

TECHNICAL WHITE PAPER · VERSION 3.6.5

NexaVM nHPS

Hyper-Performance Distributed Block Storage

nHPS is built on NexaVM's fully in-house, new-generation storage engine — a lock-free, zero-copy, run-to-completion architecture that turns commodity all-flash servers into a storage platform delivering millions of IOPS per volume at microsecond latency.

No.1

SPC-1™ leaderboard —
first distributed storage to
top the benchmark

27.2M

IOPS · 0.144 ms average
latency (SPC-1)

20–40 μ s

Software IO-stack latency
— an order of magnitude
below legacy stacks

4,096

Nodes per system, scaled
with zero service
interruption

Contents

01		
Technical Background		3
Evolution of Storage Media and Networks		3
Evolution of Distributed Storage Software		3
02		
Product Introduction		5
03		
Product Architecture		6
Software Architecture		6
Hardware Architecture		7
Network Architecture		7
Key Service Flows		8
04		
Key Block Storage Features		10
05		
Elasticity		13
06		
High-Performance Technology		14
07		
High-Reliability Technology		17
08		
Efficient O&M		19
09		
Security		21
10		
Abbreviations		22

01 Technical Background

1.1 Evolution of Storage Media and Networks

NVMe SSD

Digital transformation revolves around two dimensions: applications and data. Process optimization, product iteration, and business-model innovation are all data-driven, and enterprises increasingly need to make real-time, data-based decisions. Massive and real-time analytics mean ever-larger volumes of data must be stored — and read and written — efficiently, placing enormous pressure on storage systems.

Storage media has become the bottleneck holding back IT innovation. Although NAND flash SSDs raised the performance bar, as the technology progressed from SLC to MLC and TLC, and now toward mainstream QLC, density kept rising while IOPS per GB declined — and write performance has largely stalled, a growing problem for write-intensive environments.

RDMA networks

AI and real-time analytics require data to move across the network faster than ever. The traditional TCP/IP stack needs multiple kernel context switches and data copies for every packet, keeps host CPUs persistently loaded, and imposes millisecond-scale protocol latency — a clear bottleneck for microsecond-class distributed systems.

RDMA (Remote Direct Memory Access) is a high-performance, lossless network technology. Its kernel-bypass mechanism lets applications read and write directly to the network adapter without user/kernel context switches, and its zero-copy mechanism lets the receiver pull data straight from the sender's memory without CPU involvement — cutting network-stack latency to under 10 μ s. RDMA has become the mainstream transport for NVMe ecosystems, and its replacement of TCP/IP in AI, real-time analytics, all-flash storage, and databases is an irreversible trend.

Three main RDMA implementations exist today:

■ InfiniBand

RDMA-native from the ground up, with hardware-guaranteed reliable delivery, the highest bandwidth and lowest latency — but at higher cost, requiring dedicated InfiniBand NICs and switches.

■ iWARP

RDMA over TCP. Large numbers of TCP connections consume substantial memory and demand higher system specifications; works with standard Ethernet switches but requires iWARP-capable NICs.

■ RoCE

RDMA over UDP — a lighter resource footprint than iWARP. Runs on standard Ethernet switches, and new-generation NICs support it by default. RoCE v2 is the mainstream choice for high-performance distributed storage.

1.2 Evolution of Distributed Storage Software

Distributed storage emerged around 2005 for internet-scale workloads — massive capacity and concurrency with relaxed consistency requirements. With the spread of cloud computing and explosive data growth, it entered the enterprise market around 2015, where pooled, elastically scalable storage made it the natural choice. This **software-defined storage (SDS) 1.0** era is characterized by:

- **Scale-out on standard hardware** — cluster capacity and performance grow linearly with expansion
- **Decoupled software and hardware** — commodity servers replace proprietary storage hardware

- **Pooled, elastic resources** — deployed to actual capacity and performance needs and extended on demand

The first generation of distributed storage software, typified by Ceph, was born 18 years ago in an era of slow HDDs and high-latency, low-bandwidth TCP/IP networks. Its architecture was never designed for today's media and workloads: performance is low, latency high, and consistency and reliability are insufficient for critical enterprise workloads — earning first-generation systems the "mid-to-low-end storage" label.

Meanwhile, hardware moved on dramatically. Media performance improved roughly 1,000× over magnetic disks with access latency down to the 10 µs class; networks advanced from 10/100 Mbps to 10/100 Gbps, with RDMA cutting network latency to 10–20 µs. Yet mainstream first-generation IO stacks still use system architectures that are 15–18 years old, taking hundreds of microseconds per IO — an order of magnitude slower than today's networks and media.

The software IO stack is now the primary bottleneck of the entire storage system.

The industry response — memory zero-copy, RDMA, lock-free zero-interrupt IO, kernel bypass, full user-space processing, Run-to-Completion (RTC) IO models, and asynchronous polling, complemented by Intel's DPDK and SPDK — has been partially adopted by centralized all-flash arrays. Distributed stacks, however, must additionally solve replicated state machines, consistency guarantees, and efficient fault recovery, and have not yet adopted these techniques comprehensively.

SDS 2.0

Driven by market demand and technology evolution, a new generation of distributed storage software has emerged: a fully redesigned architecture and IO processing model that extracts 10× more performance from the same hardware, and — with global metadata management — lets one core data engine serve many storage protocols while data flows intelligently across pools, clouds, and edges.

