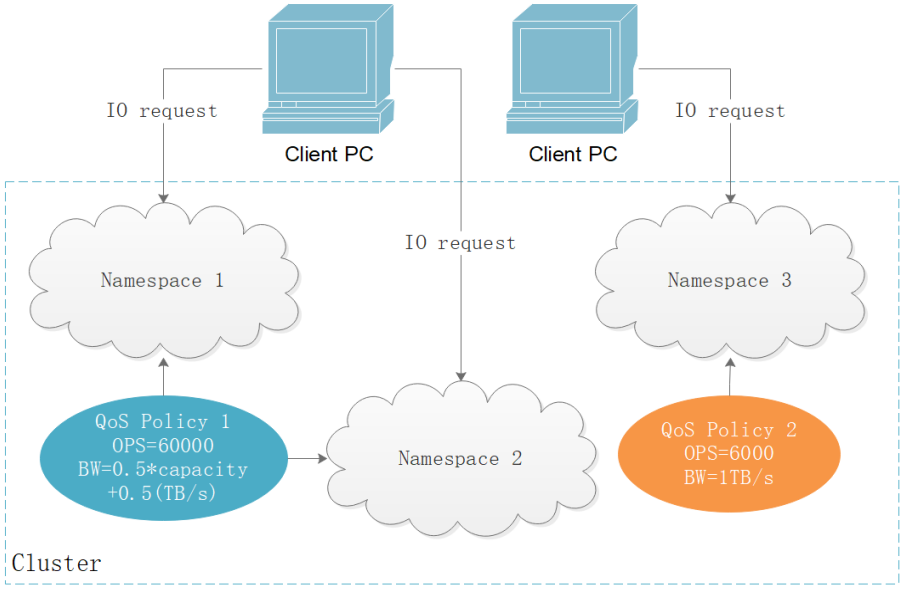
1. HBase/Hive/Spark/Presto/openLooKeng 等组件通过路径①元数据网关查询数据的位置，如果不在大数据存储池，通过路径②获得原 HDFS 上的数据位置，将以上位置信息返回给这些组件。
2. 这些组件根据位置路径获取数据。

4.7 服务质量（SmartQoS）

4.7.1 非结构化数据存储服务

HengShan Stor 系列非结构化服务的 SmartQoS 技术支持以租户/Namespace/client/用户粒度配置 QoS 策略，Namespace 级 QoS 可以基于已用容量或预设上限(带宽与 IOPS)配置，灵活满足不同场景需求。 如下图所示，命名空间 1 和命名空间 2 关联 QoS 策略 1，这两个命名空间的 OPS 各自被限制到 60000 以下，带宽上限根据各自的使用容量计算。命名空间 3 关联 QoS 策略 2，该文件系统的 OPS 将被限制在 6000 以下，带宽将被限制在 1TB/s 以内。对象服务下命名空间的读、写操作 OPS 和带宽上限支持分别单独设置。

图4-9 基于 Namespace 的 QoS 策略示意

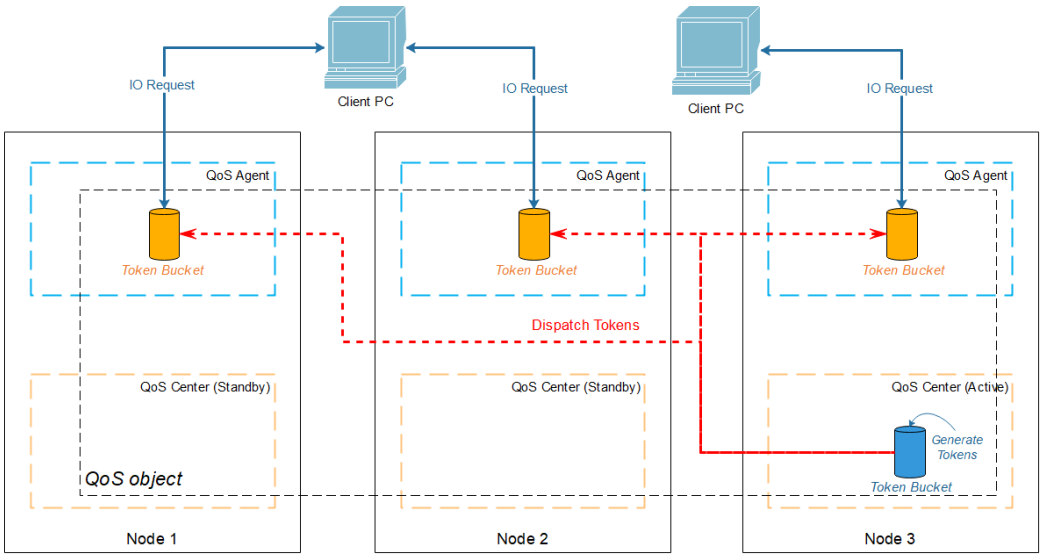


租户级 QoS 支持 OPS 和带宽的上限控制，这个租户内所有的 Namespace 共享 QoS。租户级 QoS 控制与命名空间级 QoS 控制采取叠加式控制原则，即任意命名空间的 OPS 上限和带宽上限需要同时满足该命名空间和所属租户配置的 QoS 策略规则。

客户端级的 QoS 控制，支持以客户端 IP 为粒度进行 OPS 和带宽的上限控制，对于 DPC（分布式并行客户端），该 IP 即为 DPC 的管理 IP，对于通用客户端，该 IP 指的是用户可自由配置的业务 IP。同样的，根据叠加式控制原则，除自身配置的 QoS 策略外，任意客户端 IP 下发业务的 OPS 与带宽上限还需要同时满足发往的命名空间及其所属租户的 QoS 策略规则。对象服务客户端业务 IP 下发的读、写操作 OPS 和带宽上限支持分别单独设置。

用户级 QoS 支持以用户粒度设置带宽和 OPS 上限。SMB 协议和对象服务支持用户 QoS，但是需要配置 Windows 用户和 S3 用户到 UNIX 用户的映射。同样的，根据叠加式控制原则，除用户自身配置的 QoS 策略外，还需要同时满足用户所在命名空间及相关租户和客户端的 QoS 策略规则。

图4-10 非结构化服务 QoS 技术原理



在分布式架构下，集群中所有节点安装 QoS Agent 和 QoS Center 模块。部分节点被选出作为 center 主节点，其 QoS Center 模块被置于活跃状态（Active），其余节点的 QoS Center 置于闲置状态（Standby），作为 center 主节点的备用。center 主节点的 QoS Center 模块分组管理各节点的 QoS Agent 模块。center 主节点 QoS Center 模块根据对应流控上限和并发资源上限对 QoS 对象的令牌进行总量管理。客户端从任意节点接入，发往某一个 Namespace 的 IO 都将通过 QoS Agent 模块向与之关联的 QoS 对象申请令牌，成功获取令牌方可完成 IO 请求。

4.7.2 结构化数据存储服务

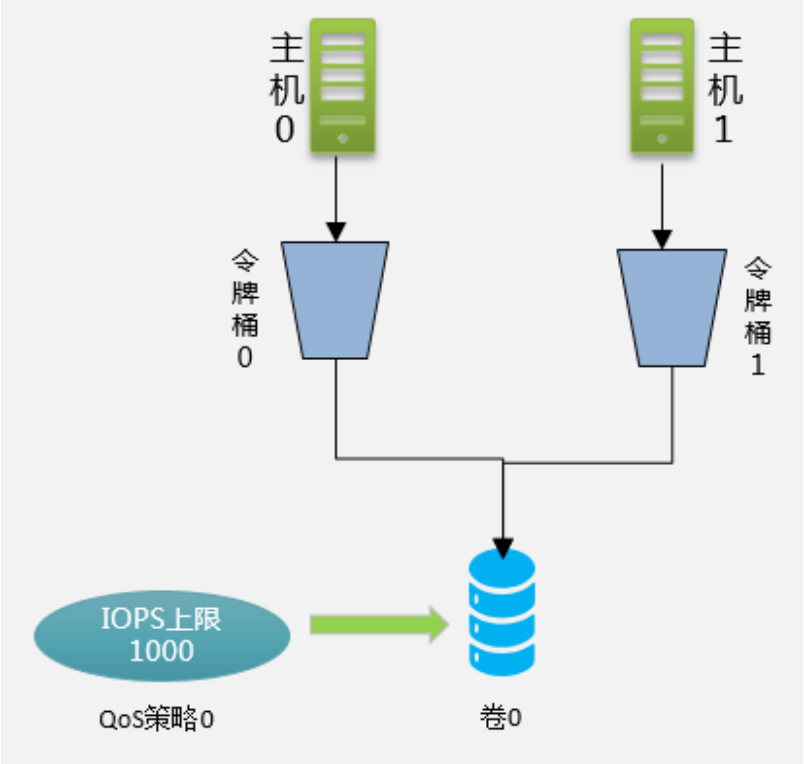
HengShan Stor 系列结构化服务的 SmartQoS 支持基于卷或池设置性能目标，支持按带宽和 IOPS 进行配置最大性能目标，可分别限定读写总性能、读性能和写性能；定时策略可以基于业务繁忙情况设置并发生作用，避免各种 IO 风暴影响生产业务。

HengShan Stor 系列结构化服务的 QoS 功能基于策略分布式流控参数协调及流控管理两部分实现，针对用户设置的性能控制目标（IOPS、带宽）进行流量限制，通过 IO 流控机制，限制某些业务由于流量过大而影响其它业务，并允许在一定时间内使用超出基准性能的配额。下面简单介绍 HengShan Stor 系列的 QoS 流量控制技术原理。

- 基于负反馈的自适应调整算法

  HengShan Stor 系列产品中存在一个卷挂载到多个 VBS 节点的场景，在此场景下为了限制该卷上的业务对系统资源的消耗。需要限制该卷在系统中的总性能，需要一个分布式流控参数协调的过程。

图4-11 基于负反馈的自适应算法

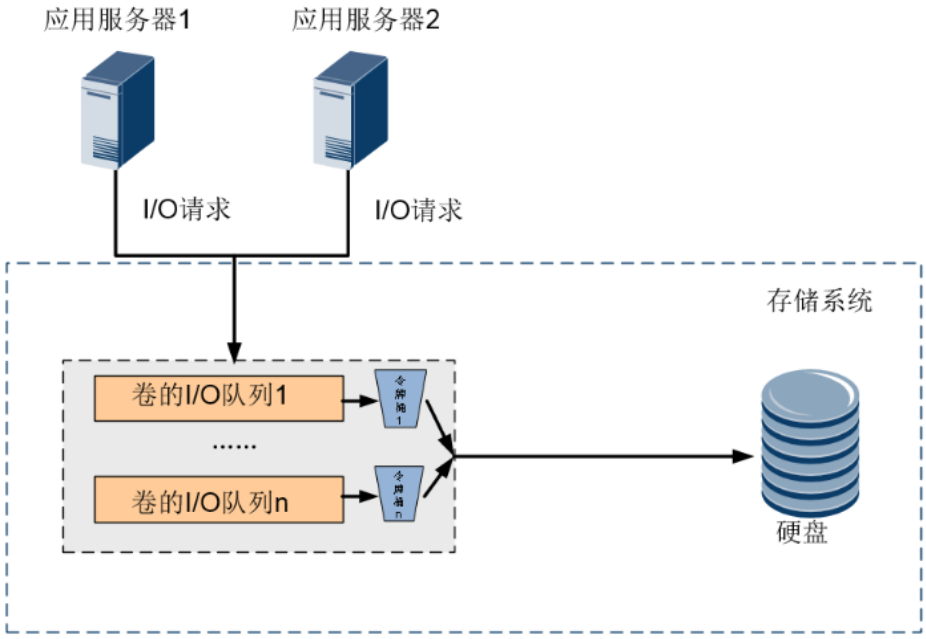


以上图举例，系统为初始状态，卷 0 的 IOPS 上限为 1000，QoS 框架会采集主机 0 和主机 1 对卷 0 的压力情况，自适应调整卷 0 在各个节点令牌桶的令牌速度，以达成整体 IOPS 被限制到 1000 的目标。

- 基于卷的 IO 流控管理算法

  QoS 流控管理通过对卷的 IO 队列管理、令牌分发和出队控制三部分实现。 当用户为某个 QoS 策略设置了性能上限目标，那么通过上文讲述的分布式流控参数协调，会确定每个 VBS 节点上的流控参数目标，进而这个性能上限会被转化成对应的令牌。在存储系统中，如果用户要限制的流量类型是 IOPS，那么一个 IO 会消耗一个令牌；如果设定的性能目标是带宽，那么一个字节对应一个令牌。基于卷的 IO 队列管理通过令牌机制实现存储资源的分配，某个卷的 IO 队列所拥有的令牌数越多，系统分配给这个卷的 IO 资源也越多。

图4-12 基于卷的 IO 流控管理



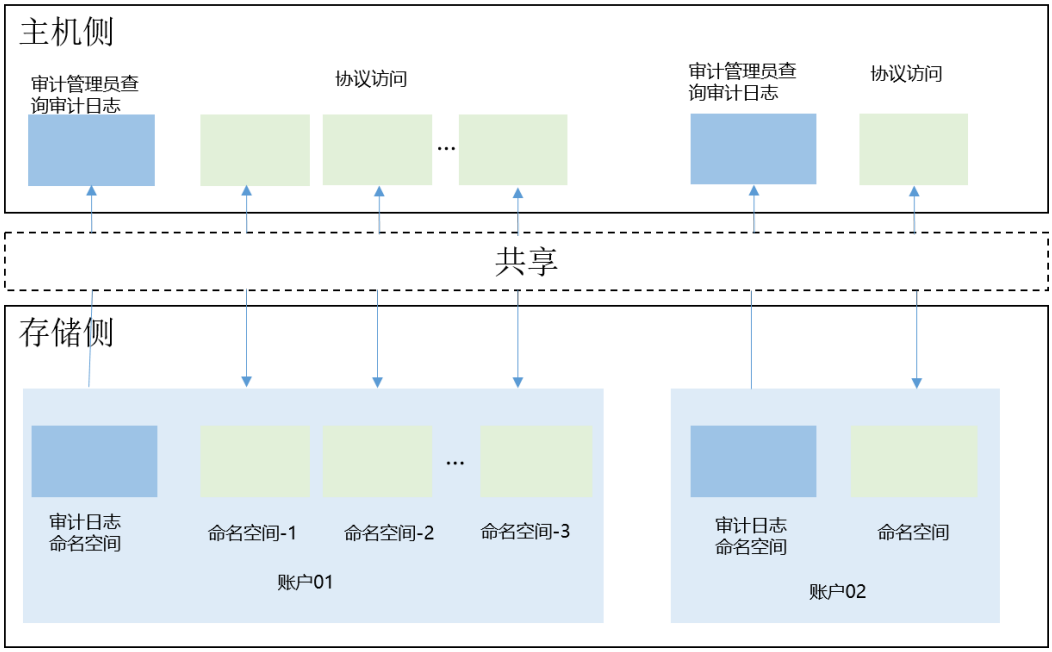
如图所示，应用程序的 IO 进入存储系统后，首先进入各自的 IO 队列。QoS 框架周期性地处理队列中的 IO 等待，将队列首元素从队列中摘除，尝试到令牌桶中获取令牌。如果令牌桶内剩余令牌满足队首元素的令牌请求，则将该元素交由存储系统其它模块继续处理，并且 QoS 框架继续处理一下个队首元素；如果桶内剩余令牌不能满足队首元素的令牌请求，将其重新挂入队列首部，QoS 框架结束本轮 IO 出队动作。

4.8 审计日志（SmartAuditlog）

审计日志特性原理

HengShan Stor 系列非结构化服务支持多协议互通访问的 Namespace，在多协议支持下，NFS，SMB，对象，HDFS 用户可以访问给定的文件。HengShan Stor 系列审计特性，通过在 Namespace 上配置审计开关和操作字，记录发生在该文件系统上的文件或目录操作行为，支持审计日志记录到审计日志命名空间中，日志文件格式支持 EVTX/XML/TXT 格式。

图4-13 审计日志特性图



1. 分布式存储节点协议层接收来自协议客户端（NFS，SMB，对象，HDFS 和 FTP）的文件访问操作时，协议层完成审计日志记录判断及日志信息组装，如果记录日志，则协议组装的审计日志信息发送给审计日志模块；

2. 审计日志模块将日志记录写入到临时存储区；

3. 审计日志后台任务收集临时存储区的日志记录，按照帐户的配置要求，以指定格式 EVTX/XML/TXT 文件存储在审计日志命名空间；

4. 完成审计日志格式转换后，删除临时存储区中日志记录；

5. 每个帐户只支持配置一个审计日志命名空间，审计日志命名空间为只读命名空间，通过配置 NFS/CIFS 共享可以访问审计日志命名空间中的日志文件。

审计日志项

HengShan Stor 系列支持的文件访问审计操作包括：“创建”、“删除”、“重命名”、“读”、“写”、“打开”、“关闭”、“列出文件夹”、“获取属性”、“设置属性”、“获取安全属性”、“设置安全属性”、“获取扩展属性”、“设置扩展属性”、“拼接文件”。

HengShan Stor 系列支持的用户审计操作包括：“登入”和“登出”。

表4-3 审计日志支持的操作说明

事件名	事件名	描述	举例
创建	Create	A file or directory was created.	Open(creat flag), mkdir, put a object
重命名	Rename	A file or directory was renamed.	Rename, mv

事件名	事件名	描述	举例
删除	Delete	A file or directory was deleted.	Unlink, rmdir, rm
读	Read	A file was read.	cat
写	Write	A file was written.	cp, hdfs dfs -put
打开	Open	A file was opened for reading or writing.	cat, cp
关闭	Close	A file was closed.	close, cp
列出文件夹	List folders	A directory was listed.	ls, ll, getListing
获取属性	Obtain properties	A file's or directory's attributes were obtained.	stat
设置属性	Set properties	A file's or directory's attributes were set.	chmod, chown, hdfs dfs -chmod
获取安全属性	Obtain security properties	A file's or directory's security attributes were obtained.	nfsv4acl-nfs4_getfacl, hdfs dfs -getfacl
设置安全属性	Set security properties	A file's or directory's security attributes were set.	nfsv4acl-nfs4_setfacl, modifyacl
获取扩展属性	Obtain extension properties	A file's or directory's extended attributes were obtained.	hdfs dfs -getfattr
设置扩展属性	Set extension properties	A file's or directory's extended attributes were set.	hdfs dfs -setfattr
拼接文件	Concatenate files	Files were concatenated.	concate
登入	Login	User login.	FTP 用户或 CIFS 用户登录共享目录
登出	Logout	User logout.	FTP 用户或 CIFS 用户登出

审计日志记录信息

支持的多协议审计日志记录包含的字段详细信息如下表所示:

表4-4 审计日志关键字段说明

事件信息	事件字段	事件描述
System	Provider Name	事件审计提供者名称

事件信息	事件字段	事件描述
	EventID	事件 ID
	Version	事件版本
	EventName	事件名称
	Level	事件级别
	Source	事件协议类型
	Result	事件结果
	Opcode	事件操作
	Keywords	事件关键字
	TimeCreated System Time	事件发生时间
	Channel	事件发生渠道
	Computer	事件发生计算机
EventData	SubjectIP	事件主体 IP
	SubjectHostName	事件主体主机名
	SubjectUnix	事件主体 UNIX 属性
	SubjectUserSid	事件主体用户 SID
	SubjectUserName	事件主体用户名
	SubjectDomainName	事件主体域名
	SubjectUserIsLocal	事件主体用户是否是本地用户
	ObjectServer	事件客体服务器
	ObjectType	事件客体类型
	ObjectName	事件客体名称
	HandleID	事件客体标识
	OldDirHandle	事件旧目录标识 **说明** 重命名操作事件时该字段有效。
	NewDirHandle	事件新目录标识 **说明** 重命名操作事件时该字段有效。

事件信息	事件字段	事件描述
	OldPath	事件客体旧路径 **说明** 重命名操作事件时该字段有效。
	NewPath	事件客体新路径 **说明** 重命名操作事件时该字段有效。
	AccessList	访问权限列表 **说明** 创建、打开、软链接操作事件时该字段有效。
	AccessMask	访问掩码 **说明** 创建、打开、软链接操作事件时该字段有效。
	DesiredAccess	所需访问权限 **说明** 创建、打开、软链接操作事件时该字段有效。
	Attributes	属性 **说明** 创建、打开、重命名、软链接操作事件时该字段有效。
	LinkName	链接名 **说明** 创建硬链接时该字段有效。
	InformationRequested	请求信息列表 **说明** 列出文件夹、获取属性、获取安全属性操作事件时该字段有效。
	InformationSet	信息列表 **说明** 设置属性操作事件时该字段有效。
	WriteOffset	写偏移 **说明** 写操作事件时该字段有效。
	WriteCount	写长度 **说明** 写操作事件时该字段有效。

事件信息	事件字段	事件描述
	ReadOffset	读偏移 **说明** 读操作事件时该字段有效。
	ReadCount	读长度 **说明** 读操作事件时该字段有效。
	oldSD	旧安全描述 **说明** 设置安全属性操作事件时该字段有效。
	newSD	新安全描述 **说明** 设置安全属性操作事件时该字段有效。
	SearchPattern	搜索模式 **说明** 列出文件夹操作事件时该字段有效。
	SearchFilter	搜索过滤 **说明** 列出文件夹操作事件时该字段有效。
	NameList	参与本次 CONCAT 操作的源文件的名字列表 **说明** NameList 长度有限制，当 CONCAT 操作的源文件的名字列表总长度超出范围时，超出部分的源文件的名字不会出现在 NameList 中。
	FileCountInList	当前名字列表中记录的文件个数 **说明** CONCAT 操作事件时该字段有效。

4.9 数据加密（SmartEncryption）

HengShan Stor 系列支持基于软件/硬件结合加密和加密盘两种静态数据加密能力，保护客户存储到系统中的数据不被泄露。

- 软硬结合加密

  支持结构化和非结构化数据的软件加密，支持 XTS-AES-128、XTS-AES-256 和国密 SM4-XTS 三种加密算法。理论上，XTS-AES-256 算法相较于 XTS-AES-128 算法对性能的影响不超过 10%。软硬结合加密通过内置密管和内置加密引擎完成静态数据加密，通过两层密钥管理提供帐户级安全。数据写入存储系统后，先通过

软硬结合的方式进行加密，再将密文写入到盘中保存；读数据时，先从盘中读取密文，再将密文通过软硬结合方式解密成明文，然后再提供给用户。

- 加密盘

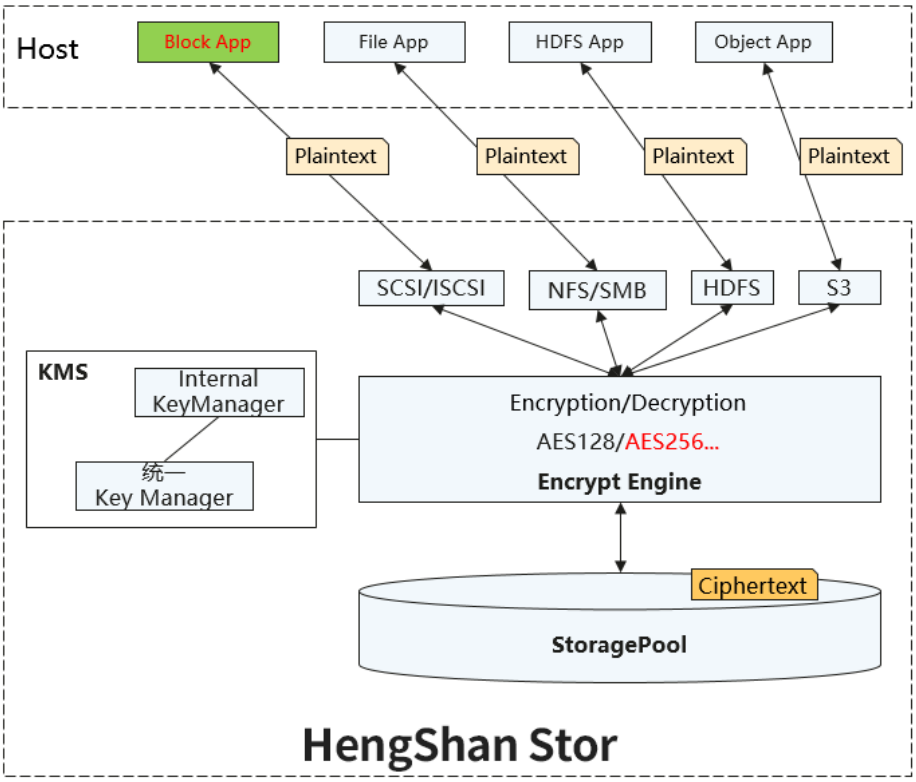
  支持结构化和非结构化服务通过配置 SED（加密盘），实现数据在盘中加密。SED 具备两层安全保护，分别使用 AK(authentication key)和 DEK(data encryption key)两个安全密钥。

4.9.1 软硬结合加密

软硬结合的加密特性，通过内置密管和内置加密引擎完成静态数据加密，通过两层秘钥管理提供租户级安全性，从而保证用户的数据加密持久化和秘钥管理安全性。内置加密引擎，为数据加解密提供基础能力，支持 AES-XTS 算法，在使用软件加解密的时候支持 ASE-NI 指令加速提升性能。两层秘钥管理是指以租户粒度分配 AK（Authentication Key），租户下的 Namespace 数据加密使用的 DEK(Data Encrypt Key)，DEK 用相应租户的 AK 进行保护。

如果想开启一个 Namespace 或卷的加密，需要在对应租户开启加密，开启租户加密后会通过加密通道从内置密管申请到租户身份认证秘钥 AK。然后创建该租户下加密的 Namespace 或卷，会申请一个数据秘钥 DEK，该 DEK 会使用所属租户的 AK 进行加密保护。

图4-14 加密技术原理



如上图所示，对于非结构化服务，加密的数据支持多协议互通访问。如文件应用客户端通过 NFS 或 SMB 协议写入明文数据，然后在 Encrypt Engine 层从 Key Manager 获取

秘钥并进行数据加密，然后持久化到存储池中。当通过 HDFS、S3 等其他协议读取时，Encrypt Engine 首先从存储池中读取密文数据，然后去 Key Manager 获取秘钥，然后解密数据，并把明文返回给协议层，由协议层将明文数据返回给客户端应用。

说明

数据加密开启后不支持再关闭。

国密算法需要单独的 License。

4.9.2 加密盘

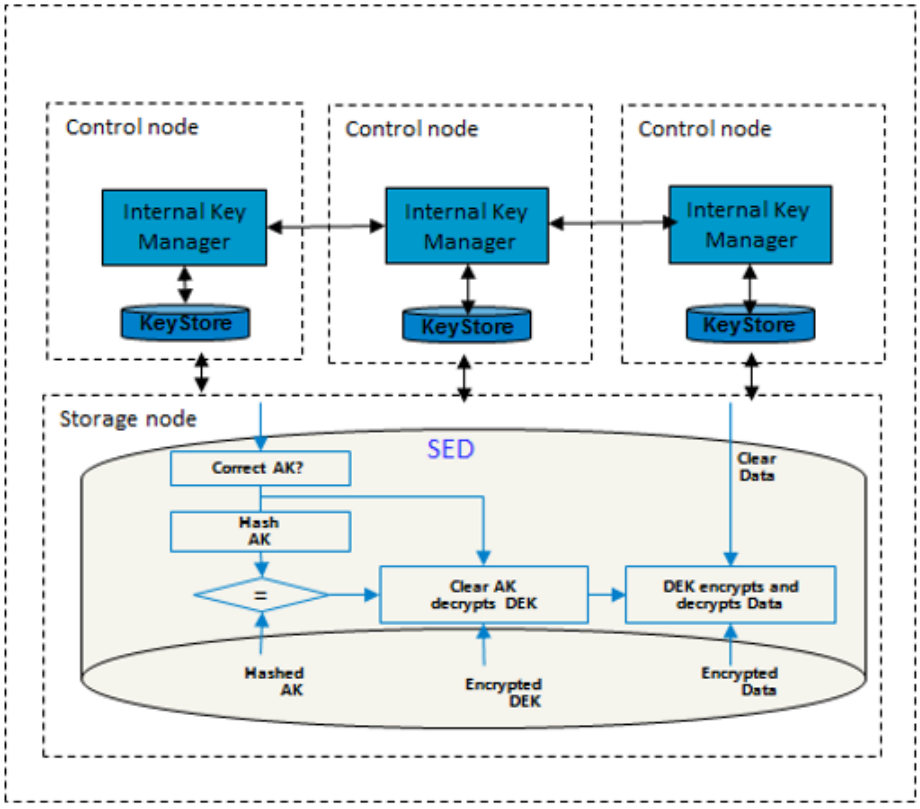
通过加密盘、内置密管和存储系统配合完成静态数据加密，从而保证数据的安全性。

使用内置密管+加密盘来保证数据的安全性，具有如下特点和优势：

- 采用 SED（Self-Encrypting Drive），数据在盘内进行加解密计算，对业务应用处理流程无影响；
- 无上层增值服务影响，由于数据加密在数据下盘后完成，因此不影响上层的数据重删压缩服务；
- 快速数据销毁，通过密钥销毁，达到快速数据销毁；
- 易部署、易配置和易管理的内嵌于存储系统的密钥管理应用（Internal Key Manager）。

HengShan Stor 系列存储系统采用分布式的架构来管理内置密管，多节点的内置密管协同为加密盘提供安全的密钥服务。参见下图：

图4-15 HengShan Stor 系列加密盘加密技术原理



- AK 认证原理：当启用数据加密特性时，存储会打开加密硬盘的 AutoLock 功能，使用由 Internal Key Manager 分配的 AK 对加密盘的接入进行认证，控制对加密硬盘的访问。此时访问已由 SED 的 AutoLock 功能进行保护，只能由存储系统本身访问。硬盘每次接入时，需要存储系统从密管服务器获取硬盘的 AK，如果与硬盘上的 AK 匹配，硬盘就将加密后的 DEK 解密，用于数据加解密。如果 AK 与硬盘上的 AK 不匹配，则任何读写操作都将失败。
- DEK 加密技术原理：当硬盘成功通过 Autolock 认证后，对硬盘进行读写时，硬盘通过自身的加密芯片和内部的数据密钥（Data Entrypt Key）完成写入数据加密和读取数据解密的功能。用户下发写操作时，明文数据通过 AES 加密引擎的 DEK 加密变成加密数据，然后被写入介质。用户下发读操作时，在介质中的加密数据通过 AES 加密引擎的 DEK 解密，被还原成明文数据取出。DEK 本身无法获取，意味着硬盘被拆除后，通过直接读取的方式无法还原原始信息。

说明

在实际部署场景下会选择若干个存储节点部署 Control node 角色服务，所有存储节点均会部署 Storage node 角色服务。

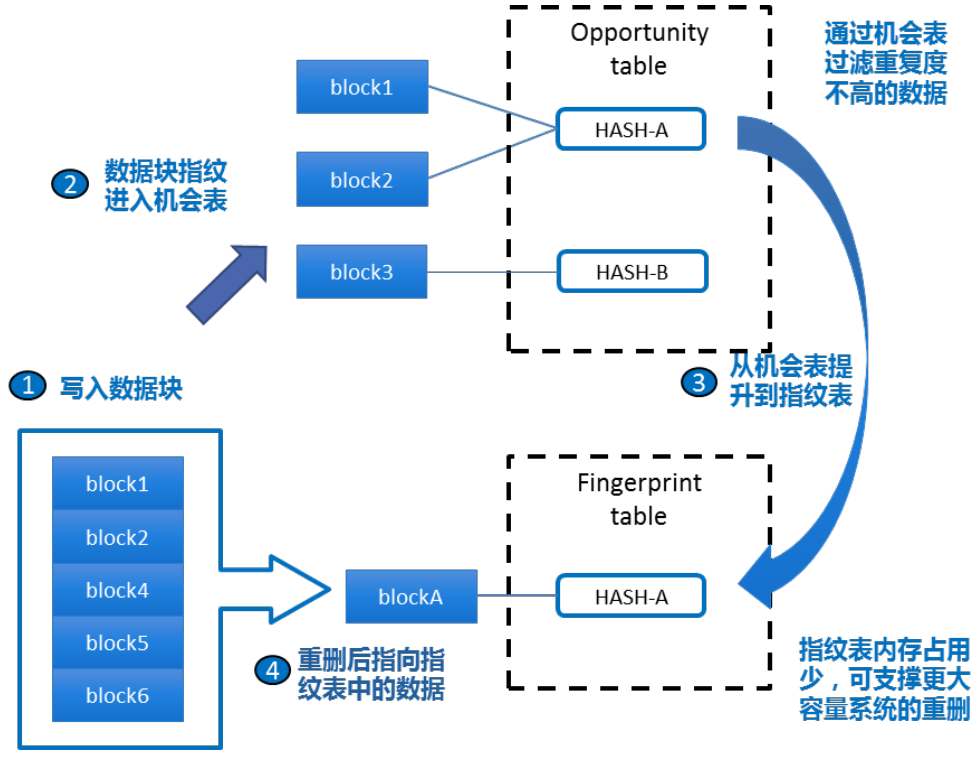
4.10 重删压缩（SmartDedupe&SmartCompression）

为了应对管理数据增加带来的运营成本的增长，各存储设备的厂商都在存储设备中增加了对应的数据缩减手段，如重复数据删除（Deduplication，重删）和数据压缩（Compression，压缩）技术以减少实际需要保存的数据量，从而降低企业的运营成本。HengShan Stor 系列结构化服务采用了自主研发的 SmartDedupe 和 SmartCompression 重删压缩特性来实现数据缩减。