02 Product Introduction

NexaVM nHPS is a hyper-performance distributed block storage product incubated on NexaVM's fully in-house, new-generation storage engine. The engine incorporates a decade of advances in distributed systems, storage technology, data processing, and system programming models, and was built from scratch on a new distributed software architecture with an end-to-end lock-free, zero-copy programming model.

Its key design goals:

- **Cloud-native by design** — deploys in private, public, hybrid, and multi-cloud as well as edge environments, delivering a seamless, consistent data experience
- **Microsecond latency, million-IOPS volumes** — solves what first-generation distributed storage cannot: databases and high-performance analytics, with read/write latency down to microseconds and over one million random IOPS per volume
- **Unleashes new-generation hardware** — fully exploits the peak performance of new storage media and network technology
- **Global metadata management** — IO-level scheduling of data across multiple pools, clouds, and edges, for higher reliability and unified, all-scenario data management

NexaVM's vision is a **unified data foundation**: fast block interfaces (NVMe-oF, vhost), classic storage interfaces (iSCSI, CSI), and data interfaces (HDFS, OLAP SQL) delivered on one engine, with storage-compute integration providing unified data management, diverse data organization and analytics, and full-lifecycle data management — driving the industry into the SDS 2.0 era.

On clusters of commodity all-SSD servers, nHPS delivers **millions of IOPS per volume with stable sub-100 µs latency per path**. Through standard interfaces it integrates with OpenStack, VMware, Kubernetes, and bare-metal environments, meeting the extreme performance demands of critical applications while adding reliability innovations for core business. Targeted at large databases, cloud-hosted core business, HPDA, and AI/ML, nHPS can in many cases directly replace all-flash arrays (AFA) — while offering horizontal scale-out, software definition, cloud-native readiness, and a unified storage and data platform that AFAs cannot match.

1M+

Random IOPS per volume, with sub-millisecond concurrent latency

<100 µs

Stable per-path latency on all-SSD commodity server clusters

10×

Performance gain achievable through software innovation on identical hardware

03 Product Architecture

3.1 Software Architecture

The data plane consists of four subsystems, bottom to top: the hardware device layer, the storage engine, the block storage service, and the USS protocol gateway. Three global control-plane subsystems run alongside: the Tianshu metadata module, the reliability subsystem, and the O&M subsystem.

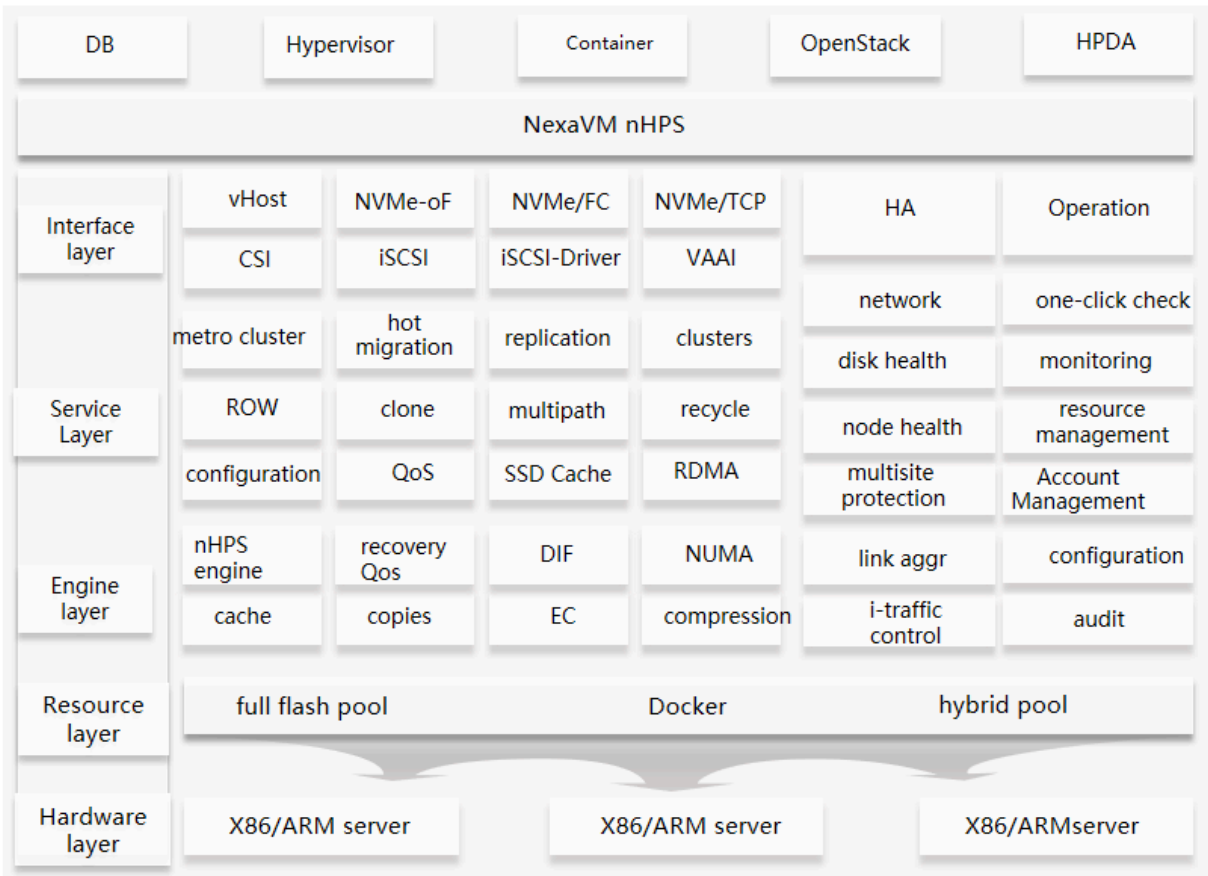


Figure 1 nHPS software architecture

■ Hardware device layer

Heterogeneous servers of different types and configurations can join a single storage system — solving hardware reuse and raising utilization.