HengShan Stor 系列结构化服务采用了智能的自适应重删技术，以用户需求为导向，在用户数据处理负载较高的情况下，前重删会自动关闭，优先确保性能，由后处理完成数据缩减。在负载较低的情况下自动开启前重删，避免了后处理的读写放大。自适应技术以用户为导向，在用户不感知的情况下根据负载自动切换重删方式，避免了在线重删和后重删两种方案的缺点。

为了获取较好的重删压缩效果，HengShan Stor 系列结构化服务采用了全局重删。同时分布式存储空间巨大，为了减少指纹表的内存空间消耗，引入指纹机会表的机制（如下图）。数据块的指纹先进入机会表计数，只有相同数据块被多次写入达到阈值（默认 3，可配置），方可进入指纹表执行重删。

图4-16 HengShan Stor 系列重删流程

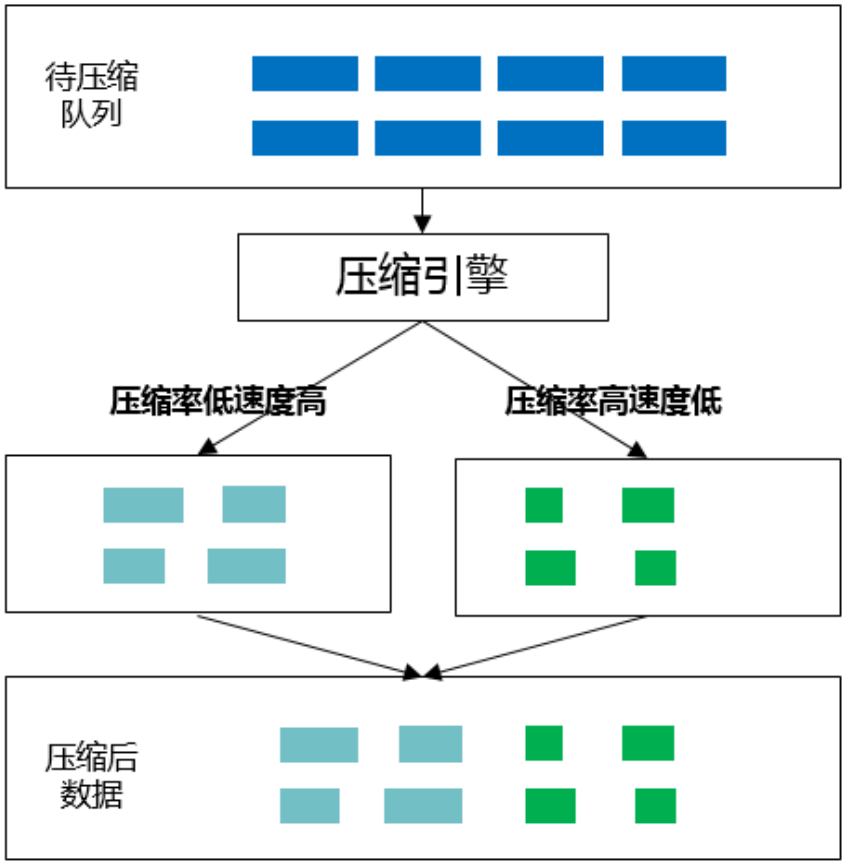


如果压缩未开启，则直接申请该数据块长度存储空间保存数据。开启了压缩的情况下，则会进行压缩。数据块将被压缩引擎压缩，然后以最小 512 字节的粒度进行保存。

HengShan Stor 系列结构化服务压缩引擎采用两种不同压缩算法组合运行，一种是压缩率低速度高的算法，一种是压缩率高但速度低的算法。通过配置两种压缩算法不同的

执行比例，可以得到不同的性能和数据缩减率。在同一个存储池只能选择一个压缩算法。存储池压缩算法的修改不会影响已经压缩的数据，已经压缩的数据在读取时会根据压缩时的算法进行解压。

图4-17 HengShan Stor 系列结构化服务多策略压缩



说明

开启数据加密后，重删压缩效率低。

4.11 卷在线迁移（SmartMove）

SmartMove 特性又称智能跨池在线迁移特性，是业务迁移的关键技术，可以实现存储系统内的业务数据跨池在线迁移。

SmartMove 的目的和受益如表 4-5 所示。

表4-5 SmartMove 的目的和受益

目的和受益	详细说明

目的和受益	详细说明
可靠的业务连续性	支持在线的业务数据迁移，避免了迁移过程中由于业务中断给客户造成的损失。
稳定的数据一致性	进行业务数据迁移的过程中，主机产生的数据变更将及时同步至建立数据迁移关系的两个 LUN 中，保证了迁移完成后数据的一致性，避免遗漏。
便捷的性能可调性	根据业务需求，实现了不同存储池间的数据迁移，完成业务性能的自由调节。

SmartMove 工作原理如下：

SmartMove 特性实现了把源 LUN 的数据完全复制到目标 LUN，并在复制完成后将目标 LUN 数据卷和源 LUN 的数据卷在线交换，最终源 LUN 的数据全部迁移至目标 LUN 的数据卷。

LUN 迁移的实现过程分为两个阶段：

1. 业务数据同步

   存储阵列通过后台任务将源 LUN 数据同步至目标 LUN，数据同步完成之后，目标 LUN 和源 LUN 上的业务数据保持完全一致。

   ① 迁移前，客户需要配置迁移的源 LUN 和目标 LUN。

   ② 迁移开始时，数据由源 LUN 后台复制到目标 LUN。

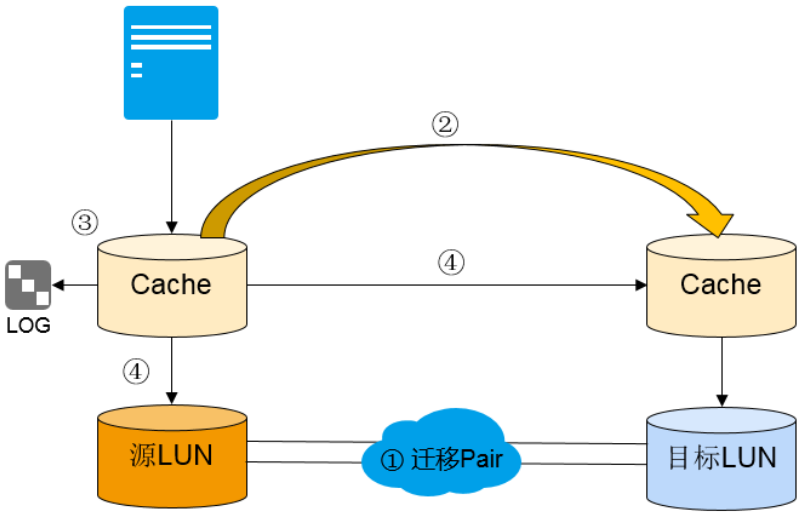
   ③ 主机此时可以继续访问源 LUN。主机写入源 LUN 数据时，将该请求记录日志。日志中只记录地址信息，不记录数据内容。

   ④ 写入的数据同时向源 LUN 和目标 LUN 双写。

   - 等待源 LUN 和目标 LUN 的写处理结果都返回。如果都写成功，清除日志；否则保留日志，进入异常断开状态，后续启动同步时重新复制该日志地址对应的数据块。
   - 返回主机写请求处理结果，以写源 LUN 的处理结果为准。例如：目标 LUN 故障时，写目标 LUN 的 IO 请求的执行结果不会导致源 LUN 上的业务受到影响。

   ⑤ 在数据完全复制到目标 LUN 之前，保持上述双写和日志机制，直到数据复制完成。

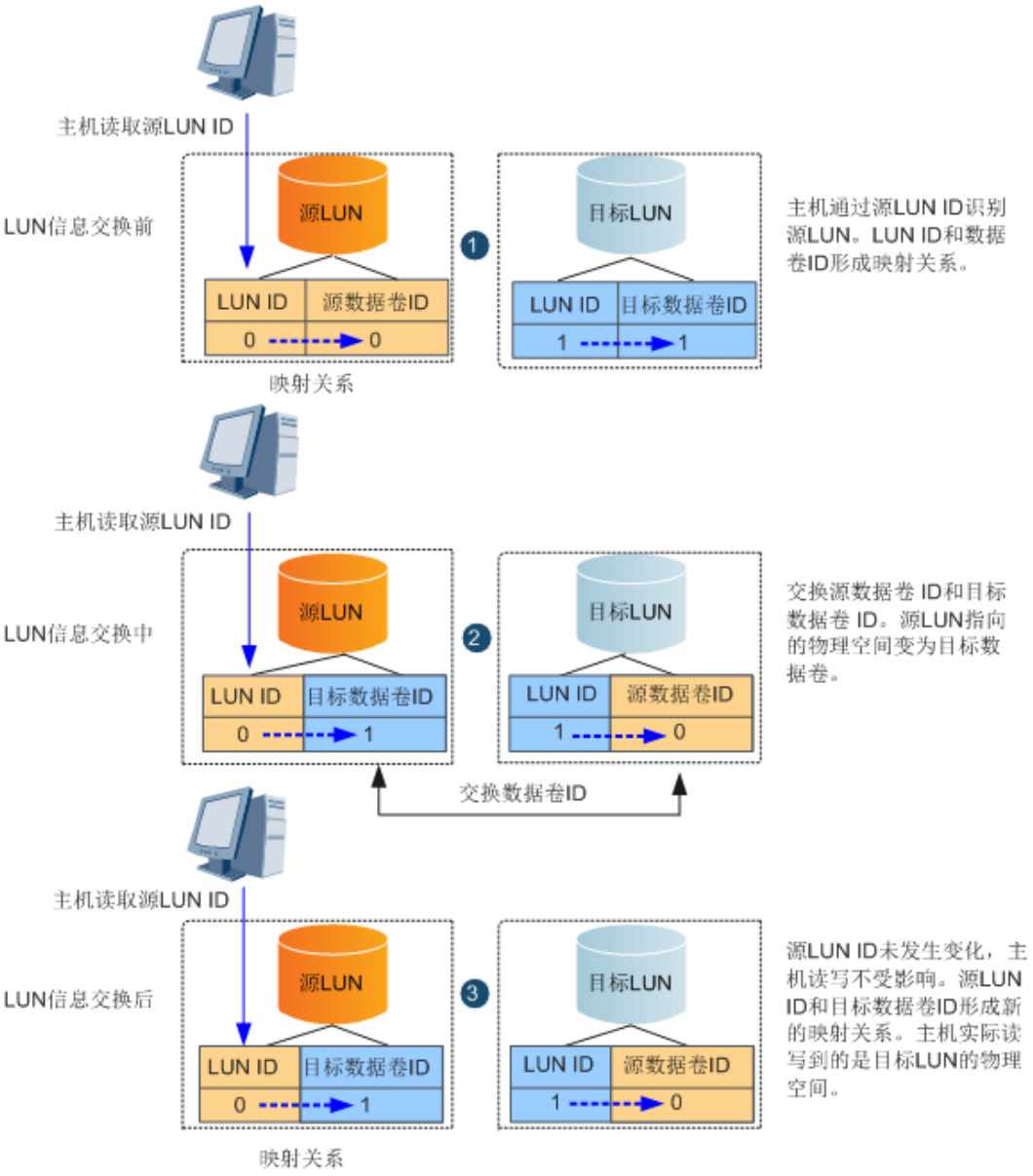
图4-18 业务迁移过程中的变更数据同步



2. LUN 信息交换

LUN 信息交换是目标 LUN 能够顺利地代替源 LUN 来承载业务的前提。通过 LUN 信息交换，可以在主机业务无感知的情况下将源 LUN 的业务迁移到目标 LUN，达到目标 LUN 完全替代源 LUN 来运行业务的目的。LUN 信息交换过程如图 4-19 所示。

图4-19 LUN 信息交换



① 存储系统中，每个 LUN 和对应的数据卷都有属于自己的唯一标识，分别为 LUN ID 和数据卷 ID。其中，源 LUN 是逻辑上的概念，源数据卷是物理上的概念，两者之间形成一一对应的映射关系。在业务运行中，主机通过源 LUN ID 来识别源 LUN，源 LUN 则通过源数据卷 ID 来识别源数据卷。

② LUN 信息交换主要是针对 LUN 和数据卷之间的映射关系，即源 LUN 和目标 LUN 的 LUN ID 保持不变的情况下，将源数据卷和目标数据卷的数据卷 ID 相互交换，这样就形成了源 LUN ID 和目标数据卷 ID、目标 LUN ID 和源数据卷 ID 之间的一一映射关系。另外除了数据卷 ID 交换外，LUN 相关的属性也会同时进行交换，例如：所属存储池 ID、所属硬盘域 ID 等。

③ 在主机未中断业务的情况下，主机与源 LUN 的映射关系一直保持不变。虽然主机所识别到的 LUN ID 依然是源 LUN ID，但由于源 LUN ID 和目标数据卷 ID

之间的映射关系，迁移完成后的源 LUN 所指向的物理空间已经变成了目标数据卷，这就实现了用户无感知的业务迁移。

4.12 vVol

VMware Virtual Volume (vVol)是 VMware ESXi 6.0 的新功能，通过 vVol 可以提供虚拟机粒度的存储资源，将更多的虚拟机对存储的操作下发到存储系统进行，充分利用存储资源。vVol 技术为虚拟机 IO 的隔离以及存储反向感知虚拟机奠定了基础。

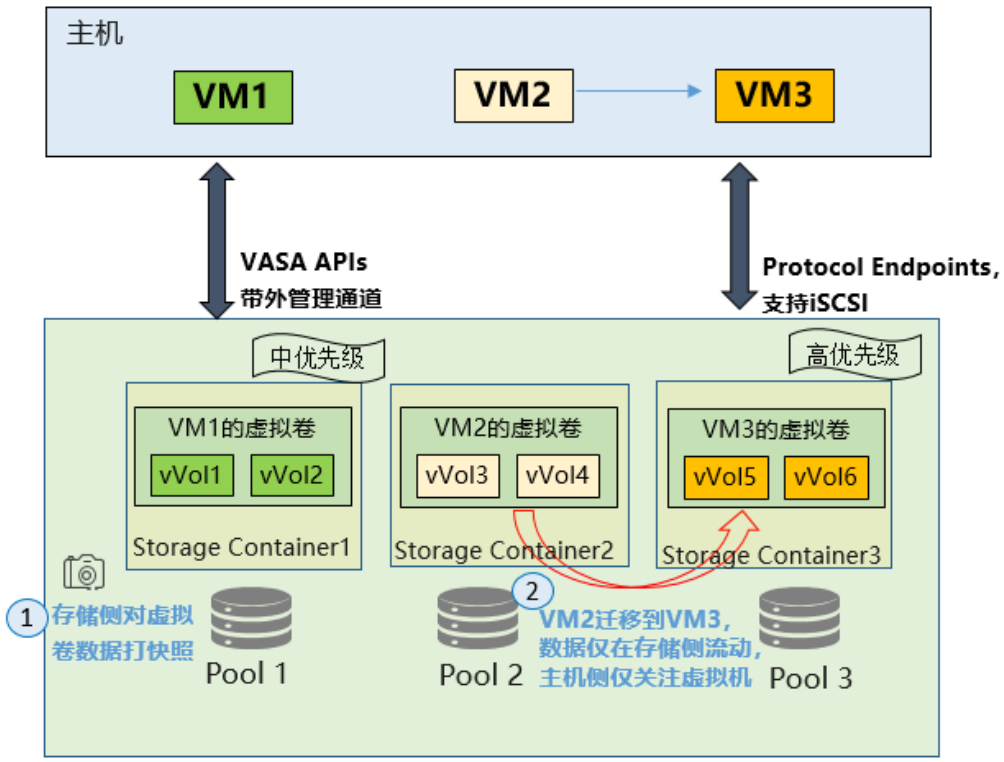
HengShan Stor 系列结构化服务支持 vVol 1.0 技术，增强了 VMware 生态兼容，简化主机侧管理，释放主机资源，针对性提升效率。vVol 技术把对虚拟卷的全生命周期数据管理操作卸载到存储侧完成，通过 VASA API 带外管理通道完成基于 vVol 粒度管理。提供如下主要功能：

表4-6 基于 vVol 的主要管理功能

基于 vVol 的主要管理功能	创建虚拟机
	扩容虚拟机磁盘
	对虚拟机打快照
	克隆虚拟机
	Fast 克隆虚拟机
	迁移虚拟机

vVol 的快照与迁移处理原理如下图所示：

图4-20 HengShan Stor 系列结构化服务 vVol 示意



1. 在存储侧可完成虚拟卷的快照，主机侧不需要关注虚拟卷，简化了主机侧的管理。
2. 如果数据要从虚拟机 2 迁移到虚拟机 3，传统方式要经过主机和网络，而采用 vVol 技术，数据仅在存储侧流动，既不占用主机 CPU 资源，也不用占用网络带宽，本地处理还可降低数据处理的时延，有利于整体性能提升。

4.13 场景化压缩（Scenario-specific SmartCompression）

场景化压缩特性是基于非结构化语义的，对部分指定的图像格式进行压缩和解压处理功能，支持文件、对象、大数据服务。针对不同场景的文件类型，识别文件格式以及文件元数据信息，匹配合适的压缩算法，以实现对单个文件的压缩功能。

非结构化数据中的文件语义是指某类型文件所属的特定非结构化场景。场景化语义压缩是先针对不同类型的文件，自动识别相应场景下的文件格式以及文件元数据信息，继而对识别出已编码数据进行无损的深度重压缩并以专用格式进行存储。

场景化压缩只支持部分场景的图像格式：冷冻电镜、基因测序、遥感高分、医疗病理图片。具体图片类型限制如下：

表4-7 图片类型限制

场景	支持文件类型(不区分大小写)	支持压缩解压缩方案
冷冻电镜	MRC、MRCS	默认解压为无压缩的原始数据后再重压缩
遥感高分	TIFF、TIF	默认解压为无压缩的原始数据后再重压缩
基因测序	FASTA、FASTQ FASTA.GZ、FASTQ.GZ	默认解压为无压缩的原始数据后再重压缩
医疗病理	KFB、SVS、SDPC	默认解压为无压缩的原始数据后再重压缩

场景化压缩支持和分级存储特性配合使用，分级存储可以将低价值或低使用率的文件放置在成本较低的、性能和可用性规格较低的设备上（温/冷存储），场景化压缩可以在分级存储进行数据搬迁时对文件进行压缩/解压。场景化压缩的周期性调度逻辑复用了 SmartTier 分级存储的实现，相关术语如硬盘池、分级等可以参考分级存储的技术白皮书。

4.14 通用压缩（Standard SmartCompression）

通用压缩特性是基于非结构化数据特征的，对用户数据提供压缩和解压的处理功能，支持文件、对象、大数据服务。针对不同场景的文件数据内容，自动选取最佳压缩策略进行压缩，除了视频、音频、图片、加密、已压缩数据、PDF、XML 等已编码数据格式，其他数据均能获得较好的压缩率。典型应用场景如表 4-8 所示。

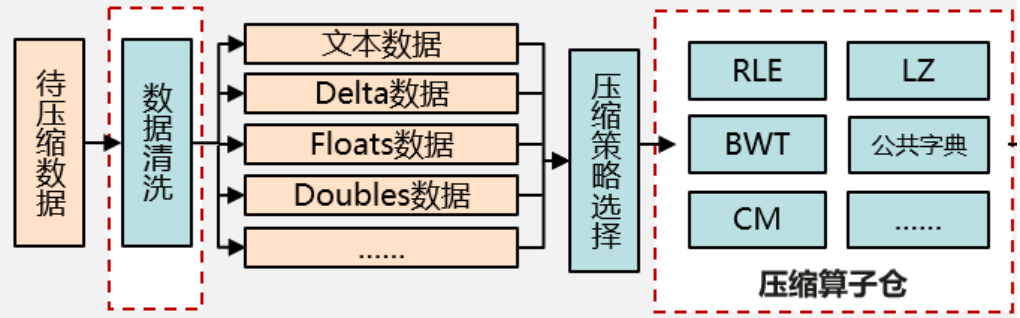
表4-8 通用压缩典型应用场景

类别	场景	子场景	典型数据格式
HPC 场景	自动驾驶	点云-产品	.bin\.pcap\.
	能源勘探	地震资料数据 SEG-Y	.sgy
		油井仿真	.res\.INC
	气象海洋	气象	.cdf\.nc\.hdf
	制造	CAD_CAE	.stl\.nc\.raw\.off\.op2
		EDA	.vpd\.fsdb\.sdb
	教育科研	计算化学，高能物理	.csv\.dcm\.tif\.tiff\.txrm
		脑科学	.tif

类别	场景	子场景	典型数据格式
		天文数据	.csv\.fit\.txt
大数据	parquet	非压缩 parquet	.parquet
资源池	动画渲染	动画渲染，后期制作	.swf\.scd
	医疗 PCAS	db	.idb\.dbf
		DCOM（非压缩格式）	.dcm

通用压缩算法由单模型向多模型方向演进，通过数据清洗、压缩策略选择、算子仓等技术，提升数据缩减收益，如图 4-21 所示。

图4-21 多模型通用压缩算法技术点



通用压缩为企业客户提供在线业务数据通用压缩功能和分级迁移通用压缩功能，两者互相配合以达成目标压缩率：

（1）按照文件系统粒度配置在线业务通用压缩策略。当开启在线业务通用压缩功能后，在接收用户业务数据时，为兼顾压缩率和效率，将用户数据按照一定粒度切分，同步调用压缩算法进行压缩，将压缩后的数据写入存储空间，保证节省空间即刻生效。

（2）按照文件系统粒度配置分级迁移通用压缩策略。当开启分级迁移通用压缩功能后，配置分级迁移策略并发生分级迁移时，当原始数据为压缩数据，则迁移压缩后的数据；当原始数据为非压缩数据，需要在迁移过程中调用通用压缩算法进行压缩，将压缩后的数据迁到目标池中。

4.15 智能数据迁移（SmartMigration）

4.15.1 文件迁移服务

智能文件迁移主要是应用于异构存储数据的迁移。目前通过主机层进行 NAS 数据迁移，是最为普遍的迁移方案。但是会面临几个问题：

- 需要使用 Windows 或 Linux 作为迁移服务器；

- 需要较长的割接时间。

为了解决以上问题，智能文件迁移提供了两种模式：

- Copy-first 模式

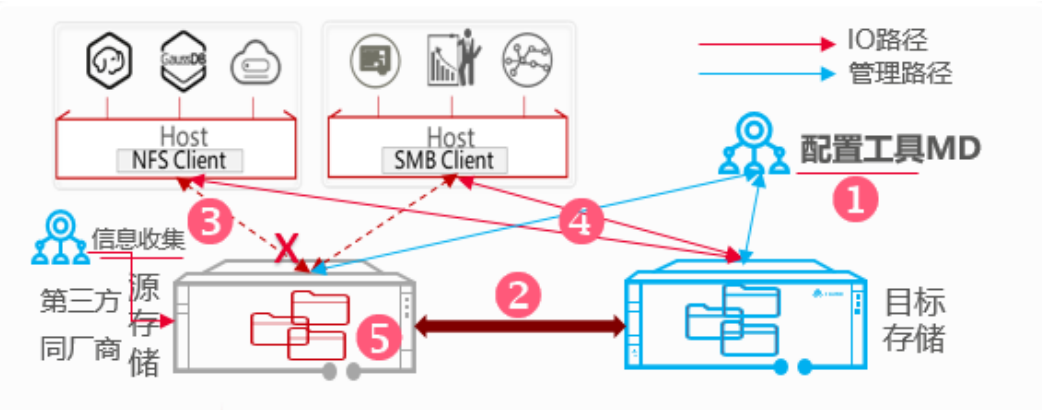
  类似于主机迁移，存储内置迁移模块读取远端数据，并同步到本端存储。但是该模式不需要客户提供 Windows 或 Linux 作为迁移服务器。该模式下存储内置数据迁移模块以及 NAS 协议客户端直接对远端存储的数据并行进行访问。类似于主机迁移，该模式，割接时间取决于最后一次增量同步时间。

- Takeover-first 模式

  不同于传统主机迁移，该模式可以保证分钟级的割接时间。客户从远端存储停止业务，建立本端存储和远端存储的同步任务以后，将 I/O 切换到本端存储。这样客户的业务可以继续进行，并且发生的 I/O 数据既写入了本端又同步到了远端存储，保证了两端存储数据的一致性。当后台数据同步任务将远端存储的数据全部搬迁到本端存储后，可以删除同步任务，从而客户业务可以无缝切换到本端存储。该模式的特点就是，可以不等待所有数据迁移完成，客户业务就可以割接到本端存储。

Copy-first 模式，具体如下所示:

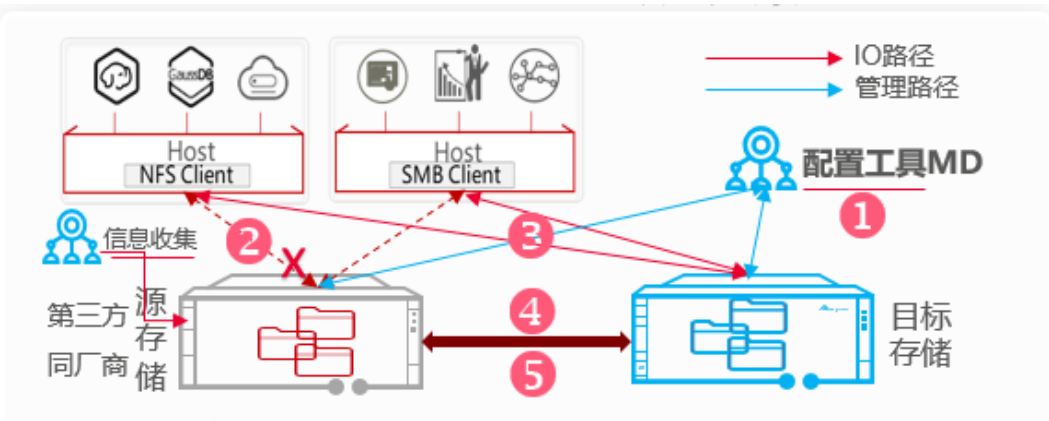
图4-22 Copy-first 模式数据迁移流程



1.准备阶段	工具收集源存储配置，通过配置工具 MD 在目标存储配置 源存储与目标存储建立连接，建立待迁移文件系统
2.初始迁移	MD 工具启动，初始迁移任务，源存储继续处理客户端业务
3.停机割接	业务客户端停机，通过差异对比启动增量迁移，迁移完成后，断开存储间连接
4.恢复业务	完成 IP 配置、DNS 修改，业务客户端重新挂载共享文件系统，采用目标存储读写业务
5.完成迁移	迁移完成，源存储从网络中移除

Takeover-first 模式，具体如下所示：

图4-23 Takeover-first 模式数据迁移流程



1.准备阶段	工具收集源存储配置，通过配置工具 MD 在目标存储配置 源存储与目标存储建立连接，建立待迁移文件系统
2.停机割接	业务客户端停机
3.恢复业务	完成 IP 配置、DNS 修改，业务客户端重新挂在共享文件系统，新业务双写（双写目标存储和源存储）
4.迁移业务	MD 工具启动源存储文件系统后台迁移功能
5.完成业务	当迁移数据全部完成后，断开源存储目标存储之间的连接，源存储从网络中移除

4.15.2 卷迁移服务

随着云、互联网及智能技术的兴起与普及，海量数据不断产生，伴随数据中心整合、存储过保等场景，存储数据迁移面临严峻挑战。以 FusionStorage 6.3 存储为例，FusionStorage 6.3 在资源池、数据库等场景应用广泛，当停止对 FusionStorage 6.3 软件版本提供服务后，就需要采用 HengShan Stor 进行替代。如何优化存储资源，实现存储系统间的数据搬迁成为亟需解决的问题。

HengShan Stor 存储系统通过 SmartMigration 提供的卷迁移技术，可以在不中断原有业务的情况下，将其兼容的异构存储系统上的数据迁移到 HengShan Stor 存储系统中。当前支持的异构存储系统为 FusionStorage 6.3 存储。

SmartMigration 卷迁移技术的主要分为纳管和在线迁移：

- 纳管技术，将异构存储系统的卷数据访问接管到 HengShan Stor 存储访问。纳管分为在线纳管和离线纳管。

- 在线迁移异构卷，通过卷在线迁移将异构存储系统的外部卷的数据完整的迁移到 HengShan Stor 存储系统。

在线纳管

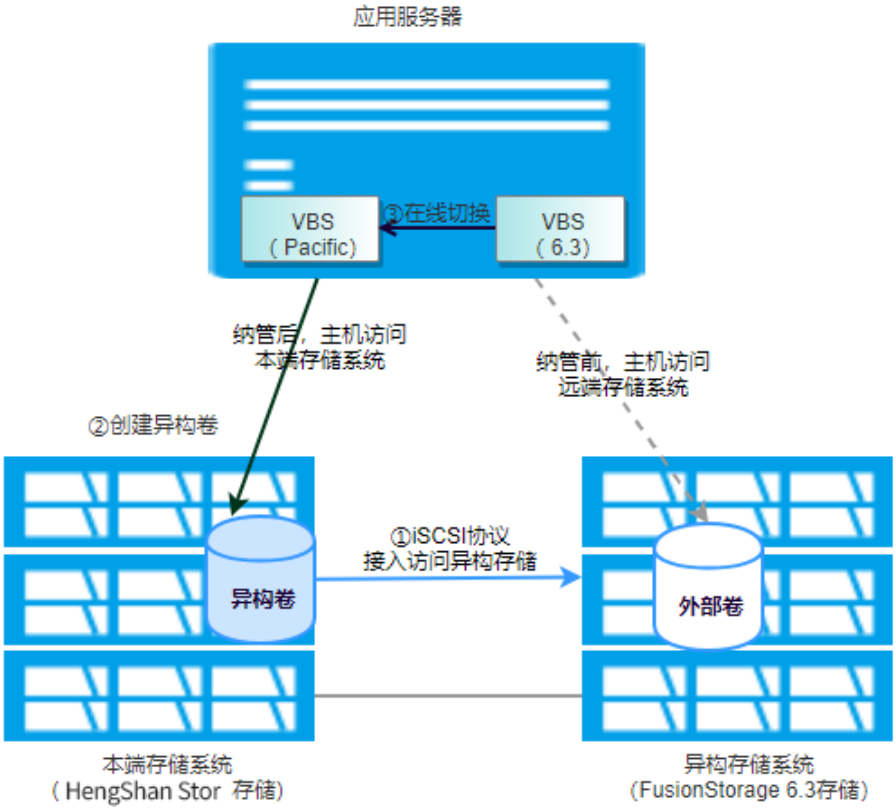
通过在线纳管技术，可以将异构存储系统的卷数据访问在线接管到 HengShan Stor 存储访问，整个过程全程不断业务，保证业务连续性。

本端存储系统通过 iSCSI 协议接入访问异构存储系统。如果应用服务器是通过存储的 VBS 客户端访问存储系统，纳管时需要进行 VBS 客户端的切换，即将应用服务器上的 FusionStorage 6.3 存储 VBS 客户端会切换为 HengShan Stor 存储 VBS 客户端。

在线纳管的纳管操作流程如下：

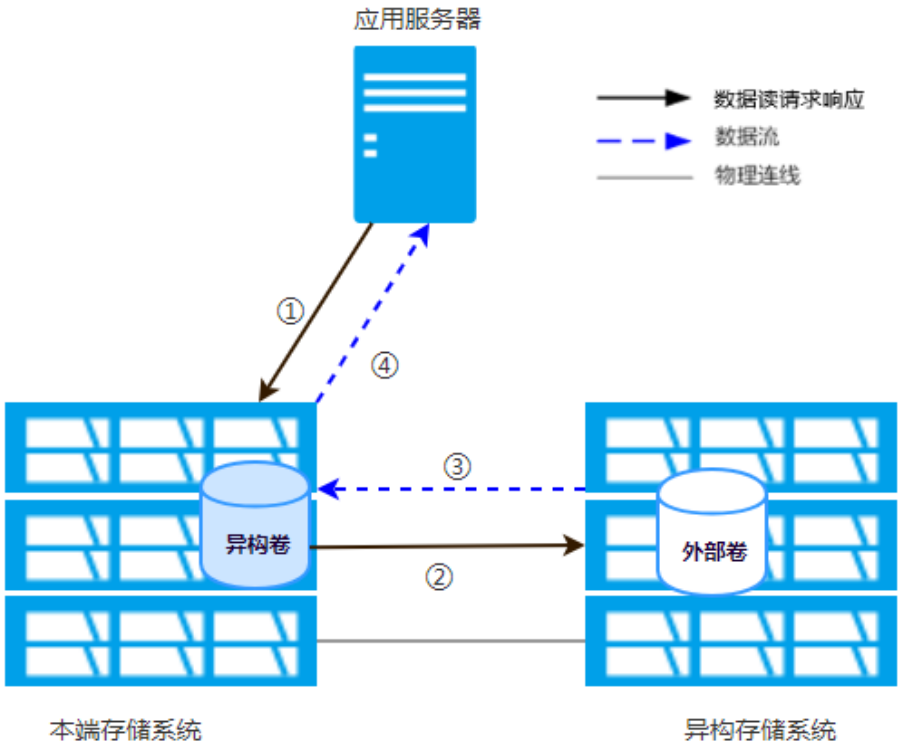
1. 异构存储系统提供 iSCSI 服务，本端存储系统以 iSCSI 协议接入，访问异构存储系统。
2. 本端存储系统创建异构卷。
3. 在线切换应用服务器的 VBS 块客户端。