■ Storage engine

B/B+ tree indexes slow as metadata grows and overflows cache. nHPS uses a Distributed Hash Table (DHT): one hash computation locates any object with O(1) complexity — constant lookup time at any scale.

■ Block storage service

Volume management, snapshot management, thin provisioning, recycle bin, online volume migration, and Ceph import.

■ USS protocol gateway

The Unified Storage Service is the engine's protocol adaptation layer — NVMe-oF, vhost, HDFS, KV, and SQL. Multiple protocols interoperate on the same data.

■ Tianshu metadata module

The brain of the engine: Raft-based global metadata management with unified scheduling and fault isolation across pools, clouds, and edges.

■ Reliability subsystem

Data-reliability and business-continuity mechanisms, including an in-house intelligent sub-health framework covering drives, networks, and nodes, plus self-healing and resource isolation.

■ O&M subsystem

Monitoring, alerting, configuration, upgrades, patching, permissions, service management, hardware replacement, log collection, and intelligent inspection — essential to fast fault detection and recovery.

3.2 Hardware Architecture

As a software-defined product, nHPS adheres to open, hardware-agnostic principles and is compatible with mainstream servers, storage media, and IO cards.

- **Server compatibility** — x86 servers from Dell, HPE, Huawei, Inspur, Lenovo, H3C, and others, plus domestic CPU platforms (Kunpeng, Hygon, Phytium). Listed models are productization-tested; others require compatibility certification.
- **Media compatibility** — main-store media: SAS/NL-SAS/SATA HDD, SATA/SAS SSD, NVMe SSD drives and AIC cards; cache media: SATA/SAS/NVMe SSD drives and AIC cards.
- **IO card compatibility** — a plugin mechanism maps RAID-card PHY numbers, drive slot numbers, and drive ESNs, and handles vendor-specific error codes. Supported chips include MegaRAID SAS-3 3108 and Tri-Mode 3508/3416. Any TCP/IP Ethernet or RoCE v2 RDMA NIC can serve as the storage-plane network.

3.3 Network Architecture

nHPS separates the **frontend protocol network** (NVMe over RDMA, NVMe over TCP, iSCSI) from the **backend engine network** that interconnects USS ↔ OSD and OSD ↔ OSD over TCP or RDMA. Frontend and backend network types combine freely to match hardware environments and business scenarios.

Network Mode	Deployment & Positioning
TCP	Select TCP for both the storage pool and the USS. Lowest performance, best universality — for standard Ethernet environments.
RDMA · RoCE v2	Select RDMA for both. Near-peak cluster performance; requires RoCE v2-capable NICs.
RDMA · InfiniBand	With InfiniBand-capable NICs, the IB protocol is engaged automatically — extreme performance, extreme IOPS, and lowest latency.
NVMe over FC	Pool network RDMA, USS protocol NVMe-oF, gateway network FC. Extreme performance where existing FC switches can be reused.

Every mode supports two topologies: **separated deployment** (USS on compute nodes, storage pools on storage nodes — balanced NIC traffic, fewer network bottlenecks) and **converged deployment** (USS and pools together on storage nodes — saves compute-node CPU).

3.4 Key Service Flows

3.4.1 System deployment

Deployment is reliability-first, with four fault-domain levels: drive (test scenarios only), server, rack (recommended), and data center. Each server assumes one or more roles, and the system launches the corresponding services:

■ Management module

Operations and maintenance: monitoring, alerting, resource creation/deletion, hardware replacement, and inspection.

■ Tianshu metadata module

Metadata services and topology view changes on failure.

■ Storage module

Data persistence via a strong-consistency replication protocol — a single service, the OSD.

■ Gateway module

The USS, exposing block storage interfaces to hosts.

3.4.2 Access protocols

iSCSI — the de facto standard block interface and nHPS's recommended standardized delivery scenario. No proprietary client software is required. Multipathing software provides link-level HA: on link failure the system fails over automatically and fails back on recovery, with no business impact.

NVMe-oF — defined by the NVMe standards body to unlock flash-era hardware. nHPS supports three transports:

■ NVMe over RDMA

Zero-copy moves data directly between system memories; remote DMA lands data straight in target memory, cutting CPU load and latency; transport bypasses the CPU; flexible topologies and rich asynchronous interfaces.

■ NVMe over TCP

Cost-effective on standard Ethernet — no special hardware. Simple configuration (no DCB needed), excellent interoperability, reuses existing infrastructure, approaches local-NVMe performance in many cases, and behaves predictably under congestion.

■ NVMe over FC

Combines NVMe performance with FC stability and investment protection: reuses FC fabrics at low upgrade cost, fast failure notification, stable low latency, and supports both legacy and new SAN workloads (big data, IoT, AI/DL).

vhost — the vhost-user protocol accelerates virtual machine IO to **1M IOPS per disk with sub-100 µs per-queue latency**:

- **Full-stack zero-copy** — huge-page shared memory plus RDMA keep VM IO zero-copy from guest memory through USS to OSD
- **Polling backend** — the USS vhost backend polls continuously, eliminating per-IO notification (kick) overhead
- **Fewer context switches** — vhost-user offloads the host-kernel part of the IO stack into the USS process
- **Multi-core concurrency** — a balancing algorithm spreads virtqueues evenly across CPU cores for massive concurrency

Cloud-native CSI — Kubernetes has become the de facto cloud-native operating system, and CSI its storage interface. nHPS's cloud-native CSI capability:

- **Full CSI support** over both iSCSI and NVMe-oF protocols
- **In-house USS operator** — deploys gateways onto any Kubernetes node; CPU pinning and huge pages are orchestrated by Kubernetes; a self-developed controller tracks storage exceptions across the full link and dynamically migrates failing pods
- **Adaptive NIC NUMA-affinity scheduling** — prioritizes the NIC's local NUMA node to raise cluster performance
- **Cloud-native block gateway** — gateways deployed on compute nodes shrink the fault domain and use loopback networking for performance
- **Cloud-native multipath** — schedules paths by physical topology, multiplying per-node volume attachment scale; Kata Containers runtime supported; full-link monitoring of storage network and cluster with automatic failover

04 Key Block Storage Features

Thin provisioning

Volumes are created with a single metadata record and no physical space; capacity is allocated only when data is actually written, dramatically improving utilization. nHPS thin provisioning is native — routed by DHT with no central metadata cluster — and incurs no performance penalty versus traditional provisioning.

Online volume expansion

Expansion completes online with no service interruption, preceded by a series of automatic risk checks.

Lossless ROW snapshots

A snapshot freezes volume data at a point in time — a "photo" of the volume for later rollback. nHPS adopts **Redirect-on-Write (ROW)** rather than Copy-on-Write: after a snapshot is taken, new writes are redirected to snapshot space and pointers are updated, so write performance is entirely unaffected. Two-phase commit guarantees crash consistency in every scenario, and intelligent snapshot-deletion scheduling with precise QoS flow control keeps merges from disturbing frontend business.

■ Second-level operations

Creation, rollback, and deletion all complete in seconds; rollback requires no extra data copy.

■ Online & lossless

Snapshot creation has zero impact on business IO; write speed is unchanged afterward.

■ Incremental

Snapshots record only deltas, minimizing space consumption.

■ Crash consistency

Every snapshot fully and accurately captures the volume's data at its point in time, under any failure.

■ Scheduled snapshots

Custom policies by hour/day/week/month with retention limits; oldest snapshots pruned automatically.

■ Chain flattening

Flattened ROW chains give cloned volumes complete data — reads carry zero snapshot-chain latency.

Consistency groups

Volumes belonging to one application — for example, database data and log volumes — are grouped and snapshotted atomically, so the whole set can be recovered to a single consistent point in time after events such as ransomware attacks.

Volume cloning

Linked clones are created instantly from a snapshot and share its data, consuming no space until new data is written — without affecting the snapshot or other clones. **Independent (full) clones** copy the snapshot's complete data at creation and are fully detached afterward. Clones inherit all volume functions, including further snapshots and rollback.

Volume QoS

Online maximum-IOPS and maximum-bandwidth settings per volume prevent noisy-neighbor interference. Multiple reusable QoS policies can be bound to individual volumes or batches, simplifying operations.

Volume recycle bin

Deleted volumes enter a recycle bin — the last line of defense against misoperation or audit-process defects. The default retention is 30 days (configurable 1–180), with manual restore or early deletion to free space.

Stretched cluster

A stretched cluster extends a storage pool across two data centers with a witness node, enabling metro active-active for industries with the strictest availability requirements — protecting against earthquakes, fire, flood, and site-level outages.