在纳管操作前，主机应用直接访问异构存储系统。纳管操作后，即 VBS 块客户端切换后，主机应用的业务就切换到本端存储系统访问。



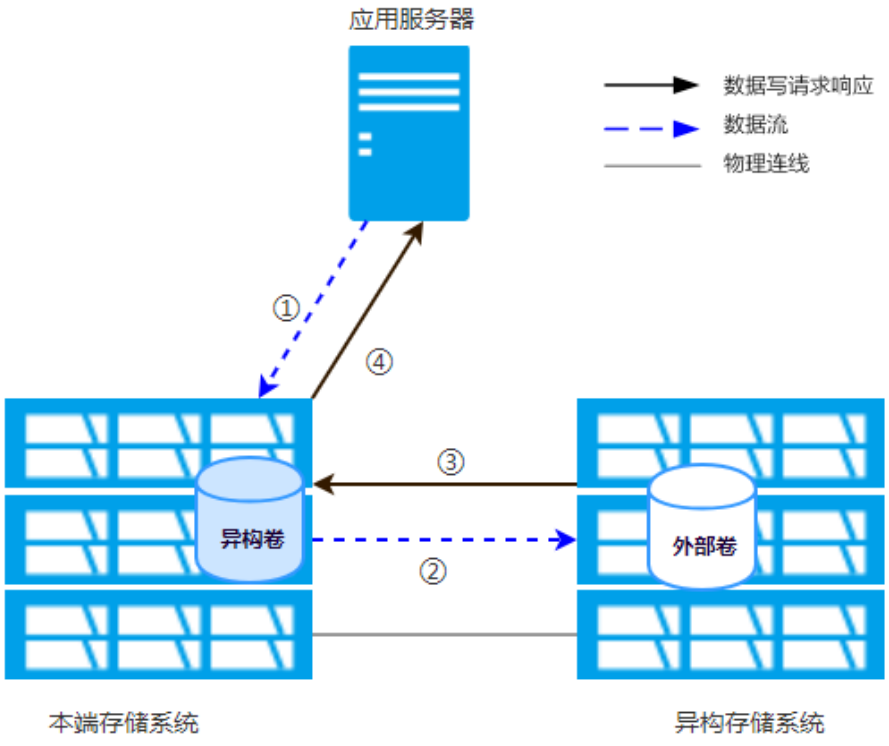
纳管期间，主机访问本端存储系统的异构卷时，都需要到异构存储系统访问。

- 数据读流程：



a. 应用服务器下发读数据请求。
b. 本端存储系统将读数据访问请求转发到异构存储系统。
c. 异构存储系统读取结果返回给本端存储系统。
d. 本端存储系统将读取的数据返回给应用服务器。

- 数据写流程



a. 应用服务器下发写数据请求。
b. 本端存储系统将写数据请求转发到异构存储系统。
c. 异构存储系统在数据写入后返回写响应给本端存储系统。
d. 本端存储系统将写成功的响应返回给应用服务器。

离线纳管

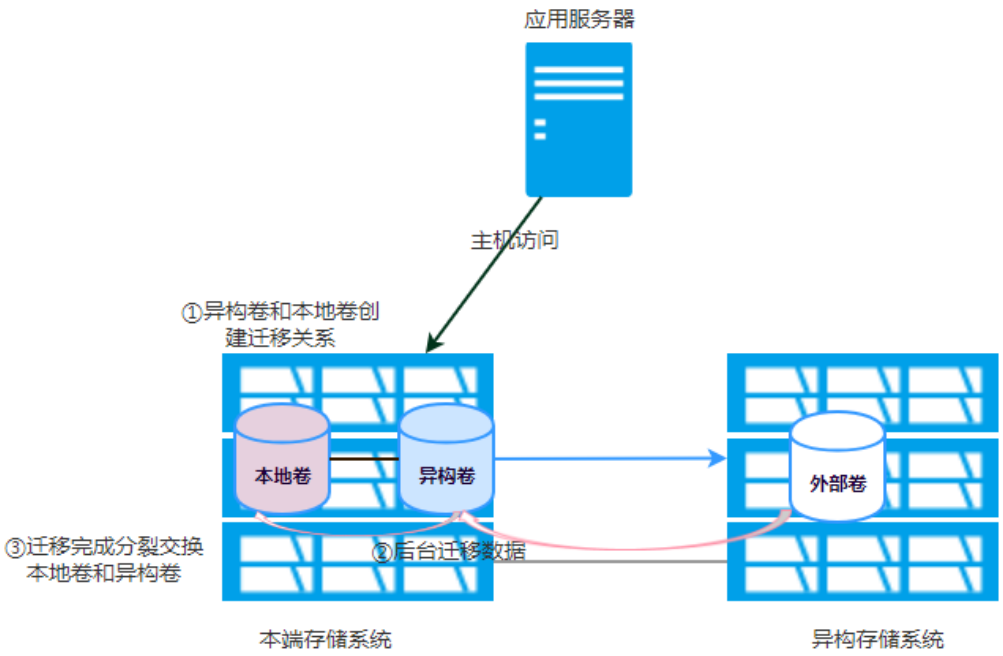
离线纳管与在线纳管流程上的主要区别在于，在线纳管时，VBS 客户端切换是进行在线切换。离线纳管时，需要先删除 FusionStorage 6.3 存储的 VBS 客户端，再安装 HengShan Stor 存储的 VBS 客户端。在离线纳管的操作过程中，业务会中断一定时间，待纳管操作完成后，业务恢复。离线纳管操作完成后，数据读写流程与在线纳管异构存储系统的数据读写流程一致。

在线迁移异构卷

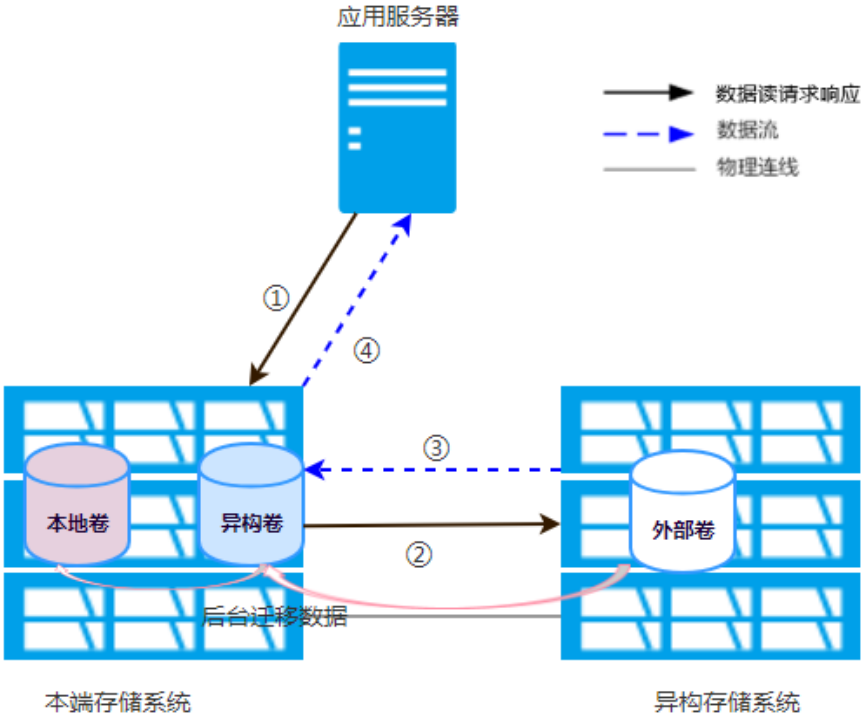
纳管将异构存储系统中的外部卷纳管到本端存储系统访问后，配合卷在线迁移技术，把异构存储系统的卷上的数据完整地同步到 HengShan Stor 存储系统。

异构卷在线迁移流程如下：

1. 在本端存储系统创建本地卷，将异构卷和本地卷创建在线迁移关系。异构卷为迁移源卷，本地卷为迁移目标卷。
2. 后台对迁移的源卷和目标卷进行数据同步，即将数据从异构存储系统的外部卷迁移到本端存储系统中。
3. 数据迁移完成后进行分裂，分裂时将异构卷和本地卷进行交换。交换后，主机的数据访问，只会访问本地卷，不再访问异构存储系统。

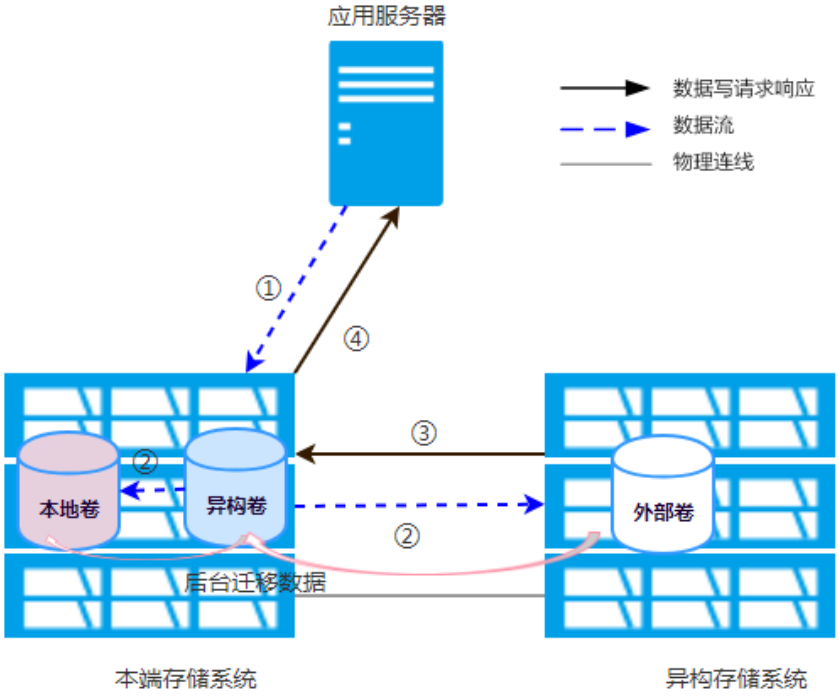


- 数据读流程：迁移期间的数据读流程与纳管期间数据读流程一致，只是同时存在后台迁移数据。



a. 应用服务器下发读数据请求。
b. 本端存储系统将读数据访问请求转发到异构存储系统。
c. 异构存储系统读取返回给本端存储系统。
d. 本端存储系统将读取的数据返回给应用服务器。

- 数据写流程：



a. 应用服务器下发写数据请求。
b. 本端存储系统收到写数据请求，并行将数据写到异构存储系统和本地卷。

c. 异构存储系统和本地卷写数据都成功后，返回写响应给本端存储系统。

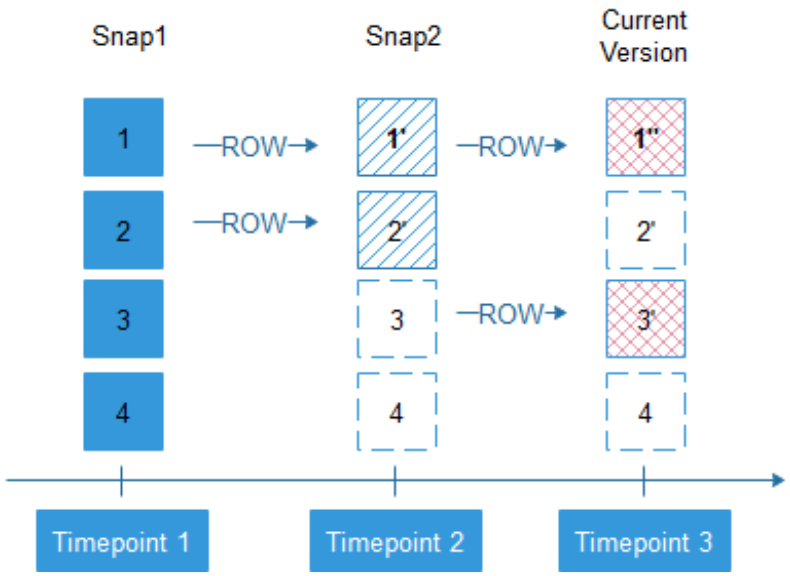
d. 本端存储系统在外部存储和本地卷数据都写成功后，将写成功的响应返回给应用服务器。

5 增值特性：数据保护

5.1 快照（HyperSnap）

HengShan Stor 系列支持秒级快照，快照数据在存储时采用 ROW（Redirect-On-Write）机制，快照不会引起原卷性能下降。

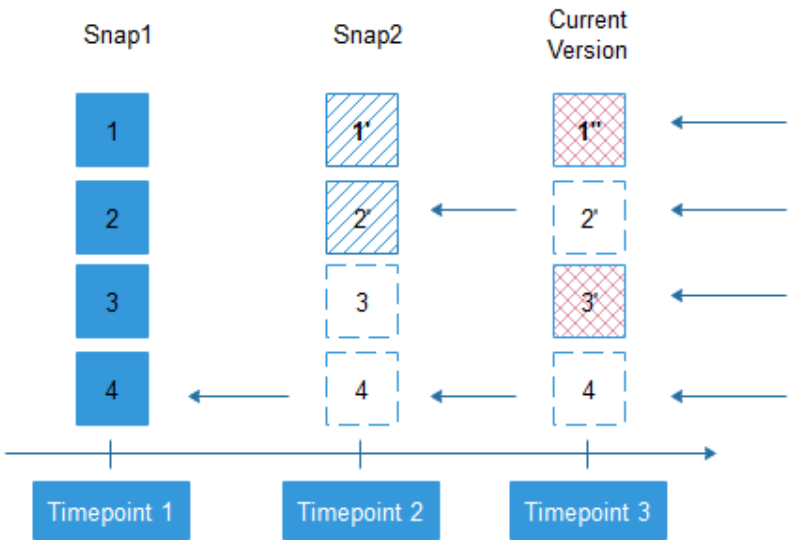
图5-1 ROW 快照原理示意



如上图所示，基于 ROW 的保护原理如下：，

- 在 Timepoint1 时刻在 Namespace 上创建快照 Snap1 后，业务对 Namespace 中的数据块 1 和数据块 2 写入新的数据，新的数据被写入新的空间中（数据块 1’ 和数据块 2’）。
- 数据块 3 和数据块 4 在 Timepoint2 时刻，数据没有变化，因此不占用新的空间。
- 如果在 Timepoint2 时刻创建新的快照 Snap2，之后对数据块 1 写入新的数据，则继续 ROW 到新的数据块 1"。
- 对数据块 3 写入新的数据，将 ROW 到新的数据块 3’，快照版本 Snap1 和 Snap2 的数据不变。

图5-2 快照数据保护读流程



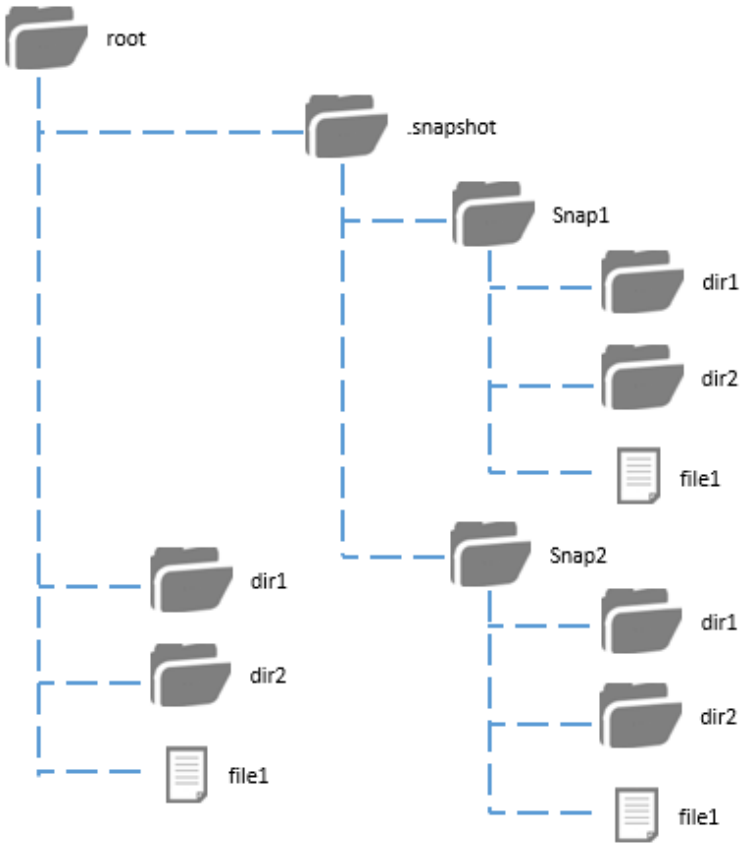
如上图所示，创建快照 Snap2 之后，访问文件系统数据时：

- 对于数据块 1，由于当前版本产生了新的数据，则直接访问数据块 1"；
- 对于数据块 2，由于当前版本并没有产生新的数据，则数据前向依赖于上一个快照版本的数据，访问快照 Snap2 的数据；
- 对于数据块 3，由于当前版本产生了新的数据，则直接访问数据块 3’；
- 对于数据块 4，由于当前版本并没有产生新的数据，则数据前向依赖于上一个快照版本的数据，由于 Snap2 同样没有产生 ROW 数据，则最终访问快照 Snap1 的数据 4。

5.1.1 非结构化数据存储服务

HengShan Stor 系列非结构化服务支持在不中断源命名空间正常业务的前提下，快速得到一份与源命名空间一致的副本。为 Namespace 创建快照，就相当于为 Namespace 创建了一份受保护的只读版本。当继续对文件系统进行写、删除等操作时，原有的数据空间将会被快照保护，保持不变。新的数据将以 ROW 的方式写入新的存储空间中。HengShan Stor 系列文件服务支持创建命名空间级和 DTree 级的快照，两种快照可以嵌套（只能嵌套 2 层），命名空间的快照可以保护该命名空间中所有的 DTree。可以为每个命名空间或 DTree 创建 4096 个用户快照和 4864 个定时快照。

图5-3 快照目录示意



创建快照后，会在快照根目录下生成.snapshot 的子目录，所有的快照会在.snapshot 下生成快照版本子目录（目录名为快照名）。各子目录为快照版本的访问入口，比如对命名空间创建 Snap1 和 Snap2 后，在命名空间根目录下生成的.snapshot 目录及其快照子目录如上图所示，访问.snapshot 中的目录或文件，即是访问快照版本的目录或文件。当访问快照版本数据时，根据快照 ID 对应的 Timepoint，访问相应时间点的数据。

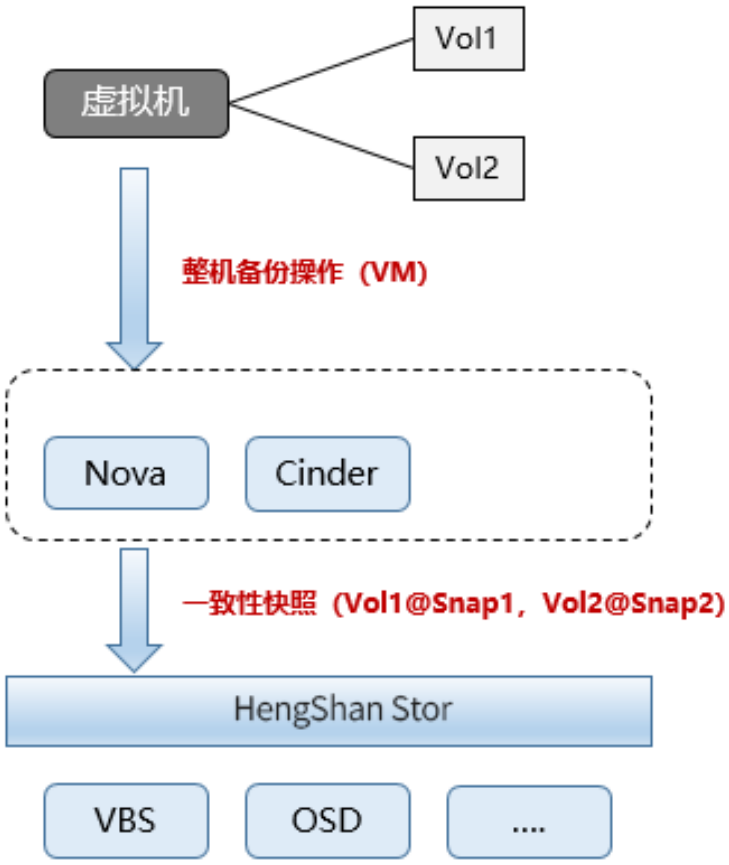
5.1.2 结构化数据存储服务

HengShan Stor 系列结构化服务提供了秒级快照机制，将用户的卷数据在某个时间点的状态保存下来，后续可以作为导出数据、恢复数据之用。

共享快照，为了支持共享卷备份能力，HengShan Stor 系列结构化服务也支持共享卷的快照；存在多个挂载点的 SCSI 卷称为共享卷，而对 iSCSI 卷而言，所有的 iSCSI 卷都是共享卷；共享卷和普通卷的快照流程相同。

一致性快照用于整机备份，一个虚拟机通常挂载了多个卷，对虚拟机做整机备份时所有卷快照的应处于同一时间点，才能保证数据恢复的可靠性；HengShan Stor 系列结构化服务支持一致性快照的能力， 对上层发起的一致性快照请求会尽量保证多个卷的快照属于同一个时间点；HengShan Stor 系列结构化服务对多个卷同时执行当前时间点失效、再更新快照信息操作，尽量保证了多个卷快照时间点的一致性。

图5-4 HengShan Stor 系列结构化一致性快照



5.2 复制（HyperReplication）

随着各行业数字化进程的推进，数据逐渐成为企业的运营核心，用户对承载数据的存储系统的稳定性要求也越来越高。虽然企业可以拥有稳定性极高的存储设备，但还是无法防止各种自然灾害对生产系统造成不可恢复的毁坏。为了保证业务数据的持续性、可恢复性和高可用性，远程容灾备份解决方案应运而生，而远程复制技术则是远程容灾备份解决方案中的关键技术之一。

HengShan Stor 系列非结构化服务提供了基于 Namespace 的异步复制，结构化服务提供了基于卷及一致性组的异步复制和同步复制。

异步复制是对主从两端存储系统的数据进行基于只读快照的周期性增量同步，实现系统容灾，从而最大限度减少由于数据远程传输的时延而造成的业务性能下降。

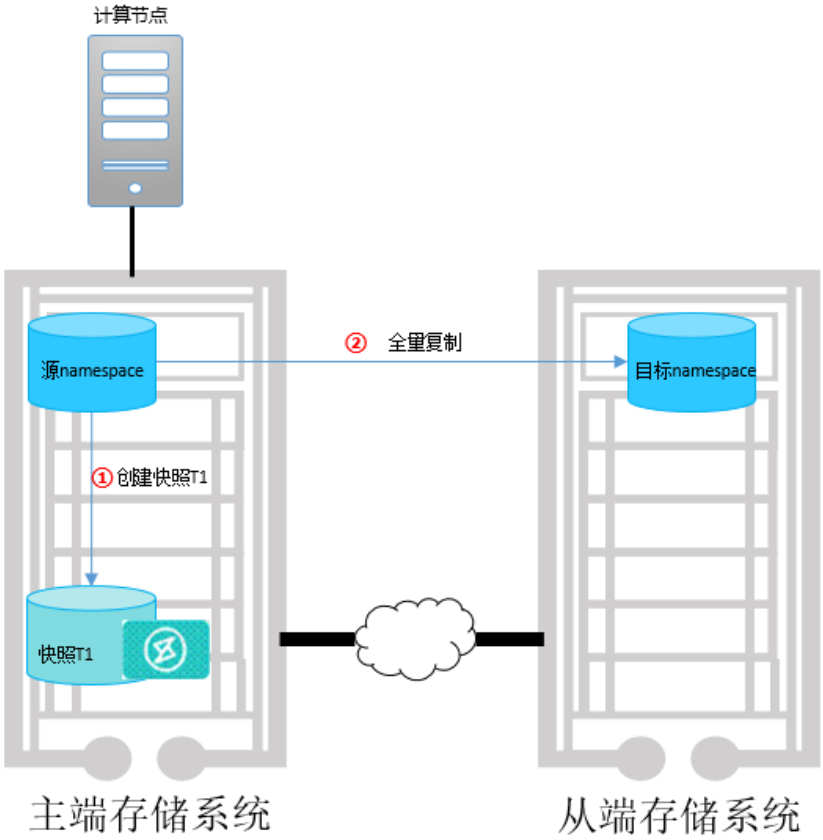
同步复制是主机写入生产的数据，会在生产存储和灾备存储都写入成功后，才返回主机写成功，能够保证 RPO 为 0。基于同步复制，能够实现灾难恢复级别较高的数据级容灾（“第 6 级：数据零丢失和远程集群支持”）。

说明

一致性组是多个远程复制 pair 的集合，可以确保单个存储系统内主机在跨多个卷进行写操作时，数据是一致的。在大中型数据库应用中，数据、日志、修改信息等存储在存储系统的不同卷中，缺少其中一个卷的数据，都将导致其他卷中的数据失效，无法继续使用。基于此场景 HengShan Stor 系列结构化服务提供了基于一致性组的异步复制，来保持多个远程复制卷的数据一致性。

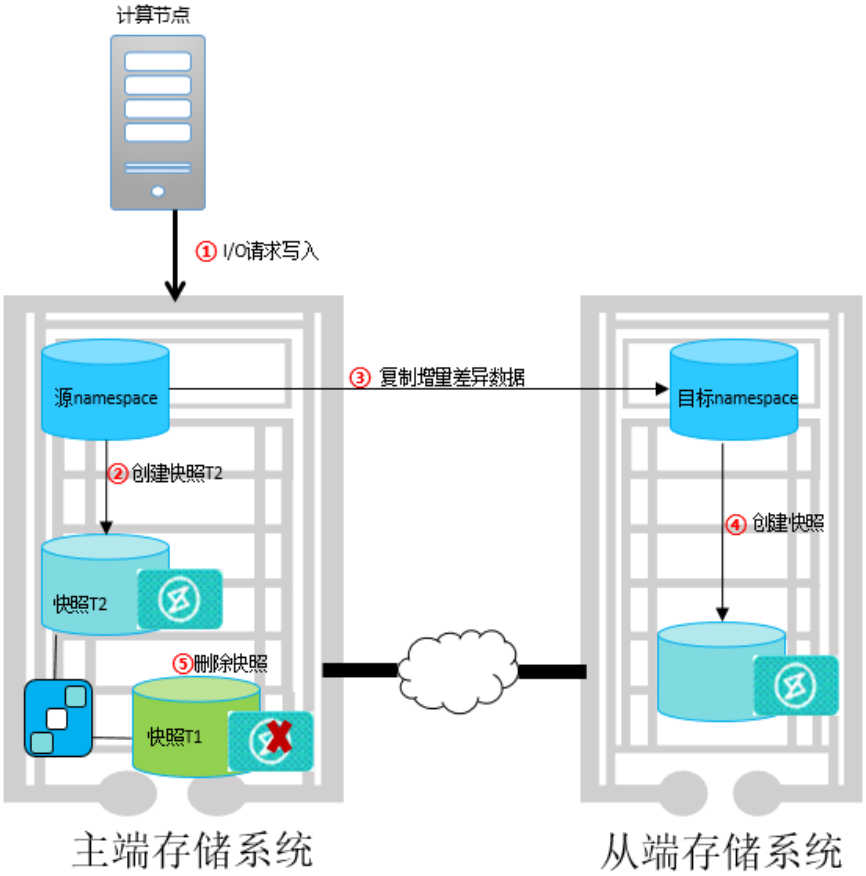
异步复制总体分为两个阶段，第一个阶段是创建复制关系时的初始同步，在初始同步完成后，进入第二个阶段，周期性的增量同步。下面以 Namespace 的异步复制为例介绍原理如下：

图5-5 异步复制初始全量同步



当源 Namespace 与目标 Namespace 建立远程复制关系后，首先会启动初始同步，初始同步前对源 Namespace 创建的快照 T1，将该快照时间点的数据全量复制到目标 Namespace，以保证目标 Namespace 与源 Namespace（某一快照时间点）数据一致。

图5-6 异步复制增量同步



初始同步完成后，后续复制通过快照差异日志同步增量数据。如上图所示：

1. 主机下发写 IO 请求到主端 Namespace。
2. 异步复制数据同步启动，启动前先创建快照 T2。
3. 将快照 T2 与前一个快照 T1 之间的差异数据同步至从端。
4. 数据同步完成后在从端对目标 Namespace 创建快照。
5. 从端删除上一次同步完成时创建的快照。主端也删除前一个快照。

同步复制（仅结构化服务支持）总体也分为两个阶段：

第一个阶段：创建复制关系时的初始同步；

第二个阶段：在初始同步完成后，进入双写状态。对于每个主机的写 I/O，都会同时写到主 LUN 和从 LUN，直到主 LUN 和从 LUN 都返回处理结果后，才会返回主机处理结果。因此，同步远程复制可以实现 RPO 为 0。

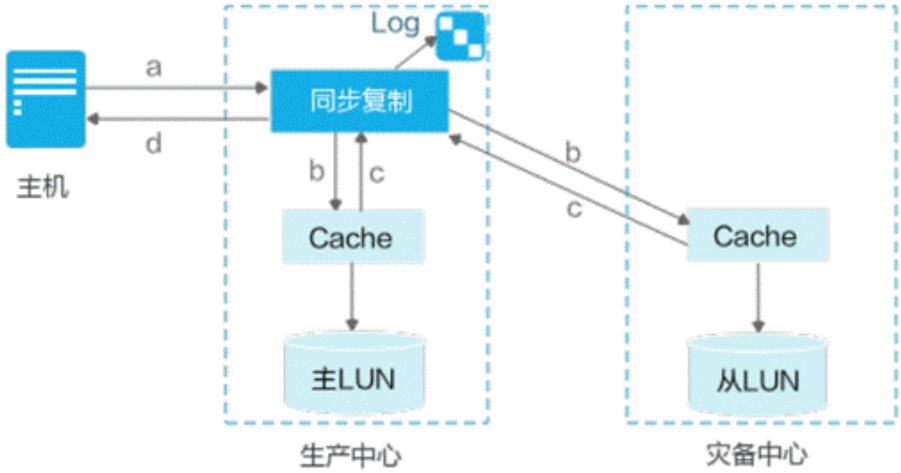
总体实现原理如下：

1. 初始同步：生产中心的主 LUN 和灾备中心的从 LUN 建立同步远程复制关系，启动初始同步。
   - 将主 LUN 数据全量拷贝到从 LUN。

- 初始同步中主 LUN 收到主机写请求也会同样写到从 LUN。

2. 双写状态：初始同步完成以后，进入正常状态，此时主、从 LUN 数据相同。正常状态下的 I/O 处理流程如下：

图5-7 同步远程复制示意图

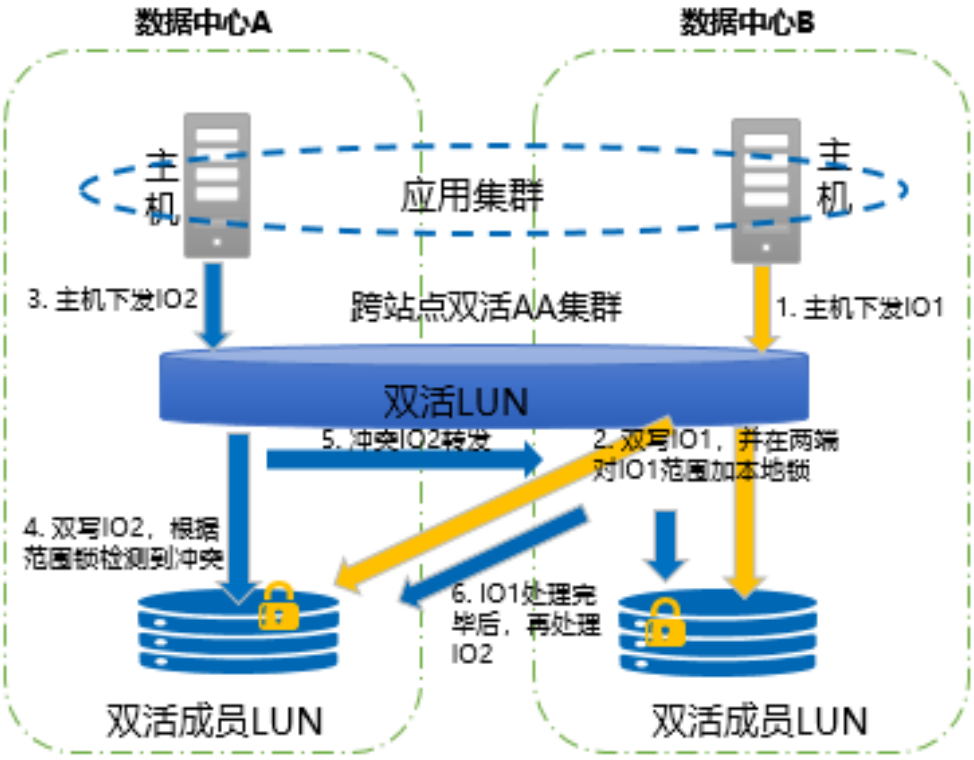


3. 生产存储收到主机写请求。同步远程复制将该请求记录日志。日志中只记录地址信息，不记录数据内容。
4. 将该请求写入主 LUN 和从 LUN。通常情况下数据会写入 Cache。
5. 同步远程复制等待主 LUN 和从 LUN 的写处理结果都返回，如果写从 LUN 超时或失败时则同步远程复制关系断开。如果都写成功，清除日志；否则保留日志，写入 DCL 中（Data Change Log 数据变更日志）元数据所在持久化空间中，进入异常断开状态，后续启动同步时重新复制该日志地址对应的数据块。
6. 返回主机写请求处理结果，以写主 LUN 的处理结果为准，如果主 LUN 写失败，即使从 LUN 写成功，仍然返回主机为失败。