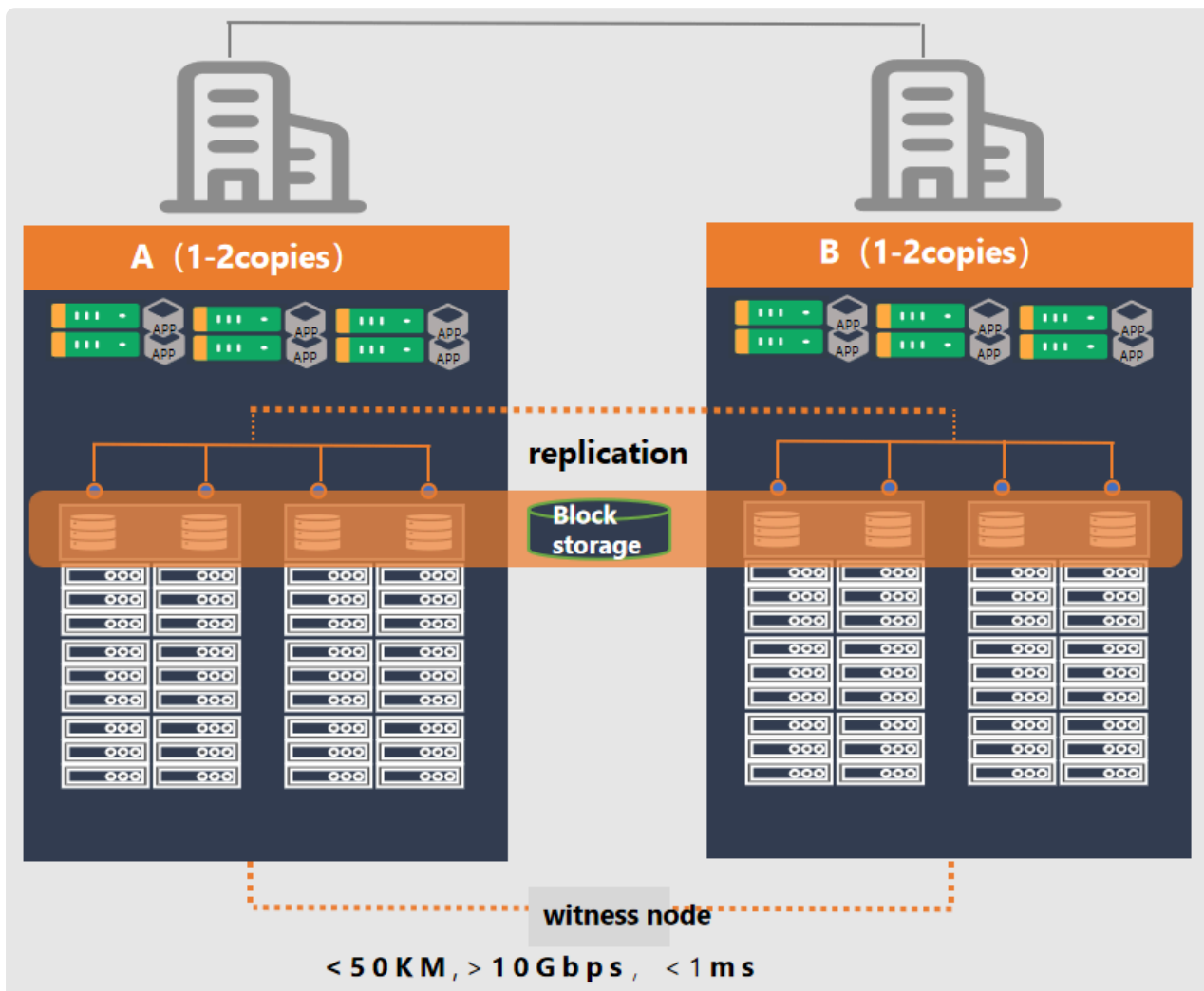


Figure 2 Stretched cluster — two data centers, one witness node

Aspect	Description
Deployment	Two data centers plus one witness node (a VM, physical machine, or cloud instance) placed at an independent site.
Network	Distance ≤ 100 km recommended; one-way latency < 2 ms (round trip < 4 ms); bandwidth > 10 GbE.
Replica placement	The replica-distribution algorithm spreads copies across data centers while pinning each volume's primary copy to one site — locality-aware "pseudo active-active," transparent to the application.
DR capability	RPO = 0; RTO ≈ 0 (application switchover dependent); automatic failover together with the application layer.

Asynchronous replication

Async remote replication is implemented either log-based (each update receives a unique, increasing sequence number and is replayed in order on the backup cluster) or snapshot-based (periodic snapshots compared to sync only the deltas). nHPS builds async replication between two nHPS clusters on the snapshot-based approach, triggered manually or by schedule (minimum hourly interval).

Online volume migration

Log-based migration dual-writes every IO to source and target (RAID1-like mirroring) and achieves second-level RPO and RTO. **Snapshot-based migration** performs a full pass plus incremental passes, reaching hour-level RPO and minute-level RTO. nHPS adopts the log-based approach for maximum availability. Typical use cases: rebalancing capacity across storage pools and rebalancing performance across storage pools.

05 Elasticity

5.1 DHT Algorithm

nHPS's DHT mechanism spreads every volume's data blocks evenly across drives and servers — no local hotspots, global load balancing. Primaries and replicas are balanced per drive; after nodes are added or removed, the reconstruction algorithm rebalances the cluster automatically. The DHT provides two key properties:

- **Balance** — data is distributed across all nodes, keeping every node evenly loaded
- **Monotonicity** — when a new node joins, only it receives migrated data; data on existing nodes barely moves

5.2 Smooth Scale-Out and Scale-In

Storage pools scale at the granularity of drives, nodes, or racks. The in-house data-distribution algorithm minimizes useless migration during expansion — data rebalances automatically afterward without large-scale movement.

5.3 Composable Architecture

The engine's fast IO stack precisely measures system resources and the performance they deliver: every process and module is quantified as "**resources + function + performance metric.**" Systems can therefore be composed precisely:

- **Scenario-defined** — e.g., a virtualization deployment needing $N \times 10K$ IOPS per volume and $M \times 10K$ cluster-wide: select NICs, media, CPU, and memory to match, then compute the software threads and resources (CPU + memory) accordingly
- **Hardware-defined** — for already-purchased hardware, compose the software stack to maximize utilization of exactly what you own

WHY IT MATTERS

Composability makes storage resources precisely quantifiable — hardware is used to its full potential, performance is predictable, and energy consumption and waste are minimized.

06 High-Performance Technology

6.1 In-House Lock-Free, High-Speed IO Stack

The storage engine replaces traditional layered forwarding, kernel-call dependency, and thread-pool concurrency with a completely new architecture: a **full user-space, lock-free data path** built on a **Run-to-Completion (RTC)** IO processing model. The stack achieves end-to-end operation with no context switches, no locks, zero interrupts, memory zero-copy, and rapid network IO. Core pinning and asynchronous polling exploit each CPU core's full capability.

20–40 μ s

nHPS software-stack latency per IO

~400 μ s

Typical first-generation distributed IO stack

10×

Order-of-magnitude improvement —
unlocking new media and networks

6.2 NUMA-Aware Framework

The in-house NUMA-aware framework is especially friendly to multi-NUMA domestic CPU platforms:

- **End-to-end zero-copy** — on the compute side, from protocol IO buffer to network send; on the storage side, from network receive to disk
- **Huge-page memory management** — centralized huge pages with an in-house allocator that stays NUMA-local when allocating
- **Lossless cross-NUMA scheduling** — combined with the lock-free IO stack, scheduling across NUMA nodes costs almost nothing on domestic platforms

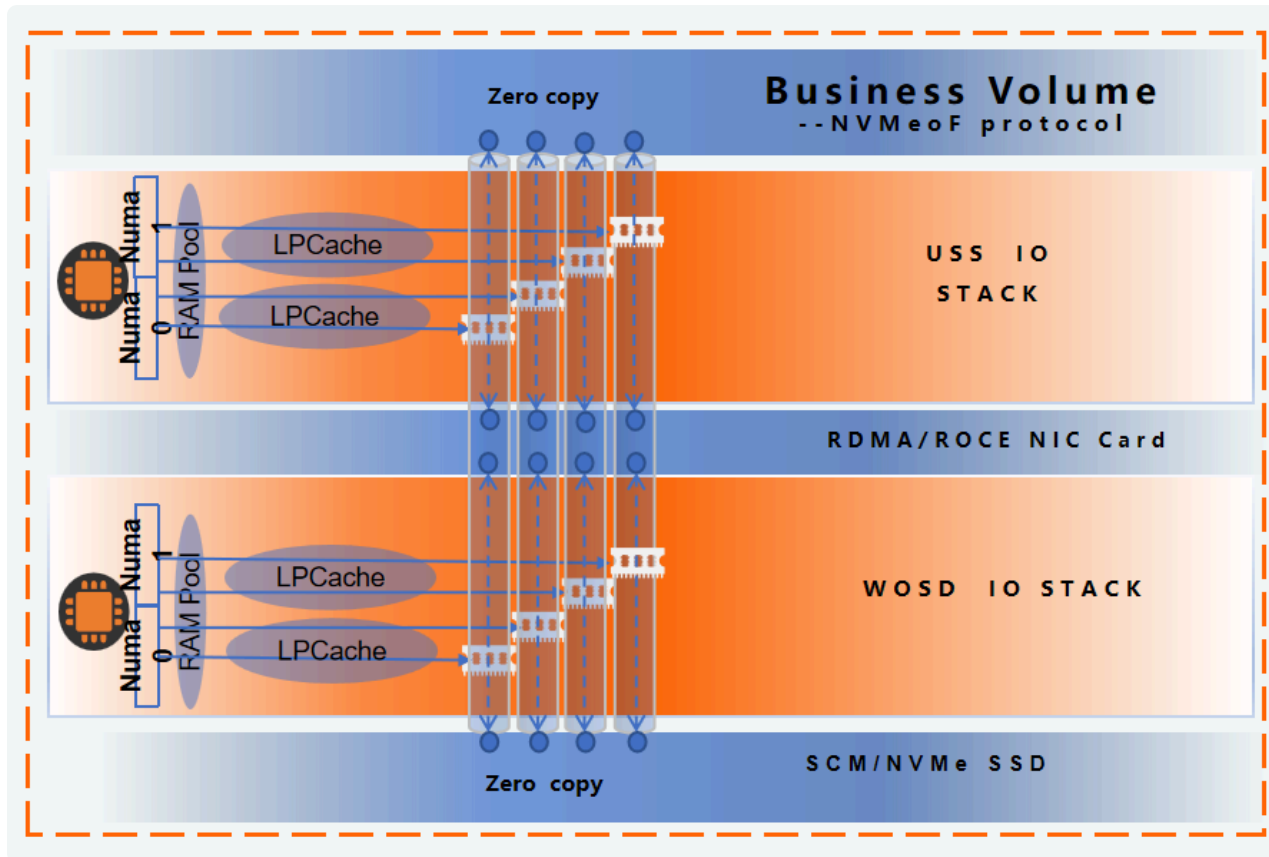


Figure 3 NUMA-aware framework with end-to-end zero-copy

6.3 Multi-Level SSD Cache

A small number of SSDs per node accelerate many HDDs, with multiple cooperating techniques:

Cache & metadata management

One SSD binds to and accelerates multiple HDDs; hash buckets plus linked lists map SSD and HDD extents in memory for fast IO-path lookups.

Reads & writes

Reads return immediately on cache hit and fall through to HDD on miss; writes land in the SSD when space allows, otherwise wait for flushing to release space.

Water-level flushing

Dirty data is flushed faster as SSD usage rises; below the 60% threshold no flushing occurs — fully exploiting SSD performance.

Large-block write-through

Requests above a configurable threshold pass through to HDD (which handles large sequential IO well) — protecting SSD life, and especially effective when SATA/SAS cache bandwidth is below aggregate HDD bandwidth.

■ Elevator scheduling

Dirty data is sorted by target LBA and flushed sequentially — random IO becomes sequential IO, boosting flush efficiency by more than 10×.

■ Memory cache & read-ahead

A sliding-window, multi-stream prediction algorithm pre-reads hot data into memory for additional acceleration.