5.3 双活（HyperMetro）

HengShan Stor 系列结构化服务的 HyperMetro 双活特性，基于 AB 两个数据中心的两套 HengShan Stor 系列存储集群构建双活容灾关系，基于两套 HengShan Stor 系列的卷虚拟出一个双活卷，两数据中心业务的主机能同时对卷进行读写。任意数据中心故障，数据零丢失，业务能自动切换到另外一个站点运行，保证业务连续型。

图5-8 HengShan Stor 系列结构化服务双活原理



HengShan Stor 系列结构化服务的双活支持增量同步，当一个站点发生故障时，按照系统的仲裁机制，会有仲裁获胜的站点提供业务，IO 请求由双写转为单写，当故障站点恢复系统，支持增加将故障过程的数据同步到恢复的站点，快速恢复系统。

同时, HengShan Stor 系列结构化服务增加了对逻辑写错误的处理，当系统正常运行，但是一端 IO 返回写失败的场景，支持将 IO 重定向写到正常的站点，写失败的站点故障修复以后，支持将增加的数据同步回来，减少逻辑写错误导致上层应用切换的问题。

HengShan Stor 系列结构化服务的 HyperMetro 双活特性可以跟上层 Oracle RAC、VMware 等应用配合，推荐数据库场景部署距离小于 100 公里，VMware 平台的部署距离小于 300KM。

5.4 对象跨站点多活（HyperGeoMetro）

跨站点容灾包括应用层的跨站点容灾和数据层面的跨站点容灾。通常，应用层的跨站点容灾由应用服务商提供，而数据层面的跨站点容灾则由存储系统提供。

典型的跨站点容灾系统由生产中心和灾备中心组成。两个中心之间的距离、网络情况、业务需求等因素直接影响复制技术的选择。同步复制技术对距离(通常不大于 100 公里)、网络时延(通常<=2ms 的 RTT 时延要求)、网络带宽要求较为苛刻，适用于高价值业务场景，典型如交易场景，成本高、站点个数有限（通常 2 站点或 2 中心）但提

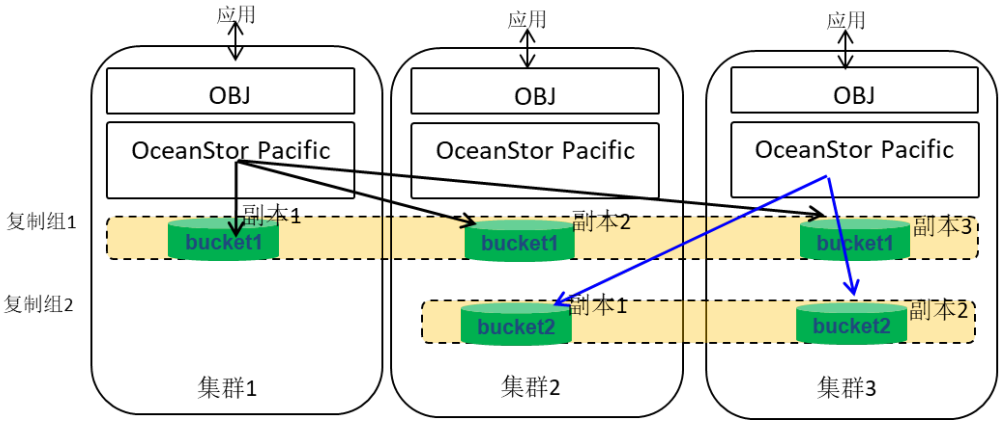
供 RPO=0 的数据容灾；异步复制技术对距离、网络时延及网络带宽的要求弱于同步复制，因此只能提供 RPO>0 的数据容灾。

随着对象应用的发展，部分应用需要同时从不同站点读写同一个桶，这使得传统的单活容灾不能满足应用需求。为了解决这些客户需求，HengShan Stor 8.1.3 对象服务跨站点容灾支持对象多活，支持不同应用从不同站点读写同一个对象桶。

跨站点容灾支持用户数据异步复制到远端站点，一旦某个业务站点故障（例如地震等自然灾害导致站点整体掉电）或站点维护导致站点无法对外提供业务的时候，上层应用可以切换到远端站点继续读写数据以满足业务连续性需求。

HyperGeoMetro 支持按桶配置冗余冗余比，如下图：bucket1 配置为 3 副本，bucket2 配置为 2 副本，给应用更多灵活选择。

图5-9 跨站点桶级多活



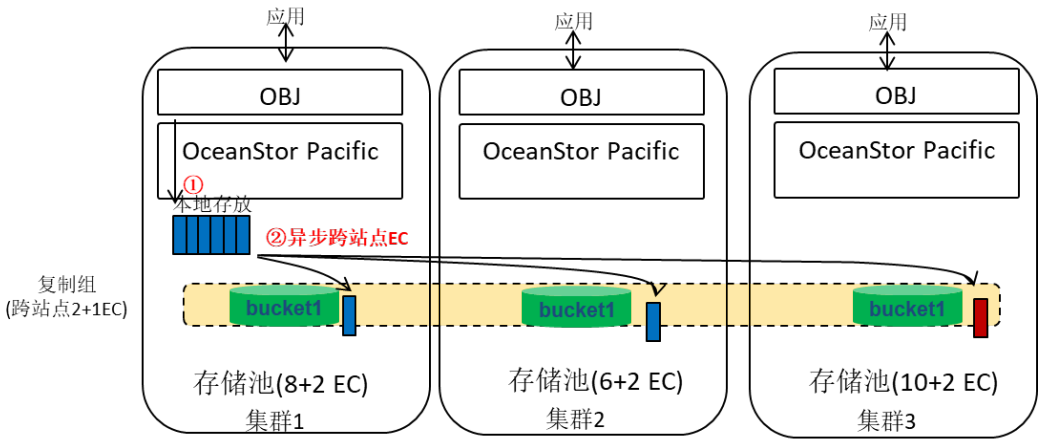
5.5 对象跨站点 EC（HyperGeoEC）

对象在归档场景对数据可靠性和成本都有较高要求，在满足数据异地容灾的同时，需要尽可能降低成本。HyperGeoEC 可以实现数据跨站点按照 EC 方式方式存储，支持多种冗余粒度，相比于副本存储，成本降低 50%以上，可以同时满足可靠性和成本的要求。

与 HyperGeoMetro 相似，HyperGeoEC 也可以按照桶级粒度设定冗余比，不同的桶可以选择不同的冗余，可以选择副本也可以选择 EC。

HyperGeoEC 除了站点间有冗余外，站点内也有冗余，站点内的硬盘和节点故障使用站点内的冗余即可，无须跨站点重构，减少了对站点间网络的依赖，同时提升了可靠性。

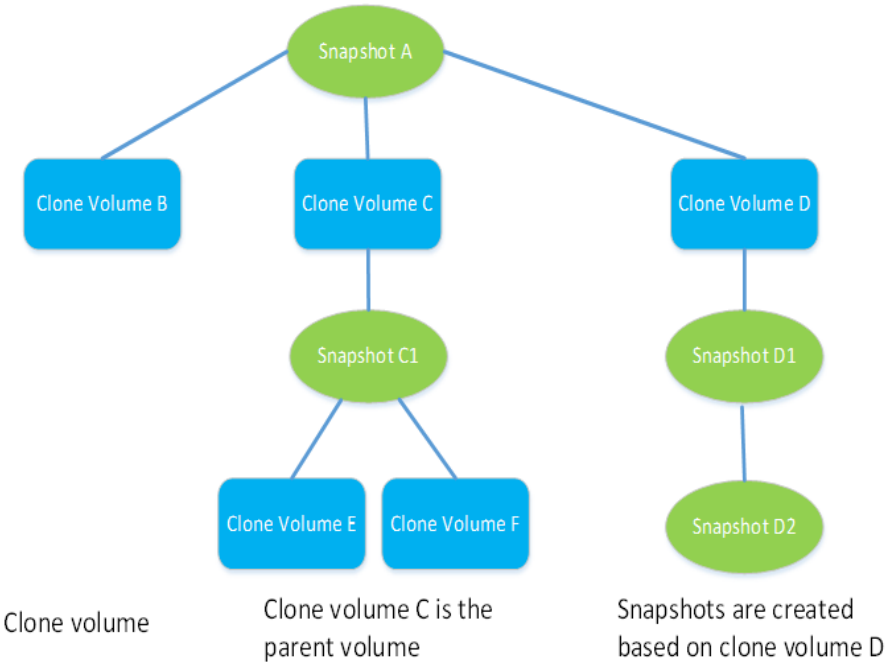
图5-10 跨站点 EC



5.6 链接克隆（HyperClone）

HengShan Stor 系列结构化服务提供基于链接克隆机制的 HyperClone 特性，支持基于一个卷快照创建出多个克隆卷，各个克隆卷刚创建出来时的数据内容与卷快照中的数据内容一致，后续对于克隆卷的修改不会影响到原始的快照和其他克隆卷。最大支持 1:2048 的链接克隆比，提升存储空间利用率。克隆卷继承普通卷所有功能：克隆卷可支持创建快照、从快照恢复以及再次作为母卷进行克隆操作。

图5-11 HengShan Stor 系列链接克隆



5.7 回收站（Recycle Bin）

HengShan Stor 系列非结构化服务提供了基于 Namesapce/DTree 的回收站，在用户删除文件时并不真正删除数据，而是文件在后台移入到回收站目录，以便用户找回数据。文件从原来的位置移入回收站，并不会真正删除也不会释放存储空间。为了及时释放空间管理员可以配置文件在回收站内的保留时长，系统会自动删除回收站内超过保留时长的文件。

每个 Namespace/DTree 创建一个回收站目录.recyclebininternal，文件在删除时不删除数据，而是在后台移到回收站中，在回收站中保持删除前的相对路径。如归属于 root 用户的文件在 Dtree 中的相对路径为/a/b/c.txt，在回收站中的路径为/.recyclebininternal/root/current/a/b/c.txt。用户可访问回收站内的目录与文件由回收站操作权限控制，普通用户权限下用户可访问归属于自己的目录与文件，回收站作为文件系统内一个特殊目录，root 用户也可以添加权限以便其他用户能访问。

系统会定期检查回收站内的目录以及数据。文件进入回收站时会首先放入/.recyclebininternal/用户目录/current 目录下，系统会首先把 current 目录 rename 到以当前时间为名称的目录下，然后再检测当前时间与创建该目录的时间差，判断是否超过保留时长，如果超过保留时长，则删除该目录。

说明

回收站功能默认关闭，用户在创建命名空间时可以根据需要开启或关闭，也可以在使用过程中修改该开关。该开关只对后续删除的文件生效。

5.8 防病毒（Antivirus）

随着网络和存储设备发展，各种病毒传播的几率也大大的增加，存储系统中保存了大量用户的文件，这些文件可能被病毒感染，这时需要有防病毒软件来清除处理。

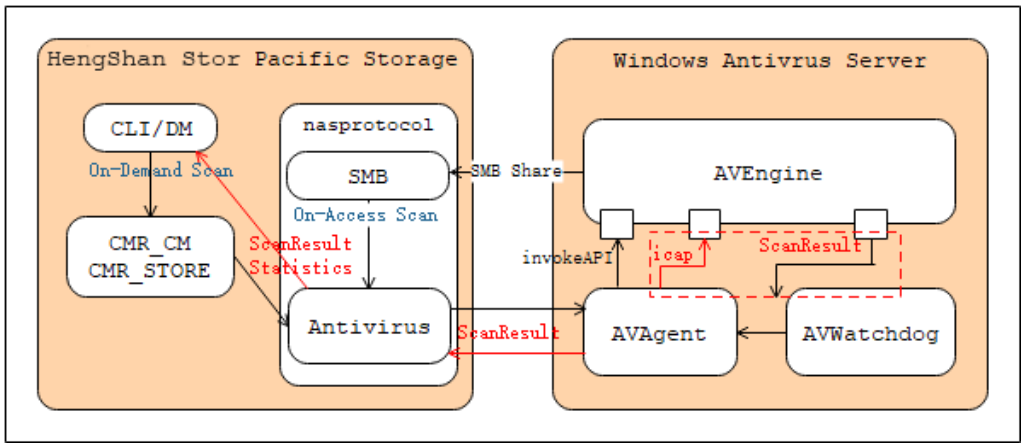
防病毒是 HengShan Stor 系列产品支持的重要的增值特性，为 NAS 存储提供安全保证。防病毒特性通常采用第三方防病毒软件，支持实时文件扫描(On-Access Scan)和按需文件扫描(On-Demand Scan)；同时支持配置扫描策略、配置扫描服务器等功能；支持防病毒日志的导出和转储。

病毒扫描类型分为提供按需病毒扫描、实时病毒扫描服务。与杀毒引擎的交互方式支持 API 调用和 ICAP 的扫描方式。

- 按需病毒扫描(On-Demand Scan)：根据用户配置的扫毒策略和周期，按照策略指定的时间段进行扫描的方式；
- 实时病毒扫描(On-Access Scan): 当实时对文件进行访问(Open/Close)时，对被访问文件进行的实时病毒扫描方式；

两种扫描方式的流程图如下：

图5-12 HengShan Stor 系列防病毒特性工作原理



与杀毒引擎的交互方式：

1. API 调用方式：安装在 AVServer 上的 AvAgent 程序调用 API 通知杀毒软件进行病毒扫描。 杀毒软件根据自身病毒特征判定逻辑，通过 SMB 协议读取必要的文件数据进行扫描，减少数据传输量，提升扫描速度。
2. ICAP 协议方式：AvAgent 收到文件扫描请求后，通过 ICAP 协议发送给杀毒软件进行病毒扫描。

5.9 防勒索（Ransomware Protection）

勒索软件攻击有两种：

1. 通过 AES、 RSA 等强加密算法加密用户数据，除非用户支付赎金，否则无法获取秘钥来恢复和访问数据。如果在指定时间内未支付赎金，这些文件数据将永久丢失。
2. 用户敏感重要数据被盗。攻击者威胁要将此数据释放到公有域，除非用户支付赎金。

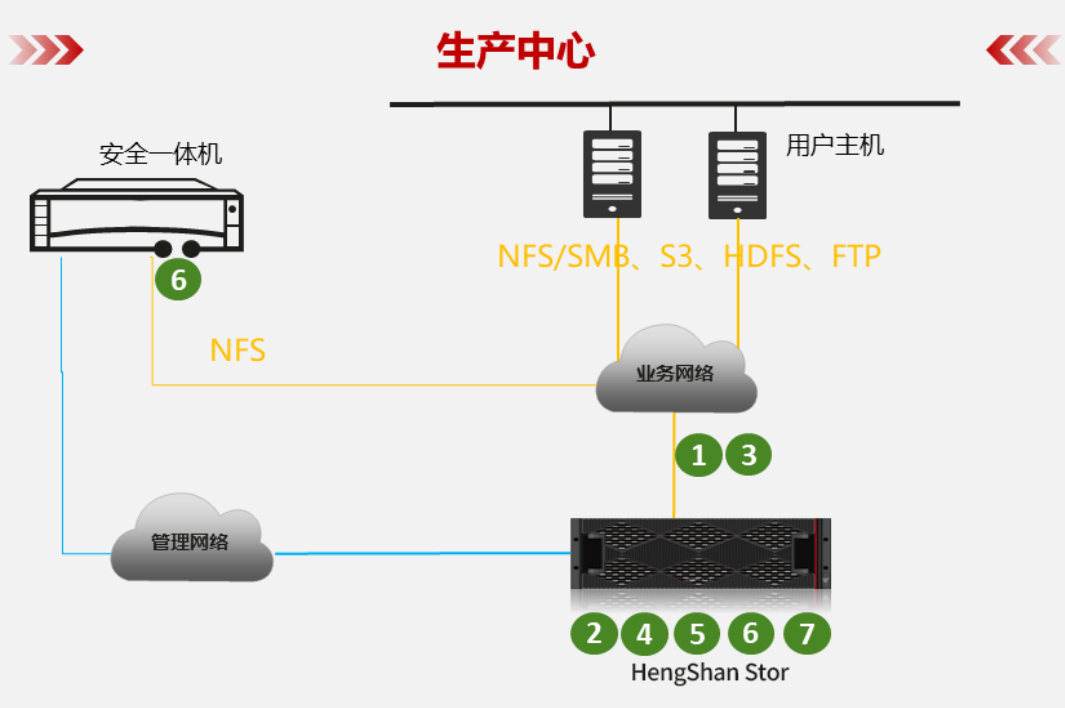
HengShan Stor 系列非结构化服务提供了对接 OceanCyber 安全一体机，协同实现完整的存储防勒索能力，通过加密技术、检测技术、数据只读技术和数据恢复技术，实现存储数据防窃取、防加密、可检测、可恢复的完整解决方案。

HengShan Stor 的非结构化服务防勒索方案符合 NIST（National Institute of Standards and Technology，美国国家标准及技术研究院）定义的 IPDRR 安全框架，在用户定义出需要保护的关键资产的情况下，提供对数据防勒索的保护、检测、响应和恢复能力。

表5-1 HengShan Stor 防勒索方案

序号	IPDRR 类别	功能	功能概要
1	保护 Protect	业务链路加密	SMB2/S3 报文以密文传输，数据防窃取
2	保护 Protect	存储加密	使用加密盘或软硬结合加密，防数据泄漏

序号	IPDRR 类别	功能	功能概要
3	保护 Protect	勒索文件拦截	设置文件后缀黑名单，拦截常见勒索软件
4	保护 Protect	WORM 文件系统	文件系统中的文件只允许一次写入，不允许修改，避免被勒索软件加密
5	保护 Protect	安全快照	快照设置保护期，在保护期内无法被修改和删除，保证副本安全
6	检测 Detect & 响应 Respond	勒索检测	分析存储中副本文件特征，及时发现勒索软件上报告警，并可以联动安全快照和快照回滚
7	恢复 Recover	快照回滚	使用安全副本恢复数据到被勒索前状态



6 端到端性能优化

百信 HengShan Stor 系列存储系统从计算节点接入均衡、IO 负载均衡、高速网络设计，同时结合基于 FlashLink 技术的混合 IO 负载优化，全 IO 路径端到端优化系统性能，为客户提供极致的高 IOPS 和低时延。HengShan Stor P9950K 产品单台整机可以提供高达 160GB/s 带宽和 640 万 IOPS 极致文件服务性能。HengShan Stor P9920K 单个节点在稳定 1ms 时延条件下，可以提供 21 万 IOPS 的极致块服务性能。

图6-1 HengShan Stor 系列端到端性能优化

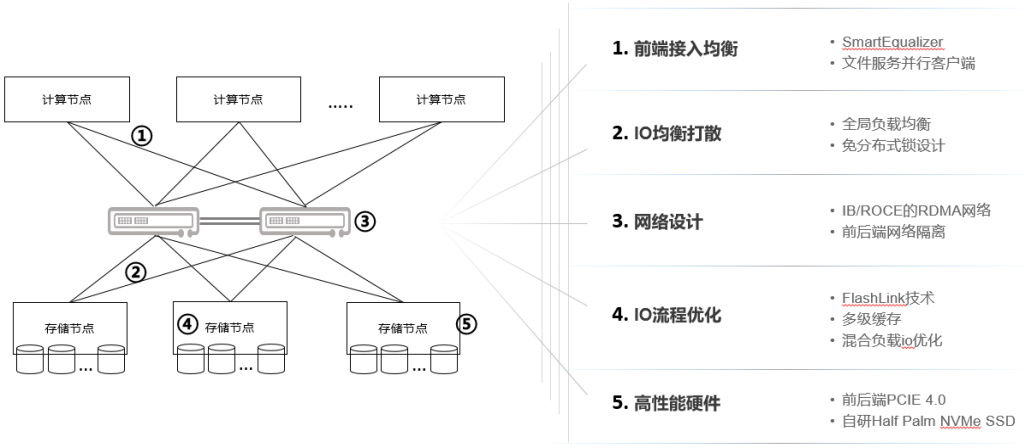


表6-1 百信 HengShan Stor 系列存储系统关键性能设计

IO 全流程	面临的挑战	关键设计	性能设计原理简述

IO 全流程	面临的挑战	关键设计	性能设计原理简述
前端接入均衡	• NAS 域名访问如何保证接入均衡 • 传统 NAS 挂载时只会连接一个节点，如何避免接入节点瓶颈	• NAS 访问负载均衡 • DPC 并行客户端接入	• HengShan Stor 系列 NAS 服务提供域名解析，通过 SmartEqualizer 均衡返回存储节点 ip。详见 4.4 负载均衡（SmartEqualizer）。 • DPC 并行客户端接入一个客户端可以连接多个存储节点，IO 级负载均衡。详见 6.3 文件服务并行客户端。
IO 均衡打散	• 单节点处理能力有限，怎么做到分布式性能随节点增加线性增长 • 分布式环境下如何保证多节点容量均衡	全局负载均衡	• 非结构化数据，按目录全局打散。 • 结构化数据，把 LUN 按 slice 切分打散。 • 通过动态智能分区与静态选盘法保证后端容量均衡。 详见 3.2 全局负载均衡。
网络设计	如何减少网络数据访问，节省带宽提升性能	• 存储前端网络与后端网络分离，支持 RDMA • DIO 数据直通，数据在 DPC 侧计算完 EC 后直接通过存储前端网络写入持久化层	• 支持 RDMA 网络，存储前后端网络分离，重构均衡 IO 走后端网络，不影响前台 IO 性能。详见 3.1.2 网络逻辑架构与 6.2 RDMA 高速网络。 • DPC 接入访问形态下，数据在 DPC 侧计算完 EC 后直接从前端网络写入持久化层。详见 6.3 文件服务并行客户端。

IO 全流程	面临的挑战	关键设计	性能设计原理简述
IO 流程优化	• 如何发挥多 CPU 多核及全闪存介质的能力 • 如何减少下盘 IO • 大小 IO 混合时如何优化	• FlashLink 技术 智能众核技术 大块顺序写 智能分条聚合技术 端到端 IO 优先级 • 多级缓存 • 免分布式锁设计 • 混合 IO 负载优化	• CPU 内按服务区分进行分区设计，减少服务相互干扰，对各类 IO 进行了优先级标识，根据标识的优先级，系统在 CPU 资源、内存资源、队列排队方面进行控制。详见 6.1 FlashLink 技术。 • 端到端多级数据缓存，提升读命中率，同时智能聚合下盘的数据，尽量满条带下盘。详见 3.3 多级缓存加速。 • 免分布式锁设计，消除分布式锁带来的分布式串行化开销。详见 6.4 免分布式锁设计。 • 大块数据，从 DPC 计算节点一次网络通信写到持久化节点,小块数据转发到归属节点 cache 聚合成大块下盘。详见 3.4.1 非结构化数据存储服务关键 IO 流程。
高性能硬件	• 如何发挥单位面积的最大性能	• 前后端 PCIE 4.0 • Half Palm NVMe SSD	详见 2.2 高密高性能硬件设计。

6.1 FlashLink 技术

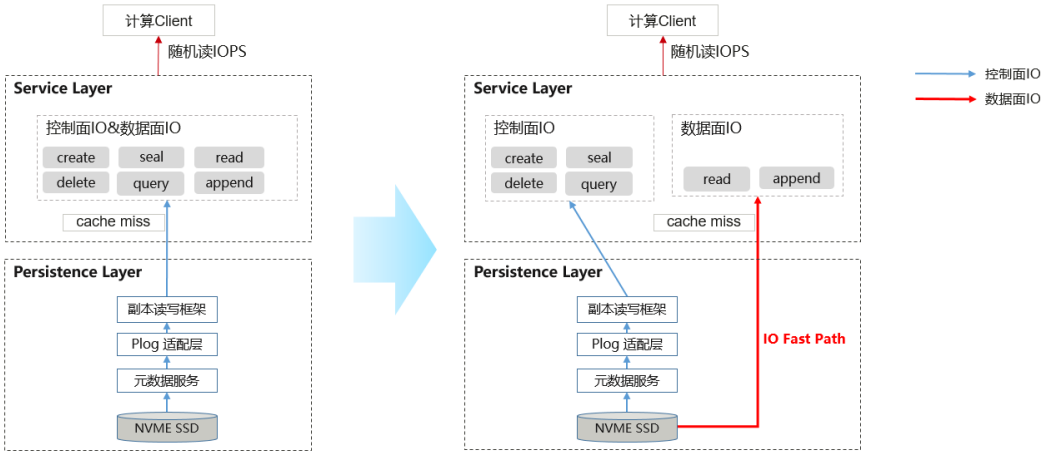
FlashLink 技术的核心是通过一系列针对闪存介质的优化技术，最大限度的发挥闪存的性能。HengShan Stor 系列的 FlashLink 针对闪存介质特点设计的关键技术主要有：数控分离技术、智能众核技术、大块顺序写、智能分条聚合技术和端到端 IO 优先级，保障了百信 HengShan Stor 系列系统的高性能。

6.1.1 数控分离技术

随机读 IOPS 场景下，读 cache 命中率较低，此时性能主要依赖盘的性能。针对该 IO 模型，HengShan Stor 系列优化了闪存盘的读写效率，存储软件的数据面 IO 路径和控制面 IO 路径分离，简称数控分离技术。

存储软件北向提供存储语义接口，具体包含数据面 IO 接口：append（追加写操作）、read（读操作），以及控制面 IO 接口：create（创建操作）、delete（删除操作）、seal（封存操作）、query（查询操作）。HengShan Stor 系列在闪存盘上将数据面 IO 路径和控制面 IO 路径分离，形成两个独立的 IO 调度框架，使得 IO 操作类接口的函数调用栈减少 50%，存储软件栈的时延消耗和 CPU 占用减少 50%，原理如下图：

图6-2 数控分离技术示意



6.1.2 智能众核技术

HengShan Stor 系列可以支持 X86 和 Kunpeng 处理器两种硬件平台，在每个节点上，最大都可以支持两个 CPU socket。如果使用 Kunpeng 处理器平台，Kunpeng 920 每颗 CPU 最大可以支持 48 个物理核；X86 平台上的 CPU，开启超线程之后，也会支持很多个逻辑核。采用智能众核技术，可以实现更好的 CPU 线性扩展性，充分发挥 CPU 的最大处理能力。

HengShan Stor 系列根据自己的业务特点，对 CPU 进行了一定的分区划分，将不同类型的业务划分在不同的 CPU 分区上。同时通过 NUMA(Non Uniform Memory Access，是一种关于多个 cpu 如何访问内存的架构模型) aware 的设计，每个 CPU 分区上运行的进程，都可以就近访问本 Socket 上所连接的物理内存，从而可以避免跨 CPU 访存时延大的问题，这样就可以减少 CPU 由于访存时延带来的开销，极大低提高软件运行效率。

通过 CPU 分区的划分和业务线程到 CPU 分区的动态绑定，不同的业务隔离在不同核上运行，避免了不同业务分组对 CPU 的争抢和冲突，这在多核系统上，可以有效减少常用的原子变量、spinlock 等互斥机制的开销，提高整系统的线性扩展能力。

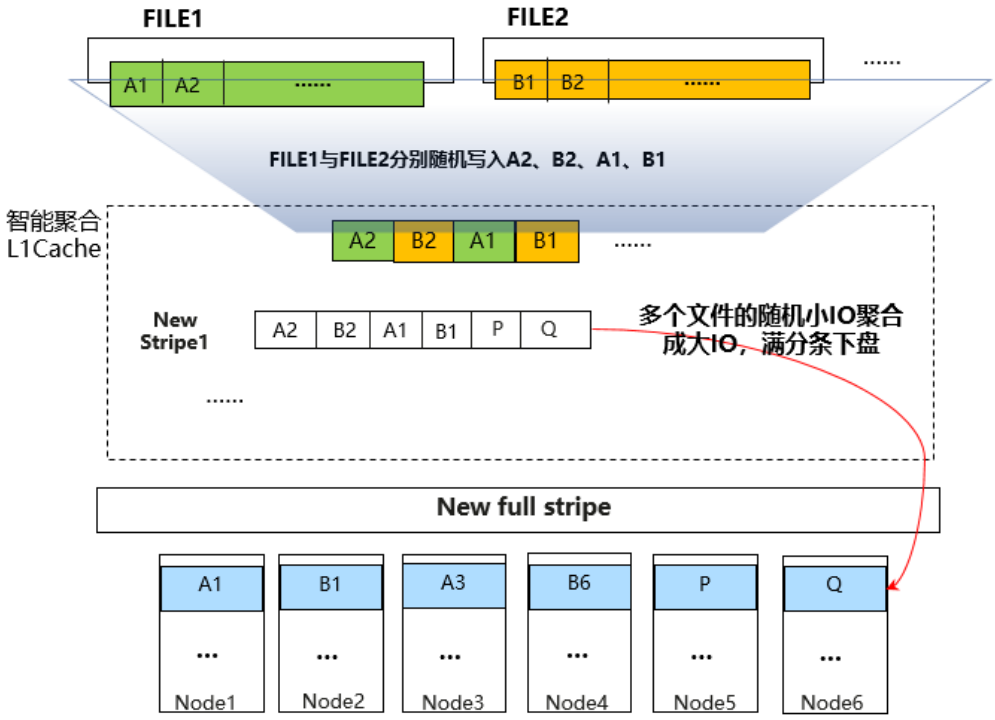
6.1.3 大块顺序写

SSD 盘相对磁盘，SSD 盘片上的 Flash 颗粒具有擦写次数的限制。在传统的 RAID 覆盖写（Write In Place）的方式下，如果某块盘上的数据成为了热点，那么对这些数据的不断改写，就会导致对应的 Flash 颗粒的擦写次数很快的耗尽。百信 HengShan Stor 系列存储全闪存配置通过对所有的写（包括新写数据和对老数据的改写等）都采用 ROW（Redirect On Write）大块顺序写的方式，每次写通过分配新的 Flash 颗粒来写，实现每个 Flash 颗粒擦写次数均衡。同时，避免了传统 RAID 写流程所需的数据读和校验修改写而产生 RAID 写惩罚，有效降低了写入过程阵列节点的 CPU 开销与对 SSD 盘的读写压力。

6.1.4 智能分条聚合技术

HengShan Stor 系列产品支持 EC 冗余保护，为提供较高的空间利用率，支持高效的大比例 EC，如 20+2 的大比例 EC 配比。在大比例的 EC 条带配比下，随机小 IO 往往很难凑成满分条下盘，传统的处理方式会产生额外的写惩罚 IO，导致性能的下降。为解决这种问题，HengShan Stor 系列产品基于 PLOG append only 的语义设计实现了针对小 IO 智能聚合的技术。Plog 是 Persistence Layer 提供的一组按照固定大小管理的物理地址的集合，提供 EC 的冗余保护能力，通过 PlogID+Offset 的方式来访问具体的物理磁盘上的 LBA 地址。PlogID+Offset 的逻辑地址到物理 LBA 地址的映射关系，是在写入数据时，才真正完成分配映射的。以文件服务为例，多个文件的随机小 IO 写入 L1Cache，在 L1Cache 进行聚合，聚合成大 IO，满分条写入 plog，减少写惩罚，提升系统性能。

图6-3 智能分条聚合示意

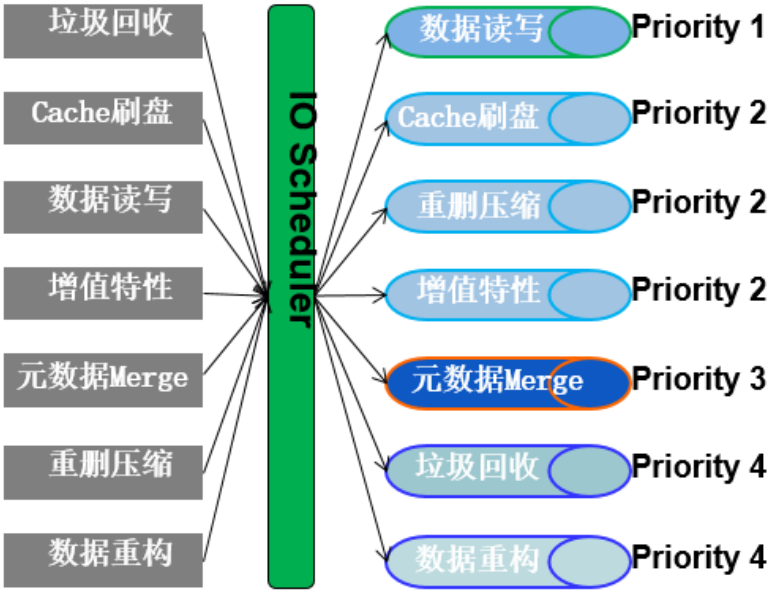


6.1.5 端到端 IO 优先级

为保证稳定时延，HengShan Stor 系列会对各类 IO 进行了优先级标识，根据标识的优先级，系统在 CPU 资源、内存资源、队列排队等方面进行控制，实现端到端的优先级保障，主机数据读写 IO 高优先级响应，后台 IO 根据类型赋予不同优先级，优先保障主机数据读写 IO 时延。