07 High-Reliability Technology

7.1 System Reliability Design

Distributed storage must tolerate hardware failures, misoperation, power loss, and network interruption. nHPS is architected so that no single point of failure exists:

- **Management nodes** — the O&M platform runs active-standby, so operations remain available through failures
- **Tianshu metadata cluster** — Raft-based strong consistency, deployed with 3, 5, or 7 nodes by cluster scale; heartbeats and automatic leader election guarantee continuity
- **Storage cluster** — replication or erasure-coding redundancy; any single drive or server failure leaves business continuity and data reliability intact; all nodes health-check each other in a full-mesh mechanism

Three additional O&M-layer protections reduce the chance of human error: **high-risk operation prompts** on every dangerous action; **fine-grained roles** (administrators perform routine operations such as drive replacement and scaling; operators perform advanced operations such as dynamically modifying reconstruction parameters); and **hidden operations** — the most dangerous actions (e.g., forced SSDCache flush, block-service re-initialization) are invisible unless a special hotkey is used.

7.2 Data Reliability

Multi-replica redundancy

nHPS's redundancy is built on the DHT algorithm: under replication policies, replicas are written to different drives with balanced capacity usage, and policies can be server- or rack-aware — the application never needs to know where replicas live. With 3 replicas the probability of data loss is already vanishingly small. Replica counts of 2–6 let users trade freely between reliability and usable space, and recovery automatically returns data to its configured redundancy.

Multi-level fault isolation

Four fault-domain levels — drive, server, rack, and data center — keep failures contained within a domain. Beyond physical isolation, storage pools add watermark protection: when written data exceeds a configured threshold, the system alarms first and then proactively stops new writes until expansion.

Parallel data reconstruction

- **No manual intervention** — recovery requires no new hardware; space is allocated on healthy devices automatically
- **Incremental recovery** — only valid data moves, not entire disks, slashing recovery traffic
- **Parallel recovery** — every healthy server participates concurrently in a many-to-many pattern
- **Scale effect** — larger clusters rebuild faster: the bigger the system, the safer it is

Reconstruction strategies

The system senses frontend business load through internal traffic state and allocates resources accordingly — busy periods favor business, idle periods favor reconstruction. Three policies — **business-first**, **balanced**, and **recovery-first** — are configurable to match different scenarios.

7.3 Business Continuity

Intelligent sub-health detection framework

Sub-health (gray failure) is among the hardest problems in distributed storage: how much slowdown qualifies as sub-health, and how to react without triggering secondary faults. nHPS's unified framework goes beyond static or dynamic thresholds — it correlates **trends in underlying monitoring metrics with trends in frontend business metrics**, raising detection accuracy and preventing false positives and negatives. Four cooperating subsystems — monitoring (hardware, pool, and business metrics), analysis (anomaly detection, clustering, business-damage assessment, event correlation), fault handling (alerting and isolation with an anti-mass-isolation algorithm that limits fault impact), and a configuration center — cover the full loop from detection to containment.

Intelligent slow-drive detection

Slow drives — elevated IO latency while still serving data — can slash business performance. nHPS detects both **horizontal** degradation (a drive falling behind peers over time) and **vertical** degradation (sudden dips that may or may not recover). The algorithm records each drive's IO model (size, latency, type, count), internal SMART metrics, and IO-path errors, then applies machine learning and clustering to build environment-specific dynamic thresholds — catching absolutely and relatively slow drives alike, then alerting and isolating them.

Intelligent network sub-health detection

Unlike hard link-down failures, network sub-health is fuzzy and business-dependent. nHPS adaptively probes different links, ports, and NICs based on the network model and scale, detecting latency anomalies, error packets, and loss across dimensions; correlates IOPS, bandwidth, and latency trend analysis with network metrics for accurate verdicts; and on detection alarms and isolates links, ports, or NICs progressively while switching business onto healthy paths.

Intelligent node sub-health detection

Node sub-health usually stems from CPU or memory starvation or an overloaded system disk rather than failed hardware — commands that normally take milliseconds take seconds. nHPS instruments IO processing stages in real time, applies election algorithms over per-node statistics to identify suspect nodes, then alarms and isolates them.

Business-aware backpressure

Under overload, most systems throttle on lower-layer metrics alone — and important workloads suffer. nHPS additionally collects cluster- and volume-level metrics and applies QoS differentially on overload — some volumes exempt — so high-priority business is protected while stability is maintained.

08 Efficient O&M

8.1 Cloud-Native Operations Platform

The O&M platform is built cloud-native: all modules — IO-path components (Tianshu, USS, OSD) and management-plane modules (Manager, Agent, DB) — are containerized and orchestrated with Kubernetes and microservices. Containerization enables low-risk rapid deployment, upgrade, and rollback: faulty container packages are simply removed without affecting the host OS, and on-premises and cloud deployments become fully consistent for hybrid multi-cloud environments.

DESIGNED FOR FAST RECOVERY

A comprehensive monitoring and alerting system detects faults quickly, while rich O&M tools and drilled emergency playbooks keep operators composed under pressure — restoring business in the shortest possible time.

8.2 Operations Capabilities

■ Graphical management console

Monitoring, alerting, and daily operations across physical resources, logical pools, and service components — nearly all O&M work completes in the GUI, reducing manual command-line errors.

■ Fine-grained role-based access control

Four roles — super administrator, administrator, operator, and auditor — with strictly separated privileges: super administrators manage users only; administrators handle routine configuration but not high-risk actions; operators hold all non-user-management rights; auditors access operation logs only.