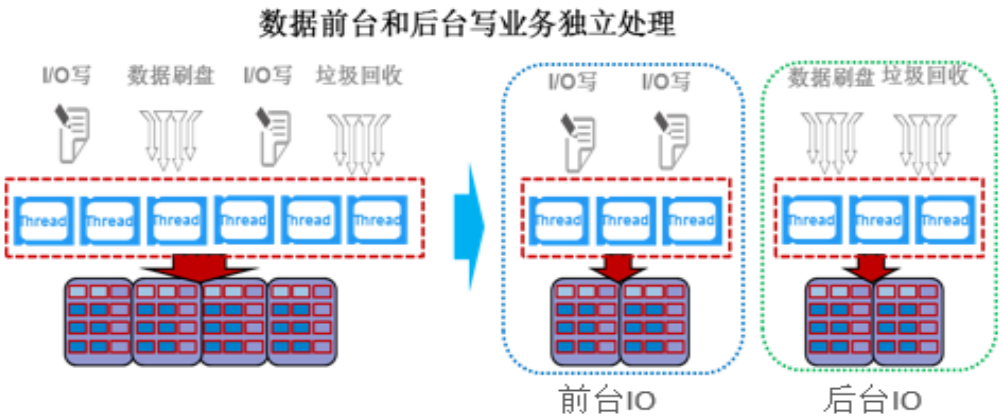
如处理主机读写 IO 请求时，会高优先处理，保证读写 IO 的请求获得稳定时延，快速响应主机请求。系统后台 IO 分为多种类别：Cache 刷盘、元数据 Merge、高级特性(远程复制、后台拷贝等)、数据垃圾回收、重删压缩、数据重构的 IO 优先级依次递减。

图6-4 IO 优先级示意



为避免后台业务对前台主机写业务影响，将前台业务和后台业务独立处理，前后台的数据业务拥有独立的线程资源、IO 并发资源和存储空间，从而降低后台刷盘、数据 GC 和元数据 Merge 业务对主机写的影响，保证系统的稳定时延。

图6-5 前台与后台区分



6.2 RDMA 高速网络

分布式存储单套集群往往会有几十甚至上千个存储节点，数据通过存储网络打散到不同节点中，这就对网络的性能和可靠性提出了更高的要求。HengShan Stor 系列文件服务所有节点之间通过 RDMA 网络互联起来，可以使用 RDMA 进行低时延通信，这是 HengShan Stor 系列能够达成高 IOPS、高带宽、低时延的基础之一。

百信 HengShan Stor 系列支持 InfiniBand 和 RoCE 网络（文件服务支持 InfiniBand 和 RoCE 网络，大数据服务支持 RoCE 网络），能使用 RDMA 协议进行 IO 数据交换。相比 TCP 协议，不存在 TCP 协议栈处理开销，RDMA 协议可以提供更快的响应。

网络芯片支持 RoCE，同时支持 RDMA 和 IP 的访问能力，在同时支持带宽、IOPS 等不同类型的业务时，可以智能调整网络的相关参数，提供更好的拥塞控制，从而可以大幅降低不同存储节点间的数据交换时延。配合交换机 AI Fabric 技术，进一步降低系统访问时延。

为了减少网络跳数，提升端到端的性能，HengShan Stor 系列非结构化服务支持从请求接入节点直接从持久化节点读取与写入大 IO 的数据，HengShan Stor 系列结构化服务支持从归属节点绕过 L1Cache 直接把大 IO 写入数据持久化节点。

- NFS over RDMA

  NFS 协议依赖 RPC 协议，RPC 协议下层为传输层。传统的 NFS 传输层采用 TCP 或 UDP 协议，NFS over RDMA 则是依赖于 RDMA 技术，实现 NFS 客户端和存储之间的数据直通，相比 TCP 传输，NFS over RDMA 可以有效降低网络时延，减少客户端和服务端的 CPU 负载，提升 NFS 协议访问性能。

6.3 文件服务并行客户端

HPC（ High Performance Computing）高性能计算，即通过高速网络将大量服务器进行互联形成计算机集群，与高性能存储一起，求解科研、工业界最复杂的科学计算问题，历经多年发展，已经在教育科研、生命科学、制造仿真、气象预报等多个领域得到广泛应用。随着 IoT 和智能技术的发展，HPC 和大数据、HPDA（High Performance Data Analysis）等新兴 HPC 场景开始出现，HPC 的应用进一步扩展到自动驾驶、金融反欺诈、个性化医疗等领域。

在 HPC 场景中存储系统将面临更复杂的 IO 模型，为应对这些性能挑战，HengShan Stor 系列文件服务推出 DPC（Distributed Parallel Client）分布式并行客户端来承载标准 POSIX 语义和 MPI-IO 语义，提升单流和单客户端性能，降低访问时延。作为存储客户端运行在计算节点上，通过网络协议与后端存储节点进行数据交换。DPC 通过兼容标准 POSIX 语义和 MPI-IO 语义，提供并行接口技术和智能数据缓存算法，使上层应用能更智能的访问存储空间。DPC 与传统的标准 NFS 协议对比如下：

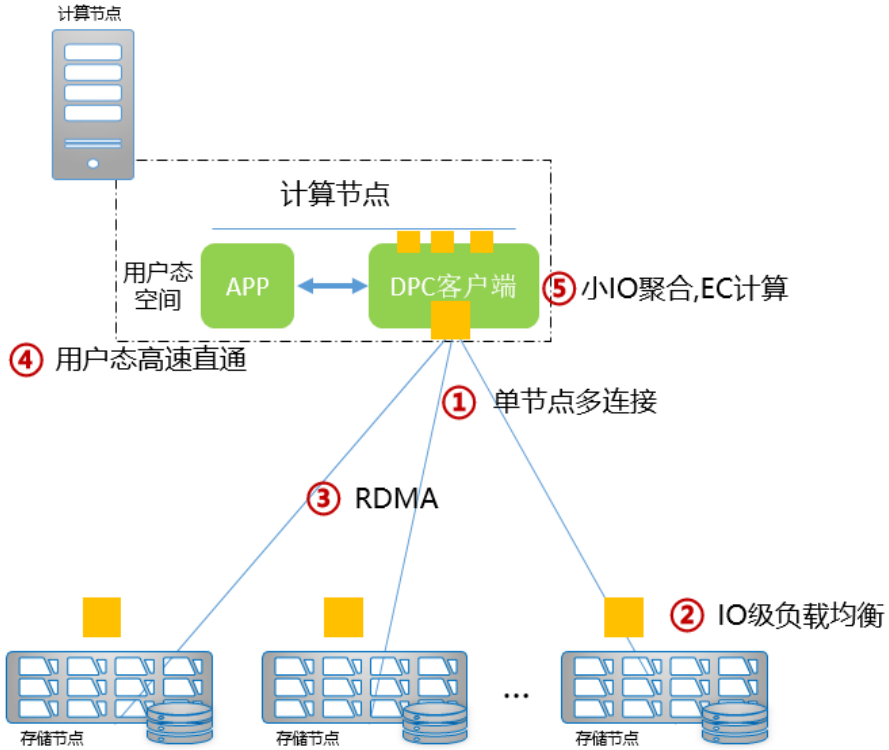
表6-2 DPC 与传统的标准 NFS 协议对比

对比项	NFS	DPC
单客户端连接模式	单客户端仅能连接一个存储节点。	单客户端可连接多个存储节点。
负载均衡模式	连接数均衡，无法实时保证存储业务压力均衡。	IO 级负载均衡，实时保证存储业务压力均衡。
数据 IO 路径长度	数据 IO 仅能写在已挂载存储节点上，若要写入其他目的存储节点，需在挂载存储节点缓存后再写入到对应存储节点。	数据 IO 可直接写入到目的存储节点上。

对比项	NFS	DPC
单流性能	受存储侧和计算侧最小单 CPU Core 性能瓶颈限制。	存储侧无性能瓶颈，仅受计算节点单 CPU Core 性能限制。
接口支持	POSIX	POSIX 和 MPI-IO
通信模式	TCP/IP	RDMA
NUMA 亲和	不支持	支持和计算应用 NUMA 亲和，发挥多 NUMA 架构的极限性能

在 POSIX 语义中，部署 DPC 并行文件客户端的计算节点在网络上与集群内所有存储节点相连，应用程序下发的 IO 数据在计算节点进行聚合，并完成 EC 计算，通过 PlogClient 直接发送给 Plog 持久化节点(persistence layer)。与传统协议 NFS/SMB 协议相比，业务层下发的数据通过 RDMA 网络直通写入持久化层，存储节点少一次网络转发并大幅降低了存储节点 CPU 利用率。如下图所示：

图6-6 DPC 侧 IO 示意图

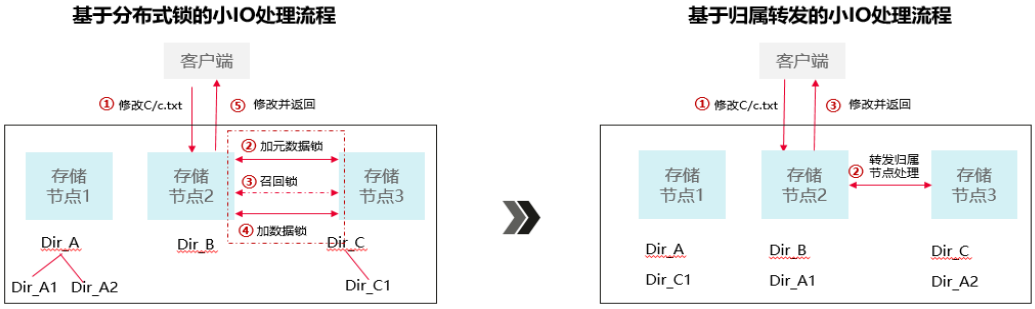


1. 单客户端连接多个存储节点，**提升单流和单客户端带宽**
2. 解决标准协议只能在 Mount 时进行负载均衡的问题，实现 **IO 级负载均衡**
3. 支持 **RDMA** 协议，可以实现更低的时延和更低的 CPU 开销

4. DPC 和 APP 实现**用户态进程间的高速直通**，避免数据从用户态到内核态的多次拷贝，异步 IO 可直接在内核态完成并保证跨节点的数据一致性。同时通过智能预读算法，自动识别文件 IO 访问规律和目录下的海量小文件访问规律等，高效预读文件元数据和数据，无需人工参数调优，即可同时满足多种多样的 HPC 应用的高性能诉求。
5. 将**随机小 IO 聚合为大 IO**，在 DPC 侧计算 EC 冗余后直接下写到持久化节点，提升数据写性能

6.4 免分布式锁设计

图6-7 免分布式锁设计示意



HengShan Stor 系列文件服务实现通过目录分区粒度的一次打散逻辑，同时引入了集中式架构的归属转发机制，小 IO 请求统一由 Owner 节点处理，实现了免分布式锁，小 IO 操作仅需一次消息交互，大幅降低了 CPU 开销和请求时延。

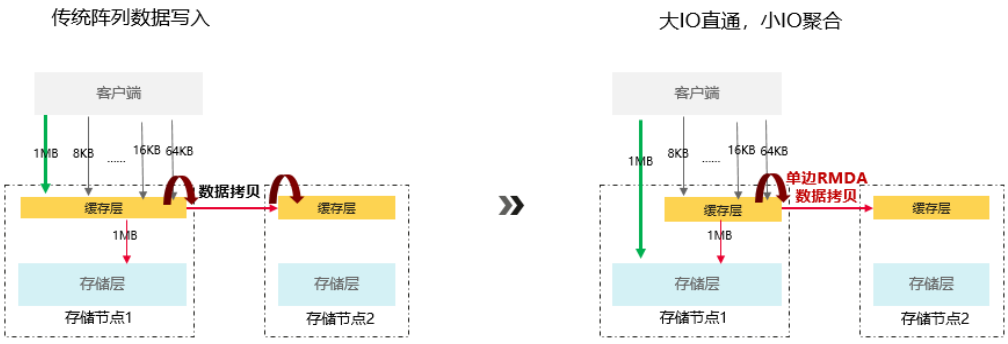
6.5 混合 IO 负载优化

HPC 中的业务负载多种多样：顺序大 IO 的带宽类、随机小 IO 的高 IOPS 类、批量元数据操作的 OPS 类、还有同时访问同一个文件的并行 IO 类，随着传统 HPC 与大数据和智能技术逐渐走向融合，以 ARM、GPU、FPGA 为代表的异构计算进一步提高对存储性能的要求，一套存储如何同时满足极致带宽、极致 IOPS 和极致时延是 HPC 存储性能面临的最大挑战。

HengShan Stor 系列文件服务，通过多维度创新来解决带宽和 IOPS 型业务共存的问题：

- 通过大 IO 直通技术减少网络放大、通过大比例 EC 技术减少磁盘带宽放大，通过基于 FlashLink 技术中的大块顺序写技术实现磁盘 IO 顺序访问，来实现极致的带宽性能。
- 通过多级智能缓存中小 IO 聚合技术、基于单边 RDMA 的技术解决 CPU 算力瓶颈和软件栈深导致的时延问题，实现高 IOPS、低时延。
- 通过 FlashLink 的智能众核技术与端到端的 IO 优先级调度关键技术解决带宽和 IOPS 共存场景下的需求冲突问题。

图6-8 大 IO 直通，小 IO 聚合示意



传统阵列数据写入，所有 IO 都先写入到内存，完成节点镜像后返回给业务，对于小 IO 来说路径最短，时延最好；但对大 IO 带宽型业务，网络和内存带宽均放大一倍，业务带宽只有网络和内存带宽的一半。面向混合负载，大 IO 走直通到盘，减少网络和内存带宽放大，实现最高的带宽性能；小 IO 写缓存聚合，完成镜像后响应，实现最低时延，但是放大了 CPU 调度的次数，进一步加剧了 CPU 的性能瓶颈。HengShan Stor 系列文件系统在此基础上又实现了单边 RDMA 方式的数据拷贝，仅占用主存储节点的 CPU 资源，将总 CPU 开销降低了 20-30%，混合负载下带宽和 IOPS 均能做到最佳。

6.6 GPU 数据读写加速

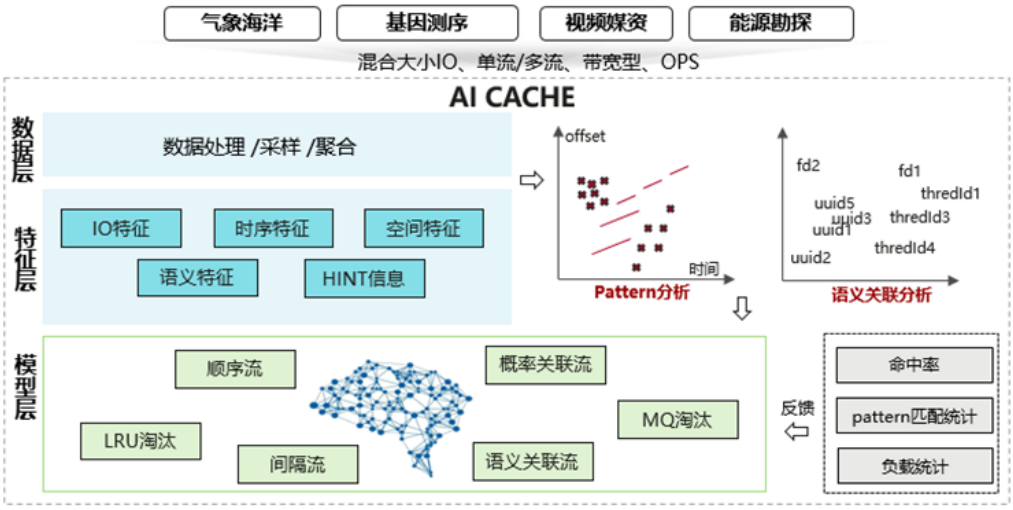
在大数据、HPC、AI 和 ML 应用中，例如图像分类、视频分析、语音识别以及自然语言处理等，应用程序将计算卸载到 GPU 上可更快地完成任务，这对 GPU 的性能有较高的要求。

HengShan Stor 系列存储系统依赖 NFS over RDMA 特性，通过 GPU Direct 功能实现 GPU 直通读写数据加速，即 GPU 可直接高速读写存储系统上的数据，无需借助 CPU 的辅助，即可增加存储系统带宽、降低 CPU 延迟，提升 GPU 数据读写性能。

6.7 AI Cache

HPC、大数据、AI 场景具有高带宽/高 IOPS/高 OPS 混合负载诉求，也面临多负载、多协议、低资源等复杂因素下文件访问乱序、串流等挑战性问题，导致访问时延高，严重影响数据处理效率。

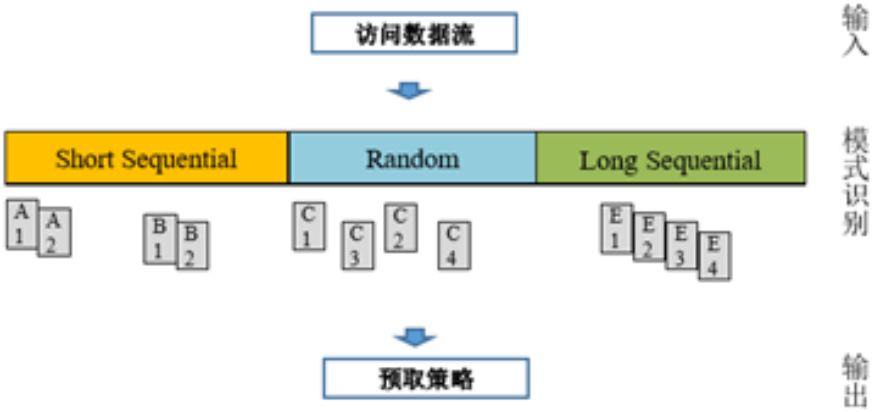
HengShan Stor 系列提供 AI Cache 技术，构建 Workload 智能感知能力，通过学习和挖掘文件 IO 特征、文件语义信息与目录操作规律达成用户行为和存储数据特征的有效关联，实现文件内、跨文件的最优预取和淘汰，大文件小 IO 性能最高提升 15 倍，小文件性能提升 2 倍。



AI Cache 采用多维智能的文件语义预取算法，构建时空、文件语义与目录操作行为等特征作为点、访问为线的相邻连接以及交互连接的特征画像，内置流式机器学习模型，兼具算力、内存资源利用率与高精度双重优势，实现自适应场景的在线策略优化和 IO 访问预测。

- **应用感知 Pattern 识别算法**

  根据文件访问的空间特征、时序特征与应用信息，识别文件内访问的顺序、间隔和随机访问模式，通过重用距离相关性模型自适应计算最佳预取长度，实现同一文件内对不同应用产生的不同 pattern 识别，解决复杂业务模型下的乱序、串流问题。



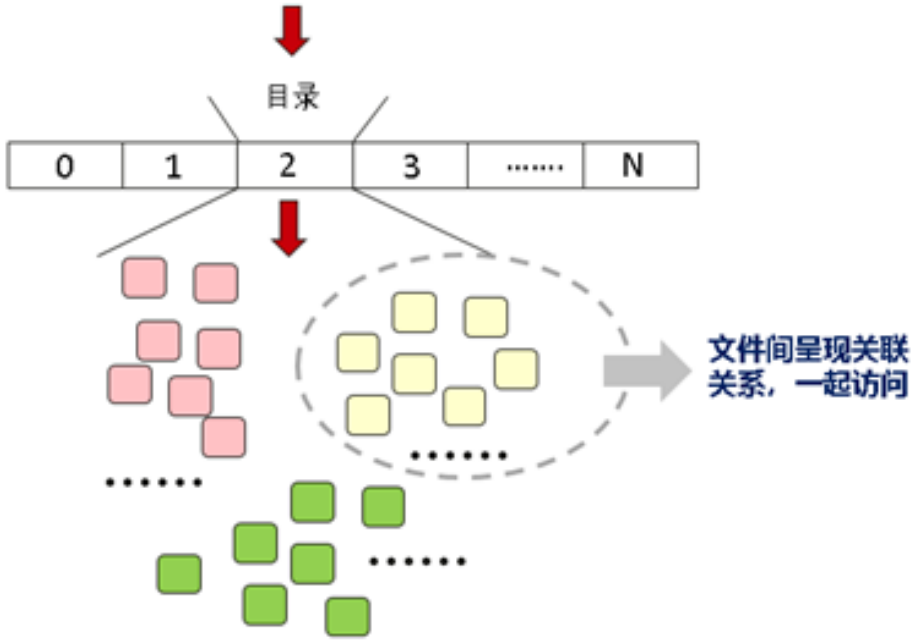
- **语义感知预取算法**

  文件名语义特征挖掘：通过挖掘字符、数字、后缀、日期等文件名特征，采用回归算法学习和推理多维预取策略，算法支持文件名中数字线性变化规律的预取、关联后缀名预测。如下图所示文件访问流中文件名数字按递增变化，在线学习文件名数字递增规律后，预测到下一个可能访问的文件。

  目录操作行为识别：采用目录操作加速算法，通过学习 IO 访问信息识别操作行为，推荐目录序、字典序预取序列，实现目录粒度预读和目录拷贝加速。

- **跨文件关联预测算法**

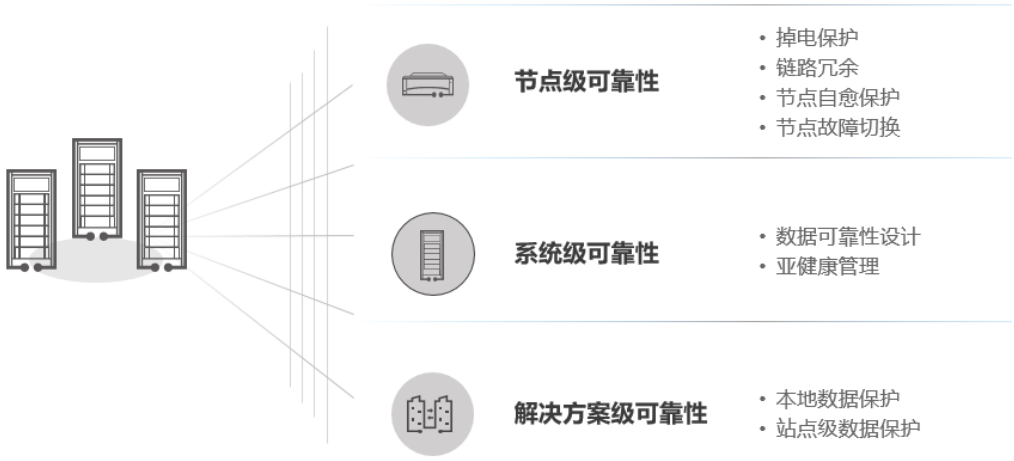
  针对文件名、目录操作无规律场景，构建文件访问之间的概率、写读关联模型，同时在低资源开销下生成紧凑型预测树，实现跨文件的时间、概率关联预取。如下图所示，文件间概率模型增量式学习与迭代，当某一文件被访问后，时间或概率关联最大文件将在后续被访问。



7 系统可靠性设计

HengShan Stor 系列存储系统，通过节点级可靠性、系统级可靠性等专业设计，为用户提供 99.999%的可靠性，再配合容灾解决方案，系统可靠性可达到 99.9999%。对于用户数据，HengShan Stor 系列实现了灵活的数据故障域安全布局和冗余策略，并通过端到端的数据完整性保护和各种故障场景下的数据保护设计，实现了数据信息的高可靠存储和业务处理。

图7-1 HengShan Stor 系列端到端可靠性保证



7.1 模块级可靠性

7.1.1 硬盘可靠性

存储系统中硬盘数量庞大，承载了用户的关键数据。因此，如何及时发现和有效处理硬盘故障，将硬盘故障影响最小化，是存储系统面临的可靠性挑战之一。百信 HengShan Stor 系列存储系统为了提升硬盘的容错能力，进行了一系列容错设计来应对这一挑战。

- 坏道智能扫描

  当硬盘出现坏道时，并不会主动上报，只有在对其位置主动进行读写或校验时才能发现。百信 HengShan Stor 系列存储系统支持在不影响业务和硬盘自身可靠性的前提下，周期性的进行扫描，在坏道出现后可自动将其识别，并立即触发从其他节点上读取冗余数据，然后写入相应位置以达到自愈修复目的。

- 硬盘故障预测

  硬盘经过长时间的工作运行后会出现部件老化等问题，故障率会随时间呈上升趋势。百信 HengShan Stor 系列存储系统通过对硬盘 SMART 信息进行周期性例测，监测会导致硬盘故障的关键指标，通过分析关键指标的变化趋势实现硬盘故障预测。对所有类型的硬盘（主存盘、缓存盘、系统盘与元数据盘），预测出盘即将失效情况下会主动发送告警。

- 慢盘检测与隔离

  硬盘由于老化、Firmware 缺陷、温度异常、震动异常等影响可能会出现 IO 响应速度比正常运行时更慢，当出现部分硬盘响应时间比其他硬盘慢时，则可能会拖累整个存储系统，进而导致使用存储系统的主机应用卡顿或中断。百信 HengShan Stor 系列存储系统在业务 IO 运行过程中，会实时统计一段时间内下发到盘上所有 IO 的平均响应时间，通过智能聚类算法（一种机器学习算法）诊断机制识别慢盘。对于主存盘/缓存盘，系统识别出慢盘后主动对该盘进行隔离并触发数据重建，对系统盘和元数据盘，系统会发送告警。

- 慢 I/O 处理

  硬盘内部或链路部件进行内部异常处理时，可能会导致访问硬盘的个别 I/O 时延异常，进而影响主机 I/O 的响应时间，导致主机业务性能波动。百信 HengShan Stor 系列存储系统对主机业务下发的每个 I/O 均进行了响应时间监控，如超过设定的阈值，对于主存盘/缓存盘，业务读操作通过其他副本或 EC 降级读方式获得数据，写操作通过重新选择其他硬盘并保持冗余写方式快速响应主机，对于系统盘和元数据盘，系统会发送告警。

- 硬盘错误处理

  百信 HengShan Stor 系列存储系统在 I/O 处理过程中主动识别硬盘写保护(WP)，命令终止(ABRT), 磁盘故障(DF)等错误，对于主存盘/缓存盘，硬盘错误会触发故障告警和自动数据重建，对于系统盘和元数据盘，系统会发送告警。

- 硬盘漫游

  通过在磁盘内部记录相关信息，并在磁盘插入时检查，百信 HengShan Stor 系列存储系统支持在同一存储节点内任意硬盘槽位间交换位置，而不影响硬盘身份识别和数据的正常读写，实现“槽位自由”。特别地，百信 HengShan Stor P9550K 是 5U 高密大容量整机，单框 120 个硬盘由 4 个扩展卡（Expander）分别管理，仅支持在同一个 Expander 内的硬盘漫游。

7.1.2 HSSD 可靠性

HSSD 具备数据保护、Die 失效管理、坏块管理、后台巡检和 Trim Mode 等可靠性技术。

7.1.3 网卡可靠性

百信 HengShan Stor 系列存储系统使用 Bond 技术来实现多网口的聚合，从而达到故障冗余和负载均衡目的，当前支持模式包含 active-backup（bond 1）、balance-xor（bond 2）、802.3ad（bond 4）。

除此之外，百信 HengShan Stor 系列存储系统对各种网卡亚健康场景进行了监控和故障切换，包含但不限于如下场景：

- 网口闪断
- 网口协商速率不匹配
- 网卡 PCIe 总线协商速率不匹配
- 网口丢包错包
- 网络时延大

7.1.4 内存可靠性

随着内存容量的增大和内存速率的提升，内存发生故障的概率升高。内存系统级的容错，主要是在内存发生错误时，进行系统层面的修复和隔离策略，提升系统容错能力，降低故障率。

系统正常运行过程中，内存控制器会对内存进行周期性巡检，发现内存可纠正错误（CE）后，通过 ECC 纠错算法，恢复数据后重新写入内存，避免错误进行积累。发现内存不可纠正错误（UCE）后，系统通过内存错误在线页面隔离技术，实现节点继续运行。

数据读写过程中，内存出现单 BIT 错误后通过 ECC 将此内存区域修复，然后对外继续提供服务。当内存硬件存在故障，可能会导致持续纠错，导致 CE 中断持续占用 CPU。为了避免此类情况发生，CPU 对内存可修复错误中断进行统计，当超过门限后，OS 针对频繁出现 CE 错误的内存区域，操作系统会以 PAGE 为单位进行隔离。如果此内存未被分配，操作系统和内存管理模块将不再分配这部分内存，如果正在使用，则会等此部分内存释放回收后，内存管理模块后不再分配这部分内存，避免频繁 ECC 出错。

数据读写过程中，如用户态进程访问到内存 UCE 错误，会杀掉并重新拉起进程。同时，支持内存 UCE 错误的在线页面隔离功能。当内存 UCE 发生时，会把对应的地址记录到非易失性介质里面。在进程重新拉起时，会根据记录的 UCE 地址，进行内存页面隔离，避免出错的页面再次被使用，造成进程反复异常。

7.2 节点级可靠性设计

7.2.1 掉电保护

系统运行过程中可能会出现服务器掉电可能，百信 HengShan Stor 系列存储系统使用保电介质来保存元数据和缓存数据，以防掉电而丢失元数据和用户数据。

HengShan Stor 系列存储支持 SSD 作为保电介质。HengShan Stor 结构化服务将 SSD 作为缓存盘形态的存储设备，程序运行过程中会把元数据和缓存数据写入保电介质中，节点异常掉电并重启后，系统自动恢复保电介质中的元数据和缓存数据。HengShan Stor 非结构化服务是通过 SSD 作为备电盘形态的存储设备，程序在运行过程中会先将元数据/缓存数据写入内存中，在节点异常掉电时，通过存储设备的备电技术将缓存数据刷入 SSD 备电盘，节点异常掉电并重启后，系统自动从备电盘恢复元数据和缓存数据。

7.2.2 链路可靠性

百信 HengShan Stor 系列存储系统使用基于链路聚合的技术实现传输链路冗余，并在链路故障和亚健康时进行链路切换（或隔离）操作，以确保业务的连续性。链路聚合是将多个物理链路组合成一个单一逻辑链路供应用使用，在某条物理链路故障时将通讯切换到剩余链路。

集群支持网络隔离，可以做到管理和存储平面在网络上物理分离（划分不同 vlan），并且为不同的组件，因此可以保证管理平面故障，不会影响业务；业务平面拥塞，不会影响管理，每个功能平面可在各自平面内部单独完成故障检测、修复和隔离，互不影响。支持以下网络配置方式，

- 两网配置：存储网络（只有前端）与管理网络。
- 三网配置：前端存储网络，后端存储网络和管理网络。

在上述网络配置下，当集群配置以太网络，IB（InfiniBand）网络或者 RoCE（RDMA over Converged Ethernet）时，链路可靠性分别通过下述方式保证：

1. 集群使用以太网络时，需要节点存储网络前端（或后端）配置两个以太网口，并分别连接两个交换机。

   集群节点存储网络的两个网口通过操作系统提供的 Bonding 驱动（Bonding Driver）实现链路聚合，支持配置的 bond 模式有：

   - Bond1，active-backup
   - Bond2，balance-xor (active-active)
   - Bond4，802.3ad (active-active)

   系统只有在配置使用 bond1 主备（active-backup）模式下时支持亚健康网口主动切换功能。在该模式下，聚合网口对应多个物理网口，其中只有一个网口是当前的活动网口，其他都是备用网口，当活动网口故障或者亚健康时，系统自动将活动网口切换为备用网口。
2. 集群使用 IB 网络时，需要节点存储网络前端（或后端）配置两个 IB 网口，并分别连接两个交换机。集群节点存储网络的两个 IPoIB 网口通过操作系统提供的 Bonding 驱动（Bonding Driver）实现链路聚合。系统业务（IO 流）不受限于 IPoIB 的 bond 配置，可以同时通过两个 RDMA 设备进行通讯，使用的是同时在线

（active-active）模式。当 IB 网口故障或者亚健康时，系统将对应 RDMA 设备进行隔离；并且对于 IPoIB 的聚合网口，如果异常网口是 IPoIB 聚合网口的活动网口，则将活动网口切换为备用网口。

3. 集群使用 RoCE 网络时，需要节点存储网络前端（或后端）配置两个以太网口，并分别连接两个交换机。系统通过操作系统提供的 Bonding 驱动和 RoCE 网卡驱动（与网卡固件等配合）实现对链路聚合的支持，支持配置的 bond 模式有：
   - Bond1，active-backup
   - Bond2，balance-xor (active-active)（RDMA 的选路方式由网卡决定）
   - Bond4，802.3ad (active-active)（RDMA 的选路方式由网卡决定）