■ Audit operation logs

Every management operation is recorded — account, role, time, user, source IP, and result — exportable as CSV, retained 180 days, tamper-proof, and permanently enabled for compliance.

■ Drive roaming

Move drives from a failed server into a replacement and rejoin the original pool — avoiding double full reconstruction and its elevated risk.

■ Drive locate LED & guided replacement

A wizard-driven, ESN-verified replacement flow covering hot-swap and chassis-open drive types across all roles (data, cache, with or without slots).

■ SSD life monitoring & alerting

Real-time tracking of write amplification, P/E cycles, and remaining life, with automatic alerts via email or instant notification — enabling preventive maintenance and fewer unplanned outages.

■ One-click intelligent inspection

Routine mode: alarm correlation and root-cause analysis, capacity growth trends, system load analysis. Deep mode: log, internal-event, and core-metric analysis to surface hidden risks. Both support manual and scheduled runs and produce a complete report.

■ One-click information collection

Gathers system configuration, cluster state, business and system logs, change history, and internal process metrics in a single action — critical when remote login is impossible.

■ Maintenance mode

During upgrades or hardware maintenance, removing or adding nodes and pulling drives triggers no reconstruction; management changes are blocked for safety.

■ System backup & recovery

Periodic automatic backup of key configuration and metadata for rapid recovery after OS reinstallation, system-disk failure, or unbootable nodes — avoiding full reconstruction even in extreme cases.

■ Online management-IP change

Re-address the management plane after network re-planning or data-center relocation without redeploying the cluster.

■ Online replica-count change

Raise replicas for critical data, lower them where space or cost dominates — balancing reliability, performance, and cost across tiers.

■ Flexible alarm masking

Temporary, planned, rule-based, and global masking with recorded reasons — reducing noise while preserving team context.

■ One-click upgrade & patching

Batch, zero-business-impact upgrades with full-process monitoring, online or offline, keeping data safe throughout.

■ Emergency response plans

Pre-released emergency tools and playbooks, drilled against real failure scenarios — even extreme field failures are handled calmly and business restored in the fastest time.

09 Security

Storage systems face data leakage, data corruption, and temporary or permanent loss of accessibility and availability — from both traditional and network-borne threats. nHPS counteracts these risks across three dimensions, protecting data integrity, confidentiality, and availability against unauthorized users and uses.

■ Management security

User roles, password composition policies, session expiry, and log/alarm auditing — layered controls on the management plane.

■ Service security

ACL-based access control, encrypted authentication, and antivirus protection through agent software cooperating with third-party engines.

■ Network security

Three isolated planes: the frontend business network (client access, external services), the backend storage network (closed, externally isolated inter-node traffic), and a dedicated O&M management network. Remote management uses SSH, SFTP, and HTTPS only.

10 Abbreviations

Term	Full Name	Description
ACL	Access Control List	A rule set — a fundamental network-security component
CSI	Container Storage Interface	Standard storage interface for Kubernetes
DHT	Distributed Hash Table	A distributed storage algorithm; O(1) object location
DNS	Domain Name System	Internet name-resolution system
DPDK	Data Plane Development Kit	Userspace packet-processing toolkit for Intel architectures
ECN	Explicit Congestion Notification	Reduces retransmission-induced latency and jitter
ESN	Equipment Serial Number	Storage-device serial number
HDFS	Hadoop Distributed File System	Distributed file system of the Apache Hadoop ecosystem
IB	InfiniBand	High-performance computing network standard
IOPS	Input/Output Operations Per Second	Read/write operations per second
iSCSI	Internet Small Computer System Interface	Block storage protocol over TCP/IP (SCSI-3 based)
iWARP	Internet Wide-area RDMA Protocol	RDMA implemented over TCP
MLC / TLC / QLC / SLC	Multi-/Trinary-/Quad-/Single-Level Cell	Flash cells storing 2 / 3 / 4 / 1 bits respectively
NUMA	Non-Uniform Memory Access	Memory access varies by distance to the CPU
NVMe	Nonvolatile Memory Express	Storage protocol designed from the ground up for performance
NVMe-oF	NVMe over Fabrics	Extends NVMe to front-end storage networks
OSD	Object Storage Device	Per-drive data persistence service in nHPS
PFC	Priority-based Flow Control	Priority-based congestion control for lossless Ethernet
QoS	Quality of Service	Resource allocation guarantees and limits
Raft	—	Distributed consensus algorithm

RAID	Redundant Array of Independent Disks	Disk subsystem for performance or fault tolerance
RDMA	Remote Direct Memory Access	Kernel-bypass, zero-copy network technology
RoCE	RDMA over Converged Ethernet	RDMA implemented over UDP/Ethernet
RTC	Run to Completion	Efficient forwarding and programming model
SMART	Self-Monitoring, Analysis and Reporting Technology	Records drive health state during operation
SPDK	Storage Performance Development Kit	Storage toolkit designed for NVMe devices
TiKV	—	Distributed transactional key-value database
USS	Unified Storage Service	nHPS protocol gateway and adaptation layer
nHPS	Hyper-IO Disk Storage	NexaVM hyper-performance distributed block storage



DATA. VELOCITY. VALUE.



NexaVM Technologies AG

W: www.nexavm.com

E: support@nexavm.com

© 2026 NEXAVM TECHNOLOGIES AG. ALL RIGHTS RESERVED.