7.2.3 节点自愈保护

7.2.3.1 高温保护

百信 HengShan Stor P9950K/P9550K 系列存储系统对环境温度和系统内的关键温度点都进行了实时监控，会根据各监控点的信息，对风扇进行动态调速，保证系统及各组件工作在正常的温度范围内。

当由于环境温度增高或者器件异常时，风扇调速无法保证系统或者组件工作在正常范围时，产品设置两级门限，超过一级门限时，进行过温告警。如果持续升高，超过二级门限时，进行系统掉电处理，避免存储系统持续高温，导致部件损坏。

百信 HengShan Stor P9920K/P9520K/P9540K 系列存储系统对硬盘温度进行实时监控，超过二级门限时，停止对硬盘进行访问，避免存储系统持续高温，导致硬盘损坏。

7.2.3.2 硬件自愈保护

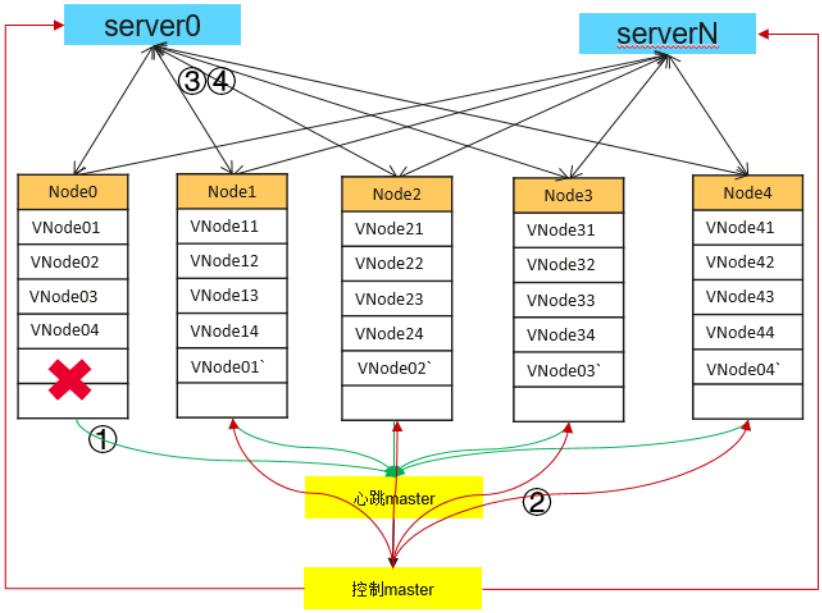
由于软件 BUG 或硬件异常导致的线程无法正常执行或 CPU 无法正常工作统称调度异常，百信 HengShan Stor P9550K/P9950K 系列存储系统在 CPU 外部有一个专用芯片（看门狗芯片 CPLD），用于看护 CPU 僵死场景。当检测到 CPU 僵死时，自动进行节点复位修复，保证存储系统的冗余度。

7.2.4 节点故障切换

节点故障切换是指当节点出现故障时（如操作系统复位、CPU 故障等），能够快速检测到故障并将其上承载的业务切换到其他正常的节点上，以确保在系统设计的冗余范围内故障不影响存储系统的可用性。同时在故障节点恢复正常并重新上线后，能够自动将其他节点上的业务均衡/回切到该节点，以恢复系统的可用性并提供更高的性能能力和更多存储空间能力等。

百信 HengShan Stor 系列存储系统通过心跳机制来检测节点故障，多节点多模块并发业务接管、延迟加载、批量加载、内存索引快速重建等机制来减少节点故障接管时长，进而缩短 IO 归零时长。

图7-2 节点故障快速切换并发业务接管示意图



1. 进程故障检测通知①：进程故障心跳断开通知。
2. 存储节点并行接管② ：软件层面，每个物理节点上有 4 个逻辑的 vnode。物理节点故障后，故障节点上业务以 vnode 为粒度在正常物理节点上进行快速接管。
3. 计算节点快速切换 DPC③④：主动通知 DPC 分区视图变更，DPC 根据视图将业务切换到新的存储节点。
4. 计算节点快速切换标准客户端④：标准客户端支持 IP 漂移，自动切换到新的存储节点。

7.3 系统级可靠性

7.3.1 数据可靠性设计

7.3.1.1 数据冗余保护

百信 HengShan Stor 系列存储系统提供了数据跨节点/机柜的保护能力，在多个硬盘、节点、机柜故障时也能继续提供服务。

7.3.1.1.1 数据安全布局策略

百信 HengShan Stor 系列存储系统提供支持灵活的数据布局策略，包括 vNode 级安全布局、节点级安全布局和机柜级安全布局，即数据跨 vNode/节点/机柜布局，在多个硬盘、vNode、节点、机柜故障时也能继续提供服务，其中非结构化服务支持 vNode 级安全布局（EC）、节点级安全布局（EC），结构化服务支持虚拟节点（vNode）级安全布局（EC）、节点级安全布局（EC 或多副本）、机柜级安全布局（EC 或多副本）。特别地，结构化服务支持节点级安全布局在线转换成机柜级安全布局。非结构化服务支持虚拟节点级安全布局通过扩容节点在线转换成节点级安全布局。

- 虚拟节点（vNode）级安全布局

  基于 HengShan Stor P9550K/P9546K 的专属单框双节点硬件架构，通过将单个节点逻辑划分为多个 vNode，将数据及其冗余分散到不同的 vNode，只要同时故障 vNode 数小于等于冗余数，就可自动恢复数据，业务不中断，数据不丢失。针对单框中的某个节点，即使同时故障 vNode 数大于冗余数（即出现单节点故障），仍可通过节点快速倒换技术将数据盘和业务切换到对端节点，实现业务不中断，数据不丢失，可靠性不降级。

- 节点级安全布局

  将数据及其冗余分散到不同节点，只要同时故障节点数小于等于冗余数，就可自动恢复数据，业务不中断，数据不丢失。

  EC 配置下，不同的 EC 分条分布在不同的服务器上，如 8 服务器存储池配置 EC 4+2 时，支持 2 服务器故障仍能继续提供服务。

- 机柜级安全布局

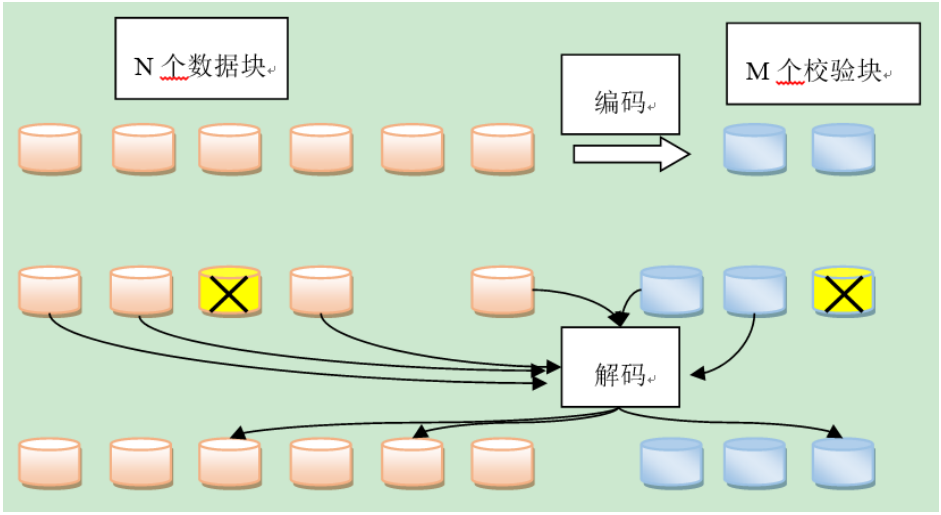
  结构化服务支持将数据及其冗余分散到不同机柜，只要同时故障机柜数小于等于冗余数，就可自动恢复数据，业务不中断，数据不丢失。

  EC 配置下，不同的 EC 分条分布在不同的机柜中的服务器上，如 8 机柜存储池配置 4+2 时，支持 2 机柜故障仍能继续提供服务。

7.3.1.1.2 Erasure Coding

百信 HengShan Stor 系列存储系统支持 Erasure Coding（以下称 EC）模式来保证数据的可靠性，支持通过不同的编码方式在存储空间利用率和数据可靠性之间取得平衡。写入百信 HengShan Stor 系列存储系统的数据，会按照固定大小划分为一个条带，将数据切分为多个原数据分片，然后对每 N 个原数据分片，计算得到 M 个校验分片，最终这 N+M 个条带组成一个分条，写入到系统中。当系统出现故障，丢失了其中的某些分片时，只要一个分条中丢失的分片数目不超过 M，就可进行正常的数据读写。通过数据恢复算法，丢失的条带可从剩余条带中计算得到。在这种方式下，空间的利用率约为 N/(N+M)，数据的可靠性由 M 值的大小决定，M 越大可靠性越高。

图7-3 百信 HengShan Stor 系列存储系统 EC



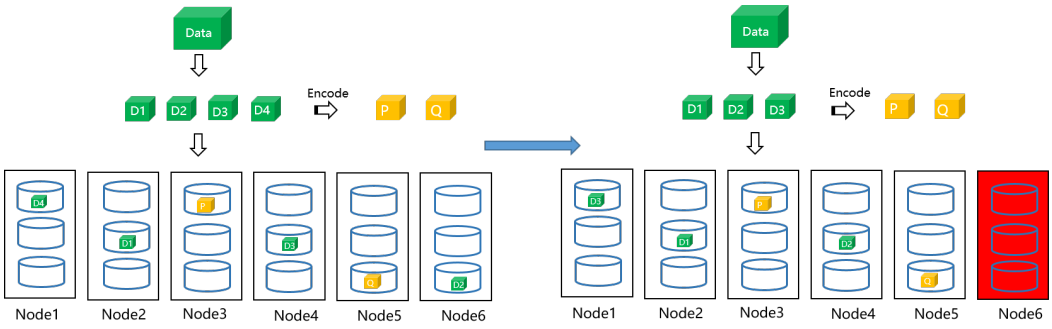
百信 HengShan Stor 系列存储系统采用的是 LDEC（Low Density Erasure Coding）算法，基于 XOR 和伽罗华域乘法相结合的一种 MDS 阵列码，编码最小粒度 512B，支持 intel 指令加速，支持各种主流配比。当节点数小于配比的 M+N 总值时，可以配置 M+N:1 模式，此时一个节点可能存在同一分条的多个条带，此时即使 M=2、3 或者 4，也只允许一个节点故障，引起可靠性下降。

EC 的具体信息请参考《百信 HengShan Stor 系列 弹性 EC 技术白皮书》。

7.3.1.1.3 数据写不降级

传统的 RAID 方式在 RAID 成员故障时，为保证 IO 不中断，只能降级写，损坏的盘只记录操作 LOG，这降低了可靠性。百信 HengShan Stor 系列存储系统的 EC 则通过“缩列”的方式实现 IO 不中断的同时，可靠性级别不变。

图7-4 单节点故障 EC 缩列



如上图所示，6 节点配置了 4+2，在遇到 1 个节点故障时，EC 的冗余配比模式自动从 4+2 缩列成 3+2，实现了新写数据 IO 不中断的同时，保证了业务的连续性。

对于 N+M 的 EC 来说，缩列将保持 M 不变，将 N 按照可以选择到的最大的 N 进行缩列，比如 4+2 6 节点故障一个节点缩列成 3+2,再故障一个节点缩列为 2+2。缩列只能保证新写数据缩列，系统中已经写入的数据并不会缩列。系统能缩列的最小比例为 N/2。

EC 缩列技术细节请参考《百信 HengShan Stor 系列 弹性 EC 技术白皮书》。

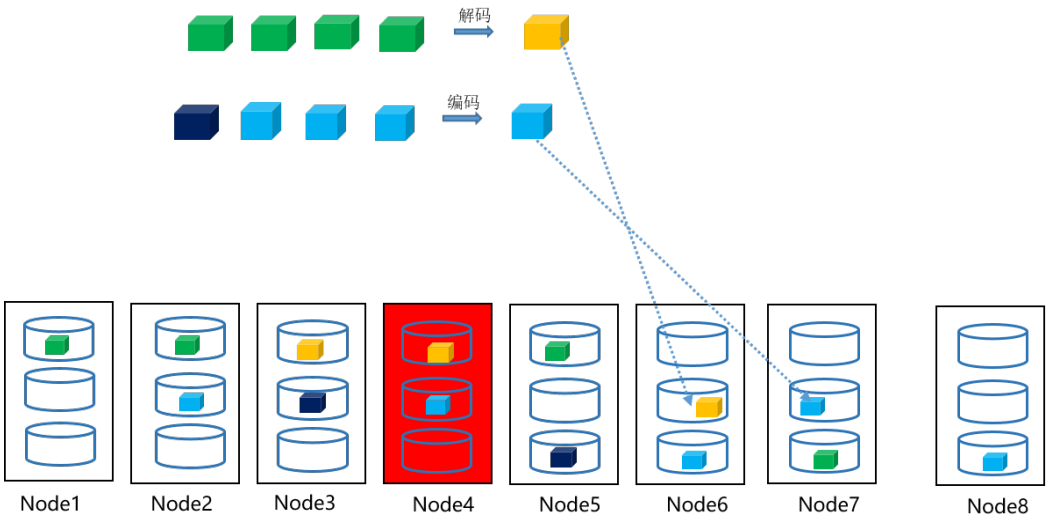
7.3.1.1.4 快速数据重构

当盘或服务器短时间离线（如盘误拔、服务器重启）后又接入存储系统时，存储系统不会进行全盘数据重构，而是直接进行状态协商（如有效数据位置），协商完成后便直接使用上面的数据；此方式相比还需要搬移新增数据到恢复的盘上而言，对系统性能的影响更小且数据可靠性更高。当盘或服务器离线时间太长时，为了数据可靠性和业务可用性，仍然会触发数据重构。

百信 HengShan Stor 系列存储系统存储池采用分区打散机制，即 EC 分条数据会按照策略打散到不同的服务器的不同硬盘。当百信 HengShan Stor 系列存储系统检测到硬盘或者节点硬件发生故障时（长时间离线），自动在后台启动数据修复。

由于 HengShan Stor 系列存储系统采用虚拟化方式，每块盘都会有一部分空间与其他盘组成 EC，一旦发生硬盘故障，参与重构的硬盘数量非常多。同时热备空间不是来自于一块硬盘，而是随机分配在存储池中。

图7-5 百信 HengShan Stor 系列数据重建示意



如上图所示，NODE4 节点故障，重构时 NODE6 和 NODE7 同时参与重构，提高重构速度。

HengShan Stor 存储系统中采用 LDEC 编码，通过最大化公共数据编码校验列，在重构时尽量多的读取公共数据块，重构相对于 Reed-Solomon 编码重构时需要的网络带宽为原来的 0.75，重构时间相对于 Reed-Solomon 编码缩短了 32.45%。

缓存盘故障增量重构：

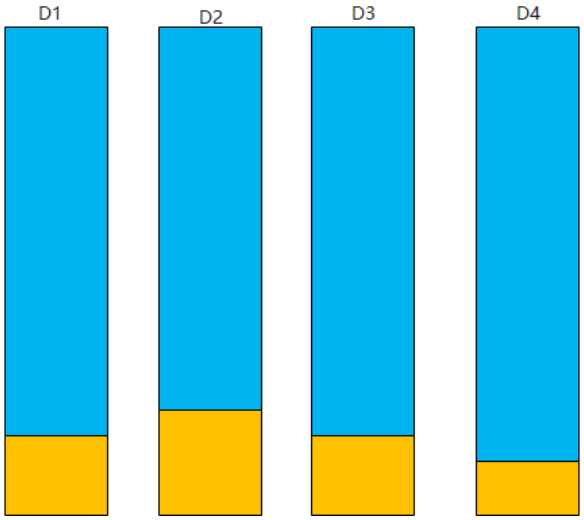
在 HDD 做主存，SSD 做缓存情况下，如果 SSD 发生故障后更换缓存盘，业界一般有两种数据重构机制：

1）全盘重构：缓存盘中的数据情况是未知的，重构时需要将写入 HDD 的数据做全盘恢复，参与重构的数据量巨大，重构时间通常按天计算。

2）增量重构。缓存盘中的数据情况是明确的，重构时仅需要对缓存盘中未写入 HDD 的数据做增量恢复，参与重构的数据量较小，通常可以在短时间内快速完成重构。

百信 HengShan Stor 系列存储系统开创性地支持缓存盘故障增量重构的方式。该方式基于 PLOG 语义，通过记录每条 plog 数据已经从 SSD 写到 HDD 的情况，当缓存盘故障时，可以从 HDD 加载 PLOG 元数据，重构时只需要对缓存盘中未写入 HDD 的数据做增量恢复。由于 SSD 缓存中未刷盘的数据只有 HDD 空间的百分之一左右，相较于全盘重构的方式，重构时间可以节约几十倍，重构效率得到了大大提升。根据实测效果，百信 HengShan Stor 系列存储系统缓存盘故障数据重构时间可以缩短至小时级别。

图7-6 增量数据重构示意图



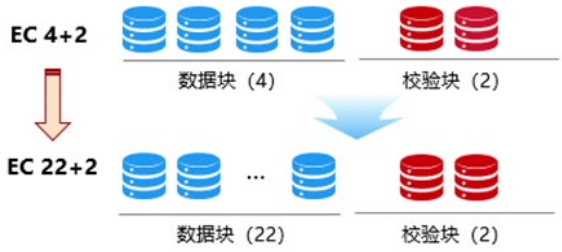
如上图所示，D1-D4 分别对应 4 个数据分片 PLOG，其中，蓝色部分代表已经从缓存盘 SSD 写入到主存盘 HDD 中的数据，黄色部分代表未写入到主存盘 HDD 中的数据，并在该分片 PLOG 的元数据中记录已经刷盘的位置。如果此时数据分片 D1 对应的缓存盘发生故障，在更换缓存盘后，按照增量数据重构的机制，只需要重构恢复图中黄色部分（已丢失）的数据即可。

7.3.1.1.5 EC 动态扩列

EC 的配比与存储池节点数强相关，如 3 节点只能配置 EC 4+2:1，4 节点只能配置 EC 6+2:1，在小规模存储池配置下，EC 容量利用率远低于大存储池的大比例 EC。

对存储池进行扩容后，为获取更高的空间利用率，系统支持 EC 动态扩列功能，如 EC 4+2 经过一次或多次扩列，可最终扩为 EC 22+2，利用率从 66.6%提升到 91.67%。同时，由于校验分片数 M 不变，扩列后可靠性不降低。

图7-7 动态扩列示意图



系统提供了灵活的扩容选项，可支持扩容不扩列，扩容扩列。扩列可支持推荐配置（根据节点数计算的最佳 EC 配比）或指定 EC 配比。

启动扩容扩列操作后，新扩容容量和原有空间中未写入数据空间立即转换为新的 EC 配比的容量，已写入的数据空间由后台任务综合考虑业务压力，容量水位等因素，智能

转换为新的 EC 配比的容量。相比节点扩缩容等操作触发的数据重构，EC 扩列产生的后台重布局耗时较长，但对用户业务影响小。

7.3.1.2 数据完整性保护

百信 HengShan Stor 系列存储系统通过 IO 实时端到端数据完整性校验、后台周期性数据校验以及损坏数据实时自愈纠错机制来解决静默数据破损场景。

7.3.1.2.1 IO 路径数据完整性保护

数据在存储系统内部传输中，经过了多个部件、多种传输通道和复杂的软件处理，其中任意一个错误都可能会导致数据错误。如果这种错误无法被立即检测出来，而是在后续访问数据过程中才发现数据已经出错，这种现象叫做静默数据破坏（Silent Data Corruption）。由于静默数据破坏无法实时检测出来，导致被破坏的数据恢复难度很大，甚至不可恢复。

产生静默数据破坏的原因有很多，主要有以下几类：

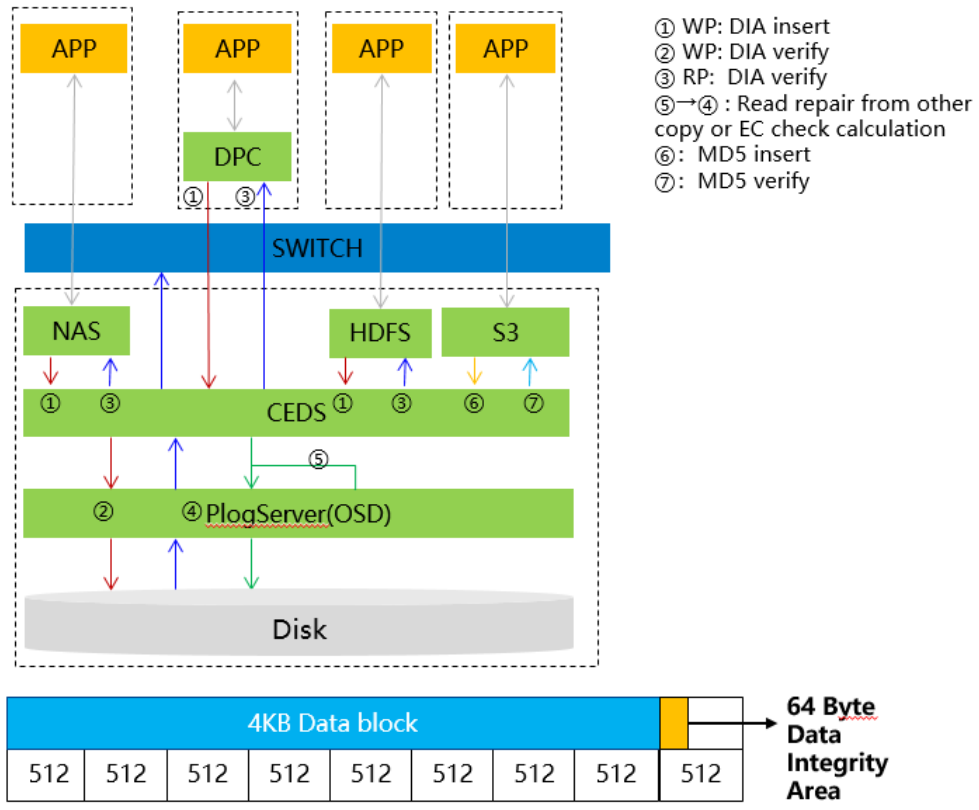
- 硬件故障：内存、CPU、硬盘、FC 或 SAS 链路等。
- Firmware 错误：HBA、硬盘等。
- 软件 bug：产品软件、操作系统、应用程序等。

硬件故障导致的数据错误通常能通过 ECC 或 CRC 等校验发现，而 Firmware 错误和软件 bug 更易产生静默数据破坏。

百信 HengShan Stor 系列存储系统提供 IO 级端到端的数据完整性保护方案，能够有效检测跳变、读写偏等各种静默数据破坏场景，当检测到数据静默破坏后会实时对数据使用冗余进行纠错自愈，避免数据损坏扩散。

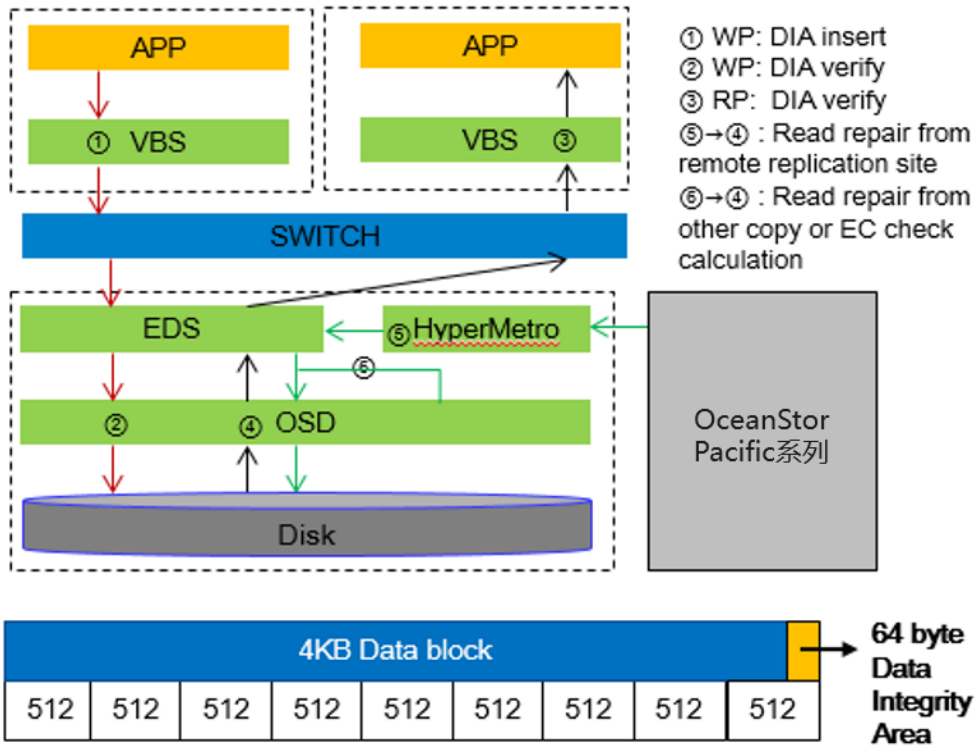
IO 路径关键静默数据错误检测位置以及磁盘布局，以文件服务为例，详见下图。

图7-8 文件服务 IO 路径关键静默数据错误检测位置以及磁盘布局



以结构化服务为例，详见下图。

图7-9 结构化服务 IO 路径关键静默数据错误检测位置以及磁盘布局



盘扇区大小支持 512、520。

DIA，即 Data Integrity Area，数据完整性校验区域。

BLOCK/NAS/HDFS/S3 均支持端到端校验，Object 支持对象级 MD5 校验。

7.3.1.2.2 持续化数据周期性校验

HDD/SSD 上的数据会由于器件老化、电磁/信号干扰、工艺缺陷等原因导致静默数据破坏。通过周期性数据校验可以提前识别风险并进行处理，能有效防止静默数据破坏累积导致数据丢失。

百信 HengShan Stor 系列存储系统采用了后台自适应周期性校验方式来防止数据出现错误，流程如下：

1. 下发读 IO，从 HDD/SSD 上读取一段数据
2. 计算读取的 Data Block 的 DIA'，并与读取上来的 DIA 进行比较
3. 如果 DIA'与 DIA 不相等，则触发再次读校验，以排除由于链路导致的临时性不一致，如果仍不相等，则插入数据纠错自愈队列进行后台自愈写修复。
4. 后台校验任务可以通过手动方式调整速率。其触发的 IO 优先级低于主机业务，当有主机业务时优先处理主机业务，从而不影响主机业务。

7.3.1.2.3 损坏数据纠错自愈

无论在主机 IO 还是后台周期性 IO 识别到静默数据破坏时，均会触发自动的用户无感知的损坏数据纠错自愈机制， 利用本系统内其他节点上存储介质上的冗余数据进行纠错自愈。

7.3.1.3 数据误删恢复

百信 HengShan Stor 系列存储系统非结构化服务提供了基于文件系统/dtree 的回收站，在用户删除文件时并不真正删除数据，而是文件在后台移入到回收站目录，以便用户找回数据。回收站功能对非结构化命名空间内的所有文件/对象都有效。

回收站的具体信息请参考《百信 HengShan Stor 系列 回收站特性技术白皮书》。

7.3.2 亚健康管理

亚健康，也称 Fail slow，是指对应硬件可以正常运行但性能低于预期的一种状态。导致亚健康的原因非常多，包含但不限于 Firmware Bug、硬件自身设计缺陷、温度、环境（如震动）、配置错误等。

多种硬件均有可能进入到亚健康状态，包含但不限于盘（SSD/HDD）、网络、CPU、内存等，一旦某硬件进入到亚健康状态，如果存储系统未采取有效监控和容错，则极有可能会导致存储系统响应主机的时延增大、IOPS/BPS 降低，甚至导致无法响应主机，进而导致主机业务中断。HengShan Stor 系列存储系统对盘、网络、服务（涵盖 CPU\内存等）等进行全面的亚健康状态监控，并进行智能诊断后实现自动隔离，实现单部件亚健康存储系统性能无感知，让主机享受一致性性能体验。

7.3.2.1 硬盘亚健康管理

百信 HengShan Stor 系列存储系统对其使用的主存盘、缓存盘、系统盘、元数据盘均实现了全方位的亚健康状态监控，主要监控内容包含但不限于：

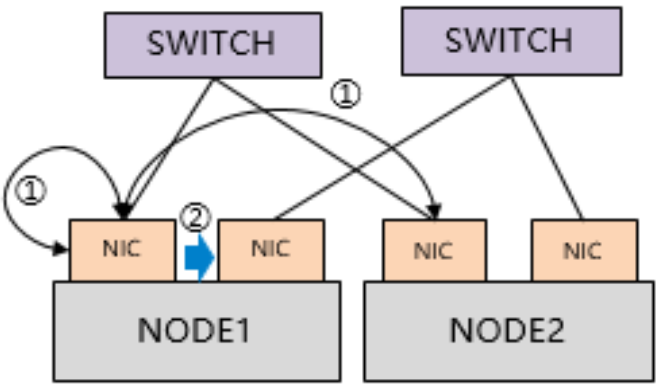
- SMART 信息（如硬盘扇区重映射数超过门限、读错误率统计超标等）
- IO 时延（含单 IO 级时延、IO 平均时延等）
- IO 错误（含静默数据错误数等）

7.3.2.2 网络亚健康管理

网卡降速，丢包/错包率增加，协商速率不匹配等都会导致集群网络性能降级，进入亚健康状态。系统通过检测网络资源状态的变化，定位受到网络亚健康影响的节点，进行 bond 主备切换或者节点隔离。基本原理是：

- 多级检测机制：节点本地网络快速检测闪断、错包、协商速率低等异常。并智能选择节点自适应发送探测包，识别链路时延异常和丢包等问题。
- 智能诊断：结合组网模型和异常信息进行智能诊断，识别网口/网卡/链路等异常。
- 逐级隔离与预警：根据诊断结果进行网口隔离、链路隔离、节点隔离等并上报告警。

图7-10 网络连接亚健康触发本地网口切换示意



以网络连接亚健康触发本地网口切换为例：

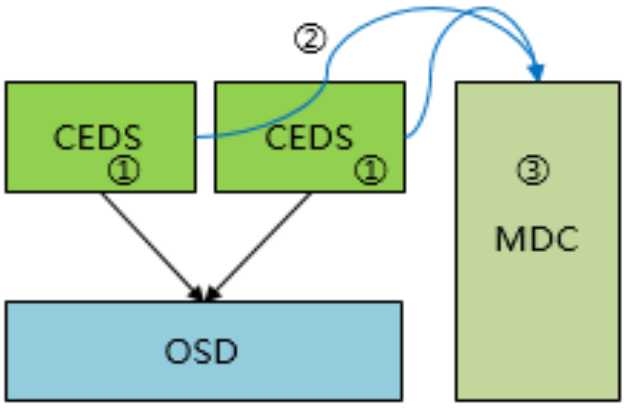
1. 节点 1 持续向集群内其他节点发送探测包，检测是否有丢包或时延增高的现象。
2. 节点 1 发现当前网口向多个目标节点发送的探测包都出现异常，进行网口切换的操作；网口切换后，网络服务恢复正常。

7.3.2.3 服务亚健康管理

分布式集群节点在运行过程中出现软硬件问题是普遍现象。由于节点软硬件问题导致节点进入亚健康状态，比如 CPU 降速，内存反复纠错导致访问降速等；在这种场景下，整系统服务时延受到单个节点影响而降级。针对这一类问题场景，系统通过收集时延信息检测出处于亚健康状态的节点，对问题节点或节点的问题资源进行隔离。基本原理是：

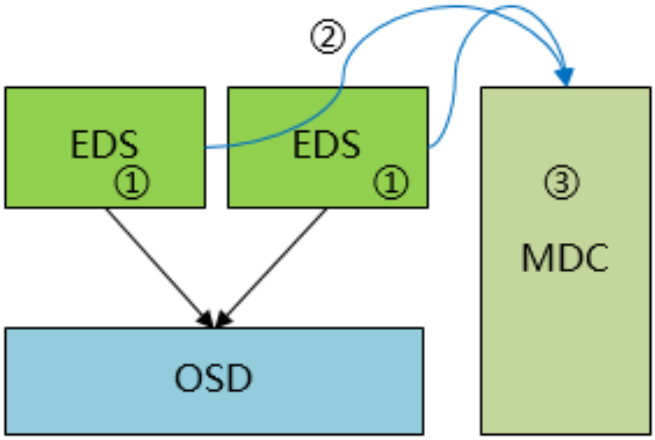
- 跨进程/服务检测：A 访问 B，在 A 上统计访问 B 的 IO 时延，如时延超阈值则上报综合诊断。
- 智能诊断：结合各个进程/服务上报的异常时延，使用基于大多数的判断、聚类算法等诊断出时延异常的进程/服务。
- 隔离与预警：将诊断出异常的进程/服务上报控制节点进行隔离（业务分摊到集群内的其他进程中）并上报告警。

图7-11 非结构化服务亚健康检测和隔离示意



1. CEDS 进程检测到 OSD 进程的时延，统计时延是否有持续异常升高的现象。
2. CEDS 检测到某 OSD 服务时延异常，上报 MDC。
3. MDC 判断是否大多数访问 OSD 的 CEDS 都上报该 OSD 服务时延异常，满足条件则对此 OSD 启动隔离操作。

图7-12 结构化服务亚健康检测和隔离示意



1. EDS 进程检测到 OSD 进程的时延，统计时延是否有持续异常升高的现象。
2. EDS 检测到某 OSD 服务时延异常，上报 MDC。
3. MDC 判断是否大多数访问 OSD 的 EDS 都上报该 OSD 服务时延异常，满足条件则对此 OSD 启动隔离操作。

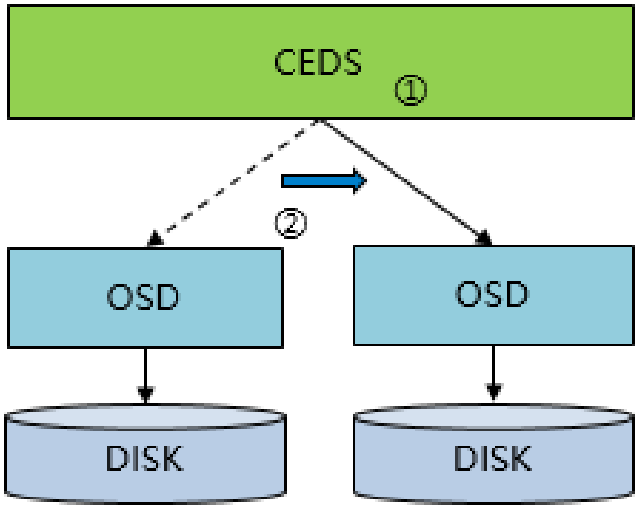
7.3.2.4 快速换路径重试机制

百信 HengShan Stor 系列存储系统设计实现了专有的快速换路重试特性，即 Fast fail 特性，确保单点亚健康的 I/O 时延可控，主要通过如下两种机制实现：

- IO 平均时延检测：检测 IO 平均时延是否超阈值时间未返回，如未返回则启动换路重试。

- 换路重试：对于读 IO 则读其他副本或降级读，对于写 IO 则在其他盘上重新分配空间来存放数据。

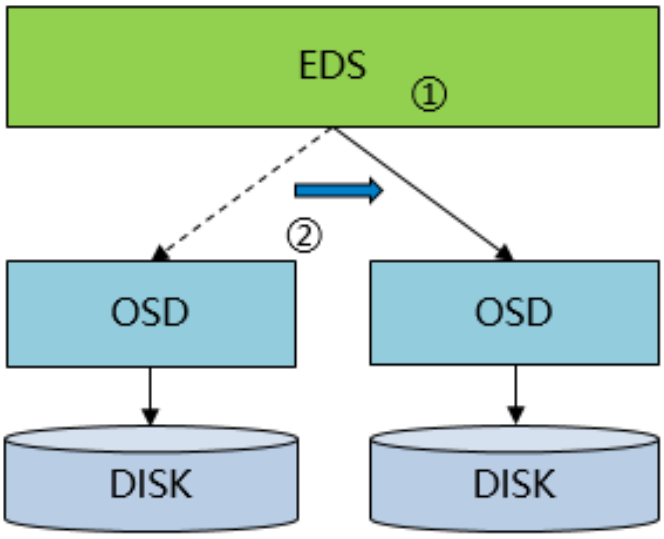
图7-13 非结构化服务快速换路重试示意



非结构化服务慢盘触发 fast-fail 换路重试为例：

1. CEDS 统计下发到盘的 IO 时延，通过聚类算法挑出访问时延比正常盘高的盘。
2. CEDS 对异常盘进行标记，将读写 IO 切换到其他健康盘，恢复读写服务时延。

图7-14 结构化服务快速换路重试示意



结构化服务慢盘触发 fast-fail 换路重试为例：

1. EDS 统计下发到盘的 IO 时延，通过聚类算法挑出访问时延比正常盘高的盘。
2. EDS 对异常盘进行标记，将读写 IO 切换到其他健康盘，恢复读写服务时延。

7.4 解决方案级可靠性设计

7.4.1 本地数据保护

本地数据保护通过在站点集群内进行数据备份保证业务的连续性。

百信 HengShan Stor 系列存储系统本地数据保护使用存储快照来实现。快照是对指定数据集合的一个完全可用的拷贝，该拷贝包含源数据在拷贝时间点的静态映像。快照生成后可以被主机读取，也可以作为某个时间点的数据备份。百信 HengShan Stor 系列存储系统快照的主要优点是：

- 快照与源数据共享存储空间，无需为快照规划独享的存储空间。
- 通过快照可生成的多份数据副本，快照之间相互独立且可供其他应用系统直接读取使用，例如，应用于数据测试、归档和数据分析等多种业务。这样既保护了源数据，又赋予了备份数据新的用途，满足企业对业务数据的多方面需求。

创建快照，就相当于创建了一份受保护的只读版本。当继续进行写、删除等操作时，原有的数据空间将会被快照保护，保持不变。新的数据将以 ROW 的方式写入新的存储空间中。

快照技术的详细原理和介绍，非结构化服务请参考《百信 HengShan Stor 系列 快照特性技术白皮书》，结构化服务请参考《百信 HengShan Stor 系列 快照特性技术白皮书（块）》。

7.4.2 站点级数据保护

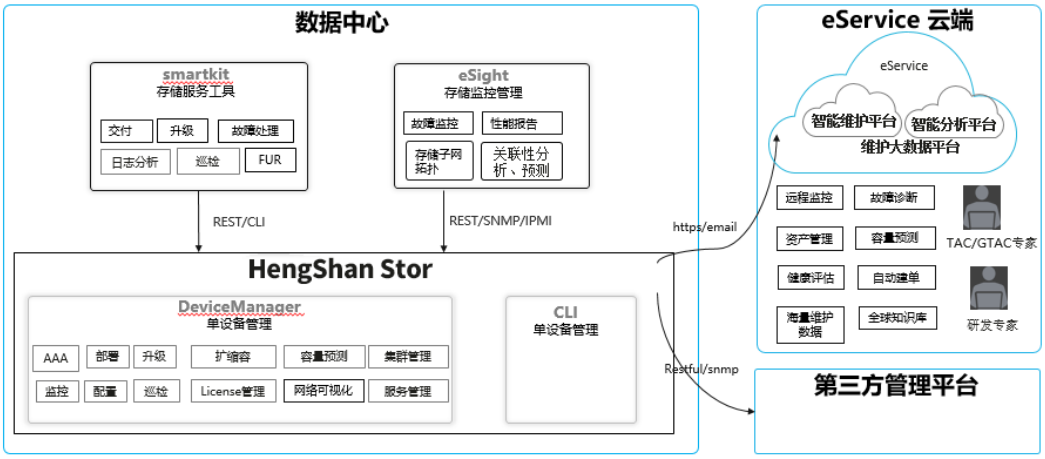
站点级数据保护通过站点间的数据复制技术保证业务的连续性。HengShan Stor 系列非结构化服务支持通过异步复制技术来构建容灾解决方案。HengShan Stor 系列结构化服务支持通过异步/同步复制与双活技术构建容灾解决方案，保证业务连续性。

HengShan Stor 系列存储服务的异步复制，对主从两端存储系统的数据进行周期性同步，实现系统容灾，从而最大限度减少由于数据远程传输的时延而造成的业务性能下降。

HengShan Stor 系列的结构化服务双活架构是基于两套百信 HengShan Stor 系列块存储集群构建 Active-Active 双活容灾关系，基于两套百信 HengShan Stor 系列的卷虚拟出一个跨站点的虚拟卷，该卷的数据在两个存储集群之间实时同步，且两套存储集群能够同时处理应用服务器的 IO 读写请求，面向应用服务器提供无差异的 Active-Active 并发访问能力。任意一个数据中心故障，数据零丢失，业务能迅速切换到另外一个站点运行，保证业务连续性。

8 系统可服务性设计

图8-1 HengShan Stor 系列管理示意图



HengShan Stor 系列支持文件、对象、大数据与块四种服务，非结构化服务支持共池部署。HengShan Stor 系列通过自带集群管理软件 DeviceManager 完成集群的管理工作，DeviceManager Client 提供初始化向导配置，引导用户完成安装管理节点、增加存储节点、配置存储网络、安装存储节点、创建控制集群。同时提供丰富的 API 接口，如 RESTful API、SNMP，支持 OpenStack SMI-S，方便第三方管理平台集成。

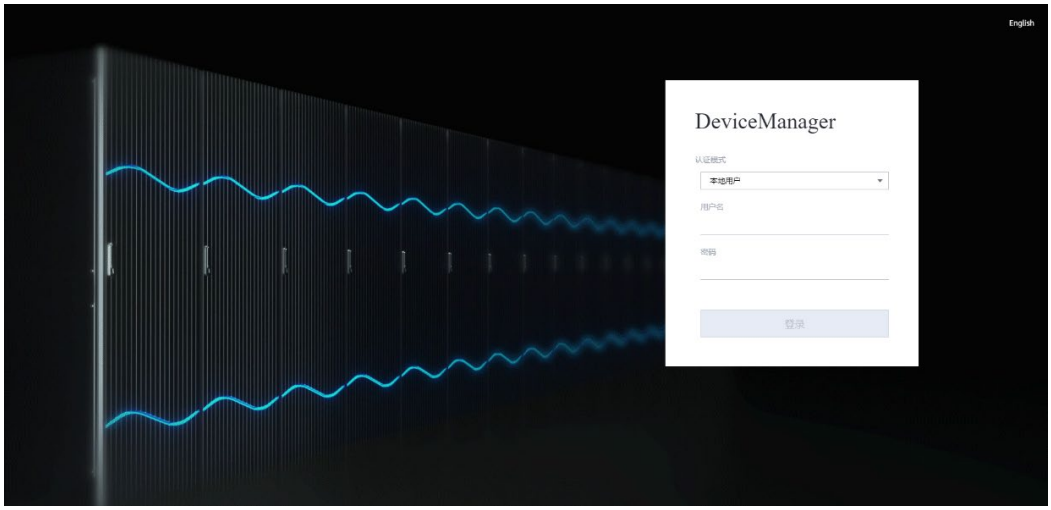
HengShan Stor 系列除了支持自带的集群管理软件 DeviceManager 进行集群管理外，还支持 eSight 数据中心级管理、eService 云化管理和 SmartKit 智能巡检等管理运维方式。

8.1 系统管理工具

8.1.1 DeviceManager

DeviceManager 是百信 HengShan Stor 系列存储系统内置的 GUI（Graphical User Interface）管理工具，用户使用普通浏览器直接访问 https://存储管理 IP:8088/ 即可使用。通过 DeviceManager，可以完成几乎所有需要的配置和管理动作。下图是 DeviceManager 登录页面：

图8-2 DeviceManager 登录页面



HengShan Stor 系列 DeviceManager 支持管理用户按角色分为“系统管理员”、“系统操作员”和“系统查看员”，DeviceManager 提供的管理功能可分为资源接入、资源管理、系统管理三类。

- 资源接入

  资源接入主要是 HengShan Stor 系列节点的接入，支持单个节点接入和批量导入两种方式（可在 DeviceManager 界面上下载批量导入模板）。节点接入前需要安装操作系统和配置管理 IP，然后在 DeviceManager 界面上输入待接入节点的管理 IP 等信息。节点接入后可在 DeviceManager 界面上为其配置存储网络和安装存储节点。

  DeviceManager 为新节点接入提供向导式操作，三步完成增加存储节点、配置存储网络、软件安装。

- 资源管理

  资源管理包括存储池管理和不同服务独特的资源管理，其中存储池管理支持创建存储池，查看选定存储池的统计信息，查看选定存储池的硬盘拓扑，为选定存储池扩容、减容、修改、增加至 QoS 策略，以及删除存储池。

- 系统管理

  系统管理包括集群基本信息监控、性能监控、告警管理、用户管理、license 管理、硬件管理、网络拓扑管理、任务中心管理。

- 集群基本信息监控：查看集群的基本信息，包括集群名称、状态、节点信息、节点进程信息。
- 性能监控：查看 CPU 利用率、内存利用率、带宽、IOPS、时延、磁盘利用率、存储池利用率统计。
- 告警管理：提供告警查看、告警通知配置（支持邮件、syslog、snmp）、告警转储、告警清除、告警屏蔽、告警导出的功能。
- 用户管理：系统管理员可以创建新的管理员，为该管理员赋予一定的管理权限，以便多个管理员按照所授权限进行系统或资源管理。对用户的操作包括：创建、查询、修改、删除、解锁、下线等。支持设置密码策略以提升系统安全。
- License 管理：提供查看已激活的 license 和导入新 license 功能。
- 硬件管理：提供节点、硬盘、网口信息查询。
  - 节点管理对系统中的所有节点集中管理，可查看节点的名称、管理 IP、存储 IP、状态、软件安装状态、机柜、槽位等信息，支持将节点设置为维护模式以方便对节点进行故障恢复处理。
  - 磁盘管理将系统中所有的磁盘集中管理，支持查看磁盘的名称、状态、槽位号、序列号、类型等。
  - 网口信息支持查看网口名称、状态、MAC 地址、协议类型和 IP 地址。
- 网络拓扑管理：提供管理网络、存储网络、复制网络拓扑信息查询。

任务中心管理：提供后台任务进度和状态查询。

8.1.2 CLI

CLI 是 Command Line Interface 的缩写，即命令行界面，允许管理员和其他系统用户执行存储系统的管理和维护操作。CLI 基于 SSH 协议，并且支持基于密钥的 SSH 访问。具体命令行详见《HengShan Stor 系列 8.1.3 命令参考》。

8.1.3 RESTful API

HengShan Stor 系列通过开放 REST API 接口与非标准 OpenStack 云管平台集成，上层云管平台可以通过 REST 接口对存储系统资源（比如存储集群资源、存储池、卷等）进行监控管理。

所有的 API 都是基于 HTTP 1.1 [RFC2616]，采用 REST 风格定义。每一个请求都包含 HTTP 的 POST、GET 等请求方法，对应于请求的每一个响应都含有标准的 HTTP 状态码(status code)。主要的管理接口如下：

- 存储池管理：支持监控存储池的相关信息，包括容量（总容量、已分配容量、已使用容量）、状态、安全级别、冗余策略等等。
- 卷管理：提供对存储卷的管理操作，包括创建卷、映射卷到主机、挂载卷、删除卷、扩容卷、卸载卷、查询卷等等接口。
- 快照管理：提供对存储快照的管理，包括创建快照、删除快照、查询快照、根据快照创建卷、对指定的多个卷创建一致性快照等接口。
- 系统用户管理：提供系统管理用户的管理，包括创建账户、删除账户、修改重置账户密码等接口。

- 性能监控：提供性能监控，包括存储池、主机和磁盘的性能，比如存储空间利用率、读写 IOPS、读写带宽、时延等。
- 安装部署管理：提供自动化部署接口，包括接入主机、查询主机、安装软件等。
- 操作日志查询：提供操作日志查询接口，支持按操作名称、操作执行结果、操作时间等查询日志。

8.1.4 SNMP

HengShan Stor 系列存储支持标准的 SNMP v1/v2/v3 管理协议接口，通过 SNMP 用户可以查询、设置系统信息，接收系统的告警信息。

8.1.5 SmartKit

SmartKit 巡检工具积累了各产品域的专家经验，降低了技能要求；多网关或多进程并发巡检，极大提升了巡检效率，主要功能如下：

- LMT 根据维护经验、预警等持续刷新巡检用例并及时发布脚本，工具自动更新；
- 融合巡检工具在记录巡检报文基础上自动输出分析、统计报告；
- 自动从网管获取网元列表，批量、远程完成巡检，效率极高。

8.1.6 eSight

eSight 存储管理软件是面向企业数据中心存储设备集中运维的管理软件，能够对多厂商存储设备统一管理，具备全局拓扑展示、容量分析、性能分析、健康度评估、故障定位，以及从主机应用到存储空间的端到端可视化等功能，显著提升了存储网络的运维效率。具体需求如下：

- 统一管理多个厂商、100+款存储设备，有效降低存储管理难度，提供统一的设备管理视图，集中展示存储硬件资源与逻辑资源；
- 实时监控，图形化展示，告警信息自动发送，管理员无需值守；
- 智能分析报表，监控关键业务服务质量，为合理规划空间提供依据。

8.1.7 eService

eService 是云端智能运维平台，利用大数据分析和智能技术，为百信存储、服务器等数据基础设施提供自动故障上报、容量预测、性能预测、硬盘风险预测、问题处理进展跟踪等服务。实现对数据基础设施随时随地主动式、预见式运维，降低运维难度，提升运维效率。eService 云端大数据智能分析系统，具备：

- 故障预测(风险盘的预测)，容量管理、性能异常分析等智能存储管理服务。
- 提供主动 O&M 服务，提供全方位的主动保护和更快速的故障处理服务， 降低事故风险和运维成本。

8.2 存储服务资源管理

8.2.1 文件服务资源管理

- 文件系统管理

  HengShan Stor 系列文件服务支持多命名空间（Namespace），即多文件系统，一个存储集群最大可支持 10 万个文件系统， 存储管理员可以在 DeviceManager 界面上创建文件系统、删除文件系统和查询文件系统。每个文件系统是一个独立的 Namespace, Namespace 间的用户数据相互隔离。

- DTree 管理

  DTree 即 Directory Tree，是一种特殊的目录，主要用于承载增值特性的配置，如快照、配额、分级等，DTree 需要通过 DeviceManager 界面、CLI 和 Restful 接口进行创建、删除等管理操作，DTree 可以在包括文件系统根目录在内的任意目录下创建，但 DTree 间不能嵌套。

- 业务 IP 管理

  HengShan Stor 系列是全对称文件系统，每个存储节点都可以对外提供存储服务，因此，每个节点都需要配置 IP 地址，在分布式存储中就需要大量的 IP 地址，为了方便客户使用存储，系统对外提供域名服务，用户通过对外提供的域名进行访问，在进行域名解析返回实现业务 IP 地址时，通过用户配置的负载均衡策略（轮询方式、CPU 利用率均衡、节点连接数均衡）来实现均衡。

- 并行客户端管理

  在部署完分布式并行客户端后，需要首先在 DeviceManager 上创建 DPC 集群，并将已经安装并行客户端软件的计算节点加入到 DPC 集群，只有加入到 DPC 集群的客户端节点才能访问存储服务。

8.2.2 对象服务资源管理

- 鉴权管理

  支持内置的 POE 鉴权，支持外置的 AD 域和 LDAP 域鉴权。

- 统一命名空间管理
  - 配置位置服务

    配置全局位置服务公网和内网地址、本地区域位置服务地址、全局域名。
  - 管理区域

    添加区域：将非默认区域添加到全局管理中，进行统一管理。

    添加集群：将集群添加到本地区域中，可实现本地区域内集群的统一管理

    域名解析服务配置： 配置对象的域名解析服务。

- 安全管理
  - 安全访问策略
    - 启用 IP 接入键策略

      如果在统计时间间隔内，同一个 IP 地址上的同一个接入键的访问失败次数大于等于错误数阈值并且访问失败的次数与总访问次数的比值大于等于错误率阈值，就对该接入键和 IP 地址的组合拒绝服务。
    - 启用接入键策略

如果在统计时间间隔内，某个接入键的访问失败次数大于等于错误数阈值，并且访问失败的次数与总访问次数的比值大于等于错误率阈值，则对该接入键拒绝服务。

- 启用 IP 访问策略

  如果在统计时间间隔内，某个 IP 地址访问失败次数大于等于错误数阈值，并且访问失败的次数与总访问次数的比值大于等于错误率阈值，则对该 IP 地址拒绝服务。

- TLS 策略

  可根据需求配置相关的 TLS 策略。

- 时间校验

  接入设置包括设置对象服务的服务域名及时间校验。

- 开启时间校验

  开启后，当客户端时间与服务端时间的差异超过 15 分钟时，该访问请求将被拒绝。

- 其他
  - 配置对象服务响应消息的 Namespace。
  - 配置访问对象服务托管的静态网站失败的重定向网址。

8.2.3 大数据服务资源管理

- 账户管理

  大数据服务基于账户实现系统资源隔离，以账户为基本管理单位，配置流程采用 AK (Access Key ID)和 SK (Secret Access Key)结合进行身份认证，数据处理流程采用 kerberos 认证、LDAP 用户管理。

- Namespace 管理

  支持针对账户创建 Namespace 以及设置配额，支持查看 Namespace 列表和配额使用情况。

- LDAP 用户管理

  支持创建本地用户、本地用户组对 Namespace 进行管理，采用 LDAP 鉴权实现 Namespace 共享。

- 负载均衡

  支持对大数据服务集群进行分区管理，实现数据流的有效控制，提升业务性能和可靠性。

8.2.4 块服务资源管理

- 存储池管理

  存储池管理可查看选定存储池的统计信息，查看选定存储池的硬盘拓扑，为选定存储池扩容、减容，以及删除存储池。还提供创建新存储池功能。

- 块客户端管理

  块客户端管理提供创建、删除客户端功能。也提供查看块客户端的挂载信息与 CPU 及内存的监控统计信息，为块客户端进行挂载和卸载卷等操作。

- 卷管理

卷管理提供卷的创建和删除功能。创建卷需指定资源池、卷名、卷大小等信息。对于创建后的卷若按 SCSI 协议使用需要挂载卷，若按 iSCSI 协议使用需要做 iSCSI 映射。还提供 iSCSI 卷映射界面完成创建主机/主机组、配置启动器、配置 CHAP 认证、为主机/主机组映射/解映射卷等操作。

说明

默认情况下 iSCSI 功能是关闭的，若要使用 iSCSI 功能需要先开启 iSCSI 功能并添加 iSCSI 绑定的 IP 地址和端口。

- QoS 策略管理

  QoS 策略管理支持创建、删除 QoS 策略，及分页查看 QoS 策略信息。

- 快照管理

  快照管理支持分页出查看快照列表，列表信息包括快照名称、容量、所属存储池和创建时间；支持创建链接克隆卷、设置 QoS 策略和删除快照。

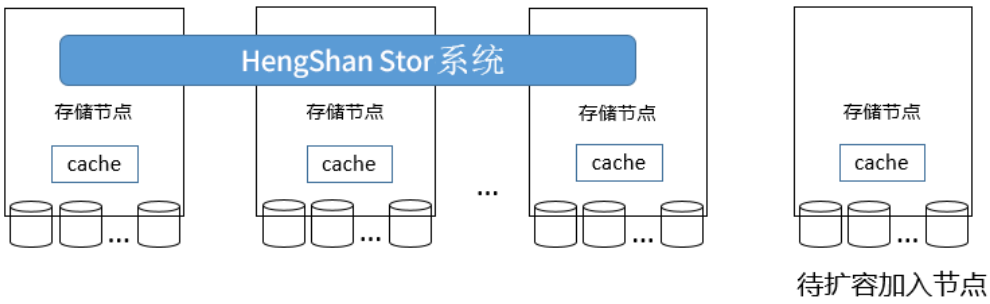
8.3 系统运维管理

8.3.1 节点扩缩容

HengShan Stor 系列产品的分布式架构具有良好的可扩展性，支持超大容量的存储：

- 扩容存储节点后不需要做大量的数据搬迁，系统可以自动达到负载均衡状态。
- 无独立的存储节点，存储接入、存储带宽和 Cache 都均匀分布到各个节点上，系统 IOPS、吞吐量和 Cache 随着节点的扩容而线性增加。

图8-3 HengShan Stor 系列扩容示意图



同理，HengShan Stor 系列产品也支持节点缩容，系统可自动对缩容节点上的有效数据进行均衡搬迁，满足用户空间管理诉求。

8.3.2 系统升级

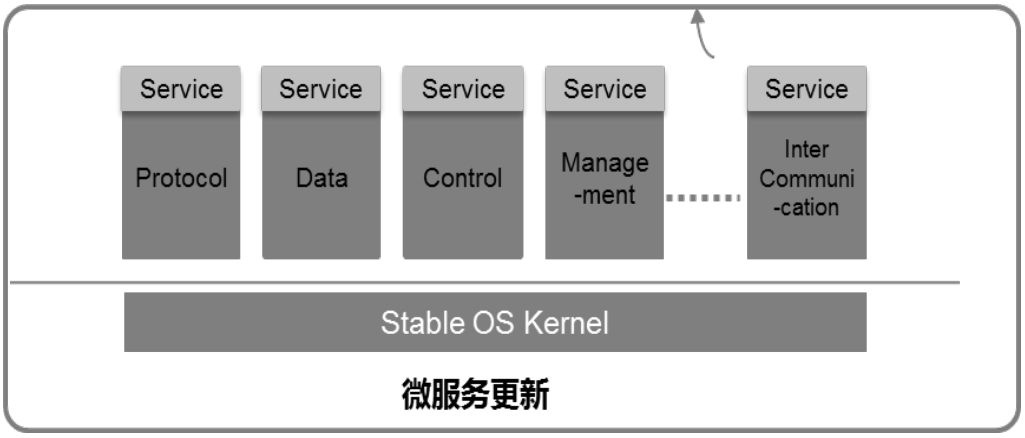
HengShan Stor 系列面向云设计，采用分布式架构，提供长期可靠的存储服务。通过微服务化自动升级保证存储服务的持续更新，提供稳定的存储服务。

HengShan Stor 系列采用模块化设计，各存储服务可以通过微服务升级和更新。

- 存储服务升级：通过去激活老的微服务，并激活新的微服务来实现存储服务升级，比如性能优化，可靠性提升等
- 存储服务更新：通过加载新的微服务来提供新的存储服务能力， 比如引入智能存储分级等

通过微服务方式，存储服务的升级和更新可以实现 5 秒钟快速激活。

图8-4 HengShan Stor 系列微服务升级示意图



HengShan Stor 系列采用分布式架构设计，实现了高扩展性以满足大规模集群部署，在系统长期运行过程中，会出现各种原因导致整个存储系统需要升级，比如引入新的特性需求、底层操作系统内核升级等等。为了尽可能降低系统的业务影响和升级风险以及提高升级效率，HengShan Stor 系列实现了滚动升级能力，即按节点和存储池批量滚动升级。

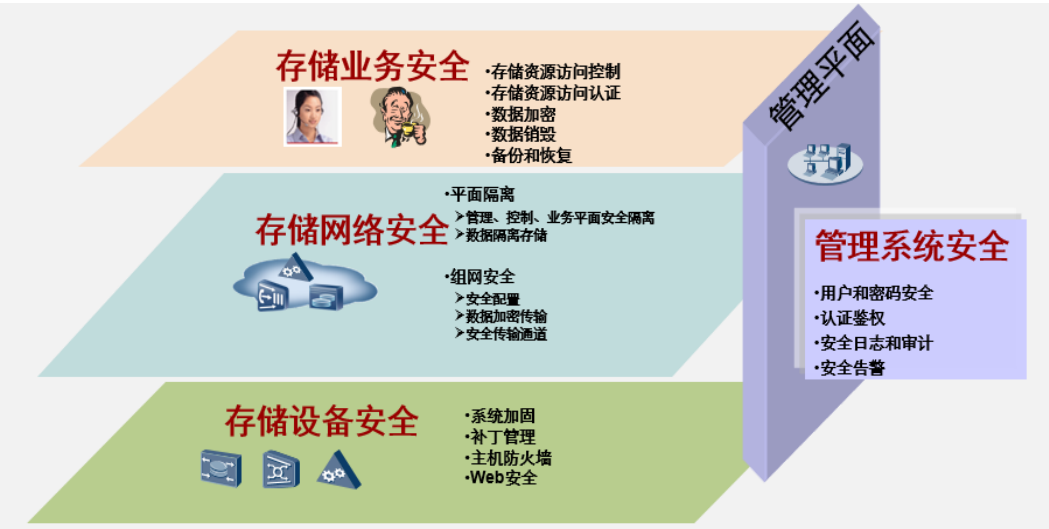
存储系统滚动升级采用一键式分批升级，操作简单高效。单存储池内每 5 分钟升级一个节点，保证业务性能的平滑稳定 。

9 系统安全性设计

随着互联网和云技术的发展，任何 IT 系统都面临着日益增多的安全威胁，既有传统安全威胁也有新兴的安全威胁。从网络安全上，分为外部网络安全威胁和内部网络安全威胁，前者包括网络 IP 攻击、软件漏洞、病毒/木马、SQL 注入攻击和钓鱼攻击等，后者包括内网 ARP 欺骗、恶意插件、非法外联、移动设备接入、应用缺乏监管等等。

具体到存储系统，则还面临着数据泄露、数据损坏以及临时或永久丧失可访问性和可用性。需要通过技术控制手段，其中包括对数据完整性、保密性和可用性监控，以防止存储资源和数据用于未经授权的用户和用途。

图9-1 HengShan Stor 系列安全框架



HengShan Stor 系列安全框架分层如下：

- 存储设备安全：操作系统加固、补丁管理、web 安全。
- 存储网络安全：平面隔离、通道安全。
- 存储业务安全：存储资源访问控制、认证鉴权、数据加密、日志审计。
- 存储管理安全：管理系统用户安全、密码安全、认证鉴权、日志和告警管理。

9.1 存储设备安全

9.1 存储设备安全

9.1.1 操作系统加固

存储系统同时支持通用操作系统和专用操作系统。通用操作系统采用系统通用的加固策略。专用操作系统配置相关的安全加固策略，主要内容如下：

- 最小化服务

  禁用危险的系统后台进程和服务；不提供的服务不会为潜在的入侵者所利用，有效地降低了风险。
- 文件和目录访问权限设置

  结合业界加固规范及应用要求，保证文件权限最小化；如果文件权限设置不当，会引起越权访问（例如普通用户能够获取管理员权限）等。
- 帐号安全

  删除不必要的账号，避免潜在的入侵者使用默认账号；为不同的账号设置不同权限，避免越权访问。
- 口令安全

  启动口令复杂度检查、密码有效期检查、登录失败重试等，避免口令的暴力破解。
- 日志和审计

  记录服务、进程运行日志；记录所有操作类日志。

9.1.2 安全启动

针对产品配套的 ARM 存储节点和欧拉 OS，支持安全启动功能，确保设备在启动时未被篡改。设备启动时首先执行被硬件保护的、不可被篡改的可信根，可信根作为信任链的第一级，对下一级启动代码进行数字签名校验，实现了从底层可信硬件到 OS 系统的数字签名校验。

9.1.3 安全补丁

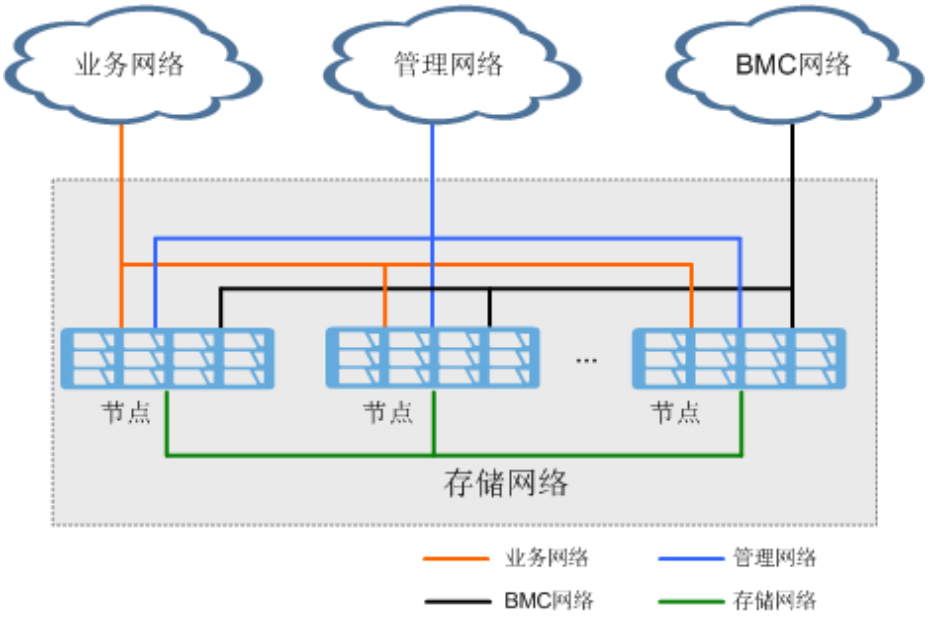
软件因自身设计缺陷而存在很多漏洞，需要定期为系统安装安全补丁以修补这些漏洞，以防止病毒、蠕虫和黑客利用操作系统漏洞对系统进行攻击。百信根据专用操作系统安全补丁和开源软件安全补丁官方的发布情况，结合实际的使用需求，为用户定期提供安全补丁。

9.2 存储网络安全

9.2.1 网络平面隔离

HengShan Stor 系列具有安全的物理组网结构，根据业务类型可划分为管理网络、BMC网络、业务网络和存储网络，可以支持通过设置 VLAN 进行逻辑隔离，也可以支持独立网口和独立交换机的物理隔离，保护系统运行的安全。

图9-2 HengShan Stor 系列集群组网



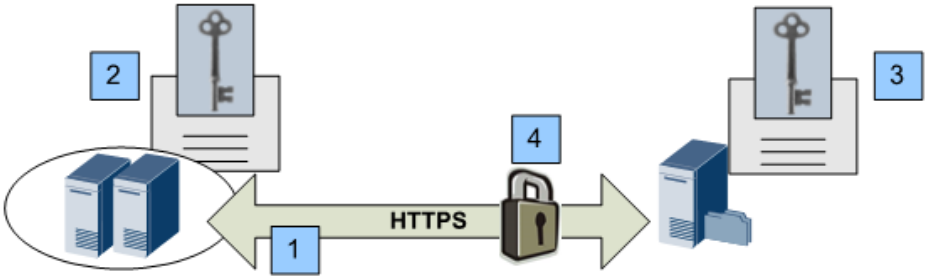
9.2.2 通道安全

- 安全系统远程管理
  采用安全的传输协议（ssh、https），如下所示。

表9-1 安全传输协议

应用	协议	端口
ssh server （系统 CLI 管理）	SSH V2	22
DeviceManager （系统 GUI 管理）	TLS1.2	8088、28443

图9-3 安全系统远程管理



- 安全远程数据传输

  HengShan Stor 系列存储系统支持数据远程复制和结构化数据的双活，数据传输有可能会跨非信任网络，在数据传输过程中跨非信任网络时，客户可以通过配置 IPSec 设备或 VPN，保障数据跨非信任网络传输时的安全。

- 对象数据安全传输

  对象服务对外提供兼容 Amazon 的 S3 API 接口，用户可通过百信或第三方提供的终端工具实现用户数据的安全上传和下载，数据传输过程中，默认采用 TLS v1.2 加密，支持兼容 TLS v1.1、TLS v1.0 可选配置。

9.3 存储业务安全

HengShan Stor 系列存储服务，针对各种服务的访问提供访问控制、数据加密、业务访问日志审计来保证存储业务的安全。

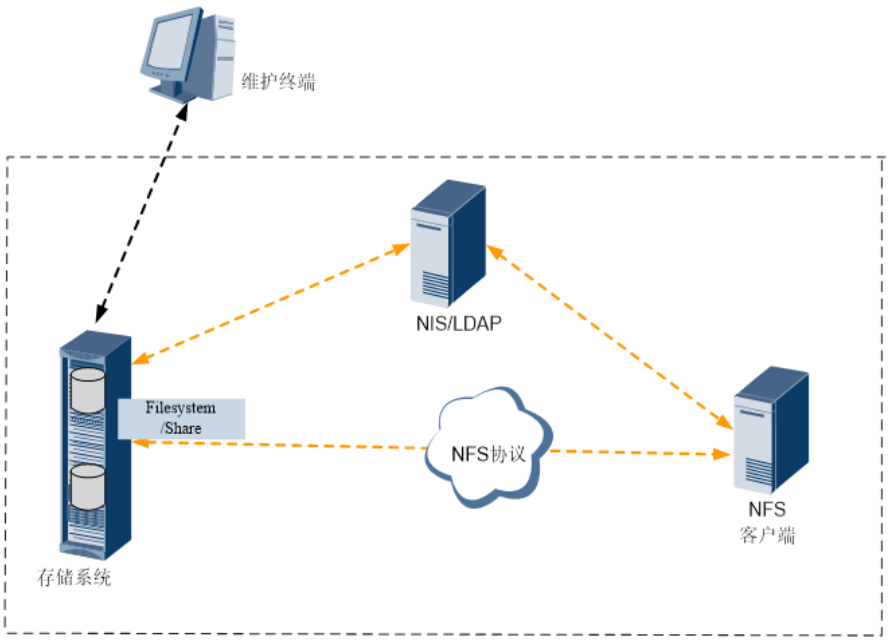
9.3.1 访问控制

9.3.1.1 文件服务访问控制

ACL 鉴权机制。利用 ACL 定义特定用户对某个资源的访问权限，只有对资源拥有相应权限的用户才能访问该资源。

当前支持 NFSV3 协议和 SMB 协议。NFS 协议文件访问支持本地认证、NIS、LDAP 域认证，本地认证需要在本地配置可以访问的主机和用户，NIS 和 LDAP 域认证是存储系统通过域服务器实现对可访问 IP 的验证。SMB 协议支持 AD 域认证。

图9-4 NFS 域认证



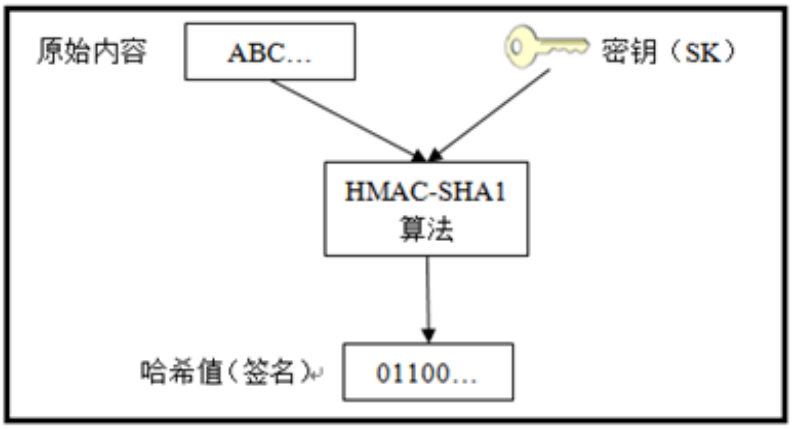
9.3.1.2 对象服务访问控制

对象服务提供灵活和安全的数据访问控制，用户可根据自己应用的需求对桶和对象设置不同的访问控制策略，包括 READ、WRITE。同时也可以指定其他用户有权访问和设置桶和对象的访问控制策略，包括 READ_ACP(可读取桶或对象当前的访问控制策略)、WRITE_ACP(可设置桶或对象当前的访问控制策略，从而授权其他用户进行数据读写)、FULL_CONTROL(完全控制权限意味着拥有 READ、WRITE、READ_ACP 和 WRITE_ACP 的权限)。

对象服务采用 AK (Access Key ID)和 SK (Secret Access Key)结合进行身份认证。认证过程采用 HMAC（keyed-Hash Message Authentication Code，密钥相关的哈希运算消息认证码）运算，HMAC 运算利用哈希算法，以一个密钥和一个消息为输入，生成一个消息摘要作为输出。

每个客户端用户都有一个 AK 和 SK，AK 用于标识唯一用户，SK 作为 key，用于计算签名，在系统中加密保存。用户需要妥善保管 SK，不能公开给其他人。客户端发送操作请求中携带用户的 AK 以及用该用户的 SK 计算得到的签名（HMAC 具体实现采用了 HMAC-SHA1 或 HMAC-SHA256）。HengShan Stor 系列收到请求，查找 AK 对应的 SK，用 SK 计算签名，与用户请求中的签名进行比较，如果一致，则鉴权通过，如图 9-5 所示，是根据用户的 SK 计算签名的过程。

图9-5 根据 SK 计算签名的过程



9.3.1.3 大数据服务访问控制

ACL 鉴权机制。利用 ACL 定义特定用户对某个资源的访问权限，只有对资源有访问权限的用户才能做对应操作。

支持 LDAP 域认证，只有租户加入了 LDAP 域，才能访问存储数据。

支持 Kerberos 认证，存储系统支持使用 Kerberos 协议对租户客户端进行认证鉴权，只有认证成功才能访问数据。

9.3.1.4 块服务访问控制

LUN 映射机制。主机和 LUN 通过映射建立访问关系，只有主机和 LUN 建立了访问关系后，对应的应用服务器才能访问该 LUN。

iSCSI 协议支持 CHAP 认证。为了建立存储系统与应用服务器的连接，需要先创建虚拟的主机。为主机添加启动器后，才能建立虚拟主机和物理上的应用服务器的对应关系。为了保证存储系统的安全，可以通过配置 CHAP（Challenge Handshake Authentication Protocol）安全性认证，限制应用服务器对存储资源的访问。

9.3.2 数据加密

HengShan Stor 系列支持基于加密盘和软件加密两种静态数据加密能力，保护客户存储到系统中的数据不被泄露。

- 加密盘

  通过配置 SED 和内置/外置密管，实现结构化（块服务）和非结构化数据（文件、大数据和对象服务）在盘中加密。SED 具备两层安全保护，分别使用 AK（authentication key）和 DEK（data encryption key）两个安全密钥。

  - 两层密钥管理：
    - 身份认证密钥（Authentication Key）：AK 用于硬盘接入时进行身份认证和对 DEK 加解密， AK 由内置密管进行管理，更新不影响加密的数据。

- 数据加密密钥（Data Encryption Key）：DEK 预置在硬盘中， DEK 使用 AES-256-XTS 进行业务数据加解密，通过更新 DEK 达到数据销毁的效果。

– 密钥管理系统：

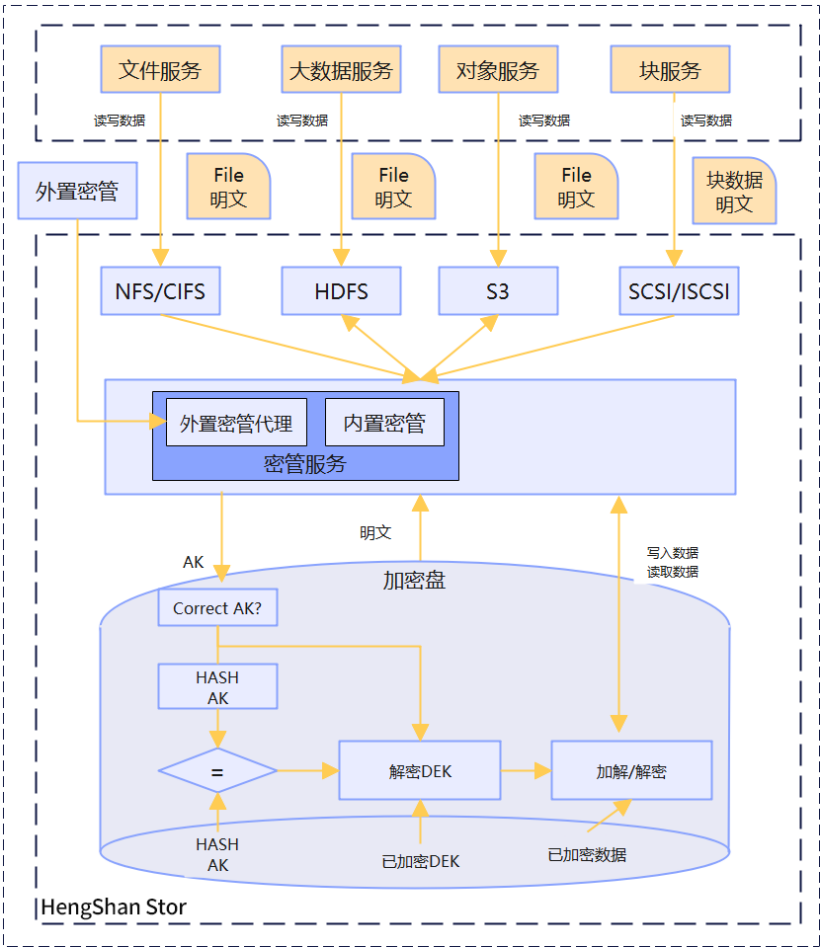
  支持内置密钥密管理和外置密钥管理。

  - 外置密钥管理：遵循标准的 KMIP 协议，支持 KMIP1.0、KMIP1.1、KMIP1.2 三个协议版本。
  - 内置密钥管理：存储系统自带的密钥管理应用，对密钥进行生命周期管理，内置密管具有易部署、易配置、易管理的特点，内置密管对身份认证密钥进行生命周期管理。AK 密钥由安全随机数产生，采用三层体系结构保护密钥安全，支持密钥的备份、恢复、更新、销毁。

– 加密引擎：

  使用加密盘的加密引擎，为数据加解密提供基础能力。

图9-6 加密盘原理图



- 软件加密

  软件加密支持结构化数据（块服务）和非结构化数据（文件、大数据和对象服务）的软件加密，支持 XTS-AES-128 和 XTS-AES-256 两种加密算法。软件加密

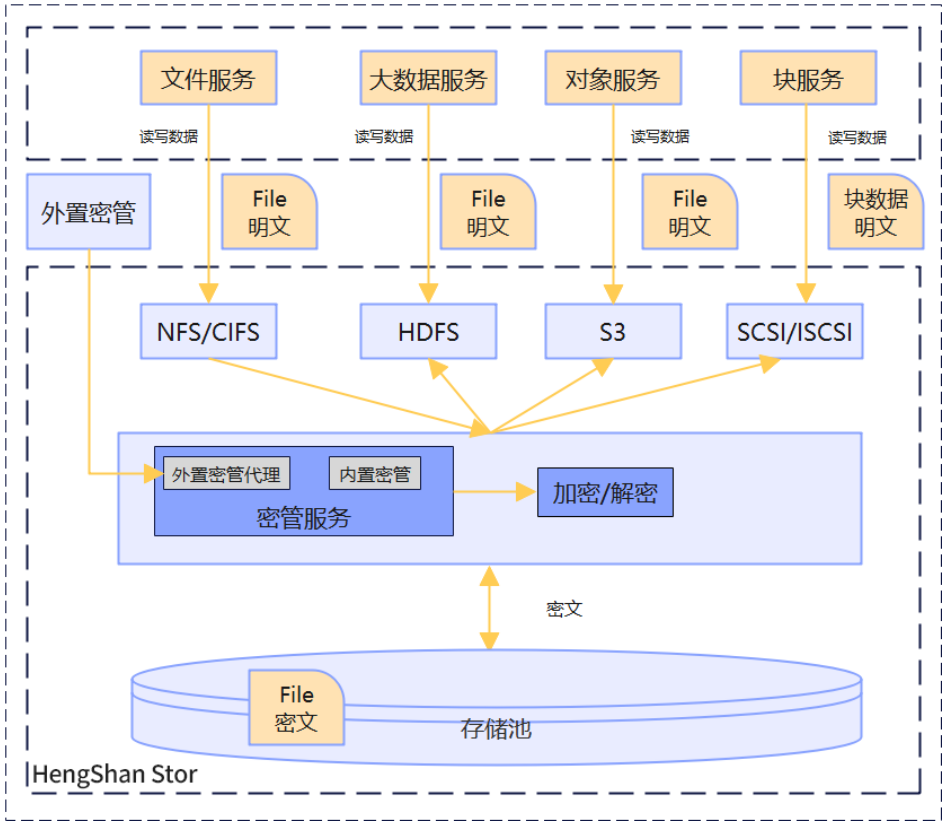
通过内置密管和内置加密引擎完成静态数据加密，通过两层密钥管理提供帐户级安全。数据写入存储系统后，先实现软件层面的加密，再将密文写入到盘中保存；读数据时，先从盘中读取密文，再在软件层面解密成明文，然后再提供给用户。

- 两层密钥管理：
  - 身份认证密钥（Authentication Key）：配置在租户上，用于身份认证，支持更新。
  - 数据加密密钥（Data Encryption Key）：用于加解密用户数据。
- 密钥管理系统：

  支持内置密钥密管理和外置密钥管理。
  - 外置密钥管理：遵循标准的 KMIP 协议，支持 KMIP1.0、KMIP1.1、KMIP1.2 三个协议版本。
  - 内置密钥管理：存储系统自带的密钥管理应用，对密钥进行生命周期管理，内置密管具有易部署、易配置、易管理的特点，内置密管对身份认证密钥进行生命周期管理。AK 密钥由安全随机数产生，采用三层体系结构保护密钥安全，支持密钥的备份、恢复、更新、销毁。
- 加密引擎：

  使用内置的加密引擎，为数据加解密提供基础能力，开启数据加密后，对数据进行读写时，通过内置加密引擎完成写入数据加密和读取数据解密。

图9-7 软加密原理图



9.3.3 业务访问日志审计

HengShan Stor 系列非结构化服务提供日志审计（SmartAuditlog）特性，通过在 Namespace 上配置审计开关和操作字，记录发生在该 Namespace 上的文件或目录操作行为。

9.3.4 WORM

WORM（write once read many）意味着一次写入多次读取，用户可以为文件设置保护期，在保护期内，文件只能读取，不能修改或删除，保护期到期后可以删除和读取文件。因此，WORM 是文件系统作为归档系统的必备特性。WORM 分为企业级和法规遵从级，其中企业级 WORM 的系统管理员可以在文件被保护期内删除 WORM 保护的文件，而法规遵从级 WORM 在文件被保护期内任何人都无法删除文件。HengShan Stor 系列的非结构化服务支持企业级 WORM 和法规遵从级 WORM，其中对象服务可兼容 Amazon S3 Object Lock 接口。

通常，WORM 特性的启用主要包括以下流程：

- 启用 WORM 时钟

  WORM 时钟，用于 WORM 文件的计时。设置了 WORM 时钟之后，系统会在该时钟基础上进行计时，避免由于用户通过修改本地时钟导致文件提前过期。

  每个目录或文件都具有 atime、ctime 和 mtime 属性，当一个目录或文件被设置了 WORM 属性后，该目录或文件将使用 WORM 时钟来计时，防止由于节点本地时间的变更导致目录或文件保护时间的变化。

  WORM 时钟只允许设置一次，设置成功后不再允许更改。

- 设置 WORM 策略

  HengShan Stor 系列支持为命名空间或 Dtree 设置 WORM 策略。WORM 策略包括默认保留期，并支持对特定文件或文件名前缀设置默认保留期。如果启用了默认保留期，那么上传的文件将自动使用默认保留期作为自己的保留期；如果没设置默认保留期，那么上传的文件默认不保护。

- 访问 WORM 文件

  访问 WORM 文件时，存储服务会检查文件的保留期是否超期（以系统的 WORM 时钟为参考，不依赖于本地系统时间）。普通用户只有读权限，如果未超期，则文件不能被覆盖或删除，如果超期则仅可以删除和读取，仍然不支持修改。由于 HengShan Stor 系列支持的是企业级 WORM，所以支持在保留期内，特权用户可以执行删除、修改等全部操作。

- 记录 WORM 审计日志

  HengShan Stor 系列支持 WORM 审计日志功能，对受 WORM 保护的文件进行操作时会记录特殊的审计日志，以便于管理员进行审计。WORM 审计日志功能默认关闭，需要手动开启。

9.4 存储管理安全

9.4.1 鉴权认证

本地认证通过系统本地用户认证登录存储系统。

AD/LDAP 域认证：支持通过外部 AD/LDAP 域认证服务器进行认证。

9.4.2 角色管理

- 基于角色的访问控制管理：支持系统定义的默认角色。

  默认角色可选择超级管理员、管理员、系统查看者、SAN 资源管理员、机机账户和安全管理员。

  - 超级管理员：对存储设备有完全的控制权限，可以创建各类角色的用户。
  - 管理员：拥有 Call Home（对接云上配置）的配置和查看权限，拥有用户安全策略和告警的查看权限。
  - 系统查看者：拥有告警、安全策略、License、Call home 等功能的查看权限。
  - SAN 资源管理员：拥有存储池、卷等资源的配置管理权限。
  - 安全管理员：拥有用户的查看权限，拥有系统安全配置权限，包括安全规则管理、安全策略管理、证书管理、KMC 管理。
  - 机机账户：系统内部机机用户。
- IP 白名单：通过 IP 白名单进行控制访问。

9.4.3 日志和告警管理

- 日志管理

  对于管理面的所有操作都有操作日志记录；日志记录的内容包括事件发生的时间、用户 ID(包括关联终端、端口、网络地址或通信设备)、事件类型、被访问的资源名称、事件的结果；日志记录提供查询机制；日志记录空间满时提供自动转储和删除机制；日志时间提供统一的时间源机制。
- 告警管理

  对于系统出现的异常状态和各种故障，实时显示在操作界面，提示用户进行告警恢复；支持 Email 告警、Syslog 告警、Trap 告警；支持 SNMP 安全策略配置；支持 SFTP 告警转储到独立的告警信息服务器。

9.4.4 Web 安全

HengShan Stor 系列设备管理模块管理系统 Web 服务，具有的安全功能如下：

- 自动将客户请求转换成 HTTPS

  DeviceManager 能够自动把客户的请求转向到 HTTPS 连接。当用户使用 HTTP 访问 DeviceManager 时，DeviceManager 能自动将用户的访问方式转向为 HTTPS，增强访问安全性。

  HTTPS 访问时，提供自签名证书，建议替换为信任机构颁发的 CA 证书。
- 防止跨站脚本攻击

跨站点脚本攻击是指攻击者利用不安全的网站作为平台，对访问本网站的用户进行攻击。通过设置 web 的安全配置防止跨站脚本攻击，如 Httponly 等。

- 防止 SQL 注入式攻击

  SQL 注入式攻击是指，攻击者把 SQL 命令插入到 Web 表单的输入域或页面请求的查询字符串，欺骗服务器执行恶意的 SQL 命令。通过对外部输入的 SQL 命令做强校验，防止 SQL 注入攻击。

- 防止跨站请求伪造

  跨站请求伪造是指用户登录 A 网站且在 Session 未超时情况下，同时登录 B 网站（含攻击程序），攻击者可在这种情况下获取 A 网站的 Session ID，登录 A 网站窃取用户的关键信息。通过在消息头增加防止 CSRF 攻击的 Token 信息来防止跨站请求伪造。

- 隐藏敏感信息

  隐藏敏感信息防止攻击者获取此类信息攻击系统。

- 限制上传和下载文件

  限制用户随意上传和下载文件，防止高安全文件泄漏，以及非安全文件被上传。

- 防止 URL 越权

  每类用户都会有特定的权限，越权指用户对系统执行超越自己权限的操作。

9.4.5 用户安全策略

用户安全策略主要为登录策略，如表 9-2 所示：

表9-2 用户登录策略

参数	说明	设置
会话超时时间（分钟）	会话设置默认的超时时间。	[默认值] 30
密码锁定	启用错误密码锁定后，在 5 分钟内，当用户连续输入错误密码的次数达到设置的“错误次数”时，该用户被锁定。	[默认值] 启用
错误次数	允许连续输入错误密码的次数。在 5 分钟内，输入密码出错次数达到已设置的“错误次数”时，系统自动锁定该用户。	[取值范围] 1～9 之间的整数。 [默认值] 3
锁定类型	用户被系统自动锁定的方式。 • 选择“永久锁定”，用户被系统永久锁定，超级管理员将在被锁定 15 分钟后由系统自动解锁。 • 选择“临时锁定”，可以设置用户被系统自动锁定的时间。	[默认值] 临时锁定

参数	说明	设置
自动解锁时间（分钟）	用户被系统自动锁定后自动解锁的时间。 • 启用“账号锁定”，并选择“锁定类型”为“临时锁定”时，才可以设置此项。 • 自动解锁时间只对系统自动锁定有效。当用户被人工锁定时，锁定时间不生效，只能人工解锁。 • 自动解锁时间只对非超级管理员用户生效，无论选择“永久锁定”还是“临时锁定”，超级管理员都将在被锁定 15 分钟后由系统自动解锁。	[取值范围] 3～2000 之间的整数。 [默认值] 15

9.4.6 密码安全

支持强密码复杂度策略，防止暴力破解; 口令必须加密存储和传输；口令修改只有认证后才可修改，仅可以修改自身口令（超级管理员除外）。具体内容如表 9-3 所示。

表9-3 密码安全策略

参数	说明	设置
最大长度	密码最大长度，避免设置过于冗长的密码。	[取值范围] 8～32 之间的整数。 [默认值] 16
最小长度	密码最小长度，避免密码过于简单。	[取值范围] 8～32 之间的整数。 [默认值] 8

参数	说明	设置
复杂度	密码的复杂度，避免设置过于简单的密码。	[取值范围] “必须包含特殊字符，并且至少包含大写字母、小写字母以及数字中任意两者的组合”或“必须包含特殊字符、大写字母、小写字母和数字”。 [默认值] 必须包含特殊字符，并且至少包含大写字母、小写字母以及数字中任意两者的组合
字符重复次数	允许密码中某一字符连续出现的最大次数。	[取值范围] 0～9 之间的整数（0 表示不限制）。 [默认值] 3
历史密码保留个数	保留历史密码的个数，设置的新密码不允许和历史密码相同，当取值为“0”时表示不做限制。	[取值范围] 0～30 之间的整数。 [默认值] 3
密码有效期（天）	是否启动密码有效期设置。建议启用“密码有效期”。 启用“密码有效期”后，需要设置密码有效天数。当账户密码超过设置的天数时，系统会提示修改密码，请及时修改。	[取值范围] 1～999 之间的整数。 [默认值] 90
密码提前提示阈值（天）	在密码失效前多少天进行提示。	[取值范围] 1～99 之间的整数。 [默认值] 7
密码修改间隔时间（分钟）	设置新密码后，新密码最少使用的时间。	[取值范围] 1～9999 之间的整数。 [默认值] 5

10 生态兼容性

HengShan Stor 系列支持 NFS、SMB、HDFS、S3、iSCSI 等多种协议接口，承载多业务应用。作为分布式存储系统，HengShan Stor 系列支持主流的开放接口，提供丰富的兼容性，与客户的云数据中心和应用平台平滑集成。

10.1 数据面生态兼容性

10.1.1 存储协议兼容性

HengShan Stor 系列提供的各种类型存储服务均遵循业界主流的协议和标准。

- 块服务接口：遵循标准的 SCSI 语义和 iSCSI 协议，提供分布式块存储业务。
- 文件服务接口：遵循标准的 NFS/SMB 协议和 POSIX 语义的并行文件客户端，提供分布式文件存储业务。
- 对象服务接口：遵循 Amazon S3 接口标准，实现了对 Amazon S3 主要功能兼容。
- 大数据服务接口：遵循原生的 HDFS 协议，提供基于 Hadoop 平台的能力。

10.1.2 块服务虚拟化平台兼容性

HengShan Stor 系列提供了分布式的块数据存储业务，实现了标准的 SCSI 和 iSCSI 存储接口，支持与业界主流的虚拟化平台对接，包括 VMware、Xen、KVM 和 Hyper-V 等，以及 FusionSphere 等。其中，针对 VMware 和 Hyper-V，还提供了详细的最佳实践。

10.1.3 块服务数据库软件兼容性

HengShan Stor 系列块服务支持主流数据库软件，包括 Oracle、IBM DB2、Sybase IQ、达梦等等。对于 Oracle RAC，HengShan Stor 系列还提供了块存储服务双活最佳实践。

10.1.4 块服务操作系统兼容性

除了给虚拟化平台提供存储服务外，百信 HengShan Stor 系列还支持通过在物理服务器 OS 中部署 HengShan Stor 系列存储驱动模块提供 SCSI 存储服务。

对于主流的操作系统，百信 HengShan Stor 系列会进行完整的兼容性验证，发布相应的兼容性认证说明，并且在每个版本发布时会同步刷新和定期更新。

对于不在兼容性网站列表的 OS，HengShan Stor 系列也可以支持，但需要做兼容性认证测试。

10.1.5 文件服务兼容性

HengShan Stor 系列文件服务通过提供标准的 NFS 协议、SMB 协议、FTP/FTPS 协议、POSIX 接口协议及标准的 MPI-IO 接口协议，可实现与主流计算服务器的对接并提供标准的 NAS 协议访问能力。其中提供 POSIX 接口协议和 MPI-IO 接口协议的 DPC 的兼容性以产品提供的兼容性列表为准，对于不在兼容性列表中的计算节点操作系统，需要通过产品适配并完成兼容性测试，才能加入到 DPC 的兼容性列表中。

10.1.6 对象服务兼容性

HengShan Stor 系列对象服务提供标准的 Amazon S3 协议接口，兼容主流的备份、归档和其它第三方软件，如 Veritas NetBackup、Commvault Simpana 和爱数等。

10.1.7 大数据服务兼容性

HengShan Stor 系列大数据服务通过提供标准的 HDFS 接口与百信及业界主流大数据应用软件平台的对接。HengShan Stor 系列大数据服务当前支持与 FusionInsight、Cloudera CDH、Hortonworks HDP、星环 TDH 等主流厂商的大数据软件产品的对接。

10.2 管控面生态兼容性

10.2.1 综合网管平台兼容性

IT 运维管理平台在数据中心环境中扮演重要角色。通过 IT 运维管理软件可以实现对整个 IT 数据中心的统一管理，并可以快捷地对各类设备的状态进行管理、监控和配置。

HengShan Stor 系列支持标准的 SNMP v2 和 v3 管理协议接口，并提供开放的 REST API 接口，实现带外集中接入管理和统一维护。

10.2.2 OpenStack 集成

OpenStack 旨在为公共及私有云的建设与管理提供软件的开源项目，是业界开放的云管理平台，覆盖了计算、存储、网络等各个方面。OpenStack 通过 Cinder、Manila 模块，实现对 Block、File 存储的管理。

百信 HengShan Stor 系列提供了分布式的块存储、对象存储和文件存储业务，并提供标准 OpenStack Cinder/Manila Driver 与 OpenStack 对接，通过 REST API 接口完成对于存储资源的管理。OpenStack 各存储模块的 Provider，实现与主流 Openstack 发布版本及

商业版本的对接。目前 HengShan Stor 系列已能兼容最新的开源 OpenStack R 版，并会持续更新后续版本的兼容性。

10.2.3 容器平台兼容性

HengShan Stor 系列结构化数据存储服务发布 CSI Plugin，支持主流容器管理平台 Kubernetes 与 Red Hat OpenShift 平台。

11 特性与服务对应关系

特性	英文描述	服务
4.1 多租户（SmartMulti-Tenant）	SmartMulti-Tenant	文件、对象、大数据
4.2 配额（SmartQuota）	SmartQuota	文件、对象、大数据
4.3 分级存储（SmartTier）	SmartTier	文件、对象、大数据
4.4 负载均衡（SmartEqualizer）	SmartEqualizer	文件、对象、大数据
4.5 元数据检索（SmartIndexing）	SmartIndexing	文件、对象、大数据
4.6 智能纳管（SmartTakeover）	SmartTakeover	大数据
4.7 服务质量（SmartQoS）	SmartQoS	文件、对象、大数据、块
4.8 审计日志（SmartAuditlog）	SmartAuditlog	文件、对象、大数据
4.9 数据加密（SmartEncryption）	SmartEncryption	文件、对象、大数据、块
4.10 重删压缩（SmartDedupe&SmartCompression）	SmartDedupe&SmartCompression	块
4.12 vVol	vVol	块
5.1 快照（HyperSnap）	HyperSnap	文件、对象、大数据、块
5.2 复制（HyperReplication）	HyperReplication	文件、对象、大数据、块
5.3 双活（HyperMetro）	HyperMetro	块
5.4 对象跨站点多活（HyperGeoMetro）	HyperGeoMetro	对象
5.5 对象跨站点 EC（HyperGeoEC）	HyperGeoEC	对象

特性	英文描述	服务
5.6 链接克隆（HyperClone）	HyperClone	块
5.7 回收站（Recycle Bin）	Recycle Bin	文件、对象、大数据
4.13 场景化压缩（Scenario-specific SmartCompression）	scenario-specific SmartCompression	文件、对象、大数据
4.14 通用压缩（Standard SmartCompression）	standard SmartCompression	文件、对象、大数据
4.11 卷在线迁移（SmartMove）	SmartMove	块
4.15 智能数据迁移（SmartMigration）	SmartMigration	文件、块

12 缩略语和术语

缩略语	描述
AD	Active Directory，活动目录
CA	Client Agent，客户端代理
CLI	Command-line Interface，命令行视图
CMS	Cluster Management Service，集群管理服务
DAS	Direct-Attached Storage，直连式存储
DHT	Distributed Hash Table，分布式哈希表
DNS	Domain Name System，域名系统
DPC	Distributed Parallel Client，分布式文件并行客户端
DS	Data Service，数据业务
DTree	Directory Tree，目录树，是一种特殊的目录
EC	Erasure Code，纠删码
FTP	File Transfer Protocol，文件传输协议
FTPS	File Transfer Protocol Secure，文件传输协议
GID	Group ID，组标识
GUI	Graphical User Interface，图形用户界面
HTTP	Hypertext Transport Protocol，超文本传输协议
IAM	Identity and Access Management，认证和授权管理
IB	InfiniBand
IPMI	Intelligent Platform Management Interface，智能平台管理接口
KV	Key Value，键值对

缩略语	描述
LDAP	Lightweight Directory Access Protocol，轻型目录访问协议
LUN	Logical Unit Number，逻辑单元号
LVS	Linux Virtual Server，Linux 虚拟服务器
MDC	MetaData Controller，元数据控制设备
MPI	Message passing interface，消息传递接口
NAS	Network Attached Storage，网络存储
NFS	Network File System， 网络文件系统
NIS	Network Information Service，网络信息服务
NS	Namespace，命名空间
NUMA	Non Uniform Memory Access 缩写，是一种关于多个 cpu 如何访问内存的架构模型
OAM	Operation Administrator and Maintenance
OSD	Object Storage Device，对象存储设备
Paxos	一种基于消息传递且具有高度容错特性的一致性算法
PLOG	Persistence LOG
POSIX	Portable Operating System Interface，可移植操作系统接口
RAID	Redundant Array of Independent Disks，独立磁盘冗余阵列
RDMA	Remote Direct Memory Access，远程直接数据存取
SAN	Storage Area Network，存储区域网络
SAS	Serial Attached SCSI，串行 SCSI
SATA	Serial Advanced Technology Attachment，串行 ATA
SCSI	Small Computer System Interface，小型计算机系统接口
SMB	Server Message Block，一种客户机/服务器、请求/响应协议
SSD	Solid State Disk，固态硬盘
TCP	Transmission Control Protocol，传输控制协议
UID	User Identity，用户身份标识
ZK	Zookeeper，缩写。